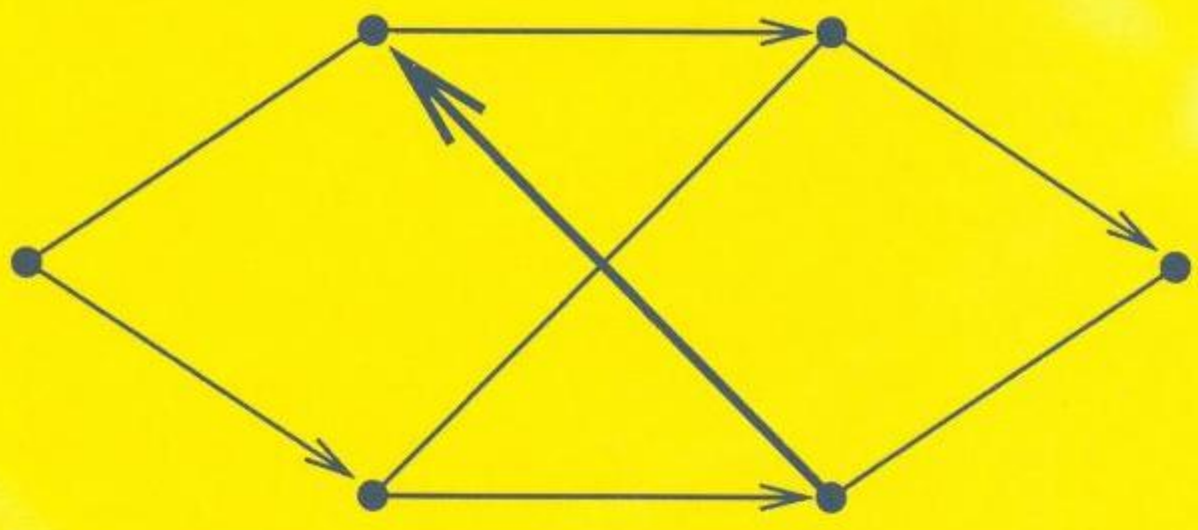
24 Algorithms and Combinatorics

Volume A

Combinatorial Optimization

Polyhedra and Efficiency

Alexander Schrijver



Springer

Algorithms and Combinatorics 24

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Alexander Schrijver

Combinatorial Optimization

Polyhedra and Efficiency

Volume A-C

Springer

Alexander Schrijver
CWI
Kruislaan 413
1098 SJ Amsterdam
The Netherlands
e-mail: lex@cwi.nl

Library of Congress Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at http://dnb.ddb.de

Mathematics Subject Classification (2000): 90C27, 05Cxx, 05C85, 68R05, 68Q25

ISSN 0937-5511
ISBN 3-540-20456-3 Springer-Verlag Berlin Heidelberg New York
ISBN 3-540-44389-4 print edition Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media
springeronline.com

Cover design: *design & production*, Heidelberg
Typesetting: by the author using a Springer LaTeX macro package
Printed on acid-free paper 46/3142ck-5 4 3 2 1 0

Preface

The book by Gene Lawler from 1976 was the first of a series of books all entitled 'Combinatorial Optimization', some embellished with a subtitle: 'Networks and Matroids', 'Algorithms and Complexity', 'Theory and Algorithms'. Why adding another book to this illustrious series? The justification is contained in the subtitle of the present book, 'Polyhedra and Efficiency'. This is shorthand for Polyhedral Combinatorics and Efficient Algorithms.

Pioneered by the work of Jack Edmonds, polyhedral combinatorics has proved to be a most powerful, coherent, and unifying tool throughout combinatorial optimization. Not only it has led to efficient (that is, polynomial-time) algorithms, but also, conversely, efficient algorithms often imply polyhedral characterizations and related min-max relations. It makes the two sides closely intertwined.

We aim at offering both an introduction to and an in-depth survey of polyhedral combinatorics and efficient algorithms. Within the span of polyhedral methods, we try to present a broad picture of polynomial-time solvable combinatorial optimization problems — more precisely, of those problems that have been proved to be polynomial-time solvable. Next to that, we go into a few prominent NP-complete problems where polyhedral methods were successful in obtaining good bounds and approximations, like the stable set and the traveling salesman problem. Nevertheless, while we obviously hope that the question 'NP=P?' will be settled soon one way or the other, we realize that in the astonishing event that NP=P will be proved, this book will be highly incomplete.

By definition, being in P means being solvable by a 'deterministic sequential polynomial-time' algorithm, and in our discussions of algorithms and complexity we restrict ourselves mainly to this characteristic. As a consequence, we do not cover (but yet occasionally touch or outline) the important work on approximative, randomized, and parallel algorithms and complexity, areas that are recently in exciting motion. We also neglect applications, modelling, and computational methods for NP-complete problems. Advanced data structures are treated only moderately. Other underexposed areas include semidefinite programming and graph decomposition. 'This all just to keep size under control.'

Although most problems that come up in practice are NP-complete or worse, recognizing those problems that are polynomial-time solvable can be very helpful: polynomial-time (and polyhedral) methods may be used in preprocessing, in obtaining approximative solutions, or as a subroutine, for instance to calculate bounds in a branch-and-bound method. A good understanding of what is in the polynomial-time tool box is essential also for the NP-hard problem solver.

* * *

This book is divided into eight main parts, each discussing an area where polyhedral methods apply:

I. Paths and Flows
II. Bipartite Matching and Covering
III. Nonbipartite Matching and Covering
IV. Matroids and Submodular Functions
V. Trees, Branchings, and Connectors
VI. Cliques, Stable Sets, and Colouring
VII. Multiflows and Disjoint Paths
VIII. Hypergraphs

Volume A contains Parts I–III, Volume B Parts IV–VI, and Volume C Parts VII and VIII, the list of References, and the Name and Subject Indices.

Each of the eight parts starts with an elementary exposition of the basic results in the area, and gradually evolves to the more elevated regions. Subsections in smaller print go into more specialized topics. We also offer several references for further exploration of the area.

Although we give elementary introductions to the various areas, this book might be less satisfactory as an introduction to combinatorial optimization. Some mathematical maturity is required, and the general level is that of graduate students and researchers. Yet, parts of the book may serve for undergraduate teaching.

The book does not offer exercises, but, to stimulate research, we collect open problems, questions, and conjectures that are mentioned throughout this book, in a separate section entitled ‘Survey of Problems, Questions, and Conjectures’ (in Volume C). It is not meant as a complete list of all open problems that may live in the field, but only of those mentioned in the text.

We assume elementary knowledge of and familiarity with graph theory, with polyhedra and linear and integer programming, and with algorithms and complexity. To support the reader, we survey the knowledge assumed in the introductory chapters, where we also give additional background references. These chapters are meant mainly just for consultation, and might be less attractive to read from front to back. Some less standard notation and terminology are given on the inside back cover of this book.

For background on polyhedra and linear and integer programming, we also refer to our earlier book *Theory of Linear and Integer Programming*

(Wiley, Chichester, 1986). This might seem a biased recommendation, but this 1986 book was partly written as a preliminary to the present book, and it covers anyway the author's knowledge on polyhedra and linear and integer programming.

Incidentally, the reader of this book will encounter a number of concepts and techniques that regularly crop up: total unimodularity, total dual integrality, duality, blocking and antiblocking polyhedra, matroids, submodularity, hypergraphs, uncrossing. It makes that the meaning of 'elementary' is not unambiguous. Especially for the basic results, several methods apply, and it is not in all cases obvious which method and level of generality should be chosen to give a proof. In some cases we therefore will give several proofs of one and the same theorem, just to open the perspective.

* * *

While I have pursued great carefulness and precision in composing this book, I am quite sure that much room for corrections and additions has remained. To inform the reader about them, I have opened a website at the address

`www.cwi.nl/~lex/co`

Any corrections (including typos) and other comments and suggestions from the side of the reader are most welcome at

`lex@cwi.nl`

I plan to provide those who have contributed most to this, with a complimentary copy of a potential revised edition.

* * *

In preparing this book I have profited greatly from the support and help of many friends and colleagues, to whom I would like to express my gratitude.

I am particularly much obliged to Sasha Karzanov in Moscow, who has helped me enormously by tracking down ancient publications in the (former) Lenin Library in Moscow and by giving explanations and interpretations of old and recent Russian papers. I also thank Sasha's sister Irina for translating Tolstoĭ's 1930 article for me.

I am very thankful to Tamás Fleiner, András Frank, Bert Gerards, Dion Gijswijt, Willem Jan van Hoeve, Sasha Karzanov, Judith Keijsper, Monique Laurent, Misha Lomonosov, Frédéric Maffay, Gábor Maróti, Coelho de Pina, Bruce Shepherd, and Bianca Spille, for carefully reading preliminary parts of this book, for giving corrections and suggestions improving the text and the layout, and for helping me with useful background information. I am also happy to thank Noga Alon, Csaba Berki, Vasek Chvátal, Michele Conforti, Bill Cook, Gérard Cornuéjols, Bill Cunningham, Guoli Ding, Jack Edmonds, Fritz Eisenbrand, Satoru Fujishige, Alan Hoffman, Tibor Jordán, Gil Kalai, Alfred Lehman, Jan Karel Lenstra, Laci Lovász, Bill Pulleyblank, Herman te Riele, Alexander Rosa, András Sebő, Paul Seymour, Bruno Simeone, Jan Smaus, Adri Steenbeek, Laci Szegő, Éva Tardos, Bjarne Toft, and David Williamson, for giving useful insights and suggestions, for providing me with precious and rare papers and translations, for advice on interpreting vintage articles, and for help in checking details.

Sincere thanks are due as well to Truus W. Koopmans for sharing with me her 'memories and stories' and sections from her late husband's war diary, and to Herb Scarf for his kind mediation in this. I am indebted to Steve Brady (RAND) and Dick Cottle for their successful efforts in obtaining classic RAND Reports for me, and to Richard Bancroft and Gustave Shubert of RAND Corporation for their help in downgrading the secret Harris-Ross report.

The assistance of my institute, CWI in Amsterdam, has been indispensable in writing this book. My special thanks go to Karin van Gemert of the CWI Library for her indefatigable efforts in obtaining rare publications from every corner of the world, always in sympathizing understanding for my often extravagant requests. Also Thea de Hoog and Rick Ooteman were very helpful in this. Other members of CWI's staff whose assistance is highly appreciated include Miente Bakker, Susanne van Dam, Lieke van den Eersten, Jacqueline de Klerk, Wouter Mettrop, Ay Ong, and Jos van der Werf.

In the technical realization of this book, I thankfully enjoyed the first-rate workmanship of the staff of Springer-Verlag in Heidelberg. I thank in particular Frank Holzwarth, Claudia Kehl, Leonie Kunz, Ute McCrory, and Martin Peters for their skilful and enthusiastic commitment in finalizing this out-size project.

As it has turned out, it was only by gravely neglecting my family that I was able to complete this project. I am extremely grateful to Monique, Nella, and Juliette for their perpetual understanding and devoted support. Now comes the time for the pleasant fulfilment of all promises I made for 'when my book will be finished'.

Amsterdam
November 2002

Alexander Schrijver

Table of Contents

Volume A

Volume B

Volume C

Chapter 1

Introduction

1.1. Introduction

Combinatorial optimization searches for an optimum object in a finite collection of objects. Typically, the collection has a concise representation (like a graph), while the number of objects is huge — more precisely, grows exponentially in the size of the representation (like all matchings or all Hamiltonian circuits). So scanning all objects one by one and selecting the best one is not an option. More efficient methods should be found.

In the 1960s, Edmonds advocated the idea to call a method efficient if its running time is bounded by a polynomial in the size of the representation. Since then, this criterion has won broad acceptance, also because Edmonds found polynomial-time algorithms for several important combinatorial optimization problems (like the matching problem). The class of polynomial-time solvable problems is denoted by P.

Further relief in the landscape of combinatorial optimization was discovered around 1970 when Cook and Karp found out that several other prominent combinatorial optimization problems (including the traveling salesman problem) are the hardest in a large natural class of problems, the class NP. The class NP includes most combinatorial optimization problems. Any problem in NP can be reduced to such 'NP-complete' problems. All NP-complete problems are equivalent in the sense that the polynomial-time solvability of one of them implies the same for all of them.

Almost every combinatorial optimization problem has since been either proved to be polynomial-time solvable or NP-complete — and none of the problems have been proved to be both. This spotlights the big mystery: are the two properties disjoint (equivalently, P$\neq$NP), or do they coincide (P=NP)?

This book focuses on those combinatorial optimization problems that have been proved to be solvable in polynomial time, that is, those that have been proved to belong to P. Next to polynomial-time solvability, we focus on the related polyhedra and min-max relations.

These three aspects have turned out to be closely related, as was shown also by Edmonds. Often a polynomial-time algorithm yields, as a by-product, a description (in terms of inequalities) of an associated polyhedron. Conversely, an appropriate description of the polyhedron often implies the polynomial-time solvability of the associated optimization problem, by applying linear programming techniques. With the duality theorem of linear programming, polyhedral characterizations yield min-max relations, and vice versa.

So the span of this book can be portrayed alternatively by those combinatorial optimization problems that yield well-described polyhedra and min-max relations. This field of discrete mathematics is called *polyhedral combinatorics*. In the following sections we give some basic, illustrative examples.[1]

1.2. Matchings

Let $G = (V, E)$ be an undirected graph and let $w : E \to \mathbb{R}_+$. For any subset F of E, denote

(1.1) $$w(F) := \sum_{e \in F} w(e).$$

We will call $w(F)$ the *weight* of F.

Suppose that we want to find a *matching* (= set of disjoint edges) M in G with weight $w(M)$ as large as possible. In notation, we want to 'solve'

(1.2) $$\max\{w(M) \mid M \text{ matching in } G\}.$$

We can formulate this problem equivalently as follows. For any matching M, denote the incidence vector of M in $\mathbb{R}^E$ by χ^M; that is,

(1.3) $$\chi^M(e) := \begin{cases} 1 & \text{if } e \in M, \\ 0 & \text{if } e \notin M, \end{cases}$$

for $e \in E$. Considering w as a *vector* in $\mathbb{R}^E$, we have $w(M) = w^{\mathsf{T}}\chi^M$. Hence problem (1.2) can be rewritten as

(1.4) $$\max\{w^{\mathsf{T}}\chi^M \mid M \text{ matching in } G\}.$$

This amounts to maximizing the linear function $w^{\mathsf{T}}x$ over a finite set of vectors. Therefore, the optimum value does not change if we maximize over the *convex hull* of these vectors:

(1.5) $$\max\{w^{\mathsf{T}}x \mid x \in \text{conv.hull}\{\chi^M \mid M \text{ matching in } G\}\}.$$

The set

(1.6) $$\text{conv.hull}\{\chi^M \mid M \text{ matching in } G\}$$

is a polytope in $\mathbb{R}^E$, called the *matching polytope* of G. As it is a polytope, there exist a matrix A and a vector b such that

[1] Terms used but not introduced yet can be found later in this book — consult the Subject Index.

(1.7) $$\text{conv.hull}\{\chi^M \mid M \text{ matching in } G\} = \{x \in \mathbb{R}^E \mid x \geq \mathbf{0}, Ax \leq b\}.$$

Then problem (1.5) is equivalent to

(1.8) $$\max\{w^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq b\}.$$

In this way we have formulated the original combinatorial problem (1.2) as a *linear programming* problem. This enables us to apply linear programming methods to study the original problem.

The question at this point is, however, how to find the matrix A and the vector b. We know that A and b do exist, but we must know them in order to apply linear programming methods.

If G is bipartite, it turns out that the matching polytope of G is equal to the set of all vectors $x \in \mathbb{R}^E$ satisfying

(1.9) $$\begin{aligned} x(e) &\geq 0 && \text{for } e \in E, \\ \sum_{e \ni v} x(e) &\leq 1 && \text{for } v \in V. \end{aligned}$$

(The sum ranges over all edges e containing v.) That is, for A we can take the $V \times E$ incidence matrix of G and for b the all-one vector $\mathbf{1}$ in $\mathbb{R}^V$.

It is not difficult to show that the matching polytope for bipartite graphs is indeed completely determined by (1.9). First note that the matching polytope is contained in the polytope determined by (1.9), since χ^M satisfies (1.9) for each matching M. To see the reverse inclusion, we note that, if G is bipartite, then the matrix A is *totally unimodular*, i.e., each square submatrix has determinant belonging to $\{0, +1, -1\}$. (This easy fact will be proved in Section 18.2.) The total unimodularity of A implies that the vertices of the polytope determined by (1.9) are *integer* vectors, i.e., belong to $\mathbb{Z}^E$. Now each integer vector satisfying (1.9) must trivially be equal to χ^M for some matching M. Hence, if G is bipartite, the matching polytope is determined by (1.9).

We therefore can apply linear programming techniques to handle problem (1.2). Thus we can find a maximum-weight matching in a bipartite graph in polynomial time, with any polynomial-time linear programming algorithm. Moreover, the duality theorem of linear programming gives

(1.10) $$\begin{aligned} &\max\{w(M) \mid M \text{ matching in } G\} = \max\{w^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq \mathbf{1}\} \\ &= \min\{y^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq w^{\mathsf{T}}\}. \end{aligned}$$

If we take for w the all-one vector $\mathbf{1}$ in $\mathbb{R}^E$, we can derive from this Kőnig's matching theorem (Kőnig [1931]):

(1.11) the maximum size of a matching in a bipartite graph is equal to the minimum size of a vertex cover,

where a *vertex cover* is a set of vertices intersecting each edge. Indeed, the left-most expression in (1.10) is equal to the maximum size of a matching. The minimum can be seen to be attained by an integer vector y, again by the total unimodularity of A. This vector y is a $0,1$ vector in $\mathbb{R}^V$, and hence is the incidence vector χ^U of some subset U of V. Then $y^\mathsf{T} A \geq \mathbf{1}^\mathsf{T}$ implies that U is a vertex cover. Therefore, the right-most expression is equal to the minimum size of a vertex cover.

Kőnig's matching theorem (1.11) is an example of a *min-max formula* that can be derived from a polyhedral characterization. Conversely, min-max formulas (in particular in a weighted form) often give polyhedral characterizations.

The polyhedral description together with linear programming duality also gives a *certificate* of optimality of a matching M: to convince your 'boss' that a certain matching M has maximum size, it is possible and sufficient to display a vertex cover of size $|M|$. In other words, it yields a *good characterization* for the maximum-size matching problem in bipartite graphs.

1.3. But what about nonbipartite graphs?

If G is *nonbipartite*, the matching polytope is not determined by (1.9): if C is an odd circuit in G, then the vector $x \in \mathbb{R}^E$ defined by $x(e) := \frac{1}{2}$ if $e \in EC$ and $x(e) := 0$ if $e \notin EC$, satisfies (1.9) but does not belong to the matching polytope of G.

A pioneering and central theorem in polyhedral combinatorics of Edmonds [1965b] gives a complete description of the inequalities needed to describe the matching polytope for arbitrary graphs: one should add to (1.9) the inequalities

(1.12) $$\sum_{e \subseteq U} x(e) \leq \lfloor \tfrac{1}{2}|U| \rfloor \text{ for each odd-size subset } U \text{ of } V.$$

Trivially, the incidence vector χ^M of any matching M satisfies (1.12). So the matching polytope of G is contained in the polytope determined by (1.9) and (1.12). The content of Edmonds' theorem is the converse inclusion. This will be proved in Chapter 25.

In fact, Edmonds designed a polynomial-time algorithm to find a maximum-weight matching in a graph, which gave this polyhedral characterization as a by-product. Conversely, from the characterization one may derive the polynomial-time solvability of the weighted matching problem, with the ellipsoid method. In applying linear programming methods for this, one will be faced with the fact that the system $Ax \leq b$ consists of exponentially many inequalities, since there exist exponentially many odd-size subsets U of V. So in order to solve the problem with linear programming methods, we cannot just list all inequalities. However, the ellipsoid method does not require that all inequalities are listed a priori. It suffices to have a polynomial-time algorithm answering the question:

(1.13) given $x \in \mathbb{R}^E$, does x belong to the matching polytope of G?

Such an algorithm indeed exists, as it has been shown that the inequalities (1.9) and (1.12) can be checked in time bounded by a polynomial in $|V|$, $|E|$, and the size of x. This method obviously should avoid testing all inequalities (1.12) one by one.

Combining the description of the matching polytope with the duality theorem of linear programming gives a min-max formula for the maximum weight of a matching. It again yields a certificate of optimality: if we have a matching M, we can convince our 'boss' that M has maximum weight, by supplying a dual solution y of objective value $w(M)$. So the maximum-weight matching problem has a good characterization — i.e., belongs to NP$\cap$co-NP.

This gives one motivation for studying polyhedral methods. The ellipsoid method proves polynomial-time solvability, it however does not yield a practical method, but rather an incentive to search for a practically efficient algorithm. The polyhedral method can be helpful also in this, e.g., by imitating the simplex method with a constraint generation technique, or by a primal-dual approach.

1.4. Hamiltonian circuits and the traveling salesman problem

As we discussed above, matching is an area where the search for an inequality system determining the corresponding polytope has been successful. This is in contrast with, for instance, Hamiltonian circuits. No full description in terms of inequalities of the convex hull of the incidence vectors of Hamiltonian circuits — the *traveling salesman polytope* — is known. The corresponding optimization problem is the traveling salesman problem: 'find a Hamiltonian circuit of minimum weight', which problem is NP-complete. This implies that, unless NP=co-NP, there exist facet-inducing inequalities for the traveling salesman polytope that have no polynomial-time certificate of validity. Otherwise, linear programming duality would yield a good characterization. So unless NP=co-NP there is no hope for an appropriate characterization of the traveling salesman polytope.

Moreover, unless NP=P, there is no polynomial-time algorithm answering the question

(1.14) given $x \in \mathbb{R}^E$, does x belong to the traveling salesman polytope?

Otherwise, the ellipsoid method would give the polynomial-time solvability of the traveling salesman problem.

Nevertheless, polyhedral combinatorics can be applied to the traveling salesman problem in a positive way. If we include the traveling salesman polytope in a larger polytope (a *relaxation*) over which we *can* optimize in polynomial time, we obtain a polynomial-time computable bound for the traveling salesman problem. The closer the relaxation is to the traveling salesman polytope, the better the bound is. This can be very useful in a branch-and-bound algorithm. This idea originates from Dantzig, Fulkerson, and Johnson [1954b].

1.5. Historical and further notes

1.5a. Historical sketch on polyhedral combinatorics

The first min-max relations in combinatorial optimization were proved by Dénes Kőnig [1916,1931], on edge-colouring and matchings in bipartite graphs, and by Karl Menger [1927], on disjoint paths in graphs. The matching theorem of Kőnig was extended to the weighted case by Egerváry [1931]. The proofs by Kőnig and Egerváry were in principal algorithmic, and also for Menger's theorem an algorithmic proof was given in the 1930s. The theorem of Egerváry may be seen as polyhedral.

Applying linear programming techniques to combinatorial optimization problems came along with the introduction of linear programming in the 1940s and 1950s. In fact, linear programming forms the hinge in the history of combinatorial optimization. Its initial conception by Kantorovich and Koopmans was motivated by combinatorial applications, in particular in transportation and transshipment.

After the formulation of linear programming as generic problem, and the development in 1947 by Dantzig of the simplex method as a tool, one has tried to attack about all combinatorial optimization problems with linear programming techniques, quite often very successfully. In the 1950s, Dantzig, Ford, Fulkerson, Hoffman, Kuhn, and others studied problems like the transportation, maximum flow, and assignment problems. These problems can be reduced to linear programming by the total unimodularity of the underlying matrix, thus yielding extensions and polyhedral and algorithmic interpretations of the earlier results of Kőnig, Egerváry, and Menger. Kuhn realized that the polyhedral methods of Egerváry for weighted bipartite matching are in fact algorithmic, and yield the efficient 'Hungarian' method for the assignment problem. Dantzig, Fulkerson, and Johnson gave a solution method for the traveling salesman problem, based on linear programming with a rudimentary, combinatorial version of a cutting plane technique.

A considerable extension and deepening, and a major justification, of the field of polyhedral combinatorics was obtained in the 1960s and 1970s by the work and pioneering vision of Jack Edmonds. He characterized basic polytopes like the matching polytope, the arborescence polytope, and the matroid intersection polytope; he introduced (with Giles) the important concept of total dual integrality; and he advocated the interconnections between polyhedra, min-max relations, good characterizations, and efficient algorithms. We give a few quotes in which Edmonds enters into these issues.

In his paper presenting a maximum-size matching algorithm, Edmonds [1965d] gave a polyhedral argument why an algorithm can lead to a min-max theorem:

> It is reasonable to hope for a theorem of this kind because any problem which involves maximizing a linear form by one of a discrete set of non-negative vectors has associated with it a dual problem in the following sense. The discrete set of vectors has a convex hull which is the intersection of a discrete set of half-spaces. The value of the linear form is as large for some vector of the discrete set

> as it is for any other vector in the convex hull. Therefore, the discrete problem is equivalent to an ordinary linear programme whose constraints, together with non-negativity, are given by the half-spaces. The dual (more precisely, a dual) of the discrete problem is the dual of this ordinary linear programme.
> For a class of discrete problems, formulated in a natural way, one may hope then that equivalent linear constraints are pleasant enough though they are not explicit in the discrete formulation.

In another paper (characterizing the matching polytope), Edmonds [1965b] stressed that the number of inequalities is not relevant:

> The results of this paper suggest that, in applying linear programming to a combinatorial problem, the number of relevant inequalities is not important but their combinatorial structure is.

Also in a discussion at the IBM Scientific Computing Symposium on Combinatorial Problems (March 1964 in Yorktown Heights, New York), Edmonds emphasized that the number of facets of a polyhedron is not a measure of the complexity of the associated optimization problem (see Gomory [1966]):

> I do not believe there is any reason for taking as a measure of the algorithmic difficulty of a class of combinatorial extremum problems the number of faces in the associated polyhedra. For example, consider the generalization of the assignment problem from bipartite graphs to arbitrary graphs. Unlike the case of bipartite graphs, the number of faces in the associated polyhedron increases exponentially with the size of the graph. On the other hand, there is an algorithm for this generalized assignment problem which has an upper bound on the work involved just as good as the upper bound for the bipartite assignment problem.

After having received support from H.W. Kuhn and referring to Kuhn's maximum-weight bipartite matching algorithm, Edmonds continued:

> This algorithm depends crucially on what amounts to knowing all the bounding inequalities of the associated convex polyhedron—and, as I said, there are many of them. The point is that the inequalities are known by an easily verifiable characterization rather than by an exhaustive listing—so their number is not important.
> This sort of thing should be expected for a class of extremum problems with a combinatorially special structure. For the traveling salesman problem, the vertices of the associated polyhedron have a simple characterization despite their number—so might the bounding inequalities have a simple characterization despite their number. At least we should hope they have, because finding a really good traveling salesman algorithm is undoubtedly equivalent to finding such a characterization.

So Edmonds was aware of the correlation of good algorithms and polyhedral characterizations, which later got further support by the ellipsoid method.

Also during the 1960s and 1970s, Fulkerson designed the clarifying framework of blocking and antiblocking polyhedra, throwing new light by the classical polarity of vertices and facets of polyhedra on combinatorial min-max relations and enabling, with a theorem of Lehman, the deduction of one polyhedral characterization from another. It stood at the basis of the solution of Berge's perfect graph conjecture in 1972 by Lovász, and it also inspired Seymour to obtain several other basic results in polyhedral combinatorics.

1.5b. Further notes

Raghavan and Thompson [1987] showed that randomized rounding of an optimum fractional solution to a combinatorial optimization problem yields, with high probability, an integer solution with objective value close to the value of the fractional solution (hence at least as close to the optimum value of the combinatorial problem). Related results were presented by Raghavan [1988], Plotkin, Shmoys, and Tardos [1991,1995], and Srinivasan [1995,1999].

Introductions to combinatorial optimization (and more than that) can be found in the books by Lawler [1976b], Papadimitriou and Steiglitz [1982], Sysło, Deo, and Kowalik [1983], Nemhauser and Wolsey [1988], Parker and Rardin [1988], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Mehlhorn and Näher [1999], and Korte and Vygen [2000]. Focusing on applying geometric algorithms in combinatorial optimization are Lovász [1986] and Grötschel, Lovász, and Schrijver [1988]. Bibliographies on combinatorial optimization were given by Kastning [1976], Golden and Magnanti [1977], Hausmann [1978b], von Randow [1982,1985,1990], and O'hEigeartaigh, Lenstra, and Rinnooy Kan [1985].

Survey papers on polyhedral combinatorics and min-max relations were presented by Hoffman [1979], Pulleyblank [1983,1989], Schrijver [1983a,1986a,1987, 1995], and Grötschel [1985], on geometric methods in combinatorial optimization by Grötschel, Lovász, and Schrijver [1984b], and on polytopes and complexity by Papadimitriou [1984].

Chapter 2

General preliminaries

We give general preliminaries on sets, numbers, orders, vectors, matrices, and functions, we discuss how to interpret maxima, minima, and infinity, and we formulate and prove Fekete's lemma.

2.1. Sets

A large part of the sets considered in this book are finite. We often neglect mentioning this when introducing a set. For instance, graphs in this book are finite graphs, except if we explicitly mention otherwise. Similarly for other structures like hypergraphs, matroids, families of sets, etc. Obvious exceptions are the sets of reals, integers, etc.

We call a subset Y of a set X *proper* if $Y \neq X$. Similarly, any other substructure like subgraph, minor, etc. is called *proper* if it is not equal to the structure of which it is a substructure.

A *family* is a set in which elements may occur more than once. More precisely, each element has a *multiplicity* associated. Sometimes, we indicate a family by $(A_1, \ldots, A_n)$ or $(A_i \mid i \in I)$.

A *collection* is synonymous with *set*, but is usually used for a set whose elements are sets. Also *class* and *system* are synonyms of set, and are usually used for sets of structures, like a set of graphs, inequalities, or curves.

A set is called *odd* (*even*) if its size is odd (even). We denote for any set X:

(2.1) $$\begin{aligned} \mathcal{P}(X) &:= \text{collection of all subsets of } X, \\ \mathcal{P}_{\text{odd}}(X) &:= \text{collection of all odd subsets } Y \text{ of } X, \\ \mathcal{P}_{\text{even}}(X) &:= \text{collection of all even subsets } Y \text{ of } X. \end{aligned}$$

Odd and even are called *parities*.

We sometimes say that if $s \in U$, then U *covers* s and s *covers* U. A set U is said to *separate* s and t if $s \neq t$ and $|U \cap \{s,t\}| = 1$. Similarly, a set U is said to *separate* sets S and T if $S \cap T = \emptyset$ and $U \cap (S \cup T) \in \{S, T\}$.

We denote the *symmetric difference* of two sets S and T by $S \triangle T$:

(2.2) $$S \triangle T = (S \setminus T) \cup (T \setminus S).$$

We sometimes use the following shorthand notation, where X is a set and y an 'element':

(2.3) $X + y := X \cup \{y\}$ and $X - y := X \setminus \{y\}$.

We say that sets $S_1, S_2, \ldots, S_k$ are *disjoint* if they are *pairwise* disjoint:

(2.4) $S_i \cap S_j = \emptyset$ for distinct $i, j \in \{1, \ldots, k\}$.

A *partition* of a set X is a collection of disjoint subsets of X with union X. The elements of the partition are called its *classes*.

As usual:

(2.5) $X \subseteq Y$ means that X is a subset of Y,
$X \subset Y$ means that X is a *proper* subset of Y, that is: $X \subseteq Y$ and $X \neq Y$.

Two sets X, Y are *comparable* if $X \subseteq Y$ or $Y \subseteq X$. A collection of pairwise comparable sets is called a *chain*.

Occasionally, we need the following inequality:

Theorem 2.1. *If T and U are subsets of a set S with $T \not\subseteq U$ and $U \not\subseteq T$, then*

(2.6) $$|T||\overline{T}| + |U||\overline{U}| > |T \cap U||\overline{T \cap U}| + |T \cup U||\overline{T \cup U}|,$$

where $\overline{X} := S \setminus X$ for any $X \subseteq S$.

Proof. Define $\alpha := |T \cap U|$, $\beta := |T \setminus U|$, $\gamma := |U \setminus T|$, and $\delta := |\overline{T \cup U}|$. Then:

(2.7) $$|T||\overline{T}| + |U||\overline{U}| = (\alpha + \beta)(\gamma + \delta) + (\alpha + \gamma)(\beta + \delta)$$
$$= 2\alpha\delta + 2\beta\gamma + \alpha\gamma + \beta\delta + \alpha\beta + \gamma\delta$$

and

(2.8) $$|T \cap U||\overline{T \cap U}| + |T \cup U||\overline{T \cup U}| = \alpha(\beta + \gamma + \delta) + (\alpha + \beta + \gamma)\delta$$
$$= 2\alpha\delta + \alpha\gamma + \beta\delta + \alpha\beta + \gamma\delta.$$

Since $\beta\gamma > 0$, (2.6) follows. ∎

A set U is called an *inclusionwise minimal* set in a collection $\mathcal{C}$ of sets if $U \in \mathcal{C}$ and there is no $T \in \mathcal{C}$ with $T \subset U$. Similarly, U is called an *inclusionwise maximal* set in $\mathcal{C}$ if $U \in \mathcal{C}$ and there is no $T \in \mathcal{C}$ with $T \supset U$.

We sometimes use the term *minimal* for *inclusionwise* minimal, and *minimum* for minimum-*size*. Similarly, we sometimes use *maximal* for *inclusionwise* maximal, and *maximum* for maximum-*size* (or maximum-*value* for flows).

A *metric* on a set V is a function $\mu : V \times V \to \mathbb{R}_+$ such that $\mu(v, v) = 0$, $\mu(u, v) = \mu(v, u)$, and $\mu(u, w) \leq \mu(u, v) + \mu(v, w)$ for all $u, v, w \in V$.

2.2. Orders

A relation $\leq$ on a set X is called a *pre-order* if it is reflexive ($x \leq x$ for all $x \in X$) and transitive ($x \leq y$ and $y \leq z$ implies $x \leq z$). It is a *partial order* if it is moreover anti-symmetric ($x \leq y$ and $y \leq x$ implies $x = y$). The pair $(X, \leq)$ is called a *partially ordered set* if $\leq$ is a partial order.

A partial order $\leq$ is a *linear order* or *total order* if $x \leq y$ or $y \leq x$ for all $x, y \in X$. If $X = \{x_1, \ldots, x_n\}$ and $x_1 < x_2 < \cdots < x_n$, we occasionally refer to the linear order $\leq$ by $x_1, \ldots, x_n$ or $x_1 < \cdots < x_n$. A linear order $\preceq$ is called a *linear extension* of a partial order $\leq$ if $x \leq y$ implies $x \preceq y$.

In a partially ordered set $(X, \leq)$, a *lower ideal* is a subset Y of X such that if $y \in Y$ and $z \leq y$, then $z \in Y$. Similarly, an *upper ideal* is a subset Y of X such that if $y \in Y$ and $z \geq y$, then $z \in Y$. Alternatively, Y is called *down-monotone* if Y is a lower ideal, and *up-monotone* if Y is an upper ideal.

If $(X, \leq)$ is a linearly ordered set, then the *lexicographic order* $\preceq$ on $\bigcup_{k \geq 0} X^k$ is defined by:

(2.9) $(v_1, \ldots, v_t) \prec (u_1, \ldots, u_s) \iff$ the smallest i with $v_i \neq u_i$ satisfies $v_i < u_i$,

where we set v_i :=void if $i > t$, u_i :=void if $i > s$, and void$< x$ for all $x \in X$.

2.3. Numbers

$\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ denote the sets of integers, rational numbers, and real numbers, respectively. The subscript $+$ restricts the sets to the nonnegative numbers:

(2.10) $\mathbb{Z}_+ := \{x \in \mathbb{Z} \mid x \geq 0\}$, $\mathbb{Q}_+ := \{x \in \mathbb{Q} \mid x \geq 0\}$,
$\mathbb{R}_+ := \{x \in \mathbb{R} \mid x \geq 0\}$.

Further we denote for any $x \in \mathbb{R}$:

(2.11) $\lfloor x \rfloor :=$ largest integer y satisfying $y \leq x$,
$\lceil x \rceil :=$ smallest integer y satisfying $y \geq x$.

2.4. Vectors, matrices, and functions

All vectors are assumed to be *column* vectors. The *components* or *entries* of a vector $x = (x_1, \ldots, x_n)^{\mathsf{T}}$ are $x_1, \ldots, x_n$. The *support* of x is the set of indices i with $x_i \neq 0$. The *size* of a vector x is the sum of its components.

A $0,1$ *vector*, or a $\{0,1\}$*-valued vector*, or a *simple vector*, is a vector with all entries in $\{0,1\}$. An *integer vector* is a vector with all entries integer.

We identify the concept of a *function* $x : V \to \mathbb{R}$ with that of a *vector* x in $\mathbb{R}^V$. Its components are denoted equivalently by $x(v)$ or x_v. An *integer function* is an integer-valued function.

For any $U \subseteq V$, the *incidence vector* of U (in $\mathbb{R}^V$) is the vector χ^U defined by:

$$(2.12) \qquad \chi^U(s) := \begin{cases} 1 & \text{if } s \in U, \\ 0 & \text{if } s \notin U. \end{cases}$$

For any $u \in V$ we set

$$(2.13) \qquad \chi^u := \chi^{\{u\}}.$$

This is the uth *unit base vector*. Given a vector space $\mathbb{R}^V$ for some set V, the all-one vector is denoted by $\mathbf{1}_V$ or just by $\mathbf{1}$, and the all-zero vector by $\mathbf{0}_V$ or just by $\mathbf{0}$. Similarly, $\mathbf{2}_V$ or $\mathbf{2}$ is the all-two vector. We use $\boldsymbol{\infty}$ for the all-∞ vector.

If $a = (a_1, \ldots, a_n)^\mathsf{T}$ and $b = (b_1, \ldots, b_n)^\mathsf{T}$ are vectors, we write $a \leq b$ if $a_i \leq b_i$ for $i = 1, \ldots, n$, and $a < b$ if $a_i < b_i$ for $i = 1, \ldots, n$.

If A is a matrix and x, b, y, and c are vectors, then when using notation like

$$(2.14) \qquad Ax = b,\ Ax \leq b,\ y^\mathsf{T}A = c^\mathsf{T},\ c^\mathsf{T}x,$$

we often implicitly assume compatibility of dimensions.

For any vector $x = (x_1, \ldots, x_n)^\mathsf{T}$:

$$(2.15) \qquad \|x\|_1 := |x_1| + \cdots + |x_n| \text{ and } \|x\|_\infty := \max\{|x_1|, \ldots, |x_n|\}.$$

A *hyperplane* in $\mathbb{R}^n$ is a set H with $H = \{x \in \mathbb{R}^n \mid c^\mathsf{T}x = \delta\}$ for some $c \in \mathbb{R}^n$ with $c \neq \mathbf{0}$ and some $\delta \in \mathbb{R}$.

If U and V are sets, then a $U \times V$ *matrix* is a matrix whose rows are indexed by the elements of U and whose columns are indexed by the elements of V. Generally, when using this terminology, the order of the rows or columns is irrelevant. For a $U \times V$ matrix M and $u \in U$, $v \in V$, the entry in position u, v is denoted by $M_{u,v}$. The all-one $U \times V$ matrix is denoted by $J_{U \times V}$, or just by J.

The *tensor product* of vectors $x \in \mathbb{R}^U$ and $y \in \mathbb{R}^V$ is the vector $x \circ y$ in $\mathbb{R}^{U \times V}$ defined by:

$$(2.16) \qquad (x \circ y)_{(u,v)} := x_u y_v$$

for $u \in U$ and $v \in V$.

The *tensor product* of a $W \times X$ matrix M and a $Y \times Z$ matrix N (where W, X, Y, Z are sets), is the $(W \times Y) \times (X \times Z)$ matrix $M \circ N$ defined by

$$(2.17) \qquad (M \circ N)_{(w,y),(x,z)} := M_{w,x} N_{y,z}$$

for $w \in W$, $x \in X$, $y \in Y$, $z \in Z$.

The $\mathcal{C} \times V$ *incidence matrix* of a collection or family $\mathcal{C}$ of subsets of a set V is the $\mathcal{C} \times V$ matrix M with $M_{C,v} := 1$ if $v \in C$ and $M_{C,v} := 0$ if $v \notin C$ (for $C \in \mathcal{C}$, $v \in V$). Similarly, the $V \times \mathcal{C}$ incidence matrix is the transpose of this matrix.

For any function $w : V \to \mathbb{R}$ and any $U \subseteq V$, we denote

$$(2.18) \qquad w(U) := \sum_{v \in U} w(v).$$

If U is a family, we take multiplicities into account (so if v occurs k times in U, $w(v)$ occurs k times in sum (2.18)).

If w is introduced as a 'weight function', then $w(v)$ is called the *weight* of v, and for any $U \subseteq V$, $w(U)$ is called the *weight* of U. Moreover, for any $x : V \to \mathbb{R}$, we call $w^{\mathsf{T}}x$ the *weight* of x. If confusion may arise, we call $w(U)$ and $w^{\mathsf{T}}x$ the *w-weight* of U and x, respectively.

The adjective 'weight' to 'function' has no mathematical meaning, and implies no restriction, but is just introduced to enable referring to $w(v)$ or $w(U)$ as the weight of v or U. Similarly, for 'length function', 'cost function', 'profit function', 'capacity function', 'demand function', etc., leading to the *length*, *cost*, *profit*, *capacity*, *demand*, etc. of elements or of subsets. Obviously, *shortest* and *longest* are synonyms for 'minimum-length' and 'maximum-length'.

A *permutation matrix* is a square $\{0, 1\}$ matrix, with exactly one 1 in each row and in each column.

Vectors $x_1, \ldots, x_k$ are called *affinely independent* if there do not exist $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$ such that $\lambda_1 x_1 + \cdots + \lambda_k x_k = \mathbf{0}$ and $\lambda_1 + \cdots + \lambda_k = 0$ and such that the λ_i are not all equal to 0.

Vectors $x_1, \ldots, x_k$ are called *linearly independent* if there do not exist $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$ such that $\lambda_1 x_1 + \cdots + \lambda_k x_k = \mathbf{0}$ and such that the λ_i are not all equal to 0. The linear hull of a set X is denoted by lin.hull$X$ or lin.hull(X).

If X and Y are subsets of a linear space L over a field $\mathbb{F}$, $z \in L$, and $\lambda \in \mathbb{F}$, then

(2.19) $z + X := \{z + x \mid x \in X\}$, $X + Y := \{x + y \mid x \in X, y \in Y\}$, and $\lambda X = \{\lambda x \mid x \in X\}$.

If X and Y are subspaces of L, then

(2.20) $X/Y := \{x + Y \mid x \in X\}$

is a *quotient space*, which is again a linear space, with addition and scalar multiplication given by (2.19). The dimension of X/Y is equal to $\dim(X) - \dim(X \cap Y)$.

A function $f : X \to Y$ is called an *injection* or an *injective function* if it is one-to-one: if $x, x' \in X$ and $x \neq x'$, then $f(x) \neq f(x')$. The function f is a *surjection* if it is onto: for each $y \in Y$ there is an $x \in X$ with $f(x) = y$. It is a *bijection* if it is both an injection and a surjection.

For a vector $x = (x_1, \ldots, x_n)^{\mathsf{T}} \in \mathbb{R}^n$, we denote

(2.21) $\lfloor x \rfloor := (\lfloor x_1 \rfloor, \ldots, \lfloor x_n \rfloor)^{\mathsf{T}}$ and $\lceil x \rceil := (\lceil x_1 \rceil, \ldots, \lceil x_n \rceil)^{\mathsf{T}}$.

If $f, g : X \to \mathbb{R}$ are functions, we say that $f(x)$ is $O(g(x))$, in notation

(2.22) $f(x) = O(g(x))$ or $O(f(x)) = O(g(x))$,

if there exists a constant $c \geq 0$ with $f(x) \leq cg(x) + c$ for all $x \in X$. Hence the relation $=$ given in (2.22) is transitive, but not symmetric. We put

(2.23) $g(x) = \Omega(f(x))$

if $f(x) = O(g(x))$.

2.5. Maxima, minima, and infinity

In this book, when speaking of a maximum or minimum, we often implicitly assume that the optimum is finite. If the optimum is not finite, consistency in min-max relations usually can be obtained by setting a minimum over the empty set to $+\infty$, a maximum over a set without upper bound to $+\infty$, a maximum over the empty set to 0 or $-\infty$ (depending on what is the universe), and a minimum over a set without lower bound to $-\infty$. This usually leads to trivial, or earlier proved, statements.

When we speak of making a value infinite, usually large enough will suffice.

If we consider maximizing a function $f(x)$ over $x \in X$, we call any $x \in X$ a *feasible solution*, and any $x \in X$ maximizing $f(x)$ an *optimum solution*. Similarly for minimizing.

2.6. Fekete's lemma

We will need the following result called Fekete's lemma, due to Pólya and Szegő [1925] (motivated by a special case proved by Fekete [1923]):

Theorem 2.2 (Fekete's lemma)**.** *Let $a_1, a_2, \ldots$ be a sequence of reals such that $a_{n+m} \geq a_n + a_m$ for all positive $n, m \in \mathbb{Z}$. Then*

$$\lim_{n\to\infty} \frac{a_n}{n} = \sup_n \frac{a_n}{n}. \tag{2.24}$$

Proof. For all $i, j, k \geq 1$ we have $a_{jk+i} \geq ja_k + a_i$, by the inequality prescribed in the theorem. Hence for all fixed $i, k \geq 1$ we have

$$\liminf_{j\to\infty} \frac{a_{jk+i}}{jk+i} \geq \liminf_{j\to\infty} \frac{ja_k + a_i}{jk+i} = \liminf_{j\to\infty} \left(\frac{a_k}{k}\frac{jk}{jk+i} + \frac{a_i}{jk+i}\right) = \frac{a_k}{k}. \tag{2.25}$$

As this is true for each i, we have for each fixed $k \geq 1$:

$$\liminf_{n\to\infty} \frac{a_n}{n} = \inf_{i=1,\ldots,k} \liminf_{j\to\infty} \frac{a_{jk+i}}{jk+i} \geq \frac{a_k}{k}. \tag{2.26}$$

So

$$\liminf_{n\to\infty} \frac{a_n}{n} \geq \sup_k \frac{a_k}{k}, \tag{2.27}$$

implying (2.24). ∎

We sometimes use the multiplicative version of Fekete's lemma:

Corollary 2.2a. *Let $a_1, a_2, \ldots$ be a sequence of positive reals such that $a_{n+m} \geq a_n a_m$ for all positive $n, m \in \mathbb{Z}$. Then*

$$\lim_{n\to\infty} \sqrt[n]{a_n} = \sup_n \sqrt[n]{a_n}. \tag{2.28}$$

Proof. Directly from Theorem 2.2 applied to the sequence $\log a_1, \log a_2, \ldots$. ∎

Chapter 3

Preliminaries on graphs

This chapter is not meant as a rush course in graph theory, but rather as a reference guide and to settle notation and terminology.
To promote readability of the book, nonstandard notation and terminology will be, besides below in this chapter, also explained on the spot in later chapters.

3.1. Undirected graphs

A *graph* or *undirected graph* is a pair $G = (V, E)$, where V is a finite set and E is a family of *unordered* pairs from V. The elements of V are called the *vertices*, sometimes the *nodes* or the *points*. The elements of E are called the *edges*, sometimes the *lines*. We use the following shorthand notation for edges:

(3.1) $\qquad uv := \{u, v\}.$

We denote

(3.2) $\qquad VG :=$ set of vertices of G,
$\qquad\qquad EG :=$ family of edges of G.

In running time estimates of algorithms, we denote:

(3.3) $\qquad n := |VG|$ and $m := |EG|$.

In the definition of graph we use the term 'family' rather than 'set', to indicate that the same pair of vertices may occur several times in E. A pair occurring more than once in E is called a *multiple* edge, and the number of times it occurs is called its *multiplicity*. Two edges are called *parallel* if they are represented by the same pair of vertices. A *parallel class* is a maximal set of pairwise parallel edges.

So distinct edges may be represented in E by the same pair of vertices. Nevertheless, we will often speak of 'an edge uv' or even of 'the edge uv', where 'an edge of type uv' would be more correct.

Also *loops* are allowed: edges that are families of the form $\{v, v\}$. Graphs without loops and multiple edges are called *simple*, and graphs without loops are called *loopless*. A vertex v is called a *loopless vertex* if $\{v, v\}$ is not a loop.

An edge uv is said to *connect* u and v. The vertices u and v are called the *ends* of the edge uv. If there exists an edge connecting vertices u and v, then u and v are called *adjacent* or *connected*, and v is called a *neighbour* of u. The edge uv is said to be *incident* with, or to *meet*, or to *cover*, the vertices u and v, and conversely. The edges e and f are said to be *incident*, or to *meet*, or to *intersect*, if they have a vertex in common. Otherwise, they are called *disjoint*.

If $U \subseteq V$ and both ends of an edge e belong to U, then we say that U *spans* e. If at least one end of e belongs to U, then U is said to be *incident with* e. An edge connecting a vertex in a set S and a vertex in a set T is said to *connect* S and T. A set F of edges is said to *cover* a vertex v if v is covered by at least one edge in F, and to *miss* v otherwise.

For a vertex v, we denote:

(3.4) $$\delta_G(v) := \delta_E(v) := \delta(v) := \text{family of edges incident with } v,$$
$$N_G(v) := N_E(v) := N(v) := \text{set of neighbours of } v.$$

Here and below, notation with the subscript deleted is used if the graph is clear from the context. We speak in the definition of $\delta(v)$ of the *family* of edges incident with v, since any loop at v occurs twice in $\delta(v)$.

The *degree* $\deg_G(v)$ of a vertex v is the number of edges incident with v. In notation,

(3.5) $$\deg_G(v) := \deg_E(v) := \deg(v) := |\delta_G(v)|.$$

A vertex of degree 0 is called *isolated*, and a vertex of degree 1 an *end vertex*. A vertex of degree k is called *k-valent*. So isolated vertices are loopless.

We denote

(3.6) $$\Delta(G) := \text{maximum degree of the vertices of } G,$$
$$\delta(G) := \text{minimum degree of the vertices of } G.$$

$\Delta(G)$ and $\delta(G)$ are called the *maximum degree* and *minimum degree* of G, respectively.

If $\Delta(G) = \delta(G)$, that is, if all degrees are equal, G is called *regular*. If all degrees are equal to k, the graph is called *k-regular*. A 3-regular graph is also called a *cubic graph*.

If $G = (V, E)$ and $G' = (V', E')$ are graphs, we denote by $G + G'$ the graph

(3.7) $$G + G' := (V \cup V', E \cup E')$$

where $E \cup E'$ is the union of E and E' as families (taking multiplicities into account).

Complementary, complete, and line graph

The *complementary graph* or *complement* of a graph $G = (V, E)$ is the simple graph with vertex set V and edges all pairs of distinct vertices that are nonadjacent in G. In notation,

(3.8) $\overline{G}$:= the complementary graph of G.

So if G is simple, then $\overline{\overline{G}} = G$.

A graph G is called *complete* if G is simple and any two distinct vertices are adjacent. In notation,

(3.9) K_n := complete graph with n vertices.

As K_n is unique up to isomorphism, we often speak of *the* complete graph on n vertices.

The *line graph* of a graph $G = (V, E)$ is the simple graph with vertex set E, where two elements of E are adjacent if and only if they meet. In notation,

(3.10) $L(G)$:= the line graph of G.

Subgraphs

A graph $G' = (V', E')$ is called a *subgraph* of a graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. If H is a subgraph of G, we say that G *contains* H. If $G' \neq G$, then G' is called a *proper subgraph* of G. If $V' = V$, then G' is called a *spanning subgraph* of G. If E' consists of all edges of G spanned by V', G' is called an *induced subgraph*, or the *subgraph induced by* V'. In notation,

(3.11) $G[V']$:= subgraph of G induced by V',
$E[V']$:= family of edges spanned by V'.

So $G[V'] = (V', E[V'])$. We further denote for any graph $G = (V, E)$ and for any vertex v, any subset U of V, any edge e, and any subset F of E,

(3.12) $G - v := G[V \setminus \{v\}]$, $G - U := G[V \setminus U]$, $G - e := (V, E \setminus \{e\})$,
$G - F := (V, E \setminus F)$.

We say that these graphs arise from G by *deleting* v, U, e, or F. (We realize that, since an edge e is a set of two vertices, the notation $G - e$ might be ambiguous (if we would consider $U := e$). We expect, however, that the appropriate interpretation will be clear from the context.)

Two subgraphs of G are called *edge-disjoint* if they have no edge in common, and *vertex-disjoint* or *disjoint*, if they have no vertex in common.

In many cases we deal with graphs *up to isomorphism.* For instance, if G and H are graphs, we say that a subgraph G' of G is *an H subgraph* if G' is isomorphic to H.

Paths and circuits

A *walk* in an undirected graph $G = (V, E)$ is a sequence

(3.13) $\quad P = (v_0, e_1, v_1, \ldots, e_k, v_k),$

where $k \geq 0$, $v_0, v_1, \ldots, v_k$ are vertices, and e_i is an edge connecting v_{i-1} and v_i (for $i = 1, \ldots, k$). If $v_0, v_1, \ldots, v_k$ are all distinct, the walk is called a *path*. (Hence $e_1, \ldots, e_k$ are distinct.)

The vertex v_0 is called the *starting vertex* or *first vertex* of P and the vertex v_k the *end vertex* or *last vertex* of P. Sometimes, both v_0 and v_k are called the *end vertices*, or just the *ends* of P. Similarly, edge e_1 is called the *starting edge* or *first edge* of P, and edge e_k the *end edge* or *last edge* of P. Sometimes, both e_1 and e_k are called the *end edges*.

The walk P is said to *connect* v_0 and v_k, to *run from* v_0 *to* v_k (or *between* v_0 *and* v_k), and to *traverse* $v_0, e_1, v_1, \ldots, e_k, v_k$. The vertices $v_1, \ldots, v_{k-1}$ are called the *internal vertices* of P. For $s, t \in V$, the walk P is called an $s - t$ *walk* if it runs from s to t, and for $S, T \subseteq V$, it is called an $S - T$ *walk* if it runs from a vertex in S to a vertex in T. Similarly, $s - T$ *walks* and $S - t$ *walks* run form s to a vertex in T and from a vertex in S to t, respectively.

The number k is called the *length* of P. (We deviate from this in case a function $l : E \to \mathbb{R}$ has been introduced as a length function. Then the *length* of P is equal to $l(e_1) + \cdots + l(e_k)$.) A walk is called *odd* (*even*, respectively) if its length is odd (even, respectively).

The minimum length of a path connecting u and v is called the *distance* of u and v. The maximum distance over all vertices u, v of G is called the *diameter* of G.

The *reverse walk* P^{-1} of P is the walk obtained from (3.13) by reversing the order of the elements:

(3.14) $\quad P^{-1} := (v_k, e_k, v_{k-1}, \ldots, e_1, v_0).$

If $P = (v_0, e_1, v_1, \ldots, e_k, v_k)$ and $Q = (u_0, f_1, u_1, \ldots, f_l, u_l)$ are walks satisfying $u_0 = v_k$, the *concatenation* PQ of P and Q is the walk

(3.15) $\quad PQ := (v_0, e_1, v_1, \ldots, e_k, v_k, f_1, u_1, \ldots, f_l, u_l).$

For any walk P, we denote by VP and EP the families of vertices and edges, respectively, occurring in P:

(3.16) $\quad VP := \{v_0, v_1, \ldots, v_k\}$ and $EP := \{e_1, \ldots, e_k\}$.

A *chord* of P is an edge of G that is not in EP and that connects two vertices of P. The path P is called *chordless* if P has no chords.

If no confusion may arise, we sometimes identify the walk P with the subgraph (VP, EP) of G, or with the set VP of vertices in P, or with the family EP of edges in P. If the graph is simple or if the edges traversed are irrelevant, we indicate the walk just by the sequence of vertices traversed:

(3.17) $\quad P = (v_0, v_1, \ldots, v_k)$ or $P = v_0, v_1, \ldots, v_k$.

A simple path may be identified by the sequence of edges:

(3.18) $P = (e_1, \ldots, e_k)$ or $P = e_1, \ldots, e_k$.

We denote

(3.19) $P_n :=$ a path with n vertices,

usually considered as the *graph* (VP_n, EP_n). This graph is unique up to isomorphism.

Two walks P and Q are called *vertex-disjoint* or *disjoint* if VP and VQ are disjoint, *internally vertex-disjoint* or *internally disjoint* if the set of internal vertices of P is disjoint from the set of internal vertices of Q, and *edge-disjoint* if EP and EQ are disjoint.

The walk P in (3.13) is called *closed* if $v_k = v_0$. It is called a *circuit* if $v_k = v_0$, $k \geq 1$, $v_1, \ldots, v_k$ are all distinct, and $e_1, \ldots, e_k$ are all distinct.

The circuit is also called a *k-circuit*. If $k = 1$, then e_1 must be a loop, and if $k = 2$, e_1 and e_2 are (distinct) parallel edges. If $k = 3$, the circuit is sometimes called a *triangle*.

The above definition of chord of a walk implies that an edge e of G is a *chord* of a circuit C if e connects two vertices in VC but does not belong to EC. A *chordless circuit* is a circuit without chords.

We denote

(3.20) $C_n :=$ a circuit with n edges,

usually considered as the *graph* (VC_n, EC_n). Again, this graph is unique up to isomorphism.

For any graph $G = (V, E)$, a subset F of E is called a *cycle* if each degree of the subgraph (V, F) is even. One may check that for any $F \subseteq E$:

(3.21) F is a cycle $\iff$ F is the symmetric difference of the edge sets of a number of circuits.

Connectivity and components

A graph $G = (V, E)$ is *connected* if for any two vertices u and v there is a path connecting u and v. A maximal connected nonempty subgraph of G is called a *connected component*, or just a *component*, of G. Here 'maximal' is taken with respect to taking subgraphs. Each component is an induced subgraph, and each vertex and each edge of G belong to exactly one component.

We often identify a component K with the set VK of its vertices. Then the components are precisely the equivalence classes of the equivalence relation $\sim$ on V defined by: $u \sim v \iff$ there exists a path connecting u and v.

A component is called *odd* (*even*) if it has an odd (even) number of vertices.

Cuts

Let $G = (V, E)$ be a graph. For any $U \subseteq V$, we denote

(3.22) $\delta_G(U) := \delta_E(U) := \delta(U) :=$ set of edges of G connecting U and $V \setminus U$.

A subset F of E is called a *cut*, if $F = \delta(U)$ for some $U \subseteq V$. In particular, $\emptyset$ is a cut. If $\emptyset \neq U \neq V$, then $\delta(U)$ is called a *nontrivial cut*. (So $\emptyset$ is a nontrivial cut if and only if G is disconnected.) It is important to observe that for any two sets $T, U \subseteq V$:

(3.23) $\delta(T) \triangle \delta(U) = \delta(T \triangle U)$.

Hence the collection of cuts is closed under taking symmetric differences.

If $s \in U$ and $t \notin U$, then $\delta(U)$ is called an *$s-t$ cut*. If $S \subseteq U$ and $T \subseteq V \setminus U$, $\delta(U)$ is called an *$S-T$ cut*. An edge-cut of size k is called a *k-cut*.

A subset F of E is called a *disconnecting edge set* if $G-F$ is disconnected. For $s, t \in V$, if F intersects each $s-t$ path, then F is said to *disconnect* or to *separate* s and t, or to be *$s-t$ disconnecting* or *$s-t$ separating*. For $S, T \subseteq V$, if F intersects each $S-T$ path, then F is said to *disconnect* or to *separate* S and T, or to be *$S-T$ disconnecting* or *$S-T$ separating*.

One may easily check that for all $s, t \in V$:

(3.24) each $s-t$ cut is $s-t$ disconnecting; each inclusionwise minimal $s-t$ disconnecting edge set is an $s-t$ cut.

An edge e of G is called a *bridge* if $\{e\}$ is a cut. A graph having no bridges is called *bridgeless*.

For any subset U of V we denote

(3.25) $d_G(U) := d_E(U) := d(U) := |\delta(U)|$.

Moreover, for subsets U, W of V:

(3.26) $E[U, W] := \{e \in E \mid \exists u \in U, w \in W : e = uw\}$.

The following is straightforward and very useful:

Theorem 3.1. *For all $U, W \subseteq V$:*

(3.27) $d(U) + d(W) = d(U \cap W) + d(U \cup W) + 2|E[U \setminus W, W \setminus U]|$.

Proof. Directly by counting edges. ∎

This in particular gives:

Corollary 3.1a. *For all $U, W \subseteq V$:*

(3.28) $d(U) + d(W) \geq d(U \cap W) + d(U \cup W)$.

Proof. Directly from Theorem 3.1. ∎

A cut of the form $\delta(v)$ for some vertex v is called a *star*.

Neighbours and vertex-cuts

Let $G = (V, E)$ be a graph. For any $U \subseteq V$, we call a vertex v a *neighbour* of U if $v \notin U$ and v has a neighbour in U. We denote

(3.29) $\qquad N_G(U) := N_E(U) := N(U) :=$ set of neighbours of U.

We further denote

(3.30) $\qquad N^2(v) := N(N(v)) \setminus \{v\}$.

A subset U of V is called a *disconnecting vertex set*, or a *vertex-cut*, if $G - U$ is disconnected. A vertex-cut of size k is called a *k-vertex-cut*. A *cut vertex* is a vertex v of G for which $G - v$ has more components than G has.

For $s, t \in V$, if U intersects each $s - t$ path, then U is said to *disconnect* s and t, or called *$s - t$ disconnecting*. If moreover $s, t \notin U$, then U is said to *separate* s and t, or called *$s - t$ separating*, or an *$s - t$ vertex-cut*. It can be shown that if U is an inclusionwise minimal $s - t$ vertex-cut, then $U = N(K)$ for the component K of $G - U$ that contains s.

For $S, T \subseteq V$, if U intersects each $S - T$ path, then U is said to *disconnect* S and T, or called *$S - T$ disconnecting*. If moreover U is disjoint from $S \cup T$, then U is said to *separate* S and T, or called *$S - T$ separating* or an *$S - T$ vertex-cut*.

A pair of subgraphs (V_1, E_1), (V_2, E_2) of a graph $G = (V, E)$ is called a *separation* if $V_1 \cup V_2 = V$ and $E_1 \cup E_2 = E$. So G has no edge connecting $V_1 \setminus V_2$ and $V_2 \setminus V_1$. Therefore, if these sets are nonempty, $V_1 \cap V_2$ is a vertex-cut of G.

Trees and forests

A graph is called a *forest* if it has no circuits. For any forest (V, E),

(3.31) $\qquad |E| = |V| - \kappa$,

where κ is the number of components of (V, F). A *tree* is a connected forest. So for any tree (V, E),

(3.32) $\qquad |E| = |V| - 1$.

Any forest with at least one edge has an end vertex. A connected subgraph of a tree T is called a *subtree* of T.

The notions of forest and tree extend to subsets of edges of a graph $G = (V, E)$ as follows. A subset F of E is called a *forest* if (V, F) is a forest, and a *spanning tree* if (V, F) is a tree. Then for any graph $G = (V, E)$:

(3.33) $\qquad G$ has a spanning tree $\iff$ G is connected.

For any connected graph $G = (V, E)$ and any $F \subseteq E$:

(3.34) $\qquad F$ is a spanning tree $\iff$ F is an inclusionwise maximal forest $\iff$ F is an inclusionwise minimal edge set with (V, F) connected.

Cliques, stable sets, matchings, vertex covers, edge covers

Let $G = (V, E)$ be a graph. A subset C of V is called a *clique* if any two vertices in V are adjacent, a *stable set* if any two vertices in C are nonadjacent, and a *vertex cover* if C intersects each edge of G.

A subset M of E is called a *matching* if any two edges in M are disjoint, an *edge cover* if each vertex of G is covered by at least one edge in M, and a *perfect matching* if it is both a matching and an edge cover. So a perfect matching M satisfies $|M| = \frac{1}{2}|V|$.

We denote and define:

(3.35) $\omega(G) :=$ *clique number* of $G :=$ maximum size of a clique in G,
$\alpha(G) :=$ *stable set number* of $G :=$ maximum size of a stable set in G,
$\tau(G) :=$ *vertex cover number* of $G :=$ minimum size of a vertex cover in G,
$\nu(G) :=$ *matching number* of $G :=$ maximum size of a matching in G,
$\rho(G) :=$ *edge cover number* of $G :=$ minimum size of an edge cover in G.

(We will recall this notation where used.)

Given a matching M in a graph $G = (V, E)$, we will say that a vertex u is *matched to* a vertex v, or u is the *mate* of v, if $uv \in M$. A subset U of V is called *matchable* if the subgraph $G[U]$ of G induced by U has a perfect matching.

Colouring

A *vertex-colouring*, or just a *colouring*, is a partition of V into stable sets. We sometimes consider a colouring as a function $\phi : V \to \{1, \ldots, k\}$ such that $\phi^{-1}(i)$ is a stable set for each $i = 1, \ldots, k$.

Each of the stable sets in a colouring is called a *colour* of the colouring. The *vertex-colouring number*, or just the *colouring number*, is the minimum number of colours in a vertex-colouring. In notation,

(3.36) $\chi(G) :=$ vertex-colouring number of G.

A graph G is called *k-colourable*, or *k-vertex-colourable*, if $\chi(G) \leq k$, and *k-chromatic* if $\chi(G) = k$. A vertex-colouring is called a *minimum vertex-colouring*, or a *minimum colouring*, if it uses the minimum number of colours.

Similar terminology holds for edge-colouring. An *edge-colouring* is a partition of E into matchings. Each of these matchings is called a *colour* of the edge-colouring. An edge-colouring can also be described by a function $\phi : E \to \{1, \ldots, k\}$ such that $\phi^{-1}(i)$ is a matching for each $i = 1, \ldots, k$.

The *edge-colouring number* is the minimum number of colours in an edge-colouring. In notation, (3.37) $\chi'(G) :=$ edge-colouring number of G.

So $\chi'(G) = \chi(L(G))$.

A graph G is called *k-edge-colourable* if $\chi'(G) \leq k$, and *k-edge-chromatic* if $\chi'(G) = k$. An edge-colouring is called a *minimum edge-colouring* if it uses the minimum number of colours.

Bipartite graphs

A graph $G = (V, E)$ is called *bipartite* if $\chi(G) \leq 2$. Equivalently, G is bipartite if V can be partitioned into two sets U and W such that each edge of G connects U and W. We call the sets U and W the *colour classes* of G (although they generally need not be unique).

Bipartite graphs are characterized by:

(3.38) G is bipartite $\iff$ each circuit of G has even length.

A graph $G = (V, E)$ is called a *complete bipartite graph* if G is simple and V can be partitioned into sets U and W such that E consists of all pairs $\{u, w\}$ with $u \in U$ and $w \in W$. If $|U| = m$ and $|W| = n$, the graph is denoted by $K_{m,n}$:

(3.39) $K_{m,n} :=$ the complete bipartite graph with colour classes of size m and n.

The graphs $K_{1,n}$ are called *stars* or (when $n \geq 3$) *claws*.

Hamiltonian and Eulerian graphs

A *Hamiltonian circuit* in a graph G is a circuit C satisfying $VC = VG$. A graph is *Hamiltonian* if it has a Hamiltonian circuit. A *Hamiltonian path* is a path P with $VP = VG$.

A walk P is called *Eulerian* if each edge of G is traversed exactly once by P. A graph G is called *Eulerian* if it has a closed Eulerian walk. The following is usually attributed to Euler [1736] (although he only proved the 'only if' part):

(3.40) a graph $G = (V, E)$ without isolated vertices is Eulerian if and only if G is connected and all degrees of G are even.

Sometimes, we call a graph Eulerian if all degrees are even, ignoring connectivity. This will be clear from the context.

Contraction and minors

Let $G = (V, E)$ be a graph and let $e = uv \in E$. *Contracting* e means deleting e and identifying u and v. We denote (for $F \subseteq E$):

(3.41) $G/e :=$ graph obtained from G by contracting e,
$G/F :=$ graph obtained from G by contracting all edges in F.

The *image* of a vertex v of G in G/F is the vertex of G/F to which v is contracted.

A graph H is called a *minor* of a graph G if H arises from G by a series of deletions and contractions of edges and deletions of vertices. A minor H of G is called a *proper minor* if $H \neq G$. If G and H are graphs, we say that a minor G' of G is *an H minor* of G if G' is isomorphic to H.

Related is the following contraction. Let $G = (V, E)$ be a graph and let $S \subseteq V$. The graph G/S (obtained by *contracting* S) is obtained by identifying all vertices in S to one new vertex, called S, deleting all edges contained in S, and redefining any edge uv with $u \in S$ and $v \notin S$ to Sv.

Homeomorphic graphs

A graph G is called a *subdivision* of a graph H if G arises from H by replacing edges by paths of length at least 1. So it arises from H by iteratively choosing an edge uv, introducing a new vertex w, deleting edge uv, and adding edges uw and wv. If G is a subdivision of H, we call G an *H-subdivision*.

Two graphs G and G' are called *homeomorphic* if there exists a graph H such that both G and G' are subdivisions of H. The graph G is called a *homeomorph* of G' if G and G' are homeomorphic.

Homeomorphism can be described topologically. For any graph $G = (V, E)$, the *topological graph* $|G|$ associated with G is the topological space consisting of V and for each edge e of G a curve $|e|$ connecting the ends of e, such that for any two edges e, f one has $|e| \cap |f| = e \cap f$. Then

(3.42) G and H are homeomorphic graphs $\iff$ $|G|$ and $|H|$ are homeomorphic topological spaces.

Planarity

An *embedding* of a graph G in a topological space S is an embedding (continuous injection) of the topological graph $|G|$ in S. A graph G is called *planar* if it has an embedding in the plane $\mathbb{R}^2$.

Often, when dealing with a planar graph G, we assume that it *is* embedded in the plane $\mathbb{R}^2$. The topological components of $\mathbb{R}^2 \setminus |G|$ are called the *faces* of G. A vertex or edge is said to be *incident* with a face F if it is contained in the boundary of F. Two faces are called *adjacent* if they are incident with some common edge.

There is a unique *unbounded face*, all other faces are *bounded*. The boundary of the unbounded face is part of $|G|$, and is called the *outer boundary* of G.

Euler's formula states that any connected planar graph $G = (V, E)$, with face collection $\mathcal{F}$, satisfies:

(3.43) $$|V| + |\mathcal{F}| = |E| + 2.$$

Kuratowski [1930] found the following characterization of planar graphs:

Theorem 3.2 (Kuratowski's theorem)**.** *A graph G is planar $\iff$ no subgraph of G is homeomorphic to K_5 or to $K_{3,3}$.*

(See Thomassen [1981b] for three short proofs, and for history and references to other proofs.)

As Wagner [1937a] noticed, the following is an immediate consequence of Kuratowski's theorem (since planarity is maintained under taking minors, and since any graph without K_5 minor has no subgraph homeomorphic to K_5):

(3.44) A graph G is planar $\iff$ G has no K_5 or $K_{3,3}$ minor.

(In turn, with a little more work, this equivalence can be shown to imply Kuratowski's theorem.)

The *four-colour theorem* of Appel and Haken [1977] and Appel, Haken, and Koch [1977] states that each loopless planar graph is 4-colourable. (Robertson, Sanders, Seymour, and Thomas [1997] gave a shorter proof.)

Tait [1878b] showed that the four-colour theorem is equivalent to: each cubic bridgeless planar graph is 3-edge-colourable. Petersen [1898] gave the example of the now-called *Petersen graph* (Figure 3.1), to show that not every bridgeless cubic graph is 3-edge-colourable. (This graph was also given by Kempe [1886], for a different purpose.)

Wagner's theorem

We will use occasionally an extension of Kuratowski's theorem, proved by Wagner [1937a]. For this we need the notion of a k-sum of graphs.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs and let $k := |V_1 \cap V_2|$. Suppose that $(V_1 \cap V_2, E_1 \cap E_2)$ is a (simple) complete graph. Then the graph

(3.45) $$(V_1 \cup V_2, E_1 \triangle E_2)$$

is called a *k-sum* of G_1 and G_2. We allow multiple edges, so the k-sum might keep edges spanned by $V_1 \cap V_2$.

To formulate Wagner's theorem, we also need the graph denoted by V_8, given in Figure 3.2.

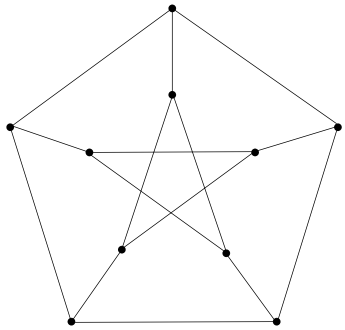


Figure 3.1
The Petersen graph

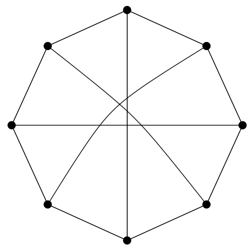


Figure 3.2
V_8

Theorem 3.3 (Wagner's theorem)**.** *A graph G has no K_5 minor $\iff$ G can be obtained from planar graphs and from copies of V_8 by taking* 1-, 2-, *and* 3-*sums.*

As Wagner [1937a] pointed out, this theorem implies that the four-colour theorem is equivalent to: each graph without K_5 minor is 4-colourable. This follows from the fact that k-colourability is maintained under taking k'-sums for all $k' \leq k$.

The dual graph

The *dual* G^* of an embedded planar graph $G = (V, E)$ is the graph having as vertex set the set of faces of G and having, for each $e \in E$, an edge e^* connecting the two faces incident with e. Then G^* again is planar, and $(G^*)^*$ is isomorphic to G, if G is connected. For any $C \subseteq E$, C is a circuit in G if and only if $C^* := \{e^* \mid e \in C\}$ is an inclusionwise minimal nonempty cut in

G^*. Moreover, C is a spanning tree in G if and only if $\{e^* \mid e \in E \setminus C\}$ is a spanning tree in G^*.

Series-parallel and outerplanar graphs

A graph is called a *series-parallel graph* if it arises from a forest by repeated replacing edges by parallel edges or by edges in series. It was proved by Duffin [1965] that a graph is series-parallel if and only if it has no K_4 minor.

A graph is called *outerplanar* if it can be embedded in the plane such that each vertex is on the outer boundary. It can be easily derived from Kuratowski's theorem that a graph is outerplanar if and only if it has no K_4 or $K_{2,3}$ minor.

Adjacency and incidence matrix

The *adjacency matrix* of a graph $G = (V, E)$ is the $V \times V$ matrix A with

(3.46) $$A_{u,v} := \text{number of edges connecting } u \text{ and } v$$

for $u, v \in V$.

The *incidence matrix*, or $V \times E$ *incidence matrix*, of G is the $V \times E$ matrix B with

(3.47) $$B_{v,e} := \begin{cases} 1 & \text{if } v \in e \text{ and } e \text{ is not a loop,} \\ 2 & \text{if } v \in e \text{ and } e \text{ is a loop,} \\ 0 & \text{if } v \notin e, \end{cases}$$

for $v \in V$ and $e \in E$. The transpose of B is called the $E \times V$ incidence matrix of G, or just the incidence matrix, if no confusion is expected.

The concepts from graph theory invite to a less formal, and more expressive language, which appeals to the intuition, and whose formalization will be often tedious rather than problematic. Thus we say 'replace the edge uv by two edges in series', which means deleting uv and introducing a new vertex, w say, and new edges uw and wv. Similarly, 'replacing the edge uv by a path' means deleting uv, and introducing new vertices $w_1, \ldots, w_k$ say, and new edges uw_1, $w_1w_2, \ldots, w_{k-1}w_k$, w_kv.

3.2. Directed graphs

A *directed graph* or *digraph* is a pair $D = (V, A)$ where V is a finite set and A is a family of *ordered* pairs from V. The elements of V are called the *vertices*, sometimes the *nodes* or the *points*. The elements of A are called the *arcs* (sometimes *directed edges*). We denote:

(3.48) $VD :=$ set of vertices of D and $AD :=$ family of arcs of D.

In running time estimates of algorithms we denote:

(3.49) $n := |VD|$ and $m := |AD|$.

Again, the term 'family' is used to indicate that the same pair of vertices may occur several times in A. A pair occurring more than once in A is called a *multiple* arc, and the number of times it occurs is called its *multiplicity*. Two arcs are called *parallel* if they are represented by the same ordered pair of vertices.

Also *loops* are allowed, that is, arcs of the form (v, v). In our discussions, loops in directed graphs will be almost always irrelevant, and it will be clear from the context if they may occur. Directed graphs without loops and multiple arcs are called *simple*, and directed graphs without loops are called *loopless*.

Each directed graph $D = (V, A)$ gives rise to an *underlying undirected graph*, which is the graph $G = (V, E)$ obtained by ignoring the orientation of the arcs:

(3.50) $E := \{\{u, v\} \mid (u, v) \in A\}$.

We often will transfer undirected terminology to the directed case. Where appropriate, the adjective 'undirected' is added to a term if it refers to the underlying undirected graph.

If G is the underlying undirected graph of a directed graph D, we call D an *orientation* of G.

An arc (u, v) is said to *connect* u and v, and to *run from* u *to* v. For an arc $a = (u, v)$, u and v are called the *ends* of a, and u is called the *tail* of a, and v the *head* of a. We say that $a = (u, v)$ *leaves* u and *enters* v. For $U \subseteq V$, an arc $a = (u, v)$ is said to *leave* U if $u \in U$ and $v \notin U$. It is said to *enter* U if $u \notin U$ and $v \in U$.

If there exists an arc connecting vertices u and v, then u and v are called *adjacent* or *connected*. If there exists an arc (u, v), then v is called an *outneighbour* of u, and u is called an *inneighbour* of v.

The arc (u, v) is said to be *incident* with, or to *meet*, or to *cover*, the vertices u and v, and conversely. The arcs a and b are said to be *incident*, or to *meet*, or to *intersect*, if they have a vertex in common. Otherwise, they are called *disjoint*. If $U \subseteq V$ and both ends of an arc a belong to U, then we say that U *spans* a.

For any vertex v, we denote:

(3.51)
$$\begin{aligned}
&\delta^{\mathrm{in}}_D(v) := \delta^{\mathrm{in}}_A(v) := \delta^{\mathrm{in}}(v) := \text{set of arcs entering } v,\\
&\delta^{\mathrm{out}}_D(v) := \delta^{\mathrm{out}}_A(v) := \delta^{\mathrm{out}}(v) := \text{set of arcs leaving } v,\\
&N^{\mathrm{in}}_D(v) := N^{\mathrm{in}}_A(v) := N^{\mathrm{in}}(v) := \text{set of inneighbours of } v,\\
&N^{\mathrm{out}}_D(v) := N^{\mathrm{out}}_A(v) := N^{\mathrm{out}}(v) := \text{set of outneighbours of } v.
\end{aligned}$$

The *indegree* $\deg^{\mathrm{in}}_D(v)$ of a vertex v is the number of arcs entering v. The *outdegree* $\deg^{\mathrm{out}}_D(v)$ of a vertex v is the number of arcs leaving v. In notation,

(3.52) $\deg^{\text{in}}_D(v) := \deg^{\text{in}}_A(v) := \deg^{\text{in}}(v) := |\delta^{\text{in}}_D(v)|$,
$\deg^{\text{out}}_D(v) := \deg^{\text{out}}_A(v) := \deg^{\text{out}}(v) := |\delta^{\text{out}}_D(v)|$.

A vertex of indegree 0 is called a *source* and a vertex of outdegree 0 a *sink*. For any arc $a = (u, v)$ we denote

(3.53) $a^{-1} := (v, u)$.

For any digraph $D = (V, A)$ the *reverse digraph* D^{-1} is defined by

(3.54) $D^{-1} = (V, A^{-1})$, where $A^{-1} := \{a^{-1} \mid a \in A\}$.

A *mixed graph* is a triple (V, E, A) where (V, E) is an undirected graph and (V, A) is a directed graph.

The complete directed graph and the line digraph

The *complete directed graph* on a set V is the simple directed graph with vertex set V and arcs all pairs (u, v) with $u, v \in V$ and $u \neq v$. A *tournament* is any simple directed graph (V, A) such that for all distinct $u, v \in V$ precisely one of (u, v) and (v, u) belongs to A.

The *line digraph* of a directed graph $D = (V, A)$ is the digraph with vertex set A and arc set

(3.55) $\{((u, v), (x, y)) \mid (u, v), (x, y) \in A, v = x\}$.

Subgraphs of directed graphs

A digraph $D' = (V', A')$ is called a *subgraph* of a digraph $D = (V, A)$ if $V' \subseteq V$ and $A' \subseteq A$. If $D' \neq D$, then D' is called a *proper subgraph* of D. If $V' = V$, then D' is called a *spanning subgraph* of D. If A' consists of all arcs of D spanned by V', D' is called an *induced subgraph*, or the *subgraph induced by* V'. In notation,

(3.56) $D[V'] :=$ subgraph of D induced by V',
$A[V'] :=$ family of arcs spanned by V'.

So $D[V'] = (V', A[V'])$. We further denote for any vertex v, any subset U of V, any arc a, and any subset B of A,

(3.57) $D - v := D[V \setminus \{v\}]$, $D - U := D[V \setminus U]$, $D - a := (V, A \setminus \{a\})$,
$D - B := (V, A \setminus B)$.

We say that these graphs arise from D by *deleting* v, U, a, or B.

Two subgraphs of D are called *arc-disjoint* if they have no arc in common, and *vertex-disjoint* or *disjoint*, if they have no vertex in common.

Directed paths and circuits

A *directed walk*, or just a *walk*, in a directed graph $D = (V, A)$ is a sequence

(3.58) $$P = (v_0, a_1, v_1, \ldots, a_k, v_k),$$

where $k \geq 0$, $v_0, v_1, \ldots, v_k \in V$, $a_1, \ldots, a_k \in A$, and $a_i = (v_{i-1}, v_i)$ for $i = 1, \ldots, k$. The path is called a *directed path*, or just a *path*, if $v_0, \ldots, v_k$ are distinct. (Hence $a_1, \ldots, a_k$ are all distinct.)

The vertex v_0 is called the *starting vertex* or the *first vertex* of P, and the vertex v_k the *end vertex* or the *last vertex* of P. Sometimes, both v_0 and v_k are called the *end vertices*, or just the *ends* of P. Similarly, arc a_1 is called the *starting arc* or *first arc* of P and arc a_k the *end arc* or *last arc* of P. Sometimes, both a_1 and a_k are called the *end arcs*.

The walk P is said to *connect* the vertices v_0 and v_k, to *run from* v_0 *to* v_k (or *between* v_0 *and* v_k), and to *traverse* $v_0, a_1, v_1, \ldots, a_k, v_k$. The vertices $v_1, \ldots, v_{k-1}$ are called the *internal vertices* of P. For $s, t \in V$, a walk P is called an $s-t$ *walk* if it runs from s to t, and for $S, T \subseteq V$, P is called an $S-T$ *walk* if it runs from a vertex in S to a vertex in T. If P is a path, we obviously speak of an $s-t$ *path* and an $S-T$ *path*, respectively.

A vertex t is called *reachable from* a vertex s (or from a set S) if there exists a directed $s-t$ path (or directed $S-t$ path). Similarly, a vertex s is said to *reach*, or to be *reachable to*, a vertex t (or to a set T) if there exists a directed $s-t$ path (or directed $s-T$ path).

The number k in (3.58) is called the *length* of P. (We deviate from this in case a function $l : A \to \mathbb{R}$ has been introduced as a length function. Then the *length* of P is equal to $l(a_1) + \cdots + l(a_k)$.)

The minimum length of a path from u to v is called the *distance* from u to v.

An *undirected walk* in a directed graph $D = (V, A)$ is a walk in the underlying undirected graph; more precisely, it is a sequence

(3.59) $$P = (v_0, a_1, v_1, \ldots, a_k, v_k)$$

where $k \geq 0$, $v_0, v_1, \ldots, v_k \in V$, $a_1, \ldots, a_k \in A$, and $a_i = (v_{i-1}, v_i)$ or $a_i = (v_i, v_{i-1})$ for $i = 1, \ldots, k$. The arcs a_i with $a_i = (v_{i-1}, v_i)$ are called the *forward arcs* of P, and the arcs a_i with $a_i = (v_i, v_{i-1})$ the *backward arcs* of P.

If $P = (v_0, a_1, v_1, \ldots, a_k, v_k)$ and $Q = (u_0, b_1, u_1, \ldots, b_l, u_l)$ are walks satisfying $u_0 = v_k$, the *concatenation* PQ of P and Q is the walk

(3.60) $$PQ := (v_0, a_1, v_1, \ldots, a_k, v_k, b_1, u_1, \ldots, b_l, u_l).$$

For any walk P, we denote by VP and AP the families of vertices and arcs, respectively, occurring in P:

(3.61) $$VP := \{v_0, v_1, \ldots, v_k\} \text{ and } AP := \{a_1, \ldots, a_k\}.$$

If no confusion may arise, we sometimes identify the walk P with the subgraph (VP, AP) of D, or with the set VP of vertices in P, or with the family AP of arcs in P.

If the digraph is simple or (more generally) if the arcs traversed are irrelevant, we indicate the walk just by the sequence of vertices traversed:

(3.62) $P = (v_0, v_1, \ldots, v_k)$ or $P = v_0, v_1, \ldots, v_k$.

A path may be identified by the sequence of arcs:

(3.63) $P = (a_1, \ldots, a_k)$ or $P = a_1, \ldots, a_k$.

Two walks P and Q are called *vertex-disjoint* or *disjoint* if VP and VQ are disjoint, *internally vertex-disjoint* or *internally disjoint* if the set of internal vertices of P is disjoint from the set of internal vertices of Q, and *arc-disjoint* if AP and AQ are disjoint.

The directed walk P in (3.13) is called a *closed directed walk* or *directed cycle* if $v_k = v_0$. It is called a *directed circuit*, or just a *circuit*, if $v_k = v_0$, $k \geq 1$, $v_1, \ldots, v_k$ are all distinct, and $a_1, \ldots, a_k$ are all distinct. An *undirected circuit* is a circuit in the underlying undirected graph.

Connectivity and components of digraphs

A digraph $D = (V, A)$ is called *strongly connected* if for each two vertices u and v there is a directed path from u to v. The digraph D is called *weakly connected* if the underlying undirected graph is connected; that is, for each two vertices u and v there is an undirected path connecting u and v.

A maximal strongly connected nonempty subgraph of a digraph $D = (V, A)$ is called a *strongly connected component*, or a *strong component*, of D. Again, 'maximal' is taken with respect to taking subgraphs. A *weakly connected component*, or a *weak component*, of D is a component of the underlying undirected graph.

Each strong component is an induced subgraph. Each vertex belongs to exactly one strong component, but there may be arcs that belong to no strong component. One has:

(3.64) arc (u, v) belongs to a strong component $\iff$ there exists a directed path in D from v to u.

We sometimes identify a strong component K with the set VK of its vertices. Then the strong components are precisely the equivalence classes of the equivalence relation $\sim$ defined on V by: $u \sim v \iff$ there exist a directed path from u to v and a directed path from v to u.

Cuts

Let $D = (V, A)$ be a directed graph. For any $U \subseteq V$, we denote:

(3.65) $\delta^{\text{in}}_D(U) := \delta^{\text{in}}_A(U) := \delta^{\text{in}}(U) :=$ set of arcs of D entering U,
$\delta^{\text{out}}_D(U) := \delta^{\text{out}}_A(U) := \delta^{\text{out}}(U) :=$ set of arcs of D leaving U.

A subset B of A is called a *cut* if $B = \delta^{\text{out}}(U)$ for some $U \subseteq V$. In particular, $\emptyset$ is a cut. If $\emptyset \neq U \neq V$, then $\delta^{\text{out}}(U)$ is called a *nontrivial cut.*

If $s \in U$ and $t \notin U$, then $\delta^{\text{out}}(U)$ is called an *$s-t$ cut.* If $S \subseteq U$ and $T \subseteq V \setminus U$, $\delta^{\text{out}}(U)$ is called an *$S-T$ cut.* A cut of size k is called a *k-cut.*

A subset B of A is called a *disconnecting arc set* if $D - B$ is not strongly connected. For $s, t \in V$, it is said to be *$s-t$ disconnecting*, if B intersects each directed $s-t$ path. For $S, T \subseteq V$, B is said to be *$S-T$ disconnecting*, if B intersects each directed $S-T$ path.

One may easily check that for all $s, t \in V$:

(3.66) each $s-t$ cut is $s-t$ disconnecting; each inclusionwise minimal $s-t$ disconnecting arc set is an $s-t$ cut.

For any subset U of V we denote

(3.67) $d^{\text{in}}_D(U) := d^{\text{in}}_A(U) := d^{\text{in}}(U) := |\delta^{\text{in}}(U)|$,
$d^{\text{out}}_D(U) := d^{\text{out}}_A(U) := d^{\text{out}}(U) := |\delta^{\text{out}}(U)|$.

The following inequalities will be often used:

Theorem 3.4. *For any digraph $D = (V, A)$ and $X, Y \subseteq V$:*

(3.68) $d^{\text{in}}(X) + d^{\text{in}}(Y) \geq d^{\text{in}}(X \cap Y) + d^{\text{in}}(X \cup Y)$ *and*
$d^{\text{out}}(X) + d^{\text{out}}(Y) \geq d^{\text{out}}(X \cap Y) + d^{\text{out}}(X \cup Y)$,

Proof. The first inequality follows directly from the equation

(3.69) $d^{\text{in}}(X) + d^{\text{in}}(Y) =$
$d^{\text{in}}(X \cap Y) + d^{\text{in}}(X \cup Y) + |A[X \setminus Y, Y \setminus X]| + |A[Y \setminus X, X \setminus Y]|$,

where $A[S, T]$ denotes the set of arcs with tail in S and head in T. The second inequality follows similarly. ∎

A cut C is called a *directed cut* if $C = \delta^{\text{in}}(U)$ for some $U \subseteq V$ with $\delta^{\text{out}}(U) = \emptyset$ and $\emptyset \neq U \neq V$. An arc is called a *cut arc* if $\{a\}$ is a directed cut; equivalently, if a is a bridge in the underlying undirected graph.

Vertex-cuts

Let $D = (V, A)$ be a digraph. A subset U of V is called a *disconnecting vertex set*, or a *vertex-cut*, if $D - U$ is disconnected. A vertex-cut of size k is called a *k-vertex-cut.*

For $s, t \in V$, if U intersects each directed $s-t$ path in D, then U is said to *disconnect* s and t, or called *$s-t$ disconnecting.* If moreover $s, t \notin U$, then U is said to *separate* s and t, or called *$s-t$ separating*, or an *$s-t$ vertex-cut.*

For $S, T \subseteq V$, if U intersects each directed $S-T$ path, then U is said to *disconnect* S and T, or called $S-T$ *disconnecting*. If moreover U is disjoint from $S \cup T$, then U is said to *separate* S and T, or called $S-T$ *separating* or an $S-T$ *vertex-cut*.

Acyclic digraphs and directed trees

A directed graph $D = (V, A)$ is called *acyclic* if it has no directed circuits. It is easy to show that

(3.70) an acyclic digraph has at least one source and at least one sink,

provided that it has at least one vertex.

A directed graph is called a *directed tree* if the underlying undirected graph is a tree; that is, if D is weakly connected and has no undirected circuits. It is called a *rooted tree* if moreover D has precisely one source, called the *root*. If r is the root, we say that the rooted tree is *rooted at* r. If a rooted tree $D = (V, A)$ has root r, then each vertex $v \neq r$ has indegree 1, and for each vertex v there is a unique directed $r-v$ path. An *arborescence* in a digraph $D = (V, A)$ is a set B of arcs such that (V, B) is a rooted tree. If the rooted tree has root r, it is called an r-*arborescence*.

A directed graph is called a *directed forest* if the underlying undirected graph is a forest; that is, if D has no undirected circuits. It is called a *rooted forest* if moreover each weak component is a rooted tree. The roots of the weak components are called the *roots* of the rooted forest. A *branching* in a digraph $D = (V, A)$ is a set B of arcs such that (V, B) is a rooted forest.

Hamiltonian and Eulerian digraphs

A *Hamiltonian circuit* in a directed graph $D = (V, A)$ is a directed circuit C with $VC = VD$. A digraph is *Hamiltonian* if it has a Hamiltonian circuit. A *Hamiltonian path* is a directed path P with $VP = VD$.

A directed walk P is called *Eulerian* if each arc of D is traversed exactly once by P. A digraph D is called *Eulerian* if it has a closed Eulerian directed walk. Then a digraph $D = (V, A)$ is Eulerian if and only if D is weakly connected and $\deg^{\mathrm{in}}(v) = \deg^{\mathrm{out}}(v)$ for each vertex v. Sometimes, we call a digraph Eulerian if each weak component is Eulerian. This will be clear from the context.

An *Eulerian orientation* of an undirected graph $G = (V, E)$ is an orientation (V, A) of G with $\deg_A^{\mathrm{in}}(v) = \deg_A^{\mathrm{out}}(v)$ for each $v \in V$. A classical theorem in graph theory states that an undirected graph G has an Eulerian orientation if and only if all degrees of G are even.

Contraction

Contraction of directed graphs is similar to contraction of undirected graphs. Let $D = (V, A)$ be a digraph and let $a = (u, v) \in A$. *Contracting* a means deleting a and identifying u and v. We denote:

(3.71) $\quad D/a :=$ digraph obtained from D by contracting a.

Related is the following contraction. Let $D = (V, A)$ be a digraph and let $S \subseteq V$. The digraph D/S (obtained by *contracting* S) is obtained by identifying all vertices in S to one new vertex, called S, deleting all arcs contained in S, and redefining any arc (u, v) to (S, v) if $u \in S$ and to (u, S) if $v \in S$.

Planar digraphs and their duals

A digraph D is called *planar* if its underlying undirected graph G is planar. There is a natural way of making the dual graph G^* of G into a directed graph D^*, the *dual*: if arc $a = (u, v)$ of D separates faces F and F', such that, when following a from u to v, F is at the left and F' is at the right of a, then the dual edge is oriented from F to F', giving the arc a^* of D^*. Then D^{**} is isomorphic to D^{-1}, if D is weakly connected. One may check that a subset C of D is a directed circuit in D if and only if the set $\{a^* \mid a \in C\}$ is an inclusionwise minimal directed cut in D^*.

Adjacency and incidence matrix

The *adjacency matrix* of a digraph $D = (V, A)$ is the $V \times V$ matrix M with

(3.72) $\quad M_{u,v} :=$ number of arcs from u to v

for $u, v \in V$.

The *incidence matrix*, or $V \times A$ *incidence matrix*, of D is the $V \times A$ matrix B with

(3.73) $$B_{v,a} := \begin{cases} -1 & \text{if } v \text{ is tail of } a, \\ +1 & \text{if } v \text{ is head of } a, \\ 0 & \text{otherwise,} \end{cases}$$

for any $v \in V$ and any nonloop $a \in A$. If a is a loop, we set $B_{v,a} := 0$ for each vertex v.

The transpose of B is called the $A \times V$ incidence matrix of D, or just the incidence matrix, if no confusion is expected.

3.3. Hypergraphs

Part VIII is devoted to hypergraphs, but we occasionally need the terminology of hypergraphs in earlier parts. A *hypergraph* is a pair $H = (V, \mathcal{E})$ where V is a finite set and $\mathcal{E}$ is a family of subsets of V. The elements of V and $\mathcal{E}$ are called the *vertices* and the *edges* respectively. If $|F| = k$ for each $F \in \mathcal{E}$, the hypergraph is called *k-uniform*.

A hypergraph $H = (V, \mathcal{E})$ is called *connected* if there is no $U \subseteq V$ such that $\emptyset \neq U \neq V$ and such that $F \subseteq U$ or $F \subseteq V \setminus U$ for each edge F. A *(connected) component* of H is a hypergraph $K = (V', \mathcal{E}')$ with $V' \subseteq V$ and $\mathcal{E}' \subseteq \mathcal{E}$, such that V' and $\mathcal{E}'$ are inclusionwise maximal with the property that K is connected. A component is uniquely identified by its set of vertices.

Packing and covering

A family $\mathcal{F}$ of sets is called a *packing* if the sets in $\mathcal{F}$ are pairwise disjoint. For $k \in \mathbb{Z}_+$, $\mathcal{F}$ is called a *k-packing* if each element of $\bigcup \mathcal{F}$ is in at most k sets in $\mathcal{F}$ (counting multiplicities). In other words, any $k+1$ sets from $\mathcal{F}$ have an empty intersection. If each set in $\mathcal{F}$ is a subset of some set S, and $c : S \to \mathbb{R}$, then $\mathcal{F}$ is called a *c-packing* if each element $s \in S$ is in at most $c(s)$ sets in $\mathcal{F}$ (counting multiplicities).

A *fractional packing* is a function $\lambda : \mathcal{F} \to \mathbb{R}_+$ such that, for each $s \in S$,

$$\sum_{\substack{U \in \mathcal{F} \\ s \in U}} \lambda_U \leq 1. \tag{3.74}$$

For $c : S \to \mathbb{R}$, the function $\lambda : \mathcal{F} \to \mathbb{R}_+$ is called a *fractional c-packing* if

$$\sum_{U \in \mathcal{F}} \lambda_U \chi^U \leq c. \tag{3.75}$$

The *size* of $\lambda : \mathcal{F} \to \mathbb{R}$ is, by definition,

$$\sum_{U \in \mathcal{F}} \lambda_U. \tag{3.76}$$

Similarly, a family $\mathcal{F}$ of sets is called a *covering* of a set S if S is contained in the union of the sets in $\mathcal{F}$. For $k \in \mathbb{Z}_+$, $\mathcal{F}$ is called a *k-covering* of S if each element of S is in at least k sets in $\mathcal{F}$ (counting multiplicities). For $c : S \to \mathbb{R}$, $\mathcal{F}$ is called a *c-covering* if each element $s \in S$ is in at least $c(s)$ sets in $\mathcal{F}$ (counting multiplicities).

A *fractional covering* of S is a function $\lambda : \mathcal{F} \to \mathbb{R}_+$ such that, for each $s \in S$,

$$\sum_{\substack{U \in \mathcal{F} \\ s \in U}} \lambda_U \geq 1. \tag{3.77}$$

For $c : S \to \mathbb{R}$, the function $\lambda : \mathcal{F} \to \mathbb{R}_+$ is called a *fractional c-covering* if

$$(3.78) \qquad \sum_{U \in \mathcal{F}} \lambda_U \chi^U \geq c.$$

Again, the *size* of $\lambda : \mathcal{F} \to \mathbb{R}$ is, by definition,

$$(3.79) \qquad \sum_{U \in \mathcal{F}} \lambda_U .$$

Cross-free and laminar families

A collection $\mathcal{C}$ of subsets of a set V is called *cross-free* if for all $T, U \in \mathcal{C}$:

$$(3.80) \qquad T \subseteq U \text{ or } U \subseteq T \text{ or } T \cap U = \emptyset \text{ or } T \cup U = V.$$

$\mathcal{C}$ is called *laminar* if for all $T, U \in \mathcal{C}$:

$$(3.81) \qquad T \subseteq U \text{ or } U \subseteq T \text{ or } T \cap U = \emptyset.$$

There is the following upper bound on the size of a laminar family:

Theorem 3.5. *If $\mathcal{C}$ is laminar and $V \neq \emptyset$, then $|\mathcal{C}| \leq 2|V|$.*

Proof. By induction on $|V|$. We can assume that $|V| \geq 2$ and that $V \in \mathcal{C}$. Let U be an inclusionwise minimal set in $\mathcal{C}$ with $|U| \geq 2$. Resetting $\mathcal{C}$ to $\mathcal{C} \setminus \{\{v\} \mid v \in U\}$, and identifying all elements in U, $|\mathcal{C}|$ decreases by at most $|U|$, and $|V|$ by $|U| - 1$. Since $|U| \leq 2(|U| - 1)$ (as $|U| \geq 2$), induction gives the required inequality. ∎

3.3a. Background references on graph theory

For background on graph theory we mention the books by Kőnig [1936] (historical), Harary [1969] (classical reference book), Wilson [1972b] (introductory), Bondy and Murty [1976], and Diestel [1997].

Chapter 4

Preliminaries on algorithms and complexity

This chapter gives an introduction to algorithms and complexity, in particular to polynomial-time solvability and NP-completeness. We restrict ourselves to a largely informal outline and keep formalisms at a low level. Most of the formalisms described in this chapter are not needed in the remaining of this book. A rough understanding of algorithms and complexity suffices.

4.1. Introduction

An informal, intuitive idea of what is an algorithm will suffice to understand the greater part of this book. An algorithm can be seen as a finite set of instructions that perform operations on certain data. The input of the algorithm will give the initial data. When the algorithm stops, the output will be found in prescribed locations of the data set. The instructions need not be performed in a linear order: an instruction determines which of the instructions should be followed next. Also, it can prescribe to stop the algorithm.

While the set of instructions constituting the algorithm is finite and fixed, the size of the data set may vary, and will depend on the input. Usually, the data are stored in arrays, that is, finite sequences. The lengths of these arrays may depend on the input, but the number of arrays is fixed and depends only on the algorithm. (A more-dimensional array like a matrix is stored in a linear fashion, in accordance with the linear order in which computer memory is organized.)

The data may consist of numbers, letters, or other symbols. In a computer model they are usually stored as finite strings of 0's and 1's (*bits*). The *size* of the data is the total length of these strings. In this context, the *size* of a rational number p/q with $p, q \in \mathbb{Z}$, $q \geq 1$, and g.c.d.$(p.q) = 1$, is equal to $1 + \lceil \log(|p| + 1) \rceil + \lceil \log q \rceil$.

4.2. The random access machine

We use the algorithmic model of the *random access machine*, sometimes abbreviated to *RAM*. It operates on entries that are $0,1$ strings, representing abstract objects (like vertices of a graph) or rational numbers. An instruction can read several (but a fixed number of) entries simultaneously, perform arithmetic operations on them, and store the answers in array positions prescribed by the instruction[2]. The array positions that should be read and written, are given in locations prescribed by the instruction.

We give a more precise description. The random access machine has a finite set of variables $z_0, \ldots, z_k$ and one array, f say, of length depending on the input. Each array entry is a $0,1$ string. They can be interpreted as rationals, in some binary encoding, but can also have a different meaning. Initially, $z_0, \ldots, z_k$ are set to 0, and f contains the input.

Each instruction is a finite sequence of resettings of one the following types, for $i, j, h \in \{1, \ldots, k\}$:

$$z_i := f(z_j);\ f(z_j) := z_i;\ z_i := z_j + z_h;\ z_i := z_j - z_h;\ z_i := z_j z_h;\\ z_i := z_j/z_h;\ z_i := z_i + 1;\ z_i := 1 \text{ if } z_j > 0 \text{ and } z_i := 0 \text{ otherwise.} \tag{4.1}$$

These include the *elementary arithmetic operations*: addition, subtraction, multiplication, division, comparison. (One may derive other arithmetic operations from this like rounding and taking logarithm or square root, by performing $O(\sigma + |\log \varepsilon|)$ elementary arithmetic operations, where σ is the size of the rational number and ε is the required precision.)

The instructions are numbered $0, 1, \ldots, t$, and z_1 is the number of the instruction to be executed. If $z_1 > t$ we stop and return the contents of the array f as output.

4.3. Polynomial-time solvability

A *polynomial-time algorithm* is an algorithm that terminates after a number of steps bounded by a polynomial in the input size. Here a *step* consists of performing one instruction. Such an algorithm is also called a *good algorithm* or an *efficient algorithm*.

In this definition, the *input size* is the size of the input, that is, the number of bits that describe the input. We say that a problem is *polynomial-time solvable*, or is *solvable in polynomial time*, if it can be solved by a polynomial-time algorithm.

This definition may depend on the chosen algorithmic model, but it has turned out that for most models the set of problems solvable by a polynomial-time algorithm is the same. However, in giving order estimates of running

[2] This property has caused the term ‘random’ in random access machine: the machine has access, in constant time, to the data in *any* (however, well-determined) position. This is in contrast with the Turing machine, which can only move to adjacent positions.

times and in considering the concept of 'strongly polynomial-time' algorithm (cf. Section 4.12), we fix the above algorithmic model of the random access machine.

4.4. P

P, NP, and co-NP are collections of *decision problems*: problems that can be answered by 'yes' or 'no', like whether a given graph has a perfect matching or a Hamiltonian circuit. An optimization problem is no decision problem, but often can be reduced to it in a certain sense — see Section 4.7 below.

A decision problem is completely described by the inputs for which the answer is 'yes'. To formalize this, fix some finite set Σ, called the *alphabet*, of size at least 2 — for instance $\{0,1\}$ or the ASCII-set of symbols. Let Σ^* denote the set of all finite strings (*words*) of letters from Σ. The *size* of a word is the number of letters (counting multiplicities) in the word. We denote the size of a word w by $\text{size}(w)$.

As an example, an undirected graph can be represented by the word

$$(\{a,b,c,d\},\{\{a,b\},\{b,c\},\{a,d\},\{b,d\},\{a,c\}\}) \tag{4.2}$$

(assuming that Σ contains each of these symbols). Its size is 43.

A *problem* is any subset Π of Σ^*. The corresponding 'informal' problem is:

$$\text{given a word } x \in \Sigma^*\text{, does } x \text{ belong to } \Pi? \tag{4.3}$$

As an example, the problem if a given graph is Hamiltonian is formalized by the collection of all strings representing a Hamiltonian graph.

The string x is called the *input* of the problem. One speaks of an *instance* of a problem Π if one asks for one concrete input x whether x belongs to Π.

A problem Π is called *polynomial-time solvable* if there exists a polynomial-time algorithm that decides whether or not a given word $x \in \Sigma^*$ belongs to Π. The collection of all polynomial-time solvable problems $\Pi \subseteq \Sigma^*$ is denoted by P.

4.5. NP

An easy way to characterize the class NP is: NP is the collection of decision problems that can be reduced in polynomial time to the satisfiability problem — that is, to checking if a Boolean expression can be satisfied. For instance, it is not difficult to describe the conditions for a perfect matching in a graph by a Boolean expression, and hence reduce the existence of a perfect matching to the satisfiability of this expression. Also the problem of finding a Hamiltonian circuit, or a clique of given size, can be treated this way.

However, this is not the definition of NP, but a theorem of Cook. Roughly speaking, NP is defined as the collection of all decision problems for which each input with positive answer, has a polynomial-time checkable 'certificate' of correctness of the answer. Consider, for instance, the question:

(4.4) Is a given graph Hamiltonian?

A positive answer can be 'certified' by giving a Hamiltonian circuit in the graph. The correctness of it can be checked in polynomial time. No such certificate is known for the opposite question:

(4.5) Is a given graph non-Hamiltonian?

Checking the certificate in polynomial time means: checking it in time bounded by a polynomial in the original input size. In particular, it implies that the certificate itself has size bounded by a polynomial in the original input size.

This can be formalized as follows. NP is the collection of problems $\Pi \subseteq \Sigma^*$ for which there is a problem $\Pi' \in$P and a polynomial p such that for each $w \in \Sigma^*$ one has:

(4.6) $w \in \Pi \iff$ there exists a word x of size at most $p(\text{size}(w))$ with $wx \in \Pi'$.

The word x is called a *certificate* for w. (NP stands for *nondeterministically polynomial-time*, since the string x could be chosen by the algorithm by guessing. So guessing well leads to a polynomial-time algorithm.)

For instance, the collection of Hamiltonian graphs belongs to NP since the collection Π' of strings GC, consisting of a graph G and a Hamiltonian circuit C in G, belongs to P. (Here we take graphs and circuits as strings like (4.2).)

Trivially, we have P⊆NP, since if $\Pi \in$P, we can take $\Pi' = \Pi$ and $p \equiv 0$ in (4.6).

About all problems that ask for the existence of a structure of a prescribed type (like a Hamiltonian circuit) belong to NP. The class NP is apparently much larger than the class P, and there might be not much reason to believe that the two classes are the same. But, as yet, nobody has been able to prove that they really are different. This is an intriguing mathematical question, but besides, answering the question might also have practical significance. If P=NP can be shown, the proof might contain a revolutionary new algorithm, or alternatively, it might imply that the concept of 'polynomial-time' is completely meaningless. If P≠NP can be shown, the proof might give us more insight in the reasons why certain problems are more difficult than other, and might guide us to detect and attack the kernel of the difficulties.

4.6. co-NP and good characterizations

The collection co-NP consists of all problems Π for which the complementary problem $\Sigma^* \setminus \Pi$ belongs to NP. Since for any problem $\Pi \in$P, also $\Sigma^* \setminus \Pi$ belongs to P, we have

(4.7) P⊆NP∩co-NP.

The problems in NP∩co-NP are those for which both a positive answer and a negative answer have a polynomial-time checkable certificate. In other words, any problem Π in NP∩co-NP has a *good characterization*: there exist $\Pi', \Pi'' \in$P and a polynomial p such that for each $w \in \Sigma^*$:

(4.8) there is an $x \in \Sigma^*$ with $wx \in \Pi'$ and $\text{size}(x) \leq p(\text{size}(w)) \iff$ there is no $y \in \Sigma^*$ with $wy \in \Pi''$ and $\text{size}(y) \leq p(\text{size}(w))$.

Therefore, the problems in NP∩co-NP are called *well-characterized*.

A typical example is Tutte's 1-factor theorem:

(4.9) a graph $G = (V, E)$ has a perfect matching if and only if there is no $U \subseteq V$ such that $G - U$ has more than $|U|$ odd components.

So in this case Π consists of all graphs having a perfect matching, Π' of all strings GM where G is a graph and M a perfect matching in G, and Π'' of all strings GU where G is a graph and U is a subset of the vertex set of G such that $G - U$ has more than $|U|$ odd components. (To be more precise, since Σ^* is the universe, we must add all strings $w\{\}$ to Π'' where w is a word in Σ^* that does not represent a graph.) This is why Tutte's theorem is said to be a good characterization.

In fact, there are very few problems known that have been proved to belong to NP∩co-NP, but that are not known to belong to P. Most problems having a good characterization, have been proved to be solvable in polynomial time. So one may ask: is P=NP∩co-NP?

4.7. Optimization problems

Optimization problems can be transformed to decision problems as follows. Consider a *minimization* problem: minimize $f(x)$ over $x \in X$, where X is a collection of elements derived from the input of the problem, and where f is a rational-valued function on X. (For instance, minimize the length of a Hamiltonian circuit in a given graph, for a given length function on the edges.) This can be transformed to the following decision problem:

(4.10) given a rational number r, is there an $x \in X$ with $f(x) \leq r$?

If we have an upper bound β on the size of the minimum value (being proportional to the sum of the logarithms of the numerator and the denominator), then by asking question (4.10) for $O(\beta)$ choices of r, we can find the optimum value (by binary search). In this way we usually can derive a polynomial-time algorithm for the minimization problem from a polynomial-time algorithm for the decision problem. Similarly, for maximization problems.

About all combinatorial optimization problems, when framed as a decision problem like (4.10), belong to NP, since a positive answer to question (4.10) can often be certified by just specifying an $x \in X$ satisfying $f(x) \leq r$.

If a combinatorial optimization problem is characterized by a min-max relation like

$$\min_{x \in X} f(x) = \max_{y \in Y} g(y), \tag{4.11}$$

this often leads to a good characterization of the corresponding decision problem. Indeed, if $\min_{x \in X} f(x) \leq r$ holds, it can be certified by an $x \in X$ satisfying $f(x) \leq r$. On the other hand, if $\min_{x \in X} f(x) > r$ holds, it can be certified by a $y \in Y$ satisfying $g(y) > r$. If these certificates can be checked in polynomial time, we say that the min-max relation is a *good characterization*, and that the optimization problem is *well-characterized*.

4.8. NP-complete problems

The NP-complete problems are the problems that are the hardest in NP: every problem in NP can be reduced to them. We make this more precise.

Problem $\Pi \subseteq \Sigma^*$ is said to be *reducible* to problem $\Lambda \subseteq \Sigma^*$ if there exists a polynomial-time algorithm that returns, for any input $w \in \Sigma^*$, an output $x \in \Sigma^*$ with the property:

$$w \in \Pi \iff x \in \Lambda. \tag{4.12}$$

This implies that if Π is reducible to Λ and Λ belongs to P, then also Π belongs to P. Similarly, one may show that if Π is reducible to Λ and Λ belongs to NP, then also Π belongs to NP.

A problem Π is said to be NP-*complete* if each problem in NP is reducible to Π. Hence

(4.13) if some NP-complete problem belongs to P, then P=NP.

Surprisingly, there exist NP-complete problems (Cook [1971]). Even more surprisingly, several prominent combinatorial optimization problems, like the traveling salesman problem, the maximum clique problem, and the maximum cut problem, are NP-complete (Karp [1972b]).

Since then one generally distinguishes between the polynomial-time solvable problems and the NP-complete problems, although there is no proof that these two concepts really are distinct. For almost every combinatorial optimization problem (and many other problems) one has been able to prove either that it is solvable in polynomial time, or that it is NP-complete — and no problem has been proved to be both. But it still has not been excluded that these two concepts are just the same!

The usual approach to prove NP-completeness of problems is to derive it from the NP-completeness of one basic problem, often the satisfiability problem. To this end, we prove NP-completeness of the satisfiability problem in the coming sections.

4.9. The satisfiability problem

To formulate the satisfiability problem, we need the notion of a *Boolean expression*. Examples are:

$$((x_2 \wedge x_3) \vee \neg(x_3 \vee x_5) \wedge x_2),\ ((\neg x_{47} \wedge x_2) \wedge x_{47}),\text{ and } \neg(x_7 \wedge \neg x_7). \tag{4.14}$$

Boolean expressions can be defined inductively. We work with an alphabet Σ containing the 'special' symbols '(', ')', '$\wedge$', '$\vee$', '$\neg$', and ',', and not containing the symbols 0 and 1. Then any word not containing any special symbol is a Boolean expression, called a *variable*. Next, if v and w are Boolean expressions, then also $(v \wedge w)$, $(v \vee w)$, and $\neg v$ are Boolean expressions. These rules give us all Boolean expressions. We denote a Boolean expression f by $f(x_1, \ldots, x_k)$ if $x_1, \ldots, x_k$ are the variables occurring in f.

A Boolean expression $f(x_1, \ldots, x_k)$ is called *satisfiable* if there exist $\alpha_1, \ldots, \alpha_k \in \{0, 1\}$ such that $f(\alpha_1, \ldots, \alpha_k) = 1$, using the well-known identities

$$\begin{aligned} &0 \wedge 0 = 0 \wedge 1 = 1 \wedge 0 = 0, 1 \wedge 1 = 1,\\ &0 \vee 0 = 0, 0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1,\\ &\neg 0 = 1, \neg 1 = 0, (0) = 0, (1) = 1. \end{aligned} \tag{4.15}$$

Now let SAT$\subseteq \Sigma^*$ be the collection of satisfiable Boolean expressions. SAT is called the *satisfiability problem*.

The satisfiability problem SAT trivially belongs to NP: to certify that $f(x_1, \ldots, x_k)$ belongs to SAT, we can take the equations $x_i = \alpha_i$ that give f the value 1.

4.10. NP-completeness of the satisfiability problem

Let an algorithm be represented by the random access machine (we use notation as in Section 4.2). Consider the performance of the algorithm for some input w of size s (in the alphabet $\{0, 1\}$). We may assume that all entries in the random access machine are stored with the same number of bits, α say, only depending on s. Let q be the length of the array f. We may assume that q is invariant throughout the algorithm, and that q only depends on s. (So the initial input w is extended to an array f of length q.) Let r be the number of iterations performed by the algorithm. We may assume that r only depends on s.

Let m_i be the following word in $\{0, 1\}^*$:

(4.16) $z_0 z_1 \dots z_k f(0) f(1) \dots f(q)$

after performing i iterations (where each z_j and each $f(j)$ is a word in $\{0,1\}^*$ of size α). So it is the content of the machine memory after i iterations. We call the word

(4.17) $h = m_0 m_1 \dots m_r$

the *history*. The size of h is equal to

(4.18) $T := (r+1)(k+q+2)\alpha.$

We call a word h *correct* if there is an input w of size s that leads to history h.

The following observation is basic:

(4.19) given the list of instructions describing the random access machine and given s, we can construct, in time bounded by a polynomial in T, a Boolean expression $g(x_1, \dots, x_T)$ such that any 0,1 word $h = \alpha_1 \dots \alpha_T$ is correct if and only if $g(\alpha_1, \dots, \alpha_T) = 1$.

To see this, we must observe that each of the instructions (4.1) can be described by Boolean expressions in the 0,1 variables describing the corresponding entries.

We can permute the positions in g such that the first s variables correspond to the s input bits, and that the last variable gives the output bit (0 or 1). Let it give the Boolean expression $\tilde{g}(y_1, \dots, y_T)$. Then input $\beta_1 \dots \beta_s$ leads to output 1 if and only if

(4.20) $\tilde{g}(\beta_1, \dots, \beta_s, y_{s+1}, \dots, y_{T-1}, 1) = 1$

has a solution in the variables $y_{s+1}, \dots, y_{T-1}$.

Consider now a problem Π in NP. Let Π' be a problem in P and p a polynomial satisfying (4.6). We can assume that x has size precisely $p(\text{size}(w))$. So if input w of Π has size u, then wx has size $s := u + p(u)$. Let A be a polynomial-time algorithm as described above solving Π' and let $\tilde{g}$ be the corresponding Boolean expression as above. Let $w = \beta_1 \dots \beta_u$. Then w belongs to Π if and only if

(4.21) $\tilde{g}(\beta_1, \dots, \beta_u, y_{u+1}, \dots, y_s, y_{s+1}, \dots, y_{T-1}, 1) = 1$

is solvable. This reduces Π to the satisfiability problem. Hence we have the main result of Cook [1971] (also Levin [1973]):

Theorem 4.1. *The satisfiability problem is* NP*-complete.*

Proof. See above. ∎

4.11. NP-completeness of some other problems

For later reference, we derive from Cook's theorem the NP-completeness of some other problems. First we show that the 3-*satisfiability problem* 3-SAT is NP-complete (Cook [1971], cf. Karp [1972b]). Let B_1 be the set of all words $x_1, \neg x_1, x_2, \neg x_2, \ldots$, where the x_i are words not containing the symbols '$\neg$', '$\wedge$', '$\vee$', '(', ')'. Let B_2 be the set of all words $(w_1 \vee \cdots \vee w_k)$, where $w_1, \cdots, w_k$ are words in B_1 and $1 \leq k \leq 3$. Let B_3 be the set of all words $w_1 \wedge \ldots \wedge w_k$, where $w_1, \ldots, w_k$ are words in B_2. Again, we say that a word $f(x_1, x_2, \ldots) \in B_3$ is *satisfiable* if there exists an assignment $x_i := \alpha_i \in \{0, 1\}$ $(i = 1, 2, \ldots)$ such that $f(\alpha_1, \alpha_2, \ldots) = 1$ (using the identities (4.15)).

Now the 3-satisfiability problem 3-SAT is: given a word $f \in B_3$, decide if it is satisfiable. More formally, 3-SAT is the set of all satisfiable words in B_3.

Corollary 4.1a. *The 3-satisfiability problem* 3-SAT *is* NP*-complete.*

Proof. We give a polynomial-time reduction of SAT to 3-SAT. Let $f(x_1, x_2, \ldots)$ be a Boolean expression. Introduce a variable y_g for each subword g of f that is a Boolean expression (not splitting variables).

Now f is satisfiable if and only if the following system is satisfiable:

(4.22)
$$\begin{array}{ll} y_g = y_{g'} \vee y_{g''} & (\text{if } g = (g' \vee g'')), \\ y_g = y_{g'} \wedge y_{g''} & (\text{if } g = (g' \wedge g'')), \\ y_g = \neg y_{g'} & (\text{if } g = \neg g'), \\ y_f = 1. & \end{array}$$

Now $y_g = y_{g'} \vee y_{g''}$ can be equivalently expressed by: $y_g \vee \neg y_{g'} = 1, y_g \vee \neg y_{g''} = 1, \neg y_g \vee y_{g'} \vee y_{g''} = 1$. Similarly, $y_g = y_{g'} \wedge y_{g''}$ can be equivalently expressed by: $\neg y_g \vee y_{g'} = 1, \neg y_g \vee y_{g''} = 1, y_g \vee \neg y_{g'} \vee \neg y_{g''} = 1$. The expression $y_g = \neg y_{g'}$ is equivalent to: $y_g \vee y_{g'} = 1, \neg y_g \vee \neg y_{g'} = 1$.

By renaming variables, we thus obtain words $w_1, \ldots, w_k$ in B_2, such that f is satisfiable if and only if the word $w_1 \wedge \ldots \wedge w_k$ is satisfiable. ∎

(As Cook [1971] mentioned, a method of Davis and Putnam [1960] solves the 2-satisfiability problem in polynomial time.)

We next derive that the *partition problem* is NP-complete (Karp [1972b]). This is the problem:

(4.23) Given a collection of subsets of a finite set X, does it contain a subcollection that is a partition of X?

Corollary 4.1b. *The partition problem is* NP*-complete.*

Proof. We give a polynomial-time reduction of 3-SAT to the partition problem. Let $f = w_1 \wedge \ldots \wedge w_k$ be a word in B_3, where $w_1, \ldots, w_k$ are words in B_2. Let $x_1, \ldots, x_m$ be the variables occurring in f. Make a bipartite graph G with colour classes $\{w_1, \ldots, w_k\}$ and $\{x_1, \ldots, x_m\}$, by joining w_i and x_j by an edge if and only if x_j or $\neg x_j$ occurs in w_i. Let X be the set of all vertices and edges of G.

Let $\mathcal{C}'$ be the collection of all sets $\{w_i\} \cup E'$, where E' is a nonempty subset of the edge set incident with w_i. Let $\mathcal{C}''$ be the collection of all sets $\{x_j\} \cup E'_j$ and $\{x_j\} \cup E''_j$, where E'_j is the set of all edges $\{w_i, x_j\}$ such that x_j occurs in w_i and where E''_j is the set of all edges $\{w_i, x_j\}$ such that $\neg x_j$ occurs in w_i.

Now f is satisfiable if and only if the collection $\mathcal{C}' \cup \mathcal{C}''$ contains a subcollection that partitions X. Thus we have a reduction of 3-SAT to the partition problem. ∎

In later chapters we derive from these results the NP-completeness of several other combinatorial optimization problems.

4.12. Strongly polynomial-time

Roughly speaking, an algorithm is strongly polynomial-time if the number of elementary arithmetic and other operations is bounded by a polynomial in the size of the input, where any number in the input is counted only for 1. Strong polynomial-timeness of an algorithm is of relevance only for problems that have numbers among its input data. (Otherwise, strongly polynomial-time coincides with polynomial-time.)

Consider a problem that has a number k of input parts, like a vertex set, an edge set, a length function. Let $f : \mathbb{Z}_+^{2k} \to \mathbb{R}$. We say that an algorithm takes $O(f)$ time if the algorithm terminates after

$$O(f(n_1, s_1, \ldots, n_k, s_k)) \tag{4.24}$$

operations (including elementary arithmetic operations), where the ith input part consists of n_i numbers of maximum size s_i ($i = 1, \ldots, k$), and if the numbers occurring during the execution of the algorithm have size

$$O(\max\{s_1, \ldots, s_k\}). \tag{4.25}$$

The algorithm is called a *strongly polynomial-time algorithm* if the algorithm takes $O(f)$ time for some polynomial f in the array lengths $n_1, \ldots, n_k$, where f is independent of $s_1, \ldots, s_k$. If a problem can be solved by a strongly polynomial-time algorithm, we say that it is *solvable in strongly polynomial time* or *strongly polynomial-time solvable*.

An algorithm is called *linear-time* if f can be taken linear in $n_1, \ldots, n_k$, and independent of $s_1, \ldots, s_k$. If a problem can be solved by a linear-time algorithm, we say that it is *solvable in linear time* or *linear-time solvable*.

Rounding a rational x to $\lfloor x \rfloor$ can be done in polynomial-time, by $O(\text{size}(x))$ elementary arithmetic operations. It however cannot be done in strongly polynomial time. In fact, even checking if an integer k is odd or even cannot be done in strongly polynomial time: for any strongly polynomial-time algorithm with one integer k as input, there is a number L and a rational function $q : \mathbb{Z} \to \mathbb{Q}$ such that if $k > L$, then the output equals $q(k)$. (This can be proved by induction on the number of steps of the algorithm.) However, there do not exist a rational function q and number L such that for $k > L$, $q(k) = 0$ if k is even, and $q(k) = 1$ if k is odd.

We say that an algorithm is *semi-strongly polynomial-time* if we count rounding a rational as one step (one time-unit). We sometimes say *weakly polynomial-time* for polynomial-time, to distinguish from strongly polynomial-time.

4.13. Lists and pointers

Algorithmically, sets (of vertices, edges, etc.) are often introduced and handled as *ordered* sets, called *lists*. Their elements can be indicated just by their positions (*addresses*) in the order: $1, 2, \ldots$. Then attributes (like the capacity, or the ends, of an edge) can be specified in arrays.

Arrays represent functions, and such functions are also called *pointers* if their value is taken as an address. Such functions also allow the value *void*, where the function is undefined. Pointers can be helpful to shorten the running time of an algorithm.

One way to store a list is just in an array. But then updating may take (relatively) much time, for instance, if we would like to perform operations on lists, such as removing or inserting elements or concatenating two lists.

A better way to store a list $S = \{s_1, \ldots, s_k\}$ is as a *linked list*. This is given by a pointer $f : S \setminus \{s_k\} \to S$ where $f(s_i) = s_{i+1}$ for $i = 1, \ldots, k-1$, together with the first element s_1 given by the variable b say (a fixed array of length 1). It makes that S can be scanned in time $O(|S|)$.

If we need to update the list after removing an element from S, it is convenient to store S as a *doubly linked list*. Then we keep, next to f and b, a pointer $g : S \setminus \{s_1\} \to S$ where $g(s_i) = s_{i-1}$ for $i = 2, \ldots, k$, and a variable l say, with $l := s_k$. The virtue of this data structure is that it can be restored in constant time if we remove some element s_j from S. Also concatenating two doubly linked lists can be done in constant time. It is usually easy to build up the doubly linked list along with reading the input, taking time $O(|S|)$.

A convenient (but usually too abundant) way to store a directed graph $D = (V, A)$ using these data structures is as follows. For each $v \in V$, order the sets $\delta^{\text{in}}(v)$ and $\delta^{\text{out}}(v)$. Store V as a doubly linked list. Give pointers $t, h : A \to V$, where $t(a)$ and $h(a)$ are the tail and head of a. Give four pointers $V \to A$, indicating the first and last (respectively) arc in the lists $\delta^{\text{in}}(v)$ and $\delta^{\text{out}}(v)$ (respectively). Give four pointers $A \to A$, indicating for each $a \in A$, the previous and next (respectively) arc in the lists $\delta^{\text{in}}(h(a))$ and $\delta^{\text{out}}(t(a))$ (respectively). (Values may be 'void'. One can avoid the value 'void' by merging the latter eight pointers described into four pointers $V \cup A \to V \cup A$.)

If, in the input of a problem, a directed graph is given as a string (or file), like

$$(4.26) \qquad (\{a, b, c, d\}, \{(a, c), (a, d), (b, d), (c, d)\}),$$

we can build up the above data structure in time linear in the length of the string. Often, when implementing a graph algorithm, a subset of this structure will be sufficient. Undirected graphs can be handled similarly by choosing an arbitrary orientation of the edges. (So each edge becomes a list.)

4.14. Further notes

4.14a. Background literature on algorithms and complexity

Background literature on algorithms and complexity includes Knuth [1968] (data structures), Garey and Johnson [1979] (complexity, NP-completeness), Papadimitriou and Steiglitz [1982] (combinatorial optimization and complexity), Aho, Hopcroft, and Ullman [1983] (data structures and complexity), Tarjan [1983] (data structures), Cormen, Leiserson, and Rivest [1990] (algorithms), Papadimitriou [1994] (complexity), Sipser [1997] (algorithms, complexity), and Mehlhorn and Näher [1999] (data structures, algorithms and algorithms).

In this book we restrict algorithms and complexity to deterministic, sequential, and exact. For other types of algorithms and complexity we refer to the books by Motwani and Raghavan [1995] (randomized algorithms and complexity), Leighton [1992,2001] (parallel algorithms and complexity), and Vazirani [2001] (approximation algorithms and complexity). A survey on practical problem solving with cutting planes was given by Jünger, Reinelt, and Thienel [1995].

4.14b. Efficiency and complexity historically

In the history of complexity, more precisely, in the conception of the notions 'polynomial-time' and 'NP-complete', two lines loom up: one motivated by questions in logic, recursion, computability, and theorem proving, the other more down-to-earth focusing on the complexity of some concrete problems, with background in discrete mathematics and operations research.

Until the mid-1960s, the notions of efficiency and complexity were not formalized. The notion of algorithm was often used for a method that was better than brute-force enumerating. We focus on how the ideas of polynomial-time and NP-complete got shape. We will not go into the history of data structures, abstract computational complexity, or the subtleties inside and beyond NP (for which we refer to Papadimitriou [1994]).

We quote references in chronological order. This order is quite arbitrary, since the papers mostly seem to be written isolated from each other and they react very seldom to each other.

Maybe the first paper that was concerned with the complexity of computation is an article by Lamé [1844], who showed that the number of iterations in the Euclidean g.c.d. algorithm is linear in the logarithm of the smallest of the two (natural) numbers:

> Dans les traités d'Arithmétique, on se contente de dire que le nombre des divisions à effectuer, dans la recherche du plus grand commun diviseur entre deux entiers, *ne pourra pas surpasser la moitié du plus petit.* Cette limite, qui peut être dépassée si les nombres sont petits, s'éloigne outre mesure quand ils ont plusieurs chiffres. L'exagération est alors semblable à celle qui assignerait la moitié d'un nombre comme la limite de son logarithme; l'analogie devient évidente quand on connaît le théorème suivant:
>
> THÉORÈME. *Le nombre des divisions à effectuer, pour trouver le plus grand commun diviseur entre deux entiers* A, *et* B<A, *est toujours moindre que cinq fois le nombre des chiffres de* B.[3]

The first major combinatorial optimization problem for which a polynomial-time algorithm was given is the shortest spanning tree problem, by Borůvka [1926a, 1926b] and Jarník [1930], but these papers do not discuss the complexity issue — the efficiency of the method might have been too obvious. Choquet [1938] mentioned explicitly an estimate for the number of iterations in finding a shortest spanning tree:

> Le réseau cherché sera tracé après $2n$ opérations élémentaires au plus, en appelant opération élémentaire la recherche du continu le plus voisin d'un continu donné.[4]

The traveling salesman and the assignment problem

The traveling salesman problem and the assignment problem have been long-term bench-marks that gave shape to the ideas on efficiency and complexity.

Menger might have been the first to ask attention for the complexity of the traveling salesman problem. In the session of 5 February 1930 of his *mathematische Kolloquium* in Vienna (as reported in Menger [1932a]), he introduced *das Botenproblem*, later called the traveling salesman problem and raised the question for a better-than-finite algorithm:

> Dieses Problem ist natürlich stets durch endlichviele Versuche lösbar. Regeln, welche die Anzahl der Versuche unter die Anzahl der Permutationen der gegebenen Punkte herunterdrücken würden, sind nicht bekannt.[5]

---

[3] In the handbooks of Arithmetics, one contents oneself with saying that, in the search for the greatest common divisor of two integers, the number of divisions to execute *could not surpass half of the smallest [integer].* This bound, that can be exceeded if the numbers are small, goes away beyond measure when they have several digits. The exaggeration then is similar to that which would assign half of a number as bound of its logarithm; the analogy becomes clear when one knows the following theorem:

THEOREM. *The number of divisions to execute, to find the greatest common divisor of two integers* A, *and* B<A, *is always smaller than five times the number of digits of* B.

[4] The network looked for will be traced after at most $2n$ elementary operations, calling the search for the continuum closest to a given continuum an elementary operation.

[5] Of course, this problem is solvable by finitely many trials. Rules which would push the number of trials below the number of permutations of the given points, are not known.

Ghosh [1949] observed that the problem of finding a shortest tour along n random points in the plane (which is the traveling salesman problem) is hard:

> After locating the n random points in a map of the region, it is very difficult to find out *actually* the shortest path connecting the points, unless the number n is very small, which is seldom the case for a large-scale survey.

We should realize however that at that time also the (now known to be polynomial-time solvable) assignment problem was considered to be hard. In an Address delivered on 9 September 1949 at a meeting of the American Psychological Association at Denver, Colorado, Thorndike [1950] studied the problem of the 'classification' of personnel:

> There are, as has been indicated, a finite number of permutations in the assignment of men to jobs. When the classification problem as formulated above was presented to a mathematician, he pointed to this fact and said that from the point of view of the mathematician there was no problem. Since the number of permutations was finite, one had only to try them all and choose the best. He dismissed the problem at that point. This is rather cold comfort to the psychologist, however, when one considers that only ten men and ten jobs mean over three and a half million permutations. Trying out all the permutations may be a mathematical solution to the problem, it is not a practical solution.

But, in a RAND Report dated 5 December 1949, Robinson [1949] reported that an 'unsuccessful attempt' to solve the traveling salesman problem, led her to a 'cycle-cancelling' method for the optimum assignment problem, which in fact stands at the basis of efficient algorithms for network problems. She gave an optimality criterion for the assignment problem (absence of negative-length cycles in the residual graph). As for the traveling salesman problem she mentions:

> Since there are only a finite number of paths to consider, the problem consists in finding a method for picking out the optimal path when n is moderately large, say $n = 50$. In this case, there are more than 10^{62} possible paths, so we can not simply try them all. Even for as few as 10 points, some short cuts are desirable.

She also observed that the number of feasible solutions is not a measure for the complexity (where 'it' refers to the assignment problem):

> However at first glance, it looks more difficult than the traveling salesman probl[e]m, for there are obviously many more systems of circuits than circuits.

The development of the simplex method for linear programming, and its, in practice successful, application to combinatorial optimization problems like assignment and transportation, led to much speculation on the theoretical efficiency of the simplex method. In his paper describing the application of the simplex method to the transportation problem, Dantzig [1951a] mentioned (after giving a variable selection criterion that he speculates to lead to favourable computational experience for large-scale practical problems):

> This does not mean that theoretical problems could not be "cooked up" where this criterion is weak, but that in practical problems the number of steps has not been far from $m + n - 1$.

(Here n and m are the numbers of vertices and arcs, respectively.)

At the Symposium on Linear Inequalities and Programming in Washington, D.C. in 1951, Votaw and Orden [1952] reported on early computational results with the simplex method (on the SEAC), and claimed (without proof) that the simplex method is polynomial-time for the transportation problem (a statement refuted by Zadeh [1973a]):

> As to computation time, it should be noted that for moderate size problems, say $m \times n$ up to 500, the time of computation is of the same order of magnitude as the time required to type the initial data. The computation time on a sample computation in which m and n were both 10 was 3 minutes. The time of computation can be shown by study of the computing method and the code to be proportional to $(m+n)^3$.

Another early mention of polynomial-time as efficiency criterion is by von Neumann, who considered the complexity of the assignment problem. In a talk in the Princeton University Game Seminar on 26 October 1951, he described a method which is equivalent to finding a best strategy in a certain zero-sum two-person game. According to a transcript of the talk (cf. von Neumann [1951,1953]), von Neumann noted the following on the number of steps:

> It turns out that this number is a moderate power of n, i.e., considerably smaller than the "obvious" estimate $n!$ mentioned earlier.

However, no further argumentation is given.

In a Cowles Commission Discussion Paper of 2 April 1953, also Beckmann and Koopmans [1953] asked for better-than-finite methods for the assignment problem, but no explicit complexity measure was proposed, except that the work should be reduced to 'manageable proportions':

> It should be added that in all the assignment problems discussed, there is, of course, the obvious brute force method of enumerating all assignments, evaluating the maximand at each of these, and selecting the assignment giving the highest value. This is too costly in most cases of practical importance, and by a method of solution we have meant a procedure that reduces the computational work to manageable proportions in a wider class of cases.

During the further 1950s, better-than-finite methods were developed for the assignment and several other problems like shortest path and maximum flow. These methods turned out to give polynomial-time algorithms (possibly after modification), and several speedups were found — but polynomial-time was, as yet, seldom marked as efficiency criterion. The term 'algorithm' was often used just to distinguish from complete enumeration, but no mathematical characterization was given.

Kuhn [1955b,1956] introduced the 'Hungarian method' for the assignment problem (inspired by the proof method of Egerváry [1931]). Kuhn contented himself with showing finiteness of the method, but Munkres [1957] showed that it is strongly polynomial-time:

> The final maximum on the number of operations needed is
>
> $$(11n^3 + 12n^2 + 31n)/6.$$
>
> This maximum is of theoretical interest, since it is much smaller than the $n!$ operations necessary in the most straightforward attack on the problem.

As for the maximum flow problem, Ford and Fulkerson [1955,1957b] showed that their augmenting path method is finite, but only Dinits [1970] and Edmonds and Karp [1970,1972] showed that it can be adapted to be (strongly) polynomial-time.

Several algorithms were given for finding shortest paths (Shimbel [1955], Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957], Bellman [1958], Dantzig [1958,1960], Dijkstra [1959], Moore [1959]), and most of them are obviously strongly polynomial-time. (Ford [1956] gave a liberal shortest path algorithm that may require exponential time (Johnson [1973a,1973b,1977a]).)

Similarly, the interest in the shortest spanning tree problem revived, leading to old and new strongly polynomial-time algorithms (Kruskal [1956], Loberman and Weinberger [1957], Prim [1957], and Dijkstra [1959]).

The traveling salesman problem resisted these efforts. In the words of Dantzig, Fulkerson, and Johnson [1954a,1954b]:

> Although algorithms have been devised for problems of similar nature, e.g., the optimal assignment problem,[3,7,8] little is known about the traveling-salesman problem. We do not claim that this note alters the situation very much;

The papers [3,7,8] referred to, are the papers Dantzig [1951a], Votaw and Orden [1952], and von Neumann [1953], quoted above.

The use of the word 'Although' in the above quote makes it unclear what Dantzig, Fulkerson, and Johnson considered to be an algorithm. Their algorithm uses polyhedral methods to solve the traveling salesman problem, while Dantzig [1951a] and Votaw and Orden [1952] apply the simplex method to solve the assignment and transportation problems. In a follow-up paper, Dantzig, Fulkerson, and Johnson [1959] seem to have come to the conclusion that both methods are of a comparable level:

> Neither does the example, as we have solved it, indicate how one could make the combinatorial analysis a routine procedure. This can certainly be done (by enumeration, if nothing else)—but the fundamental question is: does the use of a few linear inequalities in general reduce the combinatorial magnitude of such problems significantly?
> We do not know the answer to this question in any theoretical sense, but it is our feeling, based on our experience in using the method, that it does afford a practical means of computing optimal tours in problems that are not too huge. It should be noted that a similar question, for example, arises when one uses the simplex method to find optimal solutions to linear programs, since no one has yet proved that the simplex method cuts down the computational task significantly from the crude method of examining all basic solutions, say. Nonetheless, people do use the simplex method because of successful experience with many hundreds of practical problems.

The feeling that the traveling salesman problem is more complex than the assignment problem was stated by Tompkins [1956]:

> A traveling-salesman problem is in some respects similar to the assignment problem. It seems definitely more difficult, however.

Tompkins described a branch-and-bound scheme to the permutation problem (including assignment and traveling salesman), but said:

> It must be noted, however, that this is not a completely satisfactory scheme for solution of such problems. In a few important cases (such as the assignment problem) more efficient machine methods have been devised.

The available algorithms for the traveling salesman problem were also not acceptable to Flood [1956]:

> There are as yet no acceptable computational methods, and surprisingly few mathematical results relative to the problem.

He mentioned that the problem might be 'fundamentally complex':

> Very recent mathematical work on the traveling-salesman problem by I. Heller, H.W. Kuhn, and others indicates that the problem is fundamentally complex. It seems very likely that quite a different approach from any yet used may be required for succesful treatment of the problem. In fact, there may well be no general method for treating the problem and impossibility results would also be valuable.

Logic and computability

Parallel to those motivated by concrete combinatorial problems, interest in complexity arose in the circles of logicians and recursion theorists.

A first quote is from a letter of K. Gödel to J. von Neumann of 20 March 1956. (The letter was reviewed by Hartmanis [1989], to whose attention it was brought by G. Heise. A reproduction and full translation was given by Sipser [1992].)

Turing [1937] proved that there is no algorithm that decides if a given statement in full first-order predicate logic has a proof (the unsolvability of Hilbert's *Entscheidungsproblem* of the *engere Funktionskalkül* (which is the term originally used by Hilbert for full first-order predicate calculus; Turing [1937] translated it into restricted functional calculus)). It implies the result of Gödel [1931] that there exist propositions A such that neither A nor $\neg A$ is provable (in the formalism of the Principia Mathematica).

But a *given* proof can algorithmically be checked, hence there is a finite algorithm to check if there exists a proof of any prescribed length n (simply by enumeration). Nowadays it is known that this is in fact an NP-complete problem (the satisfiability problem is a special case). Gödel asked for the opinion of von Neumann on whether a proof could be found algorithmically in time linear (or else quadratic) in the length of the proof — quite a bold statement, which Gödel yet seemed to consider plausible:

> Man kann offenbar leicht eine Turingmaschine konstruieren, welche von jeder Formel F des engeren Funktionenkalküls u. jeder natürl. Zahl n zu entscheiden gestattet ob F einen Beweis der Länge n hat [Länge = Anzahl der Symbole]. Sei $\psi(F, n)$ die Anzahl der Schritte die die Maschine dazu benötigt u. sei $\varphi(n) = \max_F \psi(F, n)$. Die Frage ist, wie rasch $\varphi(n)$ für eine optimale Maschine wächst. Man kann zeigen $\varphi(n) \geq Kn$. Wenn es wirklich eine Maschine mit $\varphi(n) \sim K.n$ (oder auch nur $\sim Kn^2$) gäbe, hätte das Folgerungen von der grössten Tragweite. Es würde nämlich offenbar bedeuten, dass man trotz der Unlösbarkeit des Entscheidungsproblems die Denkarbeit des Mathematikers bei ja-oder-nein Fragen vollständig* durch Maschinen ersetzen könnte. Man müsste ja bloss das n so gross wählen, dass, wenn die Maschine kein Resultat liefert es auch keinen Sinn hat über das Problem nachzudenken. Nun scheint es mir aber durchaus im Bereich der Möglichkeit zu liegen, dass $\varphi(n)$ so langsam wächst. Denn 1.) scheint $\varphi(n) \geq Kn$ die einzige Abschätzung zu sein, die man durch eine Verallgemeinerung des Beweises für die Unlösbarkeit des Entscheidungsproblems erhalten kann; 2. bedeutet ja $\varphi(n) \sim K.n$ (oder $\sim Kn^2$) bloss, dass die Anzahl der Schritte gegenüber <u>dem blossen Probieren</u> von N auf $\log N$ (oder $(\log N)^2$) verringert werden kann. So starke Verringerungen kommen aber bei andern finiten Problemen durchaus vor, z.B. bei der Berechnung eines quadratischen Restsymbols durch wiederholte Anwendung des Reziprozitätsgesetzes. Es wäre interessant zu wissen, wie es damit z.B. bei der Feststellung, ob eine Zahl Primzahl ist, steht u. wie stark <u>im allgemeinen</u> bei finiten kombinatorischen Problemen die Anzahl der Schritte gegenüber dem blossen Probieren verringert werden kann.
>
> * abgesehen von der Aufstellung der Axiome[6]

[6] Clearly, one can easily construct a Turing machine, which makes it possible to decide, for each formula F of the restricted functional calculus and each natural number n, whether F has a proof of length n [length = number of symbols]. Let $\psi(F, n)$ be the number of steps that the machine needs for that and let $\varphi(n) = \max_F \psi(F, n)$. The question is, how fast $\varphi(n)$ grows for an optimal machine. One can show $\varphi(n) \geq Kn$.

(For integers a, p with p prime, the Legendre symbol $(\frac{a}{p})$ indicates if a is a quadratic residue mod p (that is, if $x^2 = a \pmod{p}$ has an integer solution x), and can be calculated by $\log a + \log p$ arithmetic operations (using the Jacobi symbol and the reciprocity law) — so Gödel took the logarithms of the numbers as size.)

The unavoidability of brute-force search for finding the smallest Boolean representation for a function was claimed by Yablonskiĭ [1959] (cf. Trakhtenbrot [1984]).

Davis and Putnam [1960] gave a method for the satisfiability problem (in reaction to earlier, exponential-time methods of Gilmore [1960] and Wang [1960] based on elimination of variables), which they claimed to have some (not exactly formulated) efficiency:

> In the present paper, a uniform proof procedure for quantification theory is given which is feasible for use with some rather complicated formulas and which does not ordinarily lead to exponentiation.

(It was noticed later by Cook [1971] that Davis and Putnam's method gives a polynomial-time method for the 2-satisfiability problem.)

A mathematical framework for computational complexity of algorithms was set up by Hartmanis and Stearns [1965]. They counted the number of steps made by a multitape Turing machine to solve a decision problem. They showed that for all 'real-time countable' functions f, g (which include all functions n^k, k^n, $n!$, and sums, products, and compositions of them) the following holds: if each problem solvable in time $O(f)$ is also solvable in time $O(g)$, then $f = O(g^2)$. This implies, for instance, that there exist problems solvable in time $O(n^5)$ but not in time $O(n^2)$, and problems solvable in time $O(2^n)$ but not in time $O(2^{n/3})$ (hence not in polynomial time).

Polynomial-time

In the summer of 1963, at a Workshop at the RAND Corporation, Edmonds discovered that shrinking leads to a polynomial-time algorithm to find a maximum-size matching in any graph — a basic result in graph algorithmics. It was described in the paper Edmonds [1965d] (received November 22, 1963), in which he also gave his views on algorithms and complexity:

When really there were a machine with $\varphi(n) \sim K.n$ (or even just $\sim Kn^2$), that would have consequences of the largest impact. In particular, it would obviously mean that, despite the unsolvability of the Entscheidungsproblem, one could replace the brainwork of the mathematician in case of yes-or-no questions fully* by machines. One should indeed only choose n so large that if the machine yields no result, there is also no sense in thinking about the problem. Now it seems to me, however, to lie completely within the range of possibility that $\varphi(n)$ grows that slowly. Because 1.) $\varphi(n) \geq Kn$ seems to be the only estimate that one can obtain by a generalization of the proof for the unsolvability of the Entscheidungsproblem; 2. $\varphi(n) \sim K.n$ (or $\sim Kn^2$) means indeed only that the number of steps can be reduced compared to mere trying from N to $\log N$ (or $(\log N)^2$). Such strong reductions occur however definitely at other finite problems, e.g. at the calculation of a quadratic residue symbol by repeated application of the reciprocity law. It would be interesting to know how this is e.g. for the decision if a number is prime, and how strong in general, for finite combinatorial problems, the number of steps can be reduced compared to mere trying.

* apart from the set-up of the axioms

> For practical purposes computational details are vital. However, my purpose is only to show as attractively as I can that there is an efficient algorithm. According to the dictionary, "efficient" means "adequate in operation or performance." This is roughly the meaning I want—in the sense that it is conceivable for maximum matching to have no efficient algorithm. Perhaps a better word is "good."
> I am claiming, as a mathematical result, the existence of a *good* algorithm for finding a maximum size matching in a graph.
> There is an obvious finite algorithm, but that algorithm increases in difficulty exponentially with the size of the graph. It is by no means obvious whether *or not* there exists an algorithm whose difficulty increases only algebraically with the size of the graph.

Moreover:

> For practical purposes the difference between algebraic and exponential order is often more crucial than the difference between finite and non-finite.

In another paper, Edmonds [1965c] introduced the term *good characterization*:

> We seek a good characterization of the minimum number of independent sets into which the columns of a matrix of M_F can be partitioned. As the criterion of "good" for the characterization we apply the "principle of the absolute supervisor." The good characterization will describe certain information about the matrix which the supervisor can require his assistant to search out along with a minimum partition and which the supervisor can then use with ease to verify with mathematical certainty that the partition is indeed minimum. Having a good characterization does not mean necessarily that there is a good algorithm. The assistant might have to kill himself with work to find the information and the partition.

Further motivation for polynomial-time solvability was given by Edmonds [1967b]:

> An algorithm which is good in the sense used here is not necessarily very good from a practical viewpoint. However, the good-versus-not-good dichotomy is useful. It is easily formalized (say, relative to a Turing machine, or relative to a typical digital computer with an unlimited supply of tape), and usually it is easily recognized informally. Within limitations it does have practical value, and it does admit refinements to "how good" and "how bad". The classes of problems which are respectively known and not known to have good algorithms are very interesting theoretically.

Edmonds [1967a] conjectured that there is no polynomial-time algorithm for the traveling salesman problem — in language developed later, this is equivalent to NP$\neq$P:

> I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same as for any mathematical conjecture: (1) It is a legitimate mathematical possibility, and (2) I do not know.

Also Cobham [1965] singled out polynomial-time as a complexity criterion, in a paper on Turing machines and computability, presented at the 1964 International Congress on Logic, Methodology and Philosophy of Science in Jerusalem (denoting the size of n by $l(n)$):

> To obtain some idea as to how we might go about the further classification of relatively simple functions, we might take a look at how we ordinarily set about computing some of the more common of them. Suppose, for example, that m and n are two numbers given in decimal notation with one written above the other and their right ends aligned. Then to add m and n we start at the right and

> proceed digit-by-digit to the left writing down the sum. No matter how large m and n, this process terminates with the answer after a number of steps equal at most to one greater than the larger of $l(m)$ and $l(n)$. Thus the process of adding m and n can be carried out in a number of steps which is bounded by a linear polynomial in $l(m)$ and $l(n)$. Similarly, we can multiply m and n in a number of steps bounded by a quadratic polynomial in $l(m)$ and $l(n)$. So, too, the number of steps involved in the extraction of square roots, calculation of quotients, etc., can be bounded by polynomials in the lengths of the numbers involved, and this seems to be a property of simple functions in general. This suggests that we consider the class, which I will call $\mathcal{L}$, of all functions having this property.

At a symposium in New York in 1966, also Rabin [1967] noted the importance of polynomial-time solvability:

> In the following we shall consider an algorithm to be practical if, for automata with n states, it requires at most cn^k (k is a fixed integer and c a fixed constant) computational steps. This stipulation is, admittedly, both vague and arbitrary. We do not, in fact cannot, define what is meant by a computational step, thus have no precise and general measure for the complexity of algorithms. Furthermore, there is no compelling reason to classify algorithms requiring cn^k steps as practical.
> Several points may be raised in defense of the above stipulation. In every given algorithm the notion of a computational step is quite obvious. Hence there is not much vagueness about the measure of complexity of existing algorithms. Another significant pragmatic fact is that all existing algorithms either require up to about n^4 steps or else require 2^n or worse steps. Thus drawing the line of practicality between algorithms requiring n^k steps and algorithms for which no such bound exists seems to be reasonable.

NP-completeness

Cook [1971] proved the NP-completeness of the satisfiability problem ('Theorem 1') and of the 3-satisfiability problem and the subgraph problem ('Theorem 2') and mentioned (the class of polynomial-time solvable problems is denoted by $\mathcal{L}_*$; $\{$ tautologies $\}$ is the satisfiability problem):

> Theorem 1 and its corollary give strong evidence that it is not easy to determine whether a given proposition formula is a tautology, even if the formula is in normal disjunctive form. Theorems 1 and 2 together suggest that it is fruitless to search for a polynomial decision procedure for the subgraph problem, since success would bring polynomial decision procedures to many other apparently intractable problems. Of course, the same remark applies to any combinatorial problem to which $\{$ tautologies $\}$ is P-reducible.
> Furthermore, the theorems suggest that $\{$ tautologies $\}$ is a good candidate for an interesting set not in $\mathcal{L}_*$, and I feel it is worth spending considerable effort trying to prove this conjecture. Such a proof would be a major breakthrough in complexity theory.

So Cook conjectured that NP$\neq$P.

Also Levin [1973] considered the distinction between NP and P:

> After the concept of the algorithm had been fully refined, the algorithmic unsolvability of a number of classical large-scale problems was proved (including the problems of the identity of elements of groups, the homeomorphism of varieties, the solvability of the Diophantine equations, etc.). These findings dispensed with the question of finding a practical technique for solving the indicated problems. However, the existence of algorithms for the solution of other problems does not

> eliminate the analogous question, because the volume of work mandated by those algorithms is fantastically large. This is the situation with so-called sequential (or exhaustive) search problems, including: the minimization of Boolean functions, the search for proofs of finite length, the determination of the isomorphism of graphs, etc. All of these problems are solved by trivial algorithms entailing the sequential scanning of all possibilities. The operating time of the algorithms, however, is exponential, and mathematicians nurture the conviction that it is impossible to find simpler algorithms.

Levin next announced that any problem in NP (in his terminology, any 'sequential search problem') can be reduced to the satisfiability problem, and to a few other problems.

The wide extent of NP-completeness was disclosed by Karp [1972b], by showing that a host of prominent combinatorial problems is NP-complete, therewith revealing the fissure in the combinatorial optimization landscape. According to Karp, his theorems

> strongly suggest, but do not imply, that these problems, as well as many others, will remain intractable perpetually.

Karp also introduced the notation P and NP, and in a subsequent paper, Karp [1975] introduced the term NP-complete.

Sipser [1992] gave an extensive account on the history of the P=NP question. Hartmanis [1989] reviewed the historic setting of 'Gödel, von Neumann and the P=?NP Problem'. Other papers on the history of complexity are Hartmanis [1981], Trakhtenbrot [1984] (Russian approaches), Karp [1986], and Iri [1987] (the Japanese view).

Chapter 5

Preliminaries on polyhedra and linear and integer programming

This chapter surveys what we need on polyhedra and linear and integer programming. Most background can be found in Chapters 7–10, 14, 16, 19, 22, and 23 of Schrijver [1986b]. We give proofs of a few easy further results that we need in later parts of the present book.
The results of this chapter are mostly formulated for real space, but are maintained when restricted to rational space. So the symbol $\mathbb{R}$ can be replaced by the symbol $\mathbb{Q}$. In applying these results, we add the adjective *rational* when we restrict ourselves to rational numbers.

5.1. Convexity and halfspaces

A subset C of $\mathbb{R}^n$ is *convex* if $\lambda x + (1-\lambda)y$ belongs to C for all $x, y \in C$ and each λ with $0 \leq \lambda \leq 1$. A *convex body* is a compact convex set.

The *convex hull* of a set $X \subseteq \mathbb{R}^n$, denoted by conv.hull$X$, is the smallest convex set containing X. Then:

(5.1) $$\text{conv.hull}X = \{\lambda_1 x_1 + \cdots + \lambda_k x_k \mid k \geq 1, x_1, \ldots, x_k \in X, \lambda_1, \ldots, \lambda_k \in \mathbb{R}_+, \lambda_1 + \cdots + \lambda_k = 1\}.$$

A useful fundamental result was proved by Carathéodory [1911]:

Theorem 5.1 (Carathéodory's theorem)**.** *For any $X \subseteq \mathbb{R}^n$ and $x \in$ conv.hull$X$, there exist affinely independent vectors $x_1, \ldots, x_k$ in X with $x \in$ conv.hull$\{x_1, \ldots, x_k\}$.*

(Corollary 7.1f in Schrijver [1986b].)

A subset H of $\mathbb{R}^n$ is called an *affine halfspace* if $H = \{x \mid c^{\mathsf{T}} x \leq \delta\}$, for some $c \in \mathbb{R}^n$ with $c \neq \mathbf{0}$ and some $\delta \in \mathbb{R}$. If $\delta = 0$, then H is called a *linear halfspace*.

Let $X \subseteq \mathbb{R}^n$. The set conv.hull$X + \mathbb{R}^n_+$ is called the *up hull* of X, and the set conv.hull$X - \mathbb{R}^n_+$ the *down hull* of X.

5.2. Cones

A subset C of $\mathbb{R}^n$ is called a *(convex) cone* if $C \neq \emptyset$ and $\lambda x + \mu y \in C$ whenever $x, y \in C$ and $\lambda, \mu \in \mathbb{R}_+$. The cone *generated by* a set X of vectors is the smallest cone containing X:

(5.2) $\quad \text{cone}X = \{\lambda_1 x_1 + \cdots + \lambda_k x_k \mid k \geq 0, \lambda_1, \ldots, \lambda_k \geq 0, x_1, \ldots, x_k \in X\}.$

There is a variant of Carathéodory's theorem:

Theorem 5.2. *For any $X \subseteq \mathbb{R}^n$ and $x \in \text{cone}X$, there exist linearly independent vectors $x_1, \ldots, x_k$ in X with $x \in \text{cone}\{x_1, \ldots, x_k\}$.*

A cone C is *polyhedral* if there is a matrix A such that

(5.3) $\quad C = \{x \mid Ax \leq \mathbf{0}\}.$

Equivalently, C is polyhedral if it is the intersection of finitely many linear halfspaces.

Results of Farkas [1898,1902], Minkowski [1896], and Weyl [1935] imply that

(5.4) $\quad$ a convex cone is polyhedral if and only if it is finitely generated,

where a cone C is *finitely generated* if $C = \text{cone}X$ for some finite set X. (Corollary 7.1a in Schrijver [1986b].)

5.3. Polyhedra and polytopes

A subset P of $\mathbb{R}^n$ is called a *polyhedron* if there exists an $m \times n$ matrix A and a vector $b \in \mathbb{R}^m$ (for some $m \geq 0$) such that

(5.5) $\quad P = \{x \mid Ax \leq b\}.$

So P is a polyhedron of and only if it is the intersection of finitely many affine halfspaces. If (5.5) holds, we say that $Ax \leq b$ *determines* P. Any inequality $c^{\mathsf{T}}x \leq \delta$ is called *valid* for P if $c^{\mathsf{T}}x \leq \delta$ holds for each $x \in P$.

A subset P of $\mathbb{R}^n$ is called a *polytope* if it is the convex hull of finitely many vectors in $\mathbb{R}^n$. Motzkin [1936] showed:

(5.6) $\quad$ a set P is a polyhedron if and only if $P = Q + C$ for some polytope Q and some cone C.

(Corollary 7.1b in Schrijver [1986b].) If $P \neq \emptyset$, then C is unique and is called the *characteristic cone* char.cone(P) of P. Then:

(5.7) $\quad \text{char.cone}(P) = \{y \in \mathbb{R}^n \mid \forall x \in P \forall \lambda \geq 0 : x + \lambda y \in P\}.$

If $P = \emptyset$, then by definition its characteristic cone is char.cone$(P) := \{\mathbf{0}\}$.

(5.6) implies the following fundamental result (Minkowski [1896], Steinitz [1916], Weyl [1935]):

(5.8) a set P is a polytope if and only if P is a bounded polyhedron.

(Corollary 7.1c in Schrijver [1986b].)

A polyhedron P is called *rational* if it is determined by a rational system of linear inequalities. Then a rational polytope is the convex hull of a finite number of rational vectors.

5.4. Farkas' lemma

A system $Ax \leq b$ is called *feasible* (or *solvable*) if it has a solution x. Feasibility of a system $Ax \leq b$ of linear inequalities is characterized by *Farkas' lemma* (Farkas [1894,1898], Minkowski [1896]):

Theorem 5.3 (Farkas' lemma)**.** *$Ax \leq b$ is feasible $\iff$ $y^{\mathsf{T}}b \geq 0$ for each $y \geq \mathbf{0}$ with $y^{\mathsf{T}}A = \mathbf{0}^{\mathsf{T}}$.*

(Corollary 7.1e in Schrijver [1986b].) Theorem 5.3 is equivalent to:

Corollary 5.3a (Farkas' lemma — variant)**.** *$Ax = b$ has a solution $x \geq \mathbf{0}$ $\iff$ $y^{\mathsf{T}}b \geq 0$ for each y with $y^{\mathsf{T}}A \geq \mathbf{0}^{\mathsf{T}}$.*

(Corollary 7.1d in Schrijver [1986b].) A second equivalent variant is:

Corollary 5.3b (Farkas' lemma — variant)**.** *$Ax \leq b$ has a solution $x \geq \mathbf{0}$ $\iff$ $y^{\mathsf{T}}b \geq 0$ for each $y \geq \mathbf{0}$ with $y^{\mathsf{T}}A \geq \mathbf{0}^{\mathsf{T}}$.*

(Corollary 7.1f in Schrijver [1986b].) A third equivalent, affine variant of Farkas' lemma is:

Corollary 5.3c (Farkas' lemma — affine variant)**.** *Let $Ax \leq b$ be a feasible system of inequalities and let $c^{\mathsf{T}}x \leq \delta$ be an inequality satisfied by each x with $Ax \leq b$. Then for some $\delta' \leq \delta$, the inequality $c^{\mathsf{T}}x \leq \delta'$ is a nonnegative linear combination of the inequalities in $Ax \leq b$.*

(Corollary 7.1h in Schrijver [1986b].)

5.5. Linear programming

Linear programming, abbreviated to *LP*, concerns the problem of maximizing or minimizing a linear function over a polyhedron. Examples are

(5.9) $\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$ and $\min\{c^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \geq b\}$.

If a supremum of a linear function over a polyhedron is finite, then it is attained as a maximum. So a maximum is finite if the value set is nonempty and has an upper bound. Similarly for infimum and minimum.

The *duality theorem of linear programming* says (von Neumann [1947], Gale, Kuhn, and Tucker [1951]):

Theorem 5.4 (duality theorem of linear programming)**.** *Let A be a matrix and b and c be vectors. Then*

(5.10) $$\max\{c^{\mathsf{T}}x \mid Ax \leq b\} = \min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\},$$

if at least one of these two optima is finite.

(Corollary 7.1g in Schrijver [1986b].) So, in particular, if at least one of the optima is finite, then both are finite.

Note that the inequality $\leq$ in (5.10) is easy, since $c^{\mathsf{T}}x = y^{\mathsf{T}}Ax \leq y^{\mathsf{T}}b$. This is called *weak duality.*

There are several equivalent forms of the duality theorem of linear programming, like

(5.11) $$\begin{aligned} &\max\{c^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq b\} = \min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq c^{\mathsf{T}}\},\\ &\max\{c^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax = b\} = \min\{y^{\mathsf{T}}b \mid y^{\mathsf{T}}A \geq c^{\mathsf{T}}\},\\ &\min\{c^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \geq b\} = \max\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \leq c^{\mathsf{T}}\},\\ &\min\{c^{\mathsf{T}}x \mid Ax \geq b\} = \max\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\}. \end{aligned}$$

Any of these equalities holds if at least one of the two optima is finite (implying that both are finite).

A most general formulation is: let $A, B, C, D, E, F, G, H, K$ be matrices and let a, b, c, d, e, f be vectors; then

(5.12) $$\begin{aligned} &\max\{d^{\mathsf{T}}x + e^{\mathsf{T}}y + f^{\mathsf{T}}z \mid x \geq \mathbf{0}, z \leq \mathbf{0},\\ &Ax + By + Cz \leq a,\\ &Dx + Ey + Fz = b,\\ &Gx + Hy + Kz \geq c\}\\ &= \min\{u^{\mathsf{T}}a + v^{\mathsf{T}}b + w^{\mathsf{T}}c \mid u \geq \mathbf{0}, w \leq \mathbf{0},\\ &u^{\mathsf{T}}A + v^{\mathsf{T}}D + w^{\mathsf{T}}G \geq d^{\mathsf{T}},\\ &u^{\mathsf{T}}B + v^{\mathsf{T}}E + w^{\mathsf{T}}H = e^{\mathsf{T}},\\ &u^{\mathsf{T}}C + v^{\mathsf{T}}F + w^{\mathsf{T}}K \leq f^{\mathsf{T}}\}, \end{aligned}$$

provided that at least one of the two optima is finite (cf. Section 7.4 in Schrijver [1986b]).

So there is a one-to-one relation between constraints in a problem and variables in its dual problem. The objective function in one problem becomes the right-hand side in the dual problem. We survey these relations in the following table:

maximize	minimize
$\leq$ constraint	variable ≥ 0
$\geq$ constraint	variable ≤ 0
$=$ constraint	unconstrained variable
variable ≥ 0	$\geq$ constraint
variable ≤ 0	$\leq$ constraint
unconstrained variable	$=$ constraint
right-hand side	objective function
objective function	right-hand side

Some LP terminology. Linear programming concerns maximizing or minimizing a linear function $c^{\mathsf{T}}x$ over a polyhedron P. The polyhedron P is called the *feasible region*, and any vector in P a *feasible solution*. If the feasible region is nonempty, the problem is called *feasible*, and *infeasible* otherwise. The function $x \to c^{\mathsf{T}}x$ is called the *objective function* or the *cost function*. Any feasible solution attaining the optimum value is called an *optimum solution*. An inequality $c^{\mathsf{T}}x \leq \delta$ is called *tight* or *active* for some x^* if $c^{\mathsf{T}}x^* = \delta$.

Equations like (5.10), (5.11), and (5.12) are called *linear programming duality equations*. The minimization problem is called the *dual problem* of the maximization problem (which problem then is called the *primal problem*), and conversely. A feasible solution of the dual problem is called a *dual solution*.

Complementary slackness. The following *complementary slackness conditions* characterize optimality of a pair of feasible solutions x, y of the linear programs (5.10):

(5.13) x and y are optimum solutions if and only if $(Ax)_i = b_i$ for each i with $y_i > 0$.

Similar conditions can be formulated for other pairs of dual linear programs (cf. Section 7.9 in Schrijver [1986b]).

Carathéodory's theorem. A consequence of Carathéodory's theorem (Theorem 5.1 above) is:

Theorem 5.5. *If the optimum value in the LP problems* (5.10) *is finite, then the minimum is attained by a vector* $y \geq \mathbf{0}$ *such that the rows of* A *corresponding to positive components of* y *are linearly independent.*

(Corollary 7.1l in Schrijver [1986b].)

5.6. Faces, facets, and vertices

Let $P = \{x \mid Ax \leq b\}$ be a polyhedron in $\mathbb{R}^n$. If c is a nonzero vector and $\delta = \max\{c^{\mathsf{T}}x \mid Ax \leq b\}$, the affine hyperplane $\{x \mid c^{\mathsf{T}}x = \delta\}$ is called a *supporting hyperplane* of P. A subset F of P is called a *face* if $F = P$ or if $F = P \cap H$ for some supporting hyperplane H of P. So

(5.14) F is a face of $P \iff F$ is the set of optimum solutions of $\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$ for some $c \in \mathbb{R}^n$.

An inequality $c^{\mathsf{T}}x \leq \delta$ is said to *determine* or to *induce* face F of P if

(5.15) $F = \{x \in P \mid c^{\mathsf{T}}x = \delta\}$.

Alternatively, F is a face of P if and only if

(5.16) $F = \{x \in P \mid A'x = b'\}$

for some subsystem $A'x \leq b'$ of $Ax \leq b$ (cf. Section 8.3 in Schrijver [1986b]). So any face of a nonempty polyhedron is a nonempty polyhedron. We say that a constraint $a^{\mathsf{T}}x \leq \beta$ from $Ax \leq b$ is *tight* or *active* in a face F if $a^{\mathsf{T}}x = \beta$ holds for each $x \in F$.

An inequality $a^{\mathsf{T}}x \leq \beta$ from $Ax \leq b$ is called an *implicit equality* if $Ax \leq b$ implies $a^{\mathsf{T}}x = \beta$. Then:

Theorem 5.6. *Let $P = \{x \mid Ax \leq b\}$ be a polyhedron in $\mathbb{R}^n$. Let $A'x \leq b'$ be the subsystem of implicit inequalities in $Ax \leq b$. Then* $\dim P = n - \operatorname{rank} A'$.

(Cf. Section 8.2 in Schrijver [1986b].)

A *facet* of P is an inclusionwise maximal face F of P with $F \neq P$. An inequality determining a facet is called *facet-determining* or *facet-inducing*. Any facet has dimension one less than the dimension of P.

A system $Ax \leq b$ is called *minimal* or *irredundant* if each proper subsystem $A'x \leq b'$ has a solution x not satisfying $Ax \leq b$. If $Ax \leq b$ is irredundant and P is full-dimensional, then $Ax \leq b$ is the unique minimal system determining P, up to multiplying inequalities by positive scalars.

If $Ax \leq b$ is irredundant, then there is a one-to-one relation between the facets F of P and those inequalities $a^{\mathsf{T}}x \leq \beta$ in $Ax \leq b$ that are not implicit equalities, given by:

(5.17) $F = \{x \in P \mid a^{\mathsf{T}}x = \beta\}$

(cf. Theorem 8.1 in Schrijver [1986b]). This implies that each face $F \neq P$ is the intersection of facets.

A face of $P = \{x \mid Ax \leq b\}$ is called a *minimal face* if it is an inclusionwise minimal face. Any minimal face is an affine subspace of $\mathbb{R}^n$, and all minimal faces of P are translates of each other. They all have dimension $n - \operatorname{rank} A$.

If each minimal face has dimension 0, P is called *pointed*. A *vertex* of P is an element z such that $\{z\}$ is a minimal face. A polytope is the convex hull of its vertices.

For any element z of $P = \{x \mid Ax \leq b\}$, let $A_z x \leq b_z$ be the system consisting of those inequalities from $Ax \leq b$ that are satisfied by z with equality. Then:

Theorem 5.7. *Let $P = \{x \mid Ax \leq b\}$ be a polyhedron in $\mathbb{R}^n$ and let $z \in P$. Then z is a vertex of P if and only if* $\operatorname{rank}(A_z) = n$.

An *edge* of P is a bounded face of dimension 1. It necessarily connects two vertices of P. Two vertices connected by an edge are called *adjacent*. An *extremal ray* is a face of dimension 1 that forms a halfline.

The 1-*skeleton* of a pointed polyhedron P is the union of the vertices, edges, and extremal rays of P. If P is a polytope, the 1-skeleton is a topological graph. The *diameter* of P is the diameter of the associated (combinatorial) graph.

The *Hirsch conjecture* states that a d-dimensional polytope with m facets has diameter at most $m - d$. Naddef [1989] proved this for polytopes with $0, 1$ vertices. We refer to Kalai [1997] for a survey of bounds on the diameter and on the number of pivot steps in linear programming.

5.7. Polarity

(For the results of this section, see Section 9.1 in Schrijver [1986b].) For any subset C of $\mathbb{R}^n$, the *polar* of C is

(5.18) $\quad C^* := \{z \in \mathbb{R}^n \mid x^{\mathsf{T}} z \leq 1 \text{ for all } x \in C\}.$

If C is a cone, then C^* is again a cone, the *polar cone* of C, and satisfies

(5.19) $\quad C^* := \{z \in \mathbb{R}^n \mid x^{\mathsf{T}} z \leq 0 \text{ for all } x \in C\}.$

Let C be a polyhedral cone; so $C = \{x \mid Ax \leq \mathbf{0}\}$ for some matrix A. Trivially, if C is generated by the vectors $x_1, \ldots, x_k$, then C^* is equal to the cone determined by the inequalities $x_i^{\mathsf{T}} z \leq 0$ for $i = 1, \ldots, k$. It is less trivial, and can be derived from Farkas' lemma, that:

(5.20) the polar cone C^* is equal to the cone generated by the transposes of the rows of A.

This implies

(5.21) $\quad C^{**} = C$ for each polyhedral cone C.

So there is a symmetric duality relation between finite sets of vectors generating a cone and finite sets of vectors generating its polar cone.

5.8. Blocking polyhedra

(For the results of this section, see Section 9.2 in Schrijver [1986b].) A duality relation similar to polarity holds between convex sets 'of blocking type', and also between convex sets 'of antiblocking type'. This was shown by Fulkerson [1970b,1971a,1972a], who found several applications in combinatorial optimization.

We say that a subset P of $\mathbb{R}^n$ is *up-monotone* if $x \in P$ and $y \geq x$ imply $y \in P$. Similarly, P is *down-monotone* if $x \in P$ and $y \leq x$ imply $y \in P$.

Moreover, P is *down-monotone in* $\mathbb{R}^n_+$ if $x \in P$ and $\mathbf{0} \leq y \leq x$ imply $y \in P$. For any $P \subseteq \mathbb{R}^n$ we define

(5.22) $$P^{\uparrow} := \{y \in \mathbb{R}^n \mid \exists x \in P : y \geq x\} = P + \mathbb{R}^n_+ \text{ and}$$
$$P^{\downarrow} := \{y \in \mathbb{R}^n \mid \exists x \in P : y \leq x\} = P - \mathbb{R}^n_+.$$

$P^{\uparrow}$ is called the *dominant* of P. So P is up-monotone if and only if $P = P^{\uparrow}$, and P is down-monotone if and only if $P = P^{\downarrow}$.

We say that a convex set $P \subseteq \mathbb{R}^n$ is of *blocking type* if P is a closed convex up-monotone subset of $\mathbb{R}^n_+$. Each polyhedron P of blocking type is pointed. Moreover, P is a polyhedron of blocking type if and only if there exist vectors $x_1, \ldots, x_k \in \mathbb{R}^n_+$ such that

(5.23) $$P = \text{conv.hull}\{x_1, \ldots, x_k\}^{\uparrow};$$

and also, if and only if

(5.24) $$P = \{x \in \mathbb{R}^n_+ \mid Ax \geq \mathbf{1}\}$$

for some nonnegative matrix A.

For any polyhedron P in $\mathbb{R}^n$, the *blocking polyhedron* $B(P)$ of P is defined by

(5.25) $$B(P) := \{z \in \mathbb{R}^n_+ \mid x^{\mathsf{T}} z \geq 1 \text{ for each } x \in P\}.$$

Fulkerson [1970b,1971a] showed:

Theorem 5.8. *Let $P \subseteq \mathbb{R}^n_+$ be a polyhedron of blocking type. Then $B(P)$ is again a polyhedron of blocking type and $B(B(P)) = P$. Moreover, for any $x_1, \ldots, x_k \in \mathbb{R}^n_+$:*

(5.26) (5.23) *holds if and only if* $B(P) = \{z \in \mathbb{R}^n_+ \mid x_i^{\mathsf{T}} z \geq 1 \text{ for } i = 1, \ldots, k\}$.

Here the only if part is trivial, while the if part requires Farkas' lemma.

Theorem 5.8 implies that for vectors $x_1, \ldots, x_k \in \mathbb{R}^n_+$ and $z_1, \ldots, z_d \in \mathbb{R}^n_+$ one has:

(5.27) $$\text{conv.hull}\{x_1, \ldots, x_k\} + \mathbb{R}^n_+ = \{x \in \mathbb{R}^n_+ \mid z_j^{\mathsf{T}} x \geq 1 \text{ for } j = 1, \ldots, d\}$$

if and only if

(5.28) $$\text{conv.hull}\{z_1, \ldots, z_d\} + \mathbb{R}^n_+ = \{z \in \mathbb{R}^n_+ \mid x_i^{\mathsf{T}} z \geq 1 \text{ for } i = 1, \ldots, k\}.$$

Two polyhedra P, R are called a *blocking pair (of polyhedra)* if they are of blocking type and satisfy $R = B(P)$. So if P, R is a blocking pair, then so is R, P.

5.9. Antiblocking polyhedra

(For the results of this section, see Section 9.3 in Schrijver [1986b].) The theory of antiblocking polyhedra is almost fully analogous to the blocking case and arises mostly by reversing inequality signs.

We say that a set $P \subseteq \mathbb{R}^n$ is of *antiblocking type* if P is a nonempty closed convex subset of $\mathbb{R}^n_+$ that is down-monotone in $\mathbb{R}^n_+$. Then P is a polyhedron of antiblocking type if and only if

(5.29) $\quad P = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$

for some nonnegative matrix A and nonnegative vector b.

For any subset P of $\mathbb{R}^n$, the *antiblocking set* $A(P)$ of P is defined by

(5.30) $\quad A(P) := \{z \in \mathbb{R}^n_+ \mid x^{\mathsf{T}} z \leq 1 \text{ for each } x \in P\}.$

If $A(P)$ is a polyhedron we speak of the *antiblocking polyhedron*, and if $A(P)$ is a convex body, of the *antiblocking body.*

Fulkerson [1971a,1972a] showed:

Theorem 5.9. *Let $P \subseteq \mathbb{R}^n_+$ be of antiblocking type. Then $A(P)$ is again of antiblocking type and $A(A(P)) = P$.*

The antiblocking analogue of (5.26) is a little more complicated to formulate, but we need it only for full-dimensional polytopes. For any full-dimensional polytope $P \subseteq \mathbb{R}^n$ of antiblocking type and $x_1, \ldots, x_k \in \mathbb{R}^n_+$ we have:

(5.31) $\quad P = \text{conv.hull}\{x_1, \ldots, x_k\}^{\downarrow} \cap \mathbb{R}^n_+$ if and only if $A(P) = \{z \in \mathbb{R}^n_+ \mid x_i^{\mathsf{T}} z \leq 1 \text{ for } i = 1, \ldots, k\}$.

Two convex sets P, R are called an *antiblocking pair (of polyhedra)* if they are of antiblocking type and satisfy $R = A(P)$. So if P, R is an antiblocking pair, then so is R, P.

5.10. Methods for linear programming

The simplex method was designed by Dantzig [1951b] to solve linear programming problems. It is in practice and on average quite efficient, but no polynomial-time worst-case running time bound has been proved (most of the pivot selection rules that have been proposed have been proved to take exponential time in the worst case).

The simplex method consists of finding a path in the 1-skeleton of the feasible region, ending at an optimum vertex (in preprocessing, the problem first is transformed to one with a pointed feasible region). An important issue when implementing this is that the LP problem is not given by vertices and edges, but by linear inequalities, and that vertices are determined by a, not necessarily unique, ‘basis’ among the inequalities.

The first polynomial-time method for linear programming was given by Khachiyan [1979,1980], by adapting the ‘ellipsoid method’ for nonlinear programming of Shor [1970a,1970b,1977] and Yudin and Nemirovskiĭ [1976]. The method consists of finding a sequence of shrinking ellipsoids each containing at least one optimum solution, until we have an ellipsoid that is small enough so as to derive an optimum solution. The method however is practically quite infeasible.

Karmarkar [1984a,1984b] showed that ‘interior point’ methods can solve linear programming in polynomial time, and moreover that they have efficient implementations, competing with the simplex method. Interior point methods make a tour not along vertices and edges, but across the feasible region.

5.11. The ellipsoid method

While the ellipsoid method is practically infeasible, it turned out to have features that are useful for deriving complexity results in combinatorial optimization. Specifically, the ellipsoid method does not require listing all constraints of an LP problem a priori, but allows that they are generated when needed. In this way, one can derive the polynomial-time solvability of a number of combinatorial optimization problems. This should be considered as existence proofs of polynomial-time algorithms — the algorithms are not practical.

This application of the ellipsoid method was described by Karp and Papadimitriou [1980,1982], Padberg and Rao [1980], and Grötschel, Lovász, and Schrijver [1981]. The book by Grötschel, Lovász, and Schrijver [1988] is devoted to it. We refer to Chapter 6 of this book or to Chapter 14 of Schrijver [1986b] for proofs of the results that we survey below.

The ellipsoid method applies to classes of polyhedra (and more generally, classes of convex sets) which are described as follows.

Let Σ be a finite alphabet and let Π be a subset of the set Σ^* of words over Σ. In applications, we take for Π very simple sets like the set of strings representing a graph or the set of strings representing a digraph.

For each $\sigma \in \Pi$, let E_σ be a finite set and let P_σ be a rational polyhedron in $\mathbb{Q}^{E_\sigma}$. (When we apply this, E_σ is often the vertex set or the edge or arc set of the (di)graph represented by σ.) We make the following assumptions:

(5.32)
(i) there is a polynomial-time algorithm that, given $\sigma \in \Sigma^*$, tests if σ belongs to Π and, if so, returns the set E_σ;
(ii) there is a polynomial p such that, for each $\sigma \in \Pi$, P_σ is determined by linear inequalities each of size at most $p(\text{size}(\sigma))$.

Here the *size* of a rational linear inequality is proportional to the sum of the sizes of its components, where the *size* of a rational number p/q (for integers

$p, q)$ is proportional to $\log(|p| + 1) + \log q$. Condition (5.32)(ii) is equivalent to (cf. Theorem 10.2 in Schrijver [1986b]):

(5.33) there is a polynomial q such that, for each $\sigma \in \Pi$, we can write $P_\sigma = Q + C$, where Q is a polytope with vertices each of input size at most $q(\text{size}(\sigma))$ and where C is a cone generated by vectors each of input size at most $q(\text{size}(\sigma))$.

(The *input size*[7] of a vector is the sum of the sizes of its components.) In most applications, the existence of the polynomial p in (5.32)(ii) or of the polynomial q in (5.33) is obvious.

We did not specify how the polyhedra P_σ are given algorithmically. In applications, they might have an exponential number of vertices or facets, so listing them would not be an algorithmic option. To handle this, we formulate two, in a sense dual, problems. An algorithm for either of them would determine the polyhedra P_σ.

First, the *optimization problem for* $(P_\sigma \mid \sigma \in \Pi)$ is the problem:

(5.34) given: $\sigma \in \Pi$ and $c \in \mathbb{Q}^{E_\sigma}$,
find: $x \in P_\sigma$ maximizing $c^\mathsf{T} x$ over P_σ or $y \in \text{char.cone}(P_\sigma)$ with $c^\mathsf{T} y > 0$, if either of them exists.

Second, the *separation problem for* $(P_\sigma \mid \sigma \in \Pi)$ is the problem:

(5.35) given: $\sigma \in \Pi$ and $z \in \mathbb{Q}^{E_\sigma}$,
find: $c \in \mathbb{Q}^{E_\sigma}$ such that $c^\mathsf{T} x < c^\mathsf{T} z$ for all $x \in P_\sigma$ (if such a c exists).

So c gives a separating hyperplane if $z \notin P_\sigma$.

Then the ellipsoid method implies that these two problems are 'polynomial-time equivalent':

Theorem 5.10. *Let* $\Pi \subseteq \Sigma^*$ *and let* $(P_\sigma \mid \sigma \in \Pi)$ *satisfy* (5.32). *Then the optimization problem for* $(P_\sigma \mid \sigma \in \Pi)$ *is polynomial-time solvable if and only if the separation problem for* $(P_\sigma \mid \sigma \in \Pi)$ *is polynomial-time solvable.*

(Cf. Theorem (6.4.9) in Grötschel, Lovász, and Schrijver [1988] or Corollary 14.1c in Schrijver [1986b].)

The equivalence in Theorem 5.10 makes that we call $(P_\sigma \mid \sigma \in \Pi)$ *polynomial-time solvable* if it satisfies (5.32) and the optimization problem (equivalently, the separation problem) for it is polynomial-time solvable.

Using simultaneous diophantine approximation based on the basis reduction method given by Lenstra, Lenstra, and Lovász [1982], Frank and Tardos [1985,1987] extended these results to strong polynomial-time solvability:

[7] We will use the term *size* of a vector for the sum of its components.

Theorem 5.11. *The optimization problem and the separation problem for any polynomial-time solvable system of polyhedra are solvable in strongly polynomial time.*

(Theorem (6.6.5) in Grötschel, Lovász, and Schrijver [1988].)

For polynomial-time solvable classes of polyhedra, the separation problem can be strengthened so as to obtain a facet as separating hyperplane:

Theorem 5.12. *Let* $(P_\sigma \mid \sigma \in \Pi)$ *be a polynomial-time solvable system of polyhedra. Then the following problem is strongly polynomial-time solvable:*

(5.36) given: $\sigma \in \Pi$ and $z \in \mathbb{Q}^{E_\sigma}$,
find: $c \in \mathbb{Q}^{E_\sigma}$ and $\delta \in \mathbb{Q}$ such that $c^{\mathsf{T}}z > \delta$ and such that $c^{\mathsf{T}}x \leq \delta$ is facet-inducing for P_σ (if it exists).

(Cf. Theorem (6.5.16) in Grötschel, Lovász, and Schrijver [1988].) Also a weakening of the separation problem turns out to be equivalent, under certain conditions. The *membership problem for* $(P_\sigma \mid \sigma \in \Pi)$ is the problem:

(5.37) given $\sigma \in \Pi$ and $z \in \mathbb{Q}^{E_\sigma}$, does z belong to P_σ?

Theorem 5.13. *Let* $(P_\sigma \mid \sigma \in \Pi)$ *be a system of full-dimensional polytopes satisfying* (5.32)*, such that there is a polynomial-time algorithm that gives for each* $\sigma \in \Pi$ *a vector in the interior of* P_σ*. Then* $(P_\sigma \mid \sigma \in \Pi)$ *is polynomial-time solvable if and only if the membership problem for* $(P_\sigma \mid \sigma \in \Pi)$ *is polynomial-time solvable.*

(This follows from Corollary (4.3.12) and Theorem (6.3.2) in Grötschel, Lovász, and Schrijver [1988].)

The theorems above imply:

Theorem 5.14. *Let* $(P_\sigma \mid \sigma \in \Pi)$ *and* $(Q_\sigma \mid \sigma \in \Pi)$ *be polynomial-time solvable classes of polyhedra, such that for each* $\sigma \in \Pi$*, the polyhedra* P_σ *and* Q_σ *are in the same space* $\mathbb{R}^{E_\sigma}$*. Then also* $(P_\sigma \cap Q_\sigma \mid \sigma \in \Pi)$ *and* $(\mathrm{conv.hull}(P_\sigma \cup Q_\sigma) \mid \sigma \in \Pi)$ *are polynomial-time solvable.*

(Corollary 14.1d in Schrijver [1986b].)

Corollary 5.14a. *Let* $(P_\sigma \mid \sigma \in \Pi)$ *be a polynomial-time solvable system of polyhedra, all of blocking type. Then also the system of blocking polyhedra* $(B(P_\sigma) \mid \sigma \in \Pi)$ *is polynomial-time solvable.*

(Corollary 14.1e in Schrijver [1986b].) Similarly:

Corollary 5.14b. *Let* $(P_\sigma \mid \sigma \in \Pi)$ *be a polynomial-time solvable system of polyhedra, all of antiblocking type. Then also the system of antiblocking polyhedra* $(A(P_\sigma) \mid \sigma \in \Pi)$ *is polynomial-time solvable.*

(Corollary 14.1e in Schrijver [1986b].)

Also the following holds:

Theorem 5.15. *Let $(P_\sigma \mid \sigma \in \Pi)$ be a polynomial-time solvable system of polyhedra, where each P_σ is a polytope. Then the following problems are strongly polynomial-time solvable:*

(5.38) (i) *given $\sigma \in \Pi$, find an internal vector, a vertex, and a facet-inducing inequality of P_σ;*
(ii) *given $\sigma \in \Pi$ and $x \in P_\sigma$, find affinely independent vertices $x_1, \ldots, x_k$ of P_σ and write x as a convex combination of $x_1, \ldots, x_k$;*
(iii) *given $\sigma \in \Pi$ and $c \in \mathbb{R}^{E_\sigma}$, find facet-inducing inequalities $c_1^\mathsf{T} x \leq \delta_1, \ldots, c_k^\mathsf{T} x \leq \delta_k$ of P_σ with $c_1, \ldots, c_k$ linearly independent, and find $\lambda_1, \ldots, \lambda_k \geq 0$ such that $\lambda_1 c_1 + \cdots + \lambda_k c_k = c$ and $\lambda_1 \delta_1 + \cdots + \lambda_k \delta_k = \max\{c^\mathsf{T} x \mid x \in P_\sigma\}$ (i.e., find an optimum dual solution).*

(Corollary 14.1f in Schrijver [1986b].)

The ellipsoid method can be applied also to nonpolyhedral convex sets, in which case only approximative versions of the optimization and separation problems can be shown to be equivalent. We only need this in Chapter 67 on the convex body $\mathrm{TH}(G)$, where we refer to the appropriate theorem in Grötschel, Lovász, and Schrijver [1988].

5.12. Polyhedra and NP and co-NP

An appropriate polyhedral description of a combinatorial optimization problem relates to the question NP≠co-NP. More precisely, unless NP=co-NP, the polyhedra associated with an NP-complete problem cannot be described by 'certifiable' inequalities. (These insights go back to observations in the work of Edmonds of the 1960s.)

Again, let $(P_\sigma \mid \sigma \in \Pi)$ be a system of polyhedra satisfying (5.32). Consider the decision version of the optimization problem:

(5.39) given $\sigma \in \Pi$, $c \in \mathbb{Q}^{E_\sigma}$, and $k \in \mathbb{Q}$, is there an $x \in P_\sigma$ with $c^\mathsf{T} x > k$?

Then:

Theorem 5.16. *Problem* (5.39) *belongs to* co-NP *if and only if for each $\sigma \in \Pi$, there exists a collection $\mathcal{I}_\sigma$ of inequalities determining P_σ such that the problem:*

(5.40) *given $\sigma \in \Pi$, $c \in \mathbb{Q}^{E_\sigma}$, and $\delta \in \mathbb{Q}$, does $c^\mathsf{T} x \leq \delta$ belong to $\mathcal{I}_\sigma$,* *belongs to* NP.

Proof. To see necessity, we can take for $\mathcal{I}_\sigma$ the collection of *all* valid inequalities for P_σ. Then co-NP-membership of (5.39) is equivalent of NP-membership of (5.40).

To see sufficiency, a negative answer to question (5.39) can be certified by giving inequalities $c_i^\top x \leq \delta_i$ from $\mathcal{I}_\sigma$ and $\lambda_i \in \mathbb{Q}_+$ $(i = 1, \ldots, k)$ such that $c = \lambda_1 c_1 + \cdots + \lambda_k c_k$ and $\delta \geq \lambda_1 \delta_1 + \cdots + \lambda_k \delta_k$. As we can take $k \leq |E_\sigma|$, and as each inequality in $\mathcal{I}_\sigma$ has a polynomial-time checkable certificate (as (5.40) belongs to NP), this gives a polynomial-time checkable certificate for the negative answer. Hence (5.39) belongs to co-NP. ∎

This implies for NP-complete problems:

Corollary 5.16a. *Let* (5.39) *be* NP*-complete and suppose* NP≠co-NP. *For each* $\sigma \in \Pi$, *let* $\mathcal{I}_\sigma$ *be a collection of inequalities determining* P_σ. *Then problem* (5.40) *does not belong to* NP.

Proof. If problem (5.40) would belong to NP, then by Theorem 5.16, problem (5.39) belongs to co-NP. If (5.39) is NP-complete, this implies NP=co-NP. ∎

Roughly speaking, this implies that if (5.39) is NP-complete and NP≠co-NP, then P_σ has 'difficult' facets, that is, facets which have no polynomial-time checkable certificate of validity for P_σ.

(Related work on the complexity of facets was reported in Karp and Papadimitriou [1980,1982] and Papadimitriou and Yannakakis [1982,1984].)

5.13. Primal-dual methods

As a generalization of similar methods for network flow and transportation problems, Dantzig, Ford, and Fulkerson [1956] designed the 'primal-dual method' for linear programming. The general idea is as follows. Starting with a dual feasible solution y, the method searches for a primal feasible solution x satisfying the complementary slackness condition with respect to y. If such a primal feasible solution x is found, x and y form a pair of optimum solutions (by (5.13)). If no such primal solution is found, the method prescribes a modification of y, after which the method iterates.

The problem now is how to find a primal feasible solution x satisfying the complementary slackness condition, and how to modify the dual solution y if no such primal solution is found. For general linear programs this problem can be seen to amount to another linear program, generally simpler than the original linear program. To solve the simpler linear program we could use any LP method. In many combinatorial applications, however, this simpler linear program is a simpler combinatorial optimization problem, for which direct methods are available. Thus, if we can describe a combinatorial optimization problem as a linear program, the primal-dual method gives us a scheme for reducing one combinatorial problem to an easier combinatorial problem. The efficiency of the method depends on the complexity of the easier problem and on the number of primal-dual iterations.

We describe the primal-dual method more precisely. Suppose that we wish to solve the LP problem

(5.41) $\min\{c^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax = b\}$,

where A is an $m \times n$ matrix, with columns $a_1, \ldots, a_n$, and where $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. The dual problem is

(5.42) $\max\{y^{\mathsf{T}}b \mid y^{\mathsf{T}}A \leq c^{\mathsf{T}}\}$.

The primal-dual method consists of repeating the following *primal-dual iteration*. Suppose that we have a feasible solution y_0 for problem (5.42). Let A' be the submatrix of A consisting of those columns a_j of A for which $y_0^{\mathsf{T}}a_j = c_j$ holds. To find a feasible primal solution satisfying the complementary slackness, solve the *restricted linear program*

(5.43) $x' \geq \mathbf{0}, A'x' = b$.

If such an x' exists, by adding components 0, we obtain a vector $x \geq \mathbf{0}$ such that $Ax = b$ and such that $x_j = 0$ if $y_0^{\mathsf{T}}a_j < c_j$. By complementary slackness ((5.13)), it follows that x and y_0 are optimum solutions for problems (5.41) and (5.42).

On the other hand, if no x' satisfying (5.43) exists, by Farkas' lemma (Corollary 5.3a), there exists a y' such that $y'^{\mathsf{T}}A' \leq 0$ and $y'^{\mathsf{T}}b > 0$. Let α be the largest real number satisfying

(5.44) $(y_0 + \alpha y')^{\mathsf{T}}A \leq c^{\mathsf{T}}$.

(Note that $\alpha > 0$.) Reset $y_0 := y_0 + \alpha y'$, and start the iteration anew. (If $\alpha = \infty$, (5.42) is unbounded, hence (5.41) is infeasible.)

This describes the primal-dual method. It reduces problem (5.41) to (5.43), which often is an easier problem.

The primal-dual method can equally well be considered as a *gradient method*. Suppose that we wish to solve problem (5.42), and we have a feasible solution y_0. This y_0 is not optimum if and only if there exists a vector y' such that $y'^{\mathsf{T}}b > 0$ and y' is a *feasible direction* at y_0 (that is, $(y_0 + \alpha y')^{\mathsf{T}}A \leq c^{\mathsf{T}}$ for some $\alpha > 0$). If we let A' consist of those columns of A in which $y_0^{\mathsf{T}}A \leq c^{\mathsf{T}}$ has equality, then y' is a feasible direction if and only if $y'^{\mathsf{T}}A' \leq 0$. So y' can be found by solving (5.43).

5.14. Integer linear programming

A vector $x \in \mathbb{R}^n$ is called *integer* if each component is an integer, i.e., if x belongs to $\mathbb{Z}^n$. Many combinatorial optimization problems can be described as maximizing a linear function $c^{\mathsf{T}}x$ over the *integer* vectors in some polyhedron $P = \{x \mid Ax \leq b\}$.

So this type of problems can be described as:

(5.45) $$\max\{c^{\mathsf{T}}x \mid Ax \leq b; x \in \mathbb{Z}^n\}.$$

Such problems are called *integer linear programming*, or *ILP*, problems. They consist of maximizing a linear function over the intersection $P \cap \mathbb{Z}^n$ of a polyhedron P with the set $\mathbb{Z}^n$ of integer vectors.

Clearly, always the following inequality holds:

(5.46) $$\max\{c^{\mathsf{T}}x \mid Ax \leq b; x \text{ integer}\} \leq \max\{c^{\mathsf{T}}x \mid Ax \leq b\}.$$

It is easy to make an example where strict inequality holds. This implies, that generally one will have strict inequality in the following duality relation:

(5.47) $$\max\{c^{\mathsf{T}}x \mid Ax \leq b; x \text{ integer}\} \\ \leq \min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}; y^{\mathsf{T}}A = c^{\mathsf{T}}; y \text{ integer}\}.$$

No polynomial-time algorithm is known to exist for solving an integer linear programming problem in general. In fact, the general integer linear programming problem is NP-complete (since the satisfiability problem is easily transformed to an integer linear programming problem). However, for special classes of integer linear programming problems, polynomial-time algorithms have been found. These classes often come from combinatorial problems.

5.15. Integer polyhedra

A polyhedron P is called an *integer polyhedron* if it is the convex hull of the integer vectors contained in P. This is equivalent to: P is rational and each face of P contains an integer vector. So a polytope P is integer if and only if each vertex of P is integer. If a polyhedron $P = \{x \mid Ax \leq b\}$ is integer, then the linear programming problem

(5.48) $$\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$$

has an integer optimum solution if it is finite. Hence, in that case,

(5.49) $$\max\{c^{\mathsf{T}}x \mid Ax \leq b;\ x \text{ integer}\} = \max\{c^{\mathsf{T}}x \mid Ax \leq b\}.$$

This in fact characterizes integer polyhedra, since:

Theorem 5.17. *Let P be a rational polyhedron in $\mathbb{Q}^n$. Then P is integer if and only if for each $c \in \mathbb{Q}^n$, the linear programming problem* $\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$ *has an integer optimum solution if it is finite.*

A stronger characterization is (Edmonds and Giles [1977]):

Theorem 5.18. *A rational polyhedron P in $\mathbb{Q}^n$ is integer if and only if for each $c \in \mathbb{Z}^n$ the value of* $\max\{c^{\mathsf{T}}x \mid x \in P\}$ *is an integer if it is finite.*

(Corollary 22.1a in Schrijver [1986b].) We also will use the following observation:

Theorem 5.19. *Let P be an integer polyhedron in $\mathbb{R}^n_+$ with $P + \mathbb{R}^n_+ = P$ and let $c \in \mathbb{Z}^n_+$ be such that $x \leq c$ for each vertex x of P. Then $P \cap \{x \mid x \leq c\}$ is an integer polyhedron again.*

Proof. Let $Q := P \cap \{x \mid x \leq c\}$ and let R be the convex hull of the integer vectors in Q. We must show that $Q \subseteq R$.

Let $x \in Q$. As $P = R + \mathbb{R}^n_+$ there exists a $y \in R$ with $y \leq x$. Choose such a y with $y_1 + \cdots + y_n$ maximal. Suppose that $y_i < x_i$ for some component i. Since $y \in R$, y is a convex combination of integer vectors in Q. Since $y_i < x_i \leq c_i$, at least one of these integer vectors, z say, has $z_i < c_i$. But then the vector $z' := z + \chi^i$ belongs to R. Hence we could increase y_i, contradicting the maximality of y. ∎

We call a polyhedron P *box-integer* if $P \cap \{x \mid d \leq x \leq c\}$ is an integer polyhedron for each choice of integer vectors d, c. The set $\{x \mid d \leq x \leq c\}$ is called a *box*.

A $0,1$ *polytope* is a polytope with all vertices being 0,1 vectors.

5.16. Totally unimodular matrices

Total unimodularity of matrices is an important tool in integer programming. A matrix A is called *totally unimodular* if each square submatrix of A has determinant equal to 0, +1, or −1. In particular, each entry of a totally unimodular matrix is 0, +1, or −1.

An alternative way of characterizing total unimodularity is by requiring that the matrix is integer and that each nonsingular submatrix has an integer inverse matrix. This implies the following easy, but fundamental result:

Theorem 5.20. *Let A be a totally unimodular $m \times n$ matrix and let $b \in \mathbb{Z}^m$. Then the polyhedron*

(5.50) $$P := \{x \mid Ax \leq b\}$$

is integer.

(Cf. Theorem 19.1 in Schrijver [1986b].) It follows that each linear programming problem with integer data and totally unimodular constraint matrix has integer optimum primal and dual solutions:

Corollary 5.20a. *Let A be a totally unimodular $m \times n$ matrix, let $b \in \mathbb{Z}^m$, and let $c \in \mathbb{Z}^n$. Then both optima in the LP duality equation*

(5.51) $$\max\{c^{\mathsf{T}}x \mid Ax \leq b\} = \min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\}$$

have integer optimum solutions (if the optima are finite).

(Corollary 19.1a in Schrijver [1986b].) Hoffman and Kruskal [1956] showed that this property is close to a characterization of total unimodularity.

Corollary 5.20a implies:

Corollary 5.20b. *Let A be an $m \times n$ matrix, let $b \in \mathbb{Z}^m$, and let $c \in \mathbb{R}^n$. Suppose that*

(5.52) $$\max\{c^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq b\}$$

has an optimum solution x^* such that the columns of A corresponding to positive components of x^* form a totally unimodular matrix. Then (5.52) *has an integer optimum solution.*

Proof. Since x^* is an optimum solution, we have

(5.53) $$\max\{c^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq b\} = \max\{c'^{\mathsf{T}}x' \mid x' \geq \mathbf{0}, A'x' \leq b\},$$

where A' and c' are the parts of A and c corresponding to the support of x^*. As A' is totally unimodular, the right-hand side maximum in (5.53) has an integer optimum solution x'^*. Extending x'^* by components 0, we obtain an integer optimum solution of the left-hand side maximum in (5.53). ∎

We will use the following characterization of Ghouila-Houri [1962b] (cf. Theorem 19.3 in Schrijver [1986b]):

Theorem 5.21. *A matrix M is totally unimodular if and only if each collection R of rows of M can be partitioned into classes R_1 and R_2 such that the sum of the rows in R_1, minus the sum of the rows in R_2, is a vector with entries $0, \pm 1$ only.*

5.17. Total dual integrality

Edmonds and Giles [1977] introduced the powerful notion of total dual integrality. It is not only useful as a tool to derive combinatorial min-max relation, but also it gives an efficient way of expressing a whole bunch of min-max relations simultaneously.

A system $Ax \leq b$ in n dimensions is called *totally dual integral*, or just *TDI*, if A and b are rational and for each $c \in \mathbb{Z}^n$, the dual of maximizing $c^{\mathsf{T}}x$ over $Ax \leq b$:

(5.54) $$\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\}$$

has an integer optimum solution y, if it is finite.

By extension, a system $A'x \leq b', A''x = b''$ is defined to be TDI if the system $A'x \leq b', A''x \leq b'', -A''x \leq -b''$ is TDI. This is equivalent to requiring that A', A'', b', b'' are rational and for each $c \in \mathbb{Z}^n$ the dual of maximizing $c^{\mathsf{T}}x$ over $A'x \leq b'$, $A''x = b''$ has an integer optimum solution, if finite.

Problem (5.54) is the problem dual to $\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$, and Edmonds and Giles showed that total dual integrality implies that also this primal problem has an integer optimum solution, if b is integer. In fact, they showed Theorem 5.18, which implies (since if (5.54) has an integer optimum solution, the optimum value is an integer):

Theorem 5.22. *If $Ax \leq b$ is TDI and b is integer, then $Ax \leq b$ determines an integer polyhedron.*

So total dual integrality implies 'primal integrality'. For combinatorial applications, the following observation is useful:

Theorem 5.23. *Let A be a nonnegative integer $m \times n$ matrix such that the system $x \geq \mathbf{0}, Ax \geq \mathbf{1}$ is TDI. Then also the system $\mathbf{0} \leq x \leq \mathbf{1}, Ax \geq \mathbf{1}$ is TDI.*

Proof. Choose $c \in \mathbb{Z}^n$. Let c_+ arise from c by setting negative components to 0. By the total dual integrality of $x \geq \mathbf{0}$, $Ax \geq \mathbf{1}$, there exist integer optimum solutions x, y of

(5.55) $$\min\{c_+^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \geq \mathbf{1}\} = \max\{y^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \leq c_+^{\mathsf{T}}\}.$$

As A is nonnegative and integer and as $c_+ \geq \mathbf{0}$, we may assume that $x \leq \mathbf{1}$. Moreover, we can assume that $x_i = 1$ if $(c_+)_i = 0$, that is, if $c_i \leq 0$.

Let $z := c - c_+$. So $z \leq \mathbf{0}$. We show that x, y, z are optimum solutions of

(5.56) $$\begin{aligned}&\min\{c^{\mathsf{T}}x \mid \mathbf{0} \leq x \leq \mathbf{1}, Ax \geq \mathbf{1}\}\\ &= \max\{y^{\mathsf{T}}\mathbf{1} + z^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, z \leq \mathbf{0}, y^{\mathsf{T}}A + z^{\mathsf{T}} \leq c^{\mathsf{T}}\}.\end{aligned}$$

Indeed, x is feasible, as $\mathbf{0} \leq x \leq \mathbf{1}$ and $Ax \geq \mathbf{1}$. Moreover, y, z is feasible, as $y^{\mathsf{T}}A + z^{\mathsf{T}} \leq c_+^{\mathsf{T}} + z^{\mathsf{T}} = c^{\mathsf{T}}$. Optimality of x, y, z follows from

(5.57) $$c^{\mathsf{T}}x = c_+^{\mathsf{T}}x + z^{\mathsf{T}}x = y^{\mathsf{T}}\mathbf{1} + z^{\mathsf{T}}x = y^{\mathsf{T}}\mathbf{1} + z^{\mathsf{T}}\mathbf{1}.$$ ∎

In certain cases, to obtain total dual integrality one can restrict oneself to nonnegative objective functions:

Theorem 5.24. *Let A be a nonnegative $m \times n$ matrix and let $b \in \mathbb{R}_+^m$. Then $x \geq \mathbf{0}, Ax \leq b$ is TDI if and only if $\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq c^{\mathsf{T}}\}$ is attained by an integer optimum solution (if finite), for each $c \in \mathbb{Z}_+^n$.*

Proof. Necessity is trivial. To see sufficiency, let $c \in \mathbb{Z}^n$ with $\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq c^{\mathsf{T}}\}$ finite. Let it be attained by y. Let c_+ arise from c by setting negative components to 0. Then

(5.58) $$\min\{y^\mathsf{T} b \mid y \geq \mathbf{0}, y^\mathsf{T} A \geq c_+^\mathsf{T}\} = \min\{y^\mathsf{T} b \mid y \geq \mathbf{0}, y^\mathsf{T} A \geq c^\mathsf{T}\},$$

since $y^\mathsf{T} A \geq \mathbf{0}$ if $y \geq \mathbf{0}$. As the first minimum has an integer optimum solution, also the second minimum has an integer optimum solution. ∎

Total dual integrality is maintained under setting an inequality to an equality (Theorem 22.2 in Schrijver [1986b]):

Theorem 5.25. *Let $Ax \leq b$ be TDI and let $A'x \leq b'$ arise from $Ax \leq b$ by adding $-a^\mathsf{T} x \leq -\beta$ for some inequality $a^\mathsf{T} x \leq \beta$ in $Ax \leq b$. Then also $A'x \leq b'$ is TDI.*

Total dual integrality is also maintained under translation of the solution set, as follows directly from the definition of total dual integrality:

Theorem 5.26. *If $Ax \leq b$ is TDI and $w \in \mathbb{R}^n$, then $Ax \leq b - Aw$ is TDI.*

For future reference, we prove:

Theorem 5.27. *Let $A_{11}, A_{12}, A_{21}, A_{22}$ be matrices and let b_1, b_2 be column vectors, such that the system*

(5.59) $$\begin{aligned} &A_{1,1}x_1 + A_{1,2}x_2 = b_1, \\ &A_{2,1}x_1 + A_{2,2}x_2 \leq b_2 \end{aligned}$$

is TDI and such that $A_{1,1}$ is nonsingular. Then also the system

(5.60) $$(A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2})x_2 \leq b_2 - A_{2,1}A_{1,1}^{-1}b_1$$

is TDI.

Proof. We may assume that $b_1 = \mathbf{0}$, since by Theorem 5.26 total dual integrality is invariant under replacing (5.59) by

(5.61) $$\begin{aligned} &A_{1,1}x_1 + A_{1,2}x_2 = b_1 - A_{1,1}A_{1,1}^{-1}b_1 = \mathbf{0}, \\ &A_{2,1}x_1 + A_{2,2}x_2 \leq b_2 - A_{2,1}A_{1,1}^{-1}b_1. \end{aligned}$$

Let x_2 minimize $c^\mathsf{T} x_2$ over (5.60), for some integer vector c of appropriate dimension. Define $x_1 := -A_{1,1}^{-1}A_{1,2}x_2$. Then x_1, x_2 minimizes $c^\mathsf{T} x_2$ over (5.59), since any solution x_1', x_2' of (5.59) satisfies $x_1' = -A_{1,1}^{-1}A_{1,2}x_2'$, and therefore x_2' satisfies (5.60); hence $c^\mathsf{T} x_2' \geq c^\mathsf{T} x_2$.

Let y_1, y_2 be an integer optimum solution of the problem dual to maximizing $c^\mathsf{T} x_2$ over (5.59). So y_1, y_2 satisfy

(5.62) $$y_1^\mathsf{T} A_{1,1} + y_2^\mathsf{T} A_{2,1} = \mathbf{0},\ y_1^\mathsf{T} A_{1,2} + y_2^\mathsf{T} A_{2,2} = c^\mathsf{T},\ y_2^\mathsf{T} b_2 = c^\mathsf{T} x_2.$$

Hence

(5.63) $$y_2^\mathsf{T}(A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}) = y_2^\mathsf{T} A_{2,2} + y_1^\mathsf{T} A_{1,2} = c^\mathsf{T}$$

and

(5.64) $y_2^{\mathsf{T}} b_2 = c^{\mathsf{T}} x_2.$

So y_2 is an integer optimum solution of the problem dual to maximizing $c^{\mathsf{T}} x_2$ over (5.60). ∎

This has as consequence (where a_0 is a column vector):

Corollary 5.27a. *If $x_0 = \beta$, $a_0 x_0 + Ax \leq b$ is TDI, then $Ax \leq b - \beta a_0$ is TDI.*

Proof. This is a special case of Theorem 5.27. ∎

We also have:

Theorem 5.28. *Let $A = [a_1 \;\; a_2 \;\; A'']$ be an integer $m \times n$ matrix and let $b \in \mathbb{R}^m$. Let A' be the $m \times (n-1)$ matrix $[a_1 + a_2 \;\; A'']$. Then $A'x' \leq b$ is TDI if and only if $Ax \leq b, x_1 - x_2 = 0$ is TDI.*

Proof. To see necessity, choose $c \in \mathbb{Z}^n$. Let $c' := (c_1 + c_2, c_3, \ldots, c_n)^{\mathsf{T}}$. Then

(5.65) $\mu := \max\{c^{\mathsf{T}} x \mid Ax \leq b, x_1 - x_2 = 0\} = \max\{c'^{\mathsf{T}} x' \mid A'x' \leq b\}.$

Let $y \in \mathbb{Z}_+^m$ be an integer optimum dual solution of the second maximum. So $y^{\mathsf{T}} A' = c'$ and $y^{\mathsf{T}} b = \mu$. Then $y^{\mathsf{T}} a_1 + y^{\mathsf{T}} a_2 = c_1 + c_2$. Hence $y^{\mathsf{T}} A = c^{\mathsf{T}} + \lambda(1, -1, 0, \ldots, 0)$ for some $\lambda \in \mathbb{Z}$. So y, λ form an integer optimum dual solution of the first maximum.

To see sufficiency, choose $c' = (c_2, \ldots, c_n)^{\mathsf{T}} \in \mathbb{Z}^{n-1}$. Define $c := (0, c_2, \ldots, c_n)^{\mathsf{T}}$. Again we have (5.65). Let $y \in \mathbb{Z}_+^m$, $\lambda \in \mathbb{Z}$ constitute an integer optimum dual solution of the first maximum, where λ corresponds to the constraint $x_1 - x_2 = 0$. So $y^{\mathsf{T}} A + \lambda(1, -1, 0, \ldots, 0) = c$ and $y^{\mathsf{T}} b = \mu$. Hence $y^{\mathsf{T}} A' = c^{\mathsf{T}}$, and therefore, y is an integer optimum dual solution of the second maximum. ∎

Let A be a rational $m \times n$ matrix and let $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$. Consider the following series of inequalities (where a vector z is *half-integer* if $2z$ is integer):

(5.66)
$$\begin{aligned} &\max\{c^{\mathsf{T}} x \mid Ax \leq b, x \text{ integer}\} \leq \max\{c^{\mathsf{T}} x \mid Ax \leq b\} \\ &= \min\{y^{\mathsf{T}} b \mid y \geq \mathbf{0}, y^{\mathsf{T}} A = c^{\mathsf{T}}\} \\ &\leq \min\{y^{\mathsf{T}} b \mid y \geq \mathbf{0}, y^{\mathsf{T}} A = c^{\mathsf{T}}, y \text{ half-integer}\} \\ &\leq \min\{y^{\mathsf{T}} b \mid y \geq \mathbf{0}, y^{\mathsf{T}} A = c^{\mathsf{T}}, y \text{ integer}\}. \end{aligned}$$

Under certain circumstances, equality in the last inequality implies equality throughout:

Theorem 5.29. *Let $Ax \leq b$ be a system with A and b rational. Then $Ax \leq b$ is TDI if and only if*

(5.67) $\min\{y^{\mathsf{T}} b \mid y \geq \mathbf{0}, y^{\mathsf{T}} A = c^{\mathsf{T}}, y \text{ half-integer}\}$

is finite and is attained by an integer optimum solution y, for each integer vector c with $\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$ *finite.*

Proof. Necessity follows directly from (5.66). To see sufficiency, choose $c \in \mathbb{Z}^n$ with $\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$ finite. We must show that $\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\}$ is attained by an integer optimum solution.

For each $k \geq 1$, define

(5.68) $$\alpha_k = \min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = kc^{\mathsf{T}}, y \text{ integer}\}.$$

This is well-defined, as $\max\{kc^{\mathsf{T}}x \mid Ax \leq b\}$ is finite.

The condition in the theorem gives that, for each $t \geq 0$,

(5.69) $$\frac{\alpha_{2^t}}{2^t} = \alpha_1.$$

This can be shown by induction on t, the case $t = 0$ being trivial. If $t \geq 1$, then

(5.70) $$\begin{aligned}\alpha_{2^t} &= \min\{y^{\mathsf{T}}b \mid y^{\mathsf{T}}A = 2^t c^{\mathsf{T}}, y \in \mathbb{Z}^m_+\}\\ &= 2\min\{y^{\mathsf{T}}b \mid y^{\mathsf{T}}A = 2^{t-1}c^{\mathsf{T}}, y \in \tfrac{1}{2}\mathbb{Z}^m_+\}\\ &= 2\min\{y^{\mathsf{T}}b \mid y^{\mathsf{T}}A = 2^{t-1}c^{\mathsf{T}}, y \in \mathbb{Z}^m_+\} = 2\alpha_{2^{t-1}},\end{aligned}$$

implying (5.69) by induction.

Now $\alpha_{k+l} \leq \alpha_k + \alpha_l$ for all k, l. Hence we can apply Fekete's lemma, and get:

(5.71) $$\begin{aligned}&\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\} = \min_k \frac{\alpha_k}{k} = \lim_{k\to\infty}\frac{\alpha_k}{k} = \lim_{t\to\infty}\frac{\alpha_{2^t}}{2^t}\\ &= \alpha_1.\end{aligned}$$ ∎

The following analogue of Carathéodory's theorem holds (Cook, Fonlupt, and Schrijver [1986]):

Theorem 5.30. *Let $Ax \leq b$ be a totally dual integral system in n dimensions and let $c \in \mathbb{Z}^n$. Then $\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq c^{\mathsf{T}}\}$ has an integer optimum solution y with at most $2n-1$ nonzero components.*

(Theorem 22.12 in Schrijver [1986b].)

We also will need the following substitution property:

Theorem 5.31. *Let $A_1x \leq b_1, A_2x \leq b_2$ be a TDI system with A_1 integer, and let $A_1' \leq b_1'$ be a TDI system with*

(5.72) $$\{x \mid A_1x \leq b_1\} = \{x \mid A_1'x \leq b_1'\}.$$

Then the system $A_1'x \leq b_1', A_2x \leq b_2$ is TDI.

Proof. Let $c \in \mathbb{Z}^n$ with

(5.73) $$\begin{aligned}&\max\{c^{\mathsf{T}}x \mid A_1'x \leq b_1', A_2x \leq b_2\}\\ &= \min\{y^{\mathsf{T}}b_1' + z^{\mathsf{T}}b_2 \mid y, z \geq \mathbf{0}, y^{\mathsf{T}}A_1' + z^{\mathsf{T}}A_2 = c^{\mathsf{T}}\}\end{aligned}$$

finite. By (5.72), also

(5.74) $$\max\{c^{\mathsf{T}}x \mid A_1x \leq b_1, A_2x \leq b_2\} \\ = \min\{y^{\mathsf{T}}b_1 + z^{\mathsf{T}}b_2 \mid y, z \geq \mathbf{0}, y^{\mathsf{T}}A_1 + z^{\mathsf{T}}A_2 = c^{\mathsf{T}}\}$$

is finite. Hence, since $A_1x \leq b_1, A_2x \leq b_2$ is TDI, the minimum in (5.74) has an integer optimum solution y, z. Set $d := y^{\mathsf{T}}A_1$. Then, as d is an integer vector,

(5.75) $$y^{\mathsf{T}}b_1 = \min\{u^{\mathsf{T}}b_1 \mid u \geq \mathbf{0}, u^{\mathsf{T}}A_1 = d^{\mathsf{T}}\} \\ = \max\{d^{\mathsf{T}}x \mid A_1x \leq b_1\} = \max\{d^{\mathsf{T}}x \mid A_1'x \leq b_1'\} \\ = \min\{v^{\mathsf{T}}b_1' \mid v \geq \mathbf{0}, v^{\mathsf{T}}A_1' = d^{\mathsf{T}}\}$$

is finite. Hence, since $A_1'x \leq b_1'$ is TDI, the last minimum in (5.75) has an integer optimum solution v. Then v, z is an integer optimum solution of the minimum in (5.73). ∎

A system $Ax \leq b$ is called *totally dual half-integral* if A and b are rational and for each $c \in \mathbb{Z}^n$, the dual of maximizing $c^{\mathsf{T}}x$ over $Ax \leq b$ has a half-integer optimum solution, if it is finite. Similarly, $Ax \leq b$ is called *totally dual quarter-integral* if A and b are rational and for each $c \in \mathbb{Z}^n$, the dual of maximizing $c^{\mathsf{T}}x$ over $Ax \leq b$ has a quarter-integer optimum solution y, if it is finite.

5.18. Hilbert bases and minimal TDI systems

For any $X \subseteq \mathbb{R}^n$ we denote

(5.76) $$\text{lattice}X := \{\lambda_1x_1 + \cdots + \lambda_kx_k \mid k \geq 0, \lambda_1, \ldots, \lambda_k \in \mathbb{Z}, x_1, \ldots, x_k \in X\}.$$

A subset L of $\mathbb{R}^n$ is called a *lattice* if $L = \text{lattice}X$ for some base X of $\mathbb{R}^n$. So for general X, lattice$X$ need not be a lattice.

The *dual lattice* of X is, by definition:

(5.77) $$\{x \in \mathbb{R}^n \mid y^{\mathsf{T}}x \in \mathbb{Z} \text{ for each } y \in X\}.$$

Again, this need not be a lattice in the proper sense.

A set X of vectors is called a *Hilbert base* if each vector in lattice$X \cap$cone$X$ is a nonnegative integer combination of vectors in X. The Hilbert base is called *integer* if it consists of integer vectors only.

One may show:

(5.78) Each rational polyhedral cone C is generated by an integer Hilbert base. If C is pointed, there exists a unique inclusionwise minimal integer Hilbert base generating C.

(Theorem 16.4 in Schrijver [1986b].)

There is a close relation between Hilbert bases and total dual integrality:

Theorem 5.32. *A rational system $Ax \leq b$ is TDI if and only if for each face F of $P := \{x \mid Ax \leq b\}$, the rows of A which are active in F form a Hilbert base.*

(Theorem 22.5 in Schrijver [1986b].)

(5.78) and Theorem 5.32 imply (Giles and Pulleyblank [1979], Schrijver [1981b]):

Theorem 5.33. *Each rational polyhedron P is determined by a TDI system $Ax \leq b$ with A integer. If moreover P is full-dimensional, there exists a unique minimal such system.*

(Theorem 22.6 in Schrijver [1986b].)

5.19. The integer rounding and decomposition properties

A system $Ax \leq b$ is said to have the *integer rounding property* if $Ax \leq b$ is rational and

(5.79) $$\begin{aligned}&\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}, y \text{ integer}\}\\ &= \lceil \min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\}\rceil\end{aligned}$$

for each integer vector c for which $\min\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A = c^{\mathsf{T}}\}$ is finite. So any TDI system has the integer rounding property.

A polyhedron P is said to have the *integer decomposition property* if for each natural number k, each integer vector in $k \cdot P$ is the sum of k integer vectors in P.

Baum and Trotter [1978] showed that an integer matrix A is totally unimodular if and only if the polyhedron $\{x \mid x \geq \mathbf{0}, Ax \leq b\}$ has the integer decomposition property for each integer vector b. In another paper, Baum and Trotter [1981] observed the following relation between the integer rounding and the integer decomposition property:

(5.80) Let A be a nonnegative integer matrix. Then the system $x \geq \mathbf{0}, Ax \geq \mathbf{1}$ has the integer rounding property if and only if the blocking polyhedron $B(P)$ of $P := \{x \mid x \geq \mathbf{0}, Ax \geq \mathbf{1}\}$ has the integer decomposition property and all minimal integer vectors in $B(P)$ are transposes of rows of A (minimal with respect to $\leq$).

Similarly,

(5.81) Let A be a nonnegative integer matrix. Then the system $x \geq \mathbf{0}, Ax \leq \mathbf{1}$ has the integer rounding property if and only if the antiblocking polyhedron $A(P)$ of $P := \{x \mid x \geq \mathbf{0}, Ax \leq \mathbf{1}\}$ has the integer decomposition property and all maximal integer vectors in $A(P)$ are transposes of rows of A (maximal with respect to $\leq$).

(Theorem 22.19 in Schrijver [1986b].)

5.20. Box-total dual integrality

A system $Ax \leq b$ is called *box-totally dual integral*, or just *box-TDI*, if the system $d \leq x \leq c, Ax \leq b$ is totally dual integral for each choice of vectors $d, c \in \mathbb{R}^n$. By Theorem 5.22,

(5.82) if $Ax \leq b$ is box-totally dual integral, then the polyhedron $\{x \mid Ax \leq b\}$ is box-integer.

We will need the following two results.

Theorem 5.34. *If $Ax \leq b$ is box-TDI in n dimensions and $w \in \mathbb{R}^n$, then $Ax \leq b - Aw$ is box-TDI.*

Proof. Directly from the definition of box-total dual integrality. ∎

Theorem 5.35. *Let $Ax \leq b$ be a system of linear inequalities, with A an $m \times n$ matrix. Suppose that for each $c \in \mathbb{R}^n$, $\max\{c^{\mathsf{T}}x \mid Ax \leq b\}$ has (if finite) an optimum dual solution $y \in \mathbb{R}^m_+$ such that the rows of A corresponding to positive components of y form a totally unimodular submatrix of A. Then $Ax \leq b$ is box-TDI.*

Proof. Choose $d, c \in \mathbb{R}^n$, with $d \leq c$, and choose $c \in \mathbb{Z}^n$. Consider the dual of maximizing $c^{\mathsf{T}}x$ over $Ax \leq b$, $d \leq x \leq c$:

(5.83) $$\min\{y^{\mathsf{T}}b + z_1^{\mathsf{T}}c - z_2^{\mathsf{T}}d \mid y \in \mathbb{R}^m_+, z_1, z_2 \in \mathbb{R}^n_+, y^{\mathsf{T}}A + z_1^{\mathsf{T}} - z_2^{\mathsf{T}} = c^{\mathsf{T}}\}.$$

Let y, z_1, z_2 attain this optimum. Define $c' := c - z_1 + z_2$. By assumption, $\min\{y'^{\mathsf{T}}b \mid y' \in \mathbb{R}^m_+, y'^{\mathsf{T}}A = c'^{\mathsf{T}}\}$ has an optimum solution such that the rows of A corresponding to positive components of y' form a totally unimodular matrix. Now y', z_1, z_2 is an optimum solution of (5.83). Also, the rows in $Ax \leq b$, $d \leq x \leq c$ corresponding to positive components of y', z_1, z_2 form a totally unimodular matrix. Hence by Corollary 5.20b, (5.83) has an integer optimum solution. ∎

5.21. The integer hull and cutting planes

Let P be a rational polyhedron. The *integer hull* P_{I} of P is the convex hull of the integer vectors in P:

(5.84) $\quad P_{\mathrm{I}} = \text{conv.hull}(P \cap \mathbb{Z}^n).$

It can be shown that P_{I} is a rational polyhedron again.

Consider any rational affine halfspace $H = \{x \mid c^{\mathsf{T}}x \leq \delta\}$, where c is a nonzero integer vector such that the g.c.d. of its components is equal to 1 and where $\delta \in \mathbb{Q}$. Then it is easy to show that

(5.85) $\quad H_{\mathrm{I}} = \{x \mid c^{\mathsf{T}}x \leq \lfloor\delta\rfloor\}.$

The inequality $c^{\mathsf{T}}x \leq \lfloor\delta\rfloor$ (or, more correctly, the hyperplane $\{x \mid c^{\mathsf{T}}x = \lfloor\delta\rfloor\}$) is called a *cutting plane*.

Define for any rational polyhedron P:

(5.86) $$P' := \bigcap_{H \supseteq P} H_{\mathrm{I}},$$

where H ranges over all rational affine halfspaces H containing P. Then P' is a rational polyhedron contained in P. Since $P \subseteq H$ implies $P_{\mathrm{I}} \subseteq H_{\mathrm{I}}$, we know

(5.87) $\quad P_{\mathrm{I}} \subseteq P' \subseteq P.$

For $k \in \mathbb{Z}_+$, define $P^{(k)}$ inductively by:

(5.88) $\quad P^{(0)} := P$ and $P^{(k+1)} := (P^{(k)})'.$

Then (Gomory [1958,1960], Chvátal [1973a], Schrijver [1980b]):

Theorem 5.36. *For each rational polyhedron there exists a* $k \in \mathbb{Z}_+$ *with* $P_{\mathrm{I}} = P^{(k)}$.

(For a proof, see Theorem 23.2 in Schrijver [1986b].)

5.21a. Background literature

Most background on polyhedra and linear and integer programming needed for this book can be found in Schrijver [1986b].

More background can be found in Dantzig [1963] (linear programming), Grünbaum [1967] (polytopes), Hu [1969] (integer programming), Garfinkel and Nemhauser [1972a] (integer programming), Brøndsted [1983] (polytopes), Chvátal [1983] (linear programming), Lovász [1986] (ellipsoid method), Grötschel, Lovász, and Schrijver [1988] (ellipsoid method), Nemhauser and Wolsey [1988] (integer programming), Padberg [1995] (linear programming), Ziegler [1995] (polytopes), and Wolsey [1998] (integer programming).

Part I

Paths and Flows

Paths belong to the most basic and important objects in combinatorial optimization. First of all, paths are of direct practical use, to make connections and to search. One can imagine that even in very primitive societies, finding short paths and searching (for instance, for food) is essential. 'Short' need not be just in terms of geometric distance, but might observe factors like differences in height, crosscurrent, and wind. In modern societies, searching is also an important issue in several kinds of networks, like in communication webs and data structures. Several other nonspatial problems (like the knapsack problem, dynamic programming) can be modelled as a shortest path problem. Also planning tools like PERT and CPM are based on shortest paths.

Next to that, paths form an important tool in solving other combinatorial optimization problems. In a large part of combinatorial algorithms, finding an appropriate path is the main issue in a subroutine or in the iterations. Several combinatorial optimization problems can be solved by iteratively finding a shortest path.

Disjoint paths were first investigated in a topological setting by Menger starting in the 1920s, leading to Menger's theorem, a min-max relation equating the maximum number of disjoint $s-t$ paths and the minimum size of an $s-t$ cut. The theorem is fundamental to graph theory, and provides an important tool to handle the connectivity of graphs.

In a different environment, the notion of *flow* in a graph came up, namely at RAND in the 1950s, motivated by a study of the capacity of the Soviet and East European railway system. It inspired Ford and Fulkerson to develop a maximum flow algorithm based on augmenting paths and to prove the max-flow min-cut theorem. As flows can be considered as linear combinations of incidence vectors of paths, there is a close connection between disjoint paths and flow problems, and it turned out that the max-flow min-cut theorem and Menger's theorem can be derived from each other.

Minimum-cost flows can be considered as the common generalization of shortest paths and disjoint paths/flows. Related are minimum-cost circulations and transshipments. This connects the topic to the origins of linear programming in the 1940s, when Koopmans designed pivot-like procedures for minimum-cost transshipment in order to plan protected ship convoys during World War II.

Actually, linear programming and polyhedral methods apply very favourably to path and flow problems, by the total unimodularity of the underlying constraint matrices. They lead to fast, strongly polynomial-time algorithms for such problems, while also fast direct, combinatorial algorithms have been found.

Chapters:

Chapter 6

Shortest paths: unit lengths

The first three chapters of this part are devoted to the shortest path problem. A number of simple but fundamental methods have been developed for it.
The division into the three chapters is by increasing generality of the length function. In the present chapter we take unit lengths; that is, each edge or arc has length 1. Equivalently, we search for paths with a minimum number of edges or arcs. We also consider 'zero length', equivalently, searching for *any* path.
Next, in Chapter 7, we consider nonnegative lengths, where Dijkstra's method applies. Finally, in Chapter 8, we go over to arbitrary lengths. If we put no further constraints, the shortest path problem is NP-complete (in fact, even if all lengths are -1). But if there are no negative-length directed circuits, the problem is polynomial-time solvable, by the Bellman-Ford method.
The methods and results in this chapter generally apply to directed and undirected graphs alike; however, in case of an undirected graph with length function such that each circuit has nonnegative length, the problem is polynomial-time, but the method is much more involved. It can be solved in polynomial time with nonbipartite matching methods, and for this we refer to Section 29.2.
In this chapter, graphs can be assumed to be simple.

6.1. Shortest paths with unit lengths

Let $D = (V, A)$ be a digraph. In this chapter, the length of any path in D is the number of its arcs. For $s, t \in V$, the *distance* from s to t is the minimum length of any $s-t$ path. If no $s-t$ path exists, we set the distance from s to t equal to ∞.

There is an easy min-max relation, due to Robacker [1956b], characterizing the minimum length of an $s-t$ path. Recall that a subset C of A is an $s-t$ *cut* if $C = \delta^{\text{out}}(U)$ for some subset U of V satisfying $s \in U$ and $t \notin U$.[1]

[1] $\delta^{\text{out}}(U)$ and $\delta^{\text{in}}(U)$ denote the sets of arcs leaving and entering U, respectively.

Theorem 6.1. *Let $D = (V, A)$ be a digraph and let $s, t \in V$. Then the minimum length of an $s-t$ path is equal to the maximum number of disjoint $s-t$ cuts.*

Proof. Trivially, the minimum is at least the maximum, since each $s-t$ path intersects each $s-t$ cut in at least one arc. That the minimum is equal to the maximum follows by considering the $s-t$ cuts $\delta^{\text{out}}(U_i)$ for $i = 1, \ldots, d$, where d is the distance from s to t and where U_i is the set of vertices of distance less than i from s. ∎

(This is the proof of Robacker [1956b].)

Dantzig [1957] observed the following. Let $D = (V, A)$ be a digraph and let $s \in V$. A rooted tree $T = (V', A')$ rooted at s is called a *shortest paths tree* (*rooted at* s) if V' is the set of vertices in D reachable from s and $A' \subseteq A$, such that for each $t \in V'$, the $s-t$ path in T is a shortest $s-t$ path in D.

Theorem 6.2. *Let $D = (V, A)$ be a digraph and let $s \in V$. Then there exists a shortest paths tree rooted at s.*

Proof. Let V' be the set of vertices reachable in D from s. Choose, for each $t \in V' \setminus \{s\}$, an arc a_t that is the last arc of some shortest $s-t$ path in D. Then $A' := \{a_t \mid t \in V' \setminus \{s\}\}$ gives the required rooted tree. ∎

The above trivially applies also to *undirected* graphs.

6.2. Shortest paths with unit lengths algorithmically: breadth-first search

The following algorithm of Berge [1958b] and Moore [1959], essentially *breadth-first search*, determines the distance from s to t. Let V_i denote the set of vertices of D at distance i from s. Then $V_0 = \{s\}$, and for each i:

(6.1) V_{i+1} is equal to the set of vertices $v \in V \setminus (V_0 \cup V_1 \cup \cdots \cup V_i)$ for which $(u, v) \in A$ for some $u \in V_i$.

This gives us directly an algorithm for determining the sets V_i: we set $V_0 := \{s\}$ and next we determine with rule (6.1) the sets $V_1, V_2, \ldots$ successively, until $V_{i+1} = \emptyset$.

In fact, it gives a linear-time algorithm, and so:

Theorem 6.3. *Given a digraph $D = (V, A)$ and $s, t \in V$, a unit-length shortest $s-t$ path can be found in linear time.*

Proof. Directly from the description. ∎

In fact, the algorithm finds the distance from s to all vertices reachable from s. Moreover, it gives the shortest paths, even the shortest paths tree:

Theorem 6.4. *Given a digraph $D = (V, A)$ and $s \in V$, a shortest path tree rooted at s can be found in linear time.*

Proof. Use the algorithm described above. ∎

6.3. Depth-first search

In certain cases it is more useful to scan a graph not by breadth-first search as in Section 6.2, but by *depth-first search* (a variant of which goes back to Tarry [1895]).

Let $D = (V, A)$ be a digraph. Define the operation of *scanning* a vertex v recursively by:

(6.2) For each arc $a = (v, w) \in \delta^{\text{out}}(v)$: delete all arcs entering w and scan w.

Then depth-first search from a vertex s amounts to scanning s. If each vertex of D is reachable from s, then all arcs a chosen in (6.2) form a rooted tree with root s. This tree is called a *depth-first search tree*.

This can be applied to find a path and to sort and order the vertices of a digraph $D = (V, A)$. We say that vertices $v_1, \ldots, v_n$ are in *topological order* if $i < j$ for all i, j with $(v_i, v_j) \in A$. So a subset of V can be topologically ordered only if it induces no directed circuit.

To grasp the case where directed circuits occur, we say that vertices $v_1, \ldots, v_n$ are in *pre-topological order* if for all i, j, if v_j is reachable from v_i and $j < i$, then v_i is reachable from v_j. So if D is acyclic, any pre-topological order is topological.

We can interpret a pre-topological order as a linear extension of the partial order $\prec$ defined on V by:

(6.3) $v \prec w \iff w$ is reachable from v, but v is not reachable from w.

Thus a pre-topological ordering is one satisfying: $v_i \prec v_j \Rightarrow i < j$.

The following was shown by Knuth [1968] and Tarjan [1974d] (cf. Kahn [1962]):

Theorem 6.5. *Given a digraph $D = (V, A)$ and $s \in V$, the vertices reachable from s can be ordered pre-topologically in time $O(m')$, where m' is the number of arcs reachable from s.*

Proof. Scan s. Then recursively all vertices reachable from s will be scanned, and the order in which we *finish* scanning them is the opposite of a pre-topological order: for vertices v, w reachable from s, if D has a $v-w$ path but no $w-v$ path, then scanning w is finished before scanning v. ∎

This implies:

Corollary 6.5a. *Given a digraph $D = (V, A)$, the vertices of D can be ordered pre-topologically in linear time.*

Proof. Add a new vertex s and arcs (s, v) for each $v \in V$. Applying Theorem 6.5 gives the required pre-topological ordering. ∎

For acyclic digraphs, it gives a topological order:

Corollary 6.5b. *Given an acyclic digraph $D = (V, A)$, the vertices of D can be ordered topologically in linear time.*

Proof. Directly from Corollary 6.5a, as a pre-topological order of the vertices of an acyclic graph is topological. ∎

(The existence of a topological order for an acyclic digraph is implicit in the work of Szpilrajn [1930].)

One can also use a pre-topological order to identify the strong components of a digraph in linear time (Karzanov [1970], Tarjan [1972]). We give a method essentially due to S.R. Kosaraju (cf. Aho, Hopcroft, and Ullman [1983]) and Sharir [1981].

Theorem 6.6. *Given a digraph $D = (V, A)$, the strong components of D can be found in linear time.*

Proof. First order the vertices of D pre-topologically as $v_1, \ldots, v_n$. Next let V_1 be the set of vertices reachable to v_1. Then V_1 is the strong component containing v_1: each v_j in V_1 is reachable from v_1, by the definition of pre-topological order.

By Theorem 6.5, the set V_1 can be found in time $O(|A_1|)$, where A_1 is the set of arcs with head in V_1. Delete from D and from $v_1, \ldots, v_n$ all vertices in V_1 and all arcs in A_1, yielding the subgraph D' and the ordered vertices $v'_1, \ldots, v'_{n'}$. This is a pre-topological order for D', for suppose that $i < j$ and that v'_i is reachable from v'_j in D' while v'_j is not reachable in D' from v'_i. Then v'_j is also not reachable in D from v'_i, since otherwise V_1 would be reachable in D from v'_i, and hence $v'_i \in V_1$, a contradiction.

So recursion gives all strong components, in linear time. ∎

As a consequence one has for undirected graphs (Shirey [1969]):

Corollary 6.6a. *Given a graph $G = (V, E)$, the components of G can be found in linear time.*

Proof. Directly from Theorem 6.6. ∎

If we apply depth-first search to a connected undirected graph $G = (V, E)$, starting from a vertex s, then the depth-first search tree T has the property that

(6.4) for each edge $e = uv$ of G, u is on the $s - v$ path in T or v is on the $s - u$ path in T.

So the ends of each edge e of G are connected by a directed path in T.

6.4. Finding an Eulerian orientation

An orientation $D = (V, A)$ of an undirected graph $G = (V, E)$ is called an *Eulerian orientation* if

(6.5) $\deg_A^{\text{in}}(v) = \deg_A^{\text{out}}(v)$

for each $v \in V$. As is well-known, an undirected graph $G = (V, E)$ has an Eulerian orientation if and only if each $v \in V$ has even degree in G. (We do not require connectivity.)

An Eulerian orientation can be found in linear time:

Theorem 6.7. *Given an undirected graph $G = (V, E)$ with all degrees even, an Eulerian orientation of G can be found in $O(m)$ time.*

Proof. We assume that we have a list of vertices, and, with each vertex v, a list of edges incident with v.

Consider the first nonisolated vertex in the list, v say. Starting at v, we make a walk such that no edge is traversed more than once. We make this walk as long as possible. Since all degrees are even, we terminate at v.

We orient all edges traversed in the direction as traversed, delete them from G, find the next nonisolated vertex, and iterate the algorithm. We stop if all vertices are isolated. Then all edges are oriented as required. ∎

6.5. Further results and notes

6.5a. All-pairs shortest paths in undirected graphs

Theorem 6.3 directly gives that determining the distances in a digraph $D = (V, A)$ between all pairs of vertices can be done in time $O(nm)$; similarly for undirected graphs.

Seidel [1992,1995] gave the following faster method for dense undirected graphs $G = (V, E)$, assuming without loss of generality that G is connected (by Corollary 6.6a).

For any undirected graph $G = (V, E)$, let $\mathrm{dist}_G(v, w)$ denote the distance between v and w in G (with unit-length edges). Moreover, let G^2 denote the graph with vertex set V, where two vertices v, w are adjacent in G^2 if and only if $\mathrm{dist}_G(v, w) \leq 2$.

Let $M(n)$ denote the time needed to determine the product $A \cdot B$ of two $n \times n$ matrices A and B, each with entries in $\{0, \ldots, n\}$.

It is not difficult to see that the following problem can be solved in time $O(M(n))$:

(6.6) given: an undirected graph $G = (V, E)$;
find: the undirected graph G^2.

Moreover, also the following problem can be solved in time $O(M(n))$:

(6.7) given: an undirected graph $G = (V, E)$ and the function dist_{G^2};
find: the function dist_G.

Lemma 6.8α. *Problem* (6.7) *can be solved in time* $O(M(n))$.

Proof. Let A be the adjacency matrix of G and let T be the $V \times V$ matrix with $T_{v,w} = \mathrm{dist}_{G^2}(v, w)$ for all $v, w \in V$. Note that $\mathrm{dist}_{G^2} = \lceil \mathrm{dist}_G/2 \rceil$. Let $B := T \cdot A$. So for all $u, w \in V$:

(6.8) $$B_{u,w} = \sum_{v \in N(w)} \mathrm{dist}_{G^2}(u, v).$$

Now if $\mathrm{dist}_G(u, w) = 2\lceil \mathrm{dist}_G(u, w)/2 \rceil$, then $\lceil \mathrm{dist}_G(u, v)/2 \rceil \geq \lceil \mathrm{dist}_G(u, w)/2 \rceil$ for each neighbour v of w, and hence $B_{u,w} \geq \deg(v)\lceil \mathrm{dist}_G(u, w)/2 \rceil$. On the other hand, if $\mathrm{dist}_G(u, w) = 2\lceil \mathrm{dist}_G(u, w)/2 \rceil - 1$, then $\lceil \mathrm{dist}_G(u, v)/2 \rceil \leq \lceil \mathrm{dist}_G(u, w)/2 \rceil$ for each neighbour v of w, with strict inequality for at least one neighbour v of w (namely any neighbour of w on a shortest $u - w$ path). So we have $B_{u,w} < \deg(v)\lceil \mathrm{dist}_G(u, w)/2 \rceil$. We conclude that having G, $\mathrm{dist}_{G^2} = \lceil \mathrm{dist}_G/2 \rceil$, and B, we can derive dist_G. As we can calculate B in time $O(M(n))$, we have the lemma. ∎

The all-pairs shortest paths algorithm now is described recursively as follows:

(6.9) If G is a complete graph, the distance between any two distinct vertices is 1. If G is not complete, determine G^2 and from this (recursively) dist_{G^2}. Next determine dist_G.

Theorem 6.8. *Given an undirected graph G on n vertices, the function* dist_G *can be determined in time $O(M(n) \log n)$. Here $M(n)$ denotes the time needed to multiply two $n \times n$ matrices each with entries in* $\{0, \ldots, n\}$.

Proof. Determining G^2 from G and determining dist_G from G and dist_{G^2} can be done in time $O(M(n))$. Since the depth of the recursion is $O(\log n)$, the algorithm has running time $O(M(n) \log n)$. ∎

The results on fast matrix multiplication of Coppersmith and Winograd [1987, 1990] give $M(n) = o(n^{2.376})$ (extending earlier work of Strassen [1969]).

Seidel [1992,1995] showed in fact that also shortest paths can be found in this way. More precisely, for all $u, w \in V$ with $u \neq w$, a neighbour v of w can be found such that $\mathrm{dist}_G(u, v) = \mathrm{dist}_G(u, w) - 1$, in time $O(M(n) \log n + n^2 \log^2 n)$. Having this, one can find, for any $u, w \in V$, a shortest $u - w$ path in time $O(\mathrm{dist}_G(u, w))$.

6.5b. Complexity survey

Complexity survey for all-pairs shortest paths with unit lengths ($*$ indicates an asymptotically best bound in the table):

	$O(nm)$	Berge [1958b], Moore [1959]
	$O(n^{\frac{3+\omega}{2}} \log^3 n)$	Alon, Galil, and Margalit [1991,1997]
$*$	$O(nm \log_n(n^2/m))$	Feder and Motwani [1991,1995]
$*$	$O(n^\omega \log n)$	*undirected* Seidel [1992,1995]

Here ω is any real such that any two $n \times n$ matrices can be multiplied by $O(n^\omega)$ arithmetic operations (e.g. $\omega = 2.376$).

Alon, Galil, and Margalit [1991,1997] extended their method to digraphs with arc lengths in $\{-1, 0, 1\}$.

Related work was done by Fredman [1976], Yuval [1976], Romani [1980], Aingworth, Chekuri, and Motwani [1996], Zwick [1998,1999a,2002], Aingworth, Chekuri, Indyk, and Motwani [1999], and Shoshan and Zwick [1999].

6.5c. Ear-decomposition of strongly connected digraphs

Let $D = (V, A)$ be a directed graph. An *ear* of D is a directed path or circuit P in D such that all internal vertices of P have indegree and outdegree equal to 1 in D. The path may consist of a single arc — so any arc of D is an ear. If I is the set of internal vertices of an ear P, we say that D arises from $D - I$ by *adding ear P*. An *ear-decomposition* of D is a series of digraphs $D_0, D_1, \ldots, D_k$, where $D_0 = K_1$, $D_k = D$, and D_i arises from D_{i-1} by adding an ear ($i = 1, \ldots, k$).

Digraphs having an ear-decomposition are be characterized by:

Theorem 6.9. *A digraph $D = (V, A)$ is strongly connected if and only if D has an ear-decomposition.*

Proof. Sufficiency of the condition is easy, since adding an ear to a strongly connected graph maintains strong connectivity.

To see necessity, let $D = (V, A)$ be strongly connected. Let $D' = (V', A')$ be a subgraph of D which has an ear-decomposition and with $|V'| + |A'|$ as large as possible. (Such a subgraph exists, as any single vertex has an ear-decomposition.)

Then $D' = D$, for otherwise there exists an arc $a \in A \setminus A'$ with tail in V'. Then a is contained in a directed circuit C (as D is strongly connected). This circuit C contains a subpath (or circuit) P such that P can be added as an ear to D'. This contradicts the maximality of $|V'| + |A'|$. ∎

A related decomposition of strongly connected digraphs was described by Knuth [1974]. Related work was done by Grötschel [1979].

6.5d. Transitive closure

Complexity survey for finding the transitive closure of a directed graph ($*$ indicates an asymptotically best bound in the table):

	$O(n^3)$	Warshall [1962]
$*$	$O(nm)$	Purdom [1970], cf. Coffy [1973] (also Ebert [1981])
	$O(n^3/\log n)$	Arlazarov, Dinits, Kronrod, and Faradzhev [1970]
	$\widetilde{O}(n^\omega)$	Furman [1970], Munro [1971]
	$O(n^\omega \log n \log\log n \log\log\log n)$	Aho, Hopcroft, and Ullman [1974]
$*$	$O(n^\omega \log n \log\log\log n \log\log\log\log n)$	Adleman, Booth, Preparata, and Ruzzo [1978]

Again, ω is any real such that any two $n \times n$ matrices can be multiplied by $O(n^\omega)$ arithmetic operations (e.g. $\omega = 2.376$). Moreover, $f = \widetilde{O}(g)$ if $f = O(g \log^k g)$ for some k.

For more on finding the transitive closure we refer to Fischer and Meyer [1971], Munro [1971], O'Neil and O'Neil [1973], Dzikiewicz [1975], Sysło and Dzikiewicz [1975], Warren [1975], Eve and Kurki-Suonio [1977], Adleman, Booth, Preparata, and Ruzzo [1978], Schnorr [1978a], Schmitz [1983], Ioannidis and Ramakrishnan [1988], Jakobsson [1991], Ullman and Yannakakis [1991], and Cohen [1994a,1997].

Aho, Garey, and Ullman [1972] showed that finding a minimal directed graph having the same transitive closure as a given directed graph, has the same time complexity as finding the transitive closure.

6.5e. Further notes

For the decomposition of graphs into 3-connected graphs, see Cunningham and Edmonds [1980]. Karzanov [1970] and Tarjan [1974b] gave linear-time algorithms (based on a search method) to find the bridges of an undirected graph.

Theorem 6.1 implies the result of Moore and Shannon [1956] that if $D = (V, A)$ is a digraph, $s, t \in V$, and l is the minimum length of an $s-t$ path and w is the minimum size of an $s-t$ cut, then $|A| \geq lw$ (the *length-width inequality*).

Finding a shortest (directed) circuit in a (directed) graph can be reduced to finding a shortest path. More efficient algorithms were given by Itai and Rodeh [1978].

Barnes and Ruzzo [1991,1997] gave a polynomial-time algorithm to test if there exists an $s-t$ path in an undirected graph, using sublinear space only. This was extended to directed graphs by Barnes, Buss, Ruzzo, and Schieber [1992,1998]. Related work was done by Savitch [1970], Cook and Rackoff [1980], Beame, Borodin, Raghavan, Ruzzo, and Tompa [1990,1996], Nisan [1992,1994], Nisan, Szemeredi, and Wigderson [1992], Broder, Karlin, Raghavan, and Upfal [1994], and Armoni, Ta-Shma, Wigderson, and Zhou [1997,2000].

Karp and Tarjan [1980a,1980b] gave algorithms for finding the connected components of an undirected graph, and the strong components of a directed graph, in $O(n)$ expected time. More on finding strong components can be found in Gabow [2000a].

Books discussing algorithmic problems on paths with unit lengths (reachability, closure, etc.) include Even [1973], Aho, Hopcroft, and Ullman [1974,1983], Christofides [1975], Cormen, Leiserson, and Rivest [1990], Lengauer [1990], Jungnickel [1999], and Mehlhorn and Näher [1999]. Berge [1958b] gave an early survey on shortest paths.

Chapter 7

Shortest paths: nonnegative lengths

In this chapter we consider the shortest path problem in graphs where each arc has a nonnegative length, and describe Dijkstra's algorithm, together with a number of speedups based on heaps.
In this chapter, graphs can be assumed to be simple. If not mentioned explicitly, *length* is taken with respect to a given function l.

7.1. Shortest paths with nonnegative lengths

The methods and results discussed in Chapter 6 for unit-length arcs can be generalized to the case where arcs have a not necessarily unit length. For any 'length' function $l : A \to \mathbb{R}$ and any path $P = (v_0, a_1, v_1, \ldots, a_m, v_m)$, the *length* $l(P)$ of P is defined by:

$$l(P) := \sum_{i=1}^{m} l(a). \tag{7.1}$$

The *distance* from s to t (with respect to l), denoted by $\text{dist}_l(s,t)$, is equal to the minimum length of any $s-t$ path. If no $s-t$ path exists, $\text{dist}_l(s,t)$ is set to $+\infty$.

A weighted version of Theorem 6.1 is as follows, again due to Robacker [1956b] (sometimes called the 'max-potential min-work theorem' (Duffin [1962])):

Theorem 7.1. *Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let $l : A \to \mathbb{Z}_+$. Then the minimum length of an $s-t$ path is equal to the maximum size k of a family of $s-t$ cuts $C_1, \ldots, C_k$ such that each arc a is in at most $l(a)$ of the cuts C_i.*

Proof. Again, the minimum is not smaller than the maximum, since if P is any $s-t$ path and $C_1, \ldots, C_k$ is any collection as described in the theorem, then

$$l(P) = \sum_{a \in AP} l(a) \geq \sum_{a \in AP} (\text{number of } i \text{ with } a \in C_i)$$
$$= \sum_{i=1}^{k} |C_i \cap AP| \geq \sum_{i=1}^{k} 1 = k. \tag{7.2}$$

To see equality, let d be the distance from s to t and let U_i be the set of vertices at distance less than i from s, for $i = 1, \ldots, d$. Taking $C_i := \delta^{\text{out}}(U_i)$, we obtain a collection $C_1, \ldots, C_d$ as required. ∎

A rooted tree $T = (V', A')$, with root s, is called a *shortest paths tree* for a length function $l : A \to \mathbb{R}_+$, if V' is the set of vertices reachable from s and $A' \subseteq A$ such that for each $t \in V'$, the $s - t$ path in T is a shortest $s - t$ path in D. Again, Dantzig [1957] showed:

Theorem 7.2. *Let $D = (V, A)$ be a digraph, let $s \in V$, and let $l : A \to \mathbb{R}_+$. Then there exists a shortest paths tree for l, with root s.*

Proof. Let V' be the set of vertices reachable from s. Let A' be an inclusionwise minimal set containing for each $t \in V'$ a shortest $s - t$ path of D. Suppose that some vertex v is entered by two arcs in A'. Then at least one of these arcs can be deleted, contradicting the minimality of A'. One similarly sees that no arc in A' enters s. ∎

7.2. Dijkstra's method

Dijkstra [1959] gave an $O(n^2)$ algorithm to find a shortest $s - t$ path for nonnegative length functions — in fact, the output is a shortest paths tree with root s. We describe Dijkstra's method (the idea of this method was also described by Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957]).

We keep a subset U of V and a function $d : V \to \mathbb{R}_+$ (the *tentative distance*). Start with $U := V$ and set $d(s) := 0$ and $d(v) = \infty$ if $v \neq s$. Next apply the following iteratively:

(7.3) Find $u \in U$ minimizing $d(u)$ over $u \in U$. For each $a = (u, v) \in A$ for which $d(v) > d(u) + l(a)$, reset $d(v) := d(u) + l(a)$. Reset $U := U \setminus \{u\}$.

We stop if $d(u) = \infty$ for all $u \in U$. The final function d gives the distance from s. Moreover, if we store for each $v \neq s$ the last arc $a = (u, v)$ for which we have reset $d(v) := d(u) + l(a)$, we obtain a shortest path tree with root s.

Clearly, the number of iterations is at most $|V|$, while each iteration takes $O(n)$ time. So the algorithm has running time $O(n^2)$. Thus:

Theorem 7.3. *Given a digraph $D = (V, A)$, $s \in V$, and a length function $l : A \to \mathbb{Q}_+$, a shortest paths tree with root s can be found in time $O(n^2)$.*

Proof. We show correctness of the algorithm. Let $\text{dist}(v)$ denote the distance from s to v, for any vertex v. Trivially, $d(v) \geq \text{dist}(v)$ for all v, throughout the iterations. We prove that throughout the iterations, $d(v) = \text{dist}(v)$ for each $v \in V \setminus U$. At the start of the algorithm this is trivial (as $U = V$).

Consider any iteration (7.3). It suffices to show that $d(u) = \text{dist}(u)$ for the chosen $u \in U$. Suppose $d(u) > \text{dist}(u)$. Let $s = v_0, v_1, \ldots, v_k = u$ be a shortest $s - u$ path. Let i be the smallest index with $v_i \in U$.

Then $d(v_i) = \text{dist}(v_i)$. Indeed, if $i = 0$, then $d(v_i) = d(s) = 0 = \text{dist}(s) = \text{dist}(v_i)$. If $i > 0$, then (as $v_{i-1} \in V \setminus U$):

(7.4) $$d(v_i) \leq d(v_{i-1}) + l(v_{i-1}, v_i) = \text{dist}(v_{i-1}) + l(v_{i-1}, v_i) = \text{dist}(v_i).$$

This implies $d(v_i) \leq \text{dist}(v_i) \leq \text{dist}(u) < d(u)$, contradicting the choice of u. ∎

7.3. Speeding up Dijkstra's algorithm with *k*-heaps

If $|A|$ is asymptotically smaller than $|V|^2$, one may expect faster methods than $O(n^2)$. Such a method based on 'heaps' (introduced by Williams [1964] and Floyd [1964]), was given by Murchland [1967b] and sharpened by Johnson [1972], Johnson [1973b,1977a] and Tarjan [1983] (see Section 8.6g).

In Dijkstra's algorithm, we spend (in total) $O(m)$ time on updating the values $d(u)$, and $O(n^2)$ time on finding a $u \in U$ minimizing $d(u)$. As $m \leq n^2$, a decrease in the running time bound requires a speedup in finding a u minimizing $d(u)$.

A way of doing this is based on storing U in some order such that a $u \in U$ minimizing $d(u)$ can be found quickly and such that it does not take too much time to restore the order if we delete a u minimizing $d(u)$ or if we decrease some $d(u)$.

This can be done by using 'heaps', two forms of which we consider: k-heaps (in this section) and Fibonacci heaps (in the next section).

A *k-heap* is an ordering $u_0, \ldots, u_n$ of the elements of U such that for all i, j, if $ki < j \leq k(i+1)$, then $d(u_i) \leq d(u_j)$.

This is a convenient way of defining (and displaying) the heap, but it is helpful to imagine the heap as a rooted tree on U: its arcs are the pairs (u_i, u_j) with $ki < j \leq k(i+1)$. So u_i has outdegree k if $k(i+1) \leq n$. The root of this rooted tree is u_0.

If one has a k-heap, one easily finds a u minimizing $d(u)$: it is the root u_0. The following two theorems are basic for estimating the time needed for updating the k-heap if we change U or values of $d(u)$. To *swap* u_i and u_j means exchanging the positions of u_i and u_j in the order (that is, resetting $u_j := u_i$ and $u_i :=$ the old u_j).

Theorem 7.4. *If u_0 is deleted, the k-heap can be restored in time $O(k \log_k n)$.*

Proof. Reset $u_0 := u_n$ and $n := n - 1$. Let $i = 0$. While there is a j with $ki < j \leq ki + k$, $j \leq n - 1$, and $d(u_j) < d(u_i)$, choose such a j with smallest $d(u_j)$, swap u_i and u_j, and reset $i := j$.

The final k-heap is as required. ∎

The operation described is called *sift-down*. The following theorem describes the operation *sift-up*.

Theorem 7.5. *If $d(u_i)$ is decreased, the k-heap can be restored in time $O(\log_k n)$.*

Proof. While $i > 0$ and $d(u_j) > d(u_i)$ for $j := \lfloor \frac{i-1}{k} \rfloor$, swap u_i and u_j, and reset $i := j$. The final k-heap is as required. ∎

In Dijkstra's algorithm, we delete at most $|V|$ times a u minimizing $d(u)$ and we decrease at most $|A|$ times any $d(u)$. So using a k-heap, the algorithm can be done in time $O(nk \log_k n + m \log_k n)$. This implies:

Theorem 7.6. *Given a digraph $D = (V, A)$, $s \in V$, and a length function $l : A \to \mathbb{Q}_+$, a shortest paths tree with root s can be found in time $O(m \log_k n)$, where $k := \max\{2, \frac{m}{n}\}$.*

Proof. See above. ∎

This implies that if for some class of digraphs $D = (V, A)$ one has $|A| \geq |V|^{1+\varepsilon}$ for some fixed $\varepsilon > 0$, then there is a linear-time shortest path algorithm for these graphs.

7.4. Speeding up Dijkstra's algorithm with Fibonacci heaps

Using a more sophisticated heap, the 'Fibonacci heap', Dijkstra's algorithm can be speeded up to $O(m + n \log n)$, as was shown by Fredman and Tarjan [1984,1987].

A *Fibonacci forest* is a rooted forest (U, A), such that for each $v \in U$ the children of v can be ordered in such a way that the ith child has at least $i - 2$ children. (If $(u, v) \in A$, v is called a *child* of u, and u the *parent* of v.)

Lemma 7.7α. *In a Fibonacci forest (U, A), each vertex has outdegree at most $2 \log_2 |U|$.*

Proof. We show:

(7.5) if u has outdegree at least k, then at least $\sqrt{2}^k$ vertices are reachable from u.

This implies the lemma, since $\sqrt{2}^k \leq |U|$ is equivalent to $k \leq 2\log_2 |U|$.

In proving (7.5), we may assume that u is a root. We prove (7.5) by induction on k, the case $k = 0$ being trivial. If $k \geq 1$, let v be the highest ordered child of u. So v has outdegree at least $k-2$. Then by induction, at least $\sqrt{2}^{k-2}$ vertices are reachable from v. Next delete arc (u, v). We keep a Fibonacci forest, in which u has outdegree at least $k-1$. By induction, at least $\sqrt{2}^{k-1}$ vertices are reachable from u in the new forest. Hence at least

(7.6) $$\sqrt{2}^{k-2} + \sqrt{2}^{k-1} \geq \sqrt{2}^k$$

vertices are reachable from u in the original forest. ∎

(The recursion (7.6) shows that the Fibonacci numbers give the best bound, justifying the name Fibonacci forest. The weaker bound given, however, is sufficient for our purposes.)

A *Fibonacci heap* consists of a rooted forest $F = (U, A)$ and functions $d : U \to \mathbb{R}$ and $\phi : U \to \{0, 1\}$, such that:

(7.7)
(i) if $(u, v) \in A$, then $d(u) \leq d(v)$;
(ii) for each $u \in U$, the children of u can be ordered such that the ith child v satisfies $\deg^{\text{out}}(v) + \phi(v) \geq i - 1$;
(iii) if u and v are distinct roots of F, then $\deg^{\text{out}}(u) \neq \deg^{\text{out}}(v)$.

Condition (7.7)(ii) implies that F is a Fibonacci forest. So, by Lemma 7.7α, condition (7.7)(iii) implies that F has at most $1 + 2\log_2 |U|$ roots.

The Fibonacci heap will be specified by the following data structure, where $t := \lfloor 2\log_2 |U| \rfloor$:

(7.8)
(i) for each $u \in U$, a doubly linked list of the children of u (in any order);
(ii) the function parent $: U \to U$, where parent(u) is the parent of u if it has one, and parent$(u) = u$ otherwise;
(iii) the functions $\deg^{\text{out}} : U \to \mathbb{Z}_+$, $\phi : U \to \{0, 1\}$, and $d : U \to \mathbb{R}$;
(iv) a function $b : \{0, \ldots, t\} \to U$ with $b(\deg^{\text{out}}(u)) = u$ for each root u.

Theorem 7.7. *When inserting p times a new vertex, finding and deleting n times a root u minimizing $d(u)$, and decreasing m times the value of $d(u)$, the structure can be restored in time $O(m + p + n \log p)$.*

Proof. Inserting a new vertex v, with value $d(v)$, can be done by setting $\phi(v) := 0$ and by applying:

(7.9) *plant*(v):
Let $r := b(\deg^{\text{out}}(v))$.
If r is a root with $r \neq v$, then:

$$\begin{cases} \text{if } d(r) \leq d(v), \text{ add arc } (r,v) \text{ to } A \text{ and plant}(r); \\ \text{if } d(r) > d(v), \text{ add arc } (v,r) \text{ to } A \text{ and plant}(v); \end{cases}$$

else define $b(\deg^{\text{out}}(v)) := v$.

Throughout we update the lists of children and the functions parent, $\deg^{\text{out}}$, ϕ, and b.

A root u minimizing $d(u)$ can be found in time $O(\log p)$, by scanning $d(b(i))$ for $i = 0, \ldots, t$ where $b(i)$ is a root.

The root u can be deleted as follows. Let $v_1, \ldots, v_k$ be the children of u. First delete u and all arcs leaving u from the forest. This maintains conditions (7.7)(i) and (ii). Next, condition (7.7)(iii) can be restored by applying plant(v) for each $v = v_1, \ldots, v_k$.

If we decrease the value of $d(u)$ for some $u \in U$ we do the following:

(7.10) Determine the longest directed path P in F ending at u such that each internal vertex v of P satisfies $\phi(v) = 1$. Reset $\phi(v) := 1 - \phi(v)$ for each $v \in VP \setminus \{u\}$. Delete all arcs of P from A. Apply plant(v) to each $v \in VP$ that is a root of the new forest.

The fact that this maintains (7.7) uses that if the starting vertex q of P is not a root of the original forest, then $q \neq u$ and $\phi(q)$ is reset from 0 to 1 — hence $\deg^{\text{out}}(q) + \phi(q)$ is not changed, and we maintain (7.7)(ii).

We estimate the running time. Throughout all iterations, ϕ increases at most m times (at most once in each application of (7.10)). Hence ϕ decreases at most m times. So the sum of the lengths of the paths P in (7.10) is at most $2m$. So A decreases at most $2m + 2n \log_2 p$ times (since each time we delete a root we delete at most $2 \log_2 p$ arcs). Therefore, A increases at most $2m + 2n \log_2 p + p$ times (since the final $|A|$ is less than p). This gives the running time bound. ∎

This implies for the shortest path problem:

Corollary 7.7a. *Given a digraph $D = (V, A)$, $s \in V$, and a length function $l : A \to \mathbb{Q}_+$, a shortest paths tree with root s can be found in time $O(m + n \log n)$.*

Proof. Directly from Dijkstra's method and Theorem 7.7. ∎

7.5. Further results and notes

7.5a. Weakly polynomial-time algorithms

The above methods all give a strongly polynomial-time algorithm for the shortest path problem, with best running time bound $O(m + n \log n)$. If we allow also the size of the numbers to occur in the running time bound, some other methods are of interest that are in some cases (when the lengths are small integers) faster than the above methods.

In Dijkstra's algorithm, we must select a $u \in U$ with $d(u)$ minimum. It was observed by Dial [1969] that partitioning U into 'buckets' according to the values of d gives a competitive running time bound. The method also gives the following result of Wagner [1976]:

Theorem 7.8. *Given a digraph $D = (V, A)$, $s \in V$, $l : A \to \mathbb{Z}_+$, and an upper bound Δ on $\max\{\mathrm{dist}_l(s, v) \mid v$ reachable from $s\}$, a shortest path tree rooted at s can be found in time $O(m + \Delta)$.*

Proof. Apply Dijkstra's algorithm as follows. Next to the function $d : U \to \mathbb{Z}_+ \cup \{\infty\}$, we keep doubly linked lists $L_0, \ldots, L_\Delta$ such that if $d(u) \leq \Delta$, then u is in $L_{d(u)}$. We keep also, for each $i = 0, \ldots, \Delta$, the first element u_i of L_i. If L_i is empty, then u_i is void. Moreover, we keep a 'current minimum' $\mu \in \{0, \ldots, \Delta\}$.

The initialization follows directly from the initialization of d: we set $L_0 := \{s\}$, $u_0 := s$, while L_i is empty and u_i void for $i = 1, \ldots, \Delta$. Initially, $\mu := 0$.

The iteration is as follows. If $L_\mu = \emptyset$ and $\mu \leq \Delta$, reset $\mu := \mu + 1$. If $L_\mu \neq \emptyset$, apply Dijkstra's iteration to u_μ: We remove u_μ from L_μ. When decreasing some $d(u)$ from d to d', we delete u from L_d (if $d \leq \Delta$) and insert it into $L_{d'}$ (if $d' \leq \Delta$).

We stop if $\mu = \Delta + 1$. With each removal or insertion, we can update the lists and the u_i in constant time. Hence we have the required running time. ∎

A consequence is the bound of Dial [1969]:

Corollary 7.8a. *Given a digraph $D = (V, A)$, $s \in V$, and $l : A \to \mathbb{Z}_+$, a shortest path tree rooted at s can be found in time $O(m+nL)$ where $L := \max\{l(a) \mid a \in A\}$.*

Proof. We can take $\Delta = nL$ in Theorem 7.8. ∎

One can derive from Theorem 7.8 also the following result of Gabow [1985b]:

Theorem 7.9. *Given a digraph $D = (V, A)$, $s \in V$, and a length function $l : A \to \mathbb{Z}_+$, a shortest paths tree rooted at s can be found in time $O(m \log_d L)$, where $d = \max\{2, m/n\}$, and $L := \max\{l(a) \mid a \in A\}$.*

Proof. For each $a \in A$, let $l'(a) := \lfloor l(a)/d \rfloor$. Recursively we find $\mathrm{dist}_{l'}(s, v)$ for all $v \in V$, in time $O(m \log_d L')$ where $L' := \lfloor L/d \rfloor$. Note that $\log_d L' \leq (\log_d L) - 1$. Now set

$$\tilde{l}(a) := l(a) + d \cdot \mathrm{dist}_{l'}(s, u) - d \cdot \mathrm{dist}_{l'}(s, v) \tag{7.11}$$

for each $a = (u, v) \in A$. Then $\tilde{l}(a) \geq 0$, since

$$l(a) \geq d \cdot l'(a) \geq d(\mathrm{dist}_{l'}(s, v) - \mathrm{dist}_{l'}(s, u)). \tag{7.12}$$

Moreover, $\mathrm{dist}_{\tilde{l}}(s, v) \leq nd$ for each v reachable from s, since if P is an $s - v$ path with $l'(P) = \mathrm{dist}_{l'}(s, v)$, then $\tilde{l}(P) = l(P) - dl'(P) \leq nd$. So by Theorem 7.8 we can find $\mathrm{dist}_{\tilde{l}}(s, v)$ for all $v \in V$ in time $O(m)$, since $nd \leq 2m$. As $\mathrm{dist}_l(s, v) = \mathrm{dist}_{\tilde{l}}(s, v) - d \cdot \mathrm{dist}_{l'}(s, v)$, we find the required data. ∎

(This improves a result of Hansen [1980a].)

7.5b. Complexity survey for shortest paths with nonnegative lengths

The following gives a survey of the development of the running time bound for the shortest path problem for a digraph $D = (V, A)$, $s, t \in V$, and nonnegative length-function l, where $n := |V|$, $m := |A|$, and $L := \max\{l(a) \mid a \in A\}$ (assuming l integer). As before, $*$ indicates an asymptotically best bound in the table.

	$O(n^4)$	Shimbel [1955]
	$O(n^2 mL)$	Ford [1956]
	$O(nm)$	Bellman [1958], Moore [1959]
	$O(n^2 \log n)$	Dantzig [1958,1960], Minty (cf. Pollack and Wiebenson [1960]), Whiting and Hillier [1960]
	$O(n^2)$	Dijkstra [1959]
	$O(m + nL)$	Dial [1969] (cf. Wagner [1976], Filler [1976](=E.A. Dinits))
	$O(m \log(2 + (n^2/m)))$	Johnson [1972]
	$O(dn \log_d n + m + m \log_d(n^2/m))$	(for any $d \geq 2$) Johnson [1973b]
	$O(m \log_{m/n} n)$	Johnson [1973b,1977a], Tarjan [1983]
	$O(L + m \log\log L)$	van Emde Boas, Kaas, and Zijlstra [1977]
	$O(m \log\log L + n \log L \log\log L)$	Johnson [1977b]
	$O(\min_{k\geq 2}(nkL^{1/k} + m \log k))$	Denardo and Fox [1979]
	$O(m \log L)$	Hansen [1980a]
$*$	$O(m \log\log L)$	Johnson [1982], Karlsson and Poblete [1983]
$*$	$O(m + n \log n)$	Fredman and Tarjan [1984,1987][2]
	$O(m \log_{m/n} L)$	Gabow [1983b,1985b]
$*$	$O(m + n\sqrt{\log L})$	Ahuja, Mehlhorn, Orlin, and Tarjan [1990]

Fredman and Willard [1990,1994] gave an $O(m + n\frac{\log n}{\log\log n})$-time algorithm for shortest paths with nonnegative lengths, utilizing nonstandard capabilities of a RAM like addressing. This was extended to $O(m + n\sqrt{\log n \log\log n})$ by Raman [1996], to $O(m \log\log n)$ and $O(m \log\log L)$ by Thorup [1996,2000b] (cf. Hagerup [2000]), and to $O(m+n(\log L \log\log L)^{1/3})$ by Raman [1997]. For undirected graphs,

[2] Aho, Hopcroft, and Ullman [1983] (p. 208) claimed to give an $O(m + n \log n)$-time shortest path algorithm based on 2-heaps, but they assume that, after resetting a value, the heap can be restored in constant time.

a bound of $O(m)$ was achieved by Thorup [1997,1999,2000a]. Related results were given by Pettie and Ramachandran [2002b].

The expected complexity of Dijkstra's algorithm is investigated by Noshita, Masuda, and Machida [1978], Noshita [1985], Cherkassky, Goldberg, and Silverstein [1997,1999], Goldberg [2001a,2001b], and Meyer [2001].

In the special case of *planar* directed graphs:

	$O(n\sqrt{\log n})$	Frederickson [1983b,1987b]
*	$O(n)$	Klein, Rao, Rauch, and Subramanian [1994], Henzinger, Klein, Rao, and Subramanian [1997]

For the all-pairs shortest paths problem with nonnegative lengths one has:

	$O(n^4)$	Shimbel [1955]
	$O(n^3 \log n)$	Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957]
	$O(n^2 m)$	Bellman [1958], Moore [1959]
	$O(n^3 \log n)$	Dantzig [1958,1960], Minty (cf. Pollack and Wiebenson [1960]), Whiting and Hillier [1960]
	$O(n^3)$	Dijkstra [1959]
	$O(nm + n^2 L)$	Dial [1969] (cf. Wagner [1976])
	$O(nm \log n)$	Johnson [1972]
	$O(n^3 (\log\log n / \log n)^{1/3})$	Fredman [1976]
	$O(nm \log_{m/n} n)$	Johnson [1973b,1977a], Tarjan [1983]
	$O(nL + nm \log\log L)$	van Emde Boas, Kaas, and Zijlstra [1977]
	$O(nm \log\log L + n^2 \log L \log\log L)$	Johnson [1977b]
	$O(nm \log L)$	Hansen [1980a]
*	$O(nm \log\log L)$	Johnson [1982], Karlsson and Poblete [1983]
*	$O(n(m + n \log n))$	Fredman and Tarjan [1984,1987]
	$O(nm \log_{m/n} L)$	Gabow [1983b,1985b]
*	$O(nm + n^2 \sqrt{\log L})$	Ahuja, Mehlhorn, Orlin, and Tarjan [1990]
*	$O(n^3 (\log\log n / \log n)^{1/2})$	Takaoka [1992a,1992b]
*	$\widetilde{O}(n^{\frac{5\omega-3}{\omega+1}} L + n^{\frac{3+\omega}{2}} L^{\frac{\omega-1}{2}})$	Galil and Margalit [1997a,1997b]
*	$\widetilde{O}(n^{\omega} L^{\frac{\omega+1}{2}})$	*undirected* Galil and Margalit [1997a,1997b]

Here ω is any real such that any two $n \times n$ matrices can be multiplied by $O(n^\omega)$ arithmetic operations (e.g. $\omega = 2.376$). Moreover, $f = \widetilde{O}(g)$ if $f = O(g \log^k g)$ for some k. (At the negative side, Kerr [1970] showed that matrix multiplication of $n \times n$ matrices with addition and multiplication replaced by minimization and addition, requires time $\Omega(n^3)$.)

Spira [1973] gave an $O(n^2 \log^2 n)$ expected time algorithm for all-pairs shortest paths with nonnegative lengths. This was improved to $O(n^2 \log n \log\log n)$ by Takaoka and Moffat [1980], to $O(n^2 \log n \log^* n)$ by Bloniarz [1980,1983] (defining $\log^* n := \min\{i \mid \log^{(i)} n \leq 1\}$, where $\log^{(0)} n := n$ and $\log^{(i+1)} n := \log(\log^{(i)} n)$), and to $O(n^2 \log n)$ by Moffat and Takaoka [1985,1987]. Related work includes Carson and Law [1977], Frieze and Grimmett [1985], Hassin and Zemel [1985], Walley and Tan [1995], and Mehlhorn and Priebe [1997].

Yen [1972] (cf. Williams and White [1973]) described an all-pairs shortest paths method (based on Dijkstra's method) using $\frac{1}{2}n^3$ additions and n^3 comparisons. Nakamori [1972] gave a lower bound on the number of operations. Yao, Avis, and Rivest [1977] gave a lower bound of $\Omega(n^2 \log n)$ for the time needed for the all-pairs shortest paths problem.

Karger, Koller, and Phillips [1991,1993] and McGeoch [1995] gave an $O(n(m^* + n \log n))$ algorithm for all-pairs shortest paths, where m^* is the number of arcs that belong to at least one shortest path. See also Yuval [1976] and Romani [1980] for relations between all-pairs shortest paths and matrix multiplication.

Frederickson [1983b,1987b] showed that in a *planar* directed graph, with nonnegative lengths, the all-pairs shortest paths problem can be solved in $O(n^2)$ time.

7.5c. Further notes

Spira and Pan [1973,1975], Shier and Witzgall [1980], and Tarjan [1982] studied the sensitivity of shortest paths trees under modifying arc lengths. Fulkerson and Harding [1977] studied the problem of lengthening the arc lengths within a given budget (where each arc has a given cost for lengthening the arc length) so as to maximize the distance from a given source to a given sink. They reduced this problem to a parametric minimum-cost flow problem. Land and Stairs [1967] and Hu [1968] studied decomposition methods for finding all-pairs shortest paths (cf. Farbey, Land, and Murchland [1967], Hu and Torres [1969], Yen [1971b], Shier [1973], and Blewett and Hu [1977]).

Frederickson [1989,1995] gave a strongly polynomial-time algorithm to find an $O(n)$ encoding of shortest paths between all pairs in a directed graph with nonnegative length function. (It extends earlier work of Frederickson [1991] for planar graphs.)

Algorithms for finding the k shortest paths between pairs of vertices in a directed graph were given by Clarke, Krikorian, and Rausen [1963], Yen [1971a], Minieka [1974], Weigand [1976], Lawler [1977], Shier [1979], Katoh, Ibaraki, and Mine [1982], Byers and Waterman [1984], Perko [1986], Chen [1994], and Eppstein [1994b,1999].

Mondou, Crainic, and Nguyen [1991] gave a survey of shortest paths methods, with computational results, and Raman [1997] on 'recent' results on shortest paths with nonnegative lengths.

Books covering shortest path methods for nonnegative lengths include Berge [1973b], Aho, Hopcroft, and Ullman [1974,1983], Christofides [1975], Lawler [1976b], Minieka [1978], Even [1979], Hu [1982], Papadimitriou and Steiglitz [1982], Smith [1982], Sysło, Deo, and Kowalik [1983], Tarjan [1983], Gondran and Minoux [1984], Rockafellar [1984], Nemhauser and Wolsey [1988], Bazaraa, Jarvis, and Sherali [1990], Chen [1990], Cormen, Leiserson, and Rivest [1990], Lengauer [1990], Ahuja, Magnanti, and Orlin [1993], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], Mehlhorn and Näher [1999], and Korte and Vygen [2000].

Chapter 8

Shortest paths: arbitrary lengths

We now go over to the shortest path problem for the case where negative lengths are allowed, but where each directed circuit has nonnegative length (with no restriction, the problem is NP-complete). The basic algorithm here is the Bellman-Ford method.
In this chapter, graphs can be assumed to be simple. If not mentioned explicitly, *length* is taken with respect to a given function l.

8.1. Shortest paths with arbitrary lengths but no negative circuits

If lengths of arcs may take negative values, finding a shortest $s-t$ path is NP-complete — see Theorem 8.11 below. Negative-length directed circuits seem to be the source of the trouble: if no negative-length directed circuits exist, there is a polynomial-time algorithm — mainly due to the fact that running into loops cannot give shortcuts. So a shortest *walk* (nonsimple path) exists and yields a shortest *path*.

We first observe that if no negative-length directed circuits exists, then the existence of a shortest paths tree is easy:

Theorem 8.1. *Let $D = (V, A)$ be a digraph, let $s \in V$, and let $l : A \to \mathbb{R}$ be such that each directed circuit reachable from s has nonnegative length. Then there exists a shortest paths tree with root s.*

Proof. As the proof of Theorem 7.2. ∎

8.2. Potentials

The following observation of Gallai [1958b] is very useful. Let $D = (V, A)$ be a digraph and let $l : A \to \mathbb{R}$. A function $p : V \to \mathbb{R}$ is called a *potential* if for each arc $a = (u, v)$:

(8.1) $\qquad l(a) \geq p(v) - p(u).$

Theorem 8.2. *Let $D = (V, A)$ be a digraph and let $l : A \to \mathbb{R}$ be a length function. Then there exists a potential if and only if each directed circuit has nonnegative length. If moreover l is integer, the potential can be taken integer.*

Proof. *Suffiency.* Suppose that a function p as described exists. Let $C = (v_0, a_1, v_1, \ldots, a_m, v_m)$ be a directed circuit ($v_m = v_0$). Then

$$l(C) = \sum_{i=1}^{m} l(a_i) \geq \sum_{i=1}^{m} (p(v_i) - p(v_{i-1})) = 0. \tag{8.2}$$

Necessity. Suppose that each directed circuit has nonnegative length. For each $t \in V$, let $p(t)$ be the minimum length of any path ending at t (starting wherever). This function satisfies the required condition. ∎

Theorem 8.2 gives a good characterization for the problem of deciding if there exists a negative-length directed circuit.

A potential is useful in transforming a length function to a nonnegative length function: if we define $\tilde{l}(a) := l(a) - p(v) + p(u)$ for each arc $a = (u, v)$, then we obtain a nonnegative length function $\tilde{l}$ such that each $s - t$ path is shortest with respect to l if and only if it is shortest with respect to $\tilde{l}$. So once we have a potential p, we can find shortest paths with Dijkstra's algorithm. This can be used for instance in finding shortest paths between all pairs of vertices — see Section 8.4.

One can also formulate a min-max relation in terms of functions that are potentials on an appropriate subgraph. This result is sometimes called the 'max-potential min-work theorem' (Duffin [1962]).

Theorem 8.3. *Let $D = (V, A)$ be a digraph, let $s, t \in V$ and let $l : A \to \mathbb{R}$. Then $\mathrm{dist}_l(s, t)$ is equal to the maximum value of $p(t) - p(s)$, where $p : V \to \mathbb{R}$ is such that $l(a) \geq p(v) - p(u)$ for each arc $a = (u, v)$ traversed by at least one $s - t$ walk. If l is integer, we can restrict p to be integer.*

Proof. Let $p(v) := \mathrm{dist}_l(s, v)$ if v belongs to at least one $s - t$ walk, and $p(v) := 0$ otherwise. This p is as required. ∎

The following observation can also be of use, for instance when calculating shortest paths by linear programming:

Theorem 8.4. *Let $D = (V, A)$ be a digraph, let $s \in V$ be such that each vertex of D is reachable from s, and let $l : A \to \mathbb{R}$ be such that each directed circuit has nonnegative length. Let p be a potential with $p(s) = 0$ and $\sum_{v \in V} p(v)$ maximal. Then $p(t) = \mathrm{dist}_l(s, t)$ for each $t \in V$.*

Proof. One easily shows that for any potential p with $p(s) = 0$ one has $p(t) \leq \mathrm{dist}_l(s,t)$ for each $t \in V$. As $\mathrm{dist}_l(s,\cdot)$ is a potential, the theorem follows. ∎

8.3. The Bellman-Ford method

Also in the case of a length function without negative-length directed circuit, there is a polynomial-time shortest path algorithm, the *Bellman-Ford method* (Shimbel [1955], Ford [1956], Bellman [1958], Moore [1959]). Again, it finds a shortest paths tree for any root s.

To describe the method, define for $t \in V$ and $k \geq 0$:

(8.3) $d_k(t) :=$ minimum length of any $s-t$ walk traversing at most k arcs,

setting $d_k(t) := \infty$ if no such walk exists.

Clearly, if there is no negative-length directed circuit reachable from s, the distance from s to t is equal to $d_n(v)$, where $n := |V|$.

Algorithmically, the function d_0 is easy to set: $d_0(s) = 0$ and $d_0(t) = \infty$ if $t \neq s$. Next $d_1, d_2, \ldots$ can be successively computed by the following rule:

$$d_{k+1}(t) = \min\{d_k(t), \min_{(u,t)\in A}(d_k(u) + l(u,t))\} \tag{8.4}$$

for all $t \in V$.

This method gives us the distance from s to t. It is not difficult to derive a method finding a shortest paths tree with root s. Thus:

Theorem 8.5. *Given a digraph $D = (V, A)$, $s \in V$, and a length function $l : A \to \mathbb{Q}$ such that each directed circuit reachable from s has nonnegative length, a shortest paths tree rooted at s can be found in time $O(nm)$.*

Proof. There are at most n iterations, each of which can be performed in time $O(m)$. ∎

A negative-length directed circuit can be detected similarly:

Theorem 8.6. *Given a digraph $D = (V, A)$, $s \in V$, and a length function $l : A \to \mathbb{Q}$, a directed circuit of negative length reachable from s (if any exists) can be found in time $O(nm)$.*

Proof. If $d_n \neq d_{n-1}$, then $d_n(t) < d_{n-1}(t)$ for some $t \in V$. So the algorithm finds an $s-t$ walk P of length $d_n(t)$, traversing n arcs. As P traverses n arcs, it contains a directed circuit C. Removing C gives an $s-t$ walk P' with less than n arcs. So $l(P') \geq d_{n-1}(t) > d_n(t) = l(P)$ and hence $l(C) < 0$.

If $d_n = d_{n-1}$, then there is no negative-length directed circuit reachable from s. ∎

Also a potential can be found with the Bellman-Ford method:

Theorem 8.7. *Given a digraph $D = (V, A)$ and a length function $l : A \to \mathbb{Q}$ such that each directed circuit has nonnegative length, a potential can be found in time $O(nm)$.*

Proof. Extend D by a new vertex s and arcs (s, v) for $v \in V$, each of length 0. Then setting $p(v)$ equal to the distance from s to v (which can be determined with the Bellman-Ford method) gives a potential. ∎

We remark that the shortest path problem for *undirected* graphs, for length functions without negative-length circuits, also can be solved in polynomial time. However, the obvious reduction — replacing every undirected edge uv by two arcs (u, v) and (v, u) each of length $l(uv)$ — may yield a negative-length directed circuit. So in this case, the undirected case does not reduce to the directed case, and we cannot apply the Bellman-Ford method. The undirected problem can yet be solved in polynomial time, with the methods developed for the matching problem — see Section 29.2.

8.4. All-pairs shortest paths

Let $D = (V, A)$ be a digraph and $l : A \to \mathbb{Q}$ be a length function such that each directed circuit has nonnegative length. By applying $|V|$ times the Bellman-Ford method one can find shortest $s-t$ paths for all $s, t \in V$. As the Bellman-Ford method takes time $O(nm)$, this makes an $O(n^2m)$ algorithm.

A more efficient algorithm, the *Floyd-Warshall method* was described by Floyd [1962b], based on an idea of Warshall [1962], earlier found by Kleene [1956], Roy [1959], and McNaughton and Yamada [1960]: Order the vertices of D (arbitrarily) as $v_1, \ldots, v_n$. Define for $s, t \in V$ and $k \in \{0, \ldots, n\}$:

(8.5) $d_k(s, t) :=$ minimum length of an $s - t$ walk using only vertices in $\{s, t, v_1, \ldots, v_k\}$.

Clearly, $d_0(s, t) = l(s, t)$ if $(s, t) \in A$, while $d_0(s, t) = \infty$ otherwise. Moreover:

(8.6) $$d_{k+1}(s, t) = \min\{d_k(s, t), d_k(s, v_{k+1}) + d_k(v_{k+1}, t)\}$$

for all $s, t \in V$ and $k < n$.

This gives:

Theorem 8.8. *Given a digraph $D = (V, A)$ and a length function $l : A \to \mathbb{Q}$ with no negative-length directed circuit, all distances $\mathrm{dist}_l(s, t)$ can be determined in time $O(n^3)$.*

Proof. Note that $\mathrm{dist}_l = d_n$ and that d_n can be determined in n iterations, each taking $O(n^2)$ time. ∎

The Floyd-Warshall method can be adapted so as to find for all $s \in V$, a shortest $s-t$ paths tree rooted at s.

A faster method was observed by Johnson [1973b,1977a] and Bazaraa and Langley [1974]. Combined with the Fibonacci heap implementation of Dijkstra's algorithm, it gives all-pairs shortest paths in time $O(n(m+n\log n))$, which is, if $n\log n = O(m)$, of the same order as the Bellman-Ford for *single*-source shortest path. The idea is to preprocess the data by a potential function, so as to make the length function nonnegative, and next to apply Dijkstra's method:

Theorem 8.9. *Given a digraph $D=(V,A)$ and a length function $l : A \to \mathbb{Q}$ with no negative-length directed circuit, a family $(T_s \mid s \in V)$ of shortest paths trees T_s rooted at s can be found in time $O(n(m+n\log n))$.*

Proof. With the Bellman-Ford method one finds a potential p in time $O(nm)$ (Theorem 8.7). Set $\tilde{l}(a) := l(a) - p(v) + p(u)$ for each arc $a=(u,v)$. So $\tilde{l}(a) \geq 0$ for each arc a. Next with Dijkstra's method, using Fibonacci heaps, one can determine for each $s \in V$ a shortest paths tree T_s for $\tilde{l}$, in time $O(m+n\log n)$ (Corollary 7.7a). As these are shortest paths trees also for l, we have the current theorem. ∎

8.5. Finding a minimum-mean length directed circuit

Let $D=(V,A)$ be a directed graph (with n vertices) and let $l : A \to \mathbb{R}$. The *mean length* of a directed cycle (directed closed walk) $C=(v_0,a_1,v_1,\ldots,a_t,v_t)$ with $v_t = v_0$ and $t>0$ is $l(C)/t$. Karp [1978] gave the following polynomial-time method for finding a directed cycle of minimum mean length. For each $v \in V$ and each $k=0,1,2,\ldots$, let $d_k(v)$ be the minimum length of a walk with exactly k arcs, ending at v. So for each v one has

(8.7) $$d_0(v) = 0 \text{ and } d_{k+1}(v) = \min\{d_k(u) + l(a) \mid a=(u,v) \in \delta^{\text{in}}(v)\}.$$

Now Karp [1978] showed:

Theorem 8.10. *The minimum mean length of a directed cycle in D is equal to*

(8.8) $$\min_{v\in V}\max_{0\leq k\leq n-1}\frac{d_n(v)-d_k(v)}{n-k}.$$

Proof. We may assume that the minimum mean length is 0, since adding ε to the length of each arc increases both minima in the theorem by ε. So we must show that (8.8) equals 0.

First, let minimum (8.8) be attained by v. Let P_n be a walk with n arcs ending at v, of length $d_n(v)$. So P_n can be decomposed into a path P_k, say, with k arcs ending at v, and a directed cycle C with $n-k$ arcs (for some $k < n$). Hence $d_n(v) = l(P_n) = l(P_k) + l(C) \geq l(P_k) \geq d_k(v)$ and so $d_n(v) - d_k(v) \geq 0$. Therefore, (8.8) is nonnegative.

To see that it is 0, let $C = (v_0, a_1, v_1, \ldots, a_t, v_t)$ be a directed cycle of length 0. Then $\min_r d_r(v_0)$ is attained by some r with $n - t \leq r < n$ (as it is attained by some $r < n$ (since each circuit has nonnegative length), and as we can add C to the shortest walk ending at v_0). Fix this r.

Let $v := v_{n-r}$, and split C into walks

(8.9) $$P := (v_0, a_1, v_1, \ldots, a_{n-r}, v_{n-r}) \text{ and}$$
$$Q := (v_{n-r}, a_{n-r+1}, v_{n-r+1}, \ldots, a_t, v_t).$$

Then $d_n(v) \leq d_r(v_0) + l(P)$, and therefore for each k:

(8.10) $$d_k(v) + l(Q) \geq d_{k+(t-(n-r))}(v_0) \geq d_r(v_0) \geq d_n(v) - l(P).$$

This implies $d_n(v) - d_k(v) \leq l(C) = 0$. So the minimum (8.8) is at most 0. ∎

Algorithmically, it gives:

Corollary 8.10a. *A minimum-mean length directed circuit can be found in time $O(nm)$.*

Proof. See the method above. ∎

Notes. Karp and Orlin [1981] and Karzanov [1985c] gave generalizations. Orlin and Ahuja [1992] gave an $O(\sqrt{n}\, m \log(nL))$ algorithm for the minimum-mean length directed circuit problem (cf. McCormick [1993]). Early work on this problem includes Lawler [1967], Shapiro [1968], and Fox [1969].

8.6. Further results and notes

8.6a. Complexity survey for shortest path without negative-length circuits

The following gives a survey of the development of the running time bound for the shortest path problem for a digraph $D = (V, A)$, $s, t \in V$, and $l : A \to \mathbb{Z}$ (without negative-length directed circuits), where $n := |V|$, $m := |A|$, $L := \max\{|l(a)| \mid a \in A\}$, and $L' := \max\{-l(a) \mid a \in A\}$ (assuming $L' \geq 2$). As before, $*$ indicates an asymptotically best bound in the table.

	$O(n^4)$	Shimbel [1955]
	$O(n^2 mL)$	Ford [1956]
$*$	$O(nm)$	Bellman [1958], Moore [1959]
	$O(n^{3/4} m \log L')$	Gabow [1983b,1985b]

≫

continued

	$O(\sqrt{n}\, m \log(nL'))$	Gabow and Tarjan [1988b,1989]
*	$O(\sqrt{n}\, m \log L')$	Goldberg [1993b,1995]

(Gabow [1983b,1985b] and Gabow and Tarjan [1988b,1989] give bounds with L instead of L', but Goldberg [1995] mentioned that an anonymous referee of his paper observed that L can be replaced by L'.)

Kolliopoulos and Stein [1996,1998b] proved a bound of $o(n^3)$ for the average-case complexity.

For the special case of *planar* directed graphs:

	$O(n^{3/2})$	Lipton, Rose, and Tarjan [1979]
	$O(n^{4/3} \log(nL'))$	Klein, Rao, Rauch, and Subramanian [1994], Henzinger, Klein, Rao, and Subramanian [1997]
*	$O(n \log^3 n)$	Fakcharoenphol and Rao [2001]

For the all-pairs shortest paths problem, with no negative-length directed circuits, one has:

	$O(n^4)$	Shimbel [1955]
	$O(n^3 \log n)$	Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957]
	$O(n^2 m)$	Bellman [1958], Moore [1959]
	$O(n^3)$	Floyd [1962b]
*	$O(n \cdot \mathrm{SP}_+(n, m, L))$	Johnson [1973b,1977a], Bazaraa and Langley [1974]
	$O(nm + n^3(\log\log n / \log n)^{1/3})$	Fredman [1976]
	$O(nm \log\log L + n^2 \log L \log\log L)$	Johnson [1977b]
	$O(nL + nm \log\log L)$	van Emde Boas, Kaas, and Zijlstra [1977]
*	$O(nm \log\log L)$	Johnson [1982]
	$O(nm \log_{m/n} L)$	Gabow [1985b]
*	$O(nm + n^2\sqrt{\log L})$	Ahuja, Mehlhorn, Orlin, and Tarjan [1990]
*	$O((nL)^{\frac{3+\omega}{2}} \log^3 n)$	Alon, Galil, and Margalit [1991, 1997], Galil and Margalit [1997a, 1997b]

Here $\mathrm{SP}_+(n, m, L)$ denotes the time needed to find a shortest path in a digraph with n vertices and m arcs, with *nonnegative* integer lengths on the arcs, each at most L. ω is any real such that any two $n \times n$ matrices can be multiplied by $O(n^\omega)$ arithmetic operations (e.g. $\omega = 2.376$).

Frederickson [1983b,1987b] showed that for *planar* directed graphs, the all-pairs shortest paths problem, with no negative-length directed circuits, can be solved in $O(n^2)$ time.

8.6b. NP-completeness of the shortest path problem

In full generality — that is, not requiring that each directed circuit has nonnegative length — the shortest path problem is NP-complete, even if each arc has length -1. Equivalently, finding a longest path in a graph (with unit length arcs) is NP-complete. This is a result of E.L. Lawler and R.E. Tarjan (cf. Karp [1972b]).

This directly follows from the NP-completeness of finding a Hamiltonian path in a graph. Let $D = (V, A)$ be a digraph. (A directed path P is called *Hamiltonian* if each vertex of D is traversed exactly once.)

We show the NP-completeness of the *directed Hamiltonian path problem*: Given a digraph $D = (V, A)$ and $s, t \in V$, is there a Hamiltonian $s - t$ path?

Theorem 8.11. *The directed Hamiltonian path problem is* NP*-complete.*

Proof. We give a polynomial-time reduction of the partition problem (Section 4.11) to the directed Hamiltonian path problem. Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a collection of subsets of the set $X = \{1, \ldots, k\}$. Introduce vertices $r_0, r_1, \ldots, r_m, 0, 1, \ldots, k$.

For each $i = 1, \ldots, m$, we do the following. Let $C_i = \{j_1, \ldots, j_t\}$. We construct a digraph on the vertices r_{i-1}, r_i, $j_h - 1, j_h$ (for $h = 1, \ldots, t$) and $3t$ new vertices, as in Figure 8.1. Moreover, we make an arc from r_m to 0.

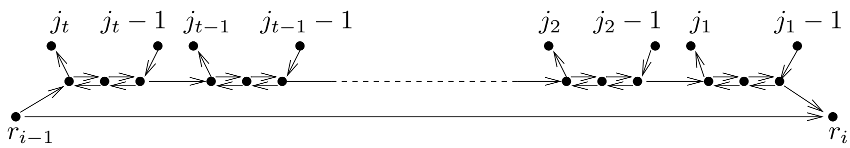


Figure 8.1

Let D be the digraph arising in this way. Then it is not difficult to check that there exists a subcollection $\mathcal{C}'$ of $\mathcal{C}$ that partitions X if and only if D has a directed Hamiltonian $r_0 - k$ path P. (Take: $(r_{i-1}, r_i) \in P \iff C_i \in \mathcal{C}'$.) ∎

Hence:

Corollary 8.11a. *Given a digraph $D = (V, A)$ and $s, t \in V$, finding a longest $s - t$ path is* NP*-complete.*

Proof. This follows from the fact that there exists an $s - t$ path of length $|V| - 1$ if and only if there is a directed Hamiltonian $s - t$ path. ∎

From this we derive the NP-completeness of the *undirected Hamiltonian path problem*: Given a graph $G = (V, E)$ and $s, t \in V$, does G have a Hamiltonian $s - t$ path? (R.E. Tarjan (cf. Karp [1972b])).

Corollary 8.11b. *The undirected Hamiltonian path problem is* NP*-complete.*

Proof. We give a polynomial-time reduction of the directed Hamiltonian path problem to the undirected Hamiltonian path problem. Let D be a digraph. Replace each vertex v by three vertices v', v'', v''', and make edges $\{v', v''\}$ and $\{v'', v'''\}$. Moreover, for each arc (v_1, v_2) of D, make an edge $\{v_1''', v_2'\}$. Delete the vertices s', s'', t'', t'''. This makes the undirected graph G. One easily checks that D has a directed Hamiltonian $s-t$ path if and only if G has an (undirected) Hamiltonian $s'''-t'$ path. ∎

Again it implies:

Corollary 8.11c. *Given an undirected graph $G = (V, E)$ and $s, t \in V$, finding a longest $s-t$ path is* NP*-complete.*

Proof. This follows from the fact that there exists an $s-t$ path of length $|V|-1$ if and only if there is a Hamiltonian $s-t$ path. ∎

Notes. Corollary 8.11b implies that finding a Hamiltonian circuit in an undirected graph is NP-complete: just add a new vertex r and edges rs and rt. This reduces finding a Hamiltonian $s-t$ path in the original graph to finding a Hamiltonian circuit in the extended graph.

Also the directed Hamiltonian circuit problem is NP-complete, as the undirected version can be reduced to it by replacing each edge uv by two oppositely oriented arcs (u, v) and (v, u).

8.6c. Nonpolynomiality of Ford's method

The method originally described by Ford [1956] consists of the following. Given a digraph $D = (V, A)$, $s, t \in V$, and a length function $l : A \to \mathbb{Q}$, define $d(s) := 0$ and $d(v) := \infty$ for all $v \neq s$; next perform the following iteratively:

(8.11) choose an arc (u, v) with $d(v) > d(u) + l(u, v)$, and reset $d(v) := d(u) + l(u, v)$.

Stop if no such arc exists.

If there are no negative-length directed circuits, this is a finite method, since at each iteration $\sum_v d(v)$ decreases, while it is at least $\sum_v \mathrm{dist}_l(s, v)$ and it is an integer multiple of the g.c.d. of the $l(a)$.

In fact, it can be shown that the number of iterations is at most $2^{|V|}$, if l is nonnegative. Moreover, if l is arbitrary (without negative directed circuit), there are at most $2|V|^2 L$ iterations, where $L := \max\{|l(a)| \mid a \in A\}$.

However, Johnson [1973a,1973b] showed that the number of iterations is $\Omega(n2^n)$, even if we prescribe to choose (u, v) in (8.11) with $d(u)$ minimal. For nonnegative l, Johnson [1973b,1977a] showed that the number of iterations is $\Omega(2^n)$ if we prescribe no selection rule of u.

8.6d. Shortest and longest paths in acyclic graphs

Let $D = (V, A)$ be a digraph. A subset C of A is called a *directed cut* if there is a subset U of V with $\emptyset \neq U \neq V$ such that $\delta^{\text{out}}(U) = C$ and $\delta^{\text{in}}(U) = \emptyset$. So each directed cut is a cut.

It is easy to see that, if D is acyclic, then a set B of arcs is contained in a directed cut if and only if no two arcs in B are contained in a directed path. Similarly, if D is acyclic, a set B of arcs is contained in a directed path if and only if no two arcs in B are contained in a directed cut.

Theorem 8.12. *Let $D = (V, A)$ be an acyclic digraph and let $s, t \in V$. Then the maximum length of an $s - t$ path is equal to the minimum number of directed $s - t$ cuts covering all arcs that are on at least one $s - t$ path.*

Proof. Any $s-t$ path of length k needs at least k directed $s-t$ cuts to be covered, so the maximum cannot exceed the minimum.

To see equality, let for each $v \in V$, $d(v)$ be equal to the length of a longest $s - v$ path. Let $k := d(t)$. For $i = 1, \ldots, k$, let $U_i := \{v \in V \mid d(v) < i\}$. Then the $\delta^{\text{out}}(U_i)$ form k directed $s - t$ cuts covering all arcs that are on at least one $s - t$ path. ∎

One similarly shows for paths not fixing its ends (Vidyasankar and Younger [1975]):

Theorem 8.13. *Let $D = (V, A)$ be an acyclic digraph. Then the maximum length of any path is equal to the minimum number of directed cuts covering A.*

Proof. Similar to the proof above. ∎

Also weighted versions hold, and may be derived similarly. A weighted version of Theorem 8.12 is:

Theorem 8.14. *Let $D = (V, A)$ be an acyclic digraph, let $s, t \in V$ and let $l : A \to \mathbb{Z}_+$ be a length function. Then the maximum length of an $s - t$ path is equal to the minimum number of directed $s - t$ cuts covering each arc a that is on at least one $s - t$ path, at least $l(a)$ times.*

Proof. Any $s-t$ path of length k needs at least k directed $s-t$ cuts to be covered appropriately, so the maximum cannot exceed the minimum.

To see equality, let for each $v \in V$, $d(v)$ be equal to the length of a longest $s - v$ path. Let $k := d(t)$. For $i = 1, \ldots, k$, let $U_i := \{v \in V \mid d(v) < i\}$. Then the $\delta^{\text{out}}(U_i)$ form k directed $s - t$ cuts covering each arc a on any $s - t$ path at least $l(a)$ times. ∎

Similarly, a weighted version of Theorem 8.13 is:

Theorem 8.15. *Let $D = (V, A)$ be an acyclic digraph and $l : A \to \mathbb{Z}_+$ be a length function. Then the maximum length of any path is equal to the minimum number of directed cuts such that any arc a is in at least $l(a)$ of these directed cuts.*

Proof. Similar to the proof above. ∎

In acyclic graphs one can find shortest paths in *linear* time (Morávek [1970]):

Theorem 8.16. *Given an acyclic digraph $D = (V, A)$, $s, t \in V$, and a length function $l : A \to \mathbb{Q}$, a shortest $s - t$ path can be found in time $O(m)$.*

Proof. First order the vertices reachable from s topologically as $v_1, \ldots, v_n$ (cf. Corollary 6.5b). So $v_1 = s$. Set $d(v_1) := 0$ and determine

(8.12) $$d(v) := \min\{d(u) + l(u, v) \mid (u, v) \in \delta^{\text{in}}(v)\}$$

for $v = v_2, \ldots, v_n$ (in this order). Then for each v reachable from s, $d(v)$ is the distance from s to v. ∎

Note that this implies that also a *longest* path in an acyclic digraph can be found in linear time.

Johnson [1973b] showed that, in a not necessarily acyclic digraph, an $O(m)$-time algorithm for the single-source shortest path problem exists if the number of directed circuits in any strongly connected component is bounded by a constant. Related work was reported by Wagner [2000].

More on longest paths and path covering in acyclic graphs can be found in Chapter 14.

8.6e. Bottleneck shortest path

Pollack [1960] observed that several of the shortest path algorithms can be modified to the following maximum-capacity path problem. For any digraph $D = (V, A)$ and 'capacity' function $c : A \to \mathbb{Q}$, the *capacity* of a path P is the minimum of the capacities of the arcs in P. (This is also called sometimes the *reliability* of P — cf. Section 50.6c.)

Then the *maximum-capacity path problem* (also called the *maximum reliability problem*), is:

(8.13) given: a digraph $D = (V, A)$, $s, t \in V$, and a 'capacity' function $c : A \to \mathbb{Q}$;
find: an $s - t$ path of maximum capacity.

To this end, one should appropriately replace min by max and + by min in these algorithms. Applying this to Dijkstra's algorithm gives, with Fibonacci heaps, a running time of $O(m + n \log n)$.

In fact, the following 'bottleneck' min-max relation holds (Fulkerson [1966]):

Theorem 8.17. *Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let $c : A \to \mathbb{R}$ be a capacity function. Then:*

(8.14) $$\max_P \min_{a \in AP} c(a) = \min_C \max_{a \in C} c(a),$$

where P ranges over all $s - t$ paths and C over all $s - t$ cuts.

Proof. To see $\leq$ in (8.14), let P be an $s-t$ path and let C be an $s-t$ cut. Since P and C have at least one arc in common, say a_0, we have $\min_{a \in AP} c(a) \leq c(a_0) \leq \max_{a \in C} c(a)$.

To see $\geq$ in (8.14), let $\gamma := \max_P \min_{a \in AP} c(a)$. Let $A' := \{a \in A \mid c(a) \leq \gamma\}$. Then A' intersects each $s-t$ path. So A' contains an $s-t$ cut C. Therefore, $c(a) \leq \gamma$ for all $a \in C$; that is, $\max_{a \in C} c(a) \leq \gamma$. Hence $\min_C \max_{a \in C} c(a) \leq \gamma$. ∎

It is easy to solve the bottleneck shortest path problem by binary search in time $O(m \log L)$, where $L := \|l\|_\infty$ assuming l integer. This was improved by Gabow [1985b] to $O(m \log_n L)$.

8.6f. Further notes

Further analyses of shortest path methods were given by Pollack and Wiebenson [1960], Hoffman and Winograd [1972], Tabourier [1973], Pape [1974], Kershenbaum [1981], Glover, Glover, and Klingman [1984], Pallottino [1984], Glover, Klingman, and Phillips [1985], Glover, Klingman, Phillips, and Schneider [1985], Desrochers [1987], Bertsekas [1991], Goldfarb, Hao, and Kai [1991], Sherali [1991], Cherkassky, Goldberg, and Radzik [1994,1996], and Cherkassky and Goldberg [1999] (negative circuit detection).

Spirakis and Tsakalidis [1986] gave an average-case analysis of an $O(nm)$-time negative circuit detecting algorithm, and Tarjan [1982] a sensitivity analysis of shortest paths trees.

Fast approximation algorithms for shortest paths were given by Klein and Sairam [1992], Cohen [1994b,2000], Aingworth, Chekuri, and Motwani [1996], Dor, Halperin, and Zwick [1996,2000], Cohen and Zwick [1997,2001], and Aingworth, Chekuri, Indyk, and Motwani [1999].

Dantzig [1957] observed that the shortest path problem can be formulated as a linear programming problem, and hence can be solved with the simplex method. Edmonds [1970a] showed that this may take exponentially many pivot steps, even for nonnegative arc lengths. On the other hand, Dial, Glover, Karney, and Klingman [1979] and Zadeh [1979] gave pivot rules that solve the shortest path problem with nonnegative arc lengths in $O(n)$ pivots. For arbitrary length Akgül [1985] and Goldfarb, Hao, and Kai [1990b] gave strongly polynomial simplex algorithms. Akgül [1993] gave an algorithm using $O(n^2)$ pivots, yielding an $O(n(m + n \log n))$-time algorithm. An improvement to $O(nm)$ (with $(n-1)(n-2)/2$ pivots) was given by Goldfarb and Jin [1999b]. Related work was done by Dantzig [1963], Orlin [1985], and Ahuja and Orlin [1988,1992].

Computational results were presented by Pape [1974,1980], Golden [1976], Carson and Law [1977], Kelton and Law [1978], van Vliet [1978], Denardo and Fox [1979], Dial, Glover, Karney, and Klingman [1979], Glover, Glover, and Klingman [1984], Imai and Iri [1984], Glover, Klingman, Phillips, and Schneider [1985], Gallo and Pallottino [1988], Mondou, Crainic, and Nguyen [1991], Goldberg and Radzik [1993] (Bellman-Ford method), Cherkassky, Goldberg, and Radzik [1996], Goldberg and Silverstein [1997], and Zhan and Noon [1998].

Frederickson [1987a,1991] gave an algorithm that gives a succint encoding of all pairs shortest path information in a directed planar graph (with arbitrary lengths, but no negative directed circuits).

Surveys and bibliographies on shortest paths were presented by Pollack and Wiebenson [1960], Murchland [1967a], Dreyfus [1969], Gilsinn and Witzgall [1973], Pierce [1975], Yen [1975], Lawler [1976b], Golden and Magnanti [1977], Deo and Pang [1984], Gallo and Pallottino [1986], and Mondou, Crainic, and Nguyen [1991].

Books covering shortest path methods include Christofides [1975], Lawler [1976b], Even [1979], Hu [1982], Papadimitriou and Steiglitz [1982], Smith [1982], Sysło, Deo, and Kowalik [1983], Tarjan [1983], Gondran and Minoux [1984], Nemhauser and Wolsey [1988], Bazaraa, Jarvis, and Sherali [1990], Chen [1990], Cormen, Leiserson, and Rivest [1990], Lengauer [1990], Ahuja, Magnanti, and Orlin [1993], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], Mehlhorn and Näher [1999], and Korte and Vygen [2000].

8.6g. Historical notes on shortest paths

Compared with other combinatorial optimization problems like the minimum spanning tree, assignment, and transportation problems, research on the shortest path problem started relatively late. This might be due to the fact that the problem is elementary and relatively easy, which is also illustrated by the fact that at the moment that the problem came into the focus of interest, several researchers independently developed similar methods. Yet, the problem has offered some substantial difficulties, as is illustrated by the fact that heuristic, nonoptimal approaches have been investigated (cf. for instance Rosenfeld [1956], who gave a heuristic approach for determining an optimal trucking route through a given traffic congestion pattern).

Search methods

Depth-first search methods were described in the 19th century in order to traverse all lanes in a maze without knowing its plan. Wiener [1873] described the following method:

> Man markire sich daher den Weg, den man zurücklegt nebst dem Sinne, in welchem es geschieht. Sobald man auf einen schon markirten Weg stösst, kehre man um und durchschreite den schon beschriebenen Weg in umgekehrtem Sinne. Da man, wenn man nicht ablenkte, denselben hierbei in seiner ganzen Ausdehnung nochmals zurücklegen würde, so muss mann nothwendig hierbei auf einen noch nicht markirten einmündenden Weg treffen, den man dann verfolge, bis man wieder auf einen markirten trifft. Hier kehre man wieder um und verfahre wie vorher. Es werden dadurch stets neue Wegtheile zu den beschriebenen zugefügt, so dass man nach einer endlichen Zeit das ganze Labyrinth durchwandern würde und so jedenfalls den Ausgang fände, wenn er nicht schon vorher erreicht worden wäre.[3]

[3] One therefore marks the road that one traverses together with the direction in which it happens. As soon as one hits a road already marked, one turns and traverses the road already followed in opposite direction. As one, if one would not deviate, would traverse it to its whole extent another time, by this one should necessarily meet a road running to a not yet marked one, which one next follows, until one again hits a marked one. Here one turns again and proceeds as before. In that road always new road parts are added to those already followed, so that after a finite time one would walk through the whole labyrinth and in this road in any case would find the exit, if it would not have been reached already before.

In his book *Recréations mathématiques*, Lucas [1882] described a method due to C.P. Trémaux to traverse all lanes of a maze exactly twice, starting at a vertex A: First traverse an arbitrary lane starting at A. Next apply the following rule iteratively when arriving through a lane L at a vertex N:

(8.15) if you did not visit N before, traverse next an arbitrary other lane at N, except if L is the only lane at N, in which case you return through L;
if you have visited N before, return through L, except if you have traversed L already twice; in that case traverse another lane at N not traversed before; if such a lane does not exist, traverse a lane at L that you have traversed before once.

The method stops if no lane can be chosen by this rule at N. It is not hard to show that then one is at the starting vertex A and that all lanes of the maze have been traversed exactly twice (if the maze is connected).

A simpler rule was given by Tarry [1895]:

> Tout labyrinthe peut être parcouru en une seule course, en passant deux fois en sens contraire par chacune des allées, sans qu'il soit nécessaire d'en connaître le plan.
> Pour résoudre ce problème, il suffit d'observer cette règle unique:
> *Ne reprendre l'allée initiale qui a conduit à un carrefour pour la première fois que lorsqu'on ne peut pas faire autrement.*[4]

This is equivalent to depth-first search.

Alternate routing

Path problems were also studied at the beginning of the 1950s in the context of 'alternate routing', that is, finding a second shortest route if the shortest route is blocked. This applies to freeway usage (cf. Trueblood [1952]), but also to telephone call routing. At that time making long-distance calls in the U.S.A. was automatized, and alternate routes for telephone calls over the U.S. telephone network nation-wide should be found automatically:

> When a telephone customer makes a long-distance call, the major problem facing the operator is how to get the call to its destination. In some cases, each toll operator has two main routes by which the call can be started towards this destination. The first-choice route, of course, is the most direct route. If this is busy, the second choice is made, followed by other available choices at the operator's discretion. When telephone operators are concerned with such a call, they can exercise choice between alternate routes. But when operator or customer toll dialing is considered, the choice of routes has to be left to a machine. Since the "intelligence" of a machine is limited to previously "programmed" operations, the choice of routes has to be decided upon, and incorporated in, an automatic alternate routing arrangement.

(Jacobitti [1955], cf. Myers [1953], Clos [1954], and Truitt [1954]).

[4] Each maze can be traversed in one single run, by passing each of the corridors twice in opposite direction, without that it is necessary to know its plan.
To solve this problem, it suffices to observe this only rule:
Retake the initial corridor that has led to a crossing for the first time only when one cannot do otherwise.

Matrix methods

Matrix methods were developed to study relations in networks, like finding the transitive closure of a relation; that is, identifying in a digraph the pairs of vertices s, t such that t is reachable from s. Such methods were studied because of their application to communication nets (including neural nets) and to animal sociology (e.g. peck rights).

The matrix methods consist of representing the relation by a matrix, and then taking iterative matrix products to calculate the transitive closure. This was studied by Landahl and Runge [1946], Landahl [1947], Luce and Perry [1949], Luce [1950], Lunts [1950,1952], and by A. Shimbel.

Shimbel's interest in matrix methods was motivated by their applications to neural networks. He analyzed with matrices which sites in a network can communicate to each other, and how much time it takes. To this end, let S be the $0,1$ matrix indicating that if $S_{i,j} = 1$, then there is direct communication from i to j (including $i = j$). Shimbel [1951] observed that the positive entries in S^t correspond to the pairs between which there exists communication in t steps. An *adequate* communication system is one for which S^t is positive for some t. One of the other observations of Shimbel [1951] is that in an adequate communication system, the time it takes that all sites have all information, is equal to the minimum value of t for which S^t is positive. (A related phenomenon was observed by Luce [1950].)

Shimbel [1953] mentioned that the distance from i to j is equal to the number of zeros in the i, j position in the matrices $S^0, S^1, S^2, \ldots, S^t$. So essentially he gave an $O(n^4)$ algorithm to find all distances in a (unit-length) digraph.

Shortest paths

The basic methods for the shortest path problem are the Bellman-Ford method and Dijkstra's method. The latter one is faster but is restricted to nonnegative length functions. The former method only requires that there is no directed circuit of negative length.

The general framework for both methods is the following scheme, described in this general form by Ford [1956]. Keep a provisional distance function d. Initially, set $d(s) := 0$ and $d(v) := \infty$ for each $v \neq s$. Next, iteratively,

(8.16) choose an arc (u, v) with $d(v) > d(u) + l(u, v)$ and reset $d(v) := d(u) + l(u, v)$.

If no such arc exists, d is the distance function.

The difference in the methods is the rule by which the arc (u, v) with $d(v) > d(u) + l(u, v)$ is chosen. The Bellman-Ford method consists of considering all arcs consecutively and applying (8.16) where possible, and repeating this (at most $|V|$ rounds suffice). This is the method described by Shimbel [1955], Bellman [1958], and Moore [1959].

Dijkstra's method prescribes to choose an arc (u, v) with $d(u)$ smallest (then each arc is chosen at most once, if the lengths are nonnegative). This was described by Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957] and Dijkstra [1959]. A related method, but slightly slower than Dijkstra's method when implemented, was given by Dantzig [1958], G.J. Minty, and Whiting and Hillier [1960], and chooses an arc (u, v) with $d(u) + l(u, v)$ smallest.

Parallel to this, a number of further results were obtained on the shortest path problem, including a linear programming approach and 'good characterizations'.

We now describe the developments in greater detail.

The Bellman-Ford method: Shimbel

In April 1954, Shimbel [1955] presented at the Symposium on Information Networks in New York some observations on calculating distances, which amount to describing a 'min-addition' algebra and a method which later became known as the Bellman-Ford method. He introduced:

> Arithmetic
> For any arbitrary real or infinite numbers x and y
>
> $$x + y \equiv \min(x, y) \text{ and}$$
> $$xy \equiv \text{the algebraic sum of } x \text{ and } y.$$

He extended this arithmetic to the matrix product. Calling the distance matrix associated with a given length matrix S the 'dispersion', he stated:

> It follows trivially that S^k $k \geq 1$ is a matrix giving the shortest paths from site to site in S given that $k-1$ other sites may be traversed in the process. It also follows that for any S there exists an integer k such that $S^k = S^{k+1}$. Clearly, the dispersion of S (let us label it $D(S)$) will be the matrix S^k such that $S^k = S^{k+1}$.

Although Shimbel did not mention it, one trivially can take $k \leq |V|$, and hence the method yields an $O(n^4)$ algorithm to find the distances between all pairs of vertices.

Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957] noted that Shimbel's method can be speeded up by calculating S^k by iteratively raising the current matrix to the square (in the min-addition matrix algebra). This solves the all-pairs shortest paths problem in time $O(n^3 \log n)$.

The Bellman-Ford method: Ford

In a RAND paper dated 14 August 1956, Ford [1956] described a method to find a shortest path from P_0 to P_N, in a network with vertices $P_0, \ldots, P_N$, where l_{ij} denotes the length of an arc from i to j:

> Assign initially $x_0 = 0$ and $x_i = \infty$ for $i \neq 0$. Scan the network for a pair P_i and P_j with the property that $x_i - x_j > l_{ji}$. For this pair replace x_i by $x_j + l_{ji}$. Continue this process. Eventually no such pairs can be found, and x_N is now minimal and represents the minimal distance from P_0 to P_N.

Ford next argues that the method terminates. It was shown by Johnson [1973a, 1973b,1977a] that Ford's liberal selection rule can require exponential time.

In their book *Studies in the Economics of Transportation*, Beckmann, McGuire, and Winsten [1956] showed that the distance matrix $D = (d_{i,j})$ is the unique matrix satisfying

(8.17)
$$d_{i,i} = 0 \text{ for all } i;$$
$$d_{i,k} = \min_j (l_{i,j} + d_{j,k}) \text{ for all } i, k \text{ with } i \neq k.$$

The Bellman-Ford method: Bellman

We next describe the work of Bellman on shortest paths. After publishing several papers on dynamic programming (in a certain sense a generalization of shortest path methods), Bellman [1958] eventually focused on the shortest path problem by itself. He described the following 'functional equation approach' (originating from dynamic programming) for the shortest path problem, which is the same as that of Shimbel [1955].

Bellman considered N cities, numbered $1, \dots, N$, every two of which are linked by a direct road, together with an $N \times N$ matrix $T = (t_{i,j})$, where $t_{i,j}$ is the time required to travel from i to j (not necessarily symmetric). Find a path between 1 and N which consumes minimum time. First, Bellman remarked that the problem is finite:

> Since there are only a finite number of paths available, the problem reduces to choosing the smallest from a finite set of numbers. This direct, or enumerative, approach is impossible to execute, however, for values of N of the order of magnitude of 20.

Next he gave a 'functional equation approach':

> The basic method is that of successive approximations. We choose an initial sequence $\{f_i^{(0)}\}$, and then proceed iteratively, setting
>
> $$f_i^{(k+1)} = \min_{j \neq i} \, (t_{ij} + f_j^{(k)}], \qquad i = 1, 2, \cdots, N-1,$$
>
> $$f_N^{(k+1)} = 0,$$
>
> for $k = 0, 1, 2 \cdots,$.

For the initial function $f_i^{(0)}$, Bellman proposed (upon a suggestion of F. Haight) to take $f_i^{(0)} = t_{i,N}$ for all i. Bellman observed that, for each fixed i, starting with this choice of $f_i^{(0)}$ gives that $f_i^{(k)}$ is monotonically nonincreasing in k, and states:

> It is clear from the physical interpretation of this iterative scheme that at most $(N-1)$ iterations are required for the sequence to converge to the solution.

Since each iteration can be done in time $O(N^2)$, the algorithm takes time $O(N^3)$. Bellman also remarks:

> It is easily seen that the iterative scheme discussed above is a feasible method for either hand or machine computation for values of N of the order of magnitude of 50 or 100.

In a footnote, Bellman says:

> *Added in proof* (*December* 1957): After this paper was written, the author was informed by Max Woodbury and George Dantzig that the particular iterative scheme discussed in Sec. 5 had been obtained by them from first principles.

Bellman [1958] mentioned that one could also start with $f_i^{(0)} = \min_{j \neq i} t_{i,j}$ (if $i \neq N$) and $f_N^{(0)} = 0$. In that case for each fixed i, the value of $f_i^{(k)}$ is monotonically *nondecreasing* in k, and converges to the distance from i to N. (Indeed, $f_i^{(k)}$ is equal to the shortest length of all those paths starting at i that have either exactly $k+1$ arcs, or have at most k arcs and end at N.)

The Bellman-Ford method: Moore

At the International Symposium on the Theory of Switching at Harvard University in April 1957, Moore [1959] of Bell Laboratories presented a paper 'The shortest path through a maze':

> The methods given in this paper require no foresight or ingenuity, and hence deserve to be called algorithms. They would be especially suited for use in a machine, either a special-purpose or a general-purpose digital computer.

The motivation of Moore was the routing of toll telephone traffic. He gave algorithms A, B, and C, and D.

First, Moore considered the case of an undirected graph $G = (V, E)$ with no length function, where a path from vertex A to vertex B should be found with a minimum number of edges. Algorithm A is: first give A label 0. Next do the following for $k = 0, 1, \ldots$: give label $k+1$ to all unlabeled vertices that are adjacent to some vertex labeled k. Stop as soon as vertex B is labeled.

> If it were done as a program on a digital computer, the steps given as single steps above would be done serially, with a few operations of the computer for each city of the maze; but, in the case of complicated mazes, the algorithm would still be quite fast compared with trial-and-error methods.

In fact, a direct implementation of the method would yield an algorithm with running time $O(m)$. It is essentially breadth-first search. Algorithms B and C differ from A in a more economical labeling (by fewer bits).

Moore's algorithm D finds a shortest route for the case where each edge of the graph has a nonnegative length. This method gives a refinement of Bellman's method described above: (i) it extends to the case that not all pairs of vertices have a direct connection; that is, if there is an underlying graph $G = (V, E)$ with length function; (ii) at each iteration only those $d_{i,j}$ are considered for which u_i has been decreased in the previous iteration.

The method has running time $O(nm)$. Moore observed that the algorithm is suitable for parallel implementation, yielding a decrease in the running time bound to $O(n\varDelta(G))$, where $\varDelta(G)$ is the maximum degree of G. He concluded:

> The origin of the present methods provides an interesting illustration of the value of basic research on puzzles and games. Although such research is often frowned upon as being frivolous, it seems plausible that these algorithms might eventually lead to savings of very large sums of money by permitting more efficient use of congested transportation or communication systems. The actual problems in communication and transportation are so much complicated by timetables, safety requirements, signal-to-noise ratios, and economic requirements that in the past those seeking to solve them have not seen the basic simplicity of the problem, and have continued to use trial-and-error procedures which do not always give the true shortest path. However, in the case of a simple geometric maze, the absence of these confusing factors permitted algorithms A, B, and C to be obtained, and from them a large number of extensions, elaborations, and modifications are obvious.
> The problem was first solved in connection with Claude Shannon's maze-solving machine. When this machine was used with a maze which had more than one solution, a visitor asked why it had not been built to always find the shortest path. Shannon and I each attempted to find economical methods of doing this by machine. He found several methods suitable for analog computation, and I obtained these algorithms. Months later the applicability of these ideas to practical problems in communication and transportation systems was suggested.

Among the further applications of his method, Moore described the example of finding the fastest connections from one station to another in a given railroad timetable (cf. also Levin and Hedetniemi [1963]). A similar method was given by Minty [1958].

Berge [1958b] described a breadth-first search method similar to Moore's algorithm A, to find the shortest paths from a given vertex a, for unit lengths, but he described it more generally for directed graphs: let $A(0) := \{a\}$; if $A(k)$ has been found, let $A(k+1)$ be the set of vertices x for which there is a $y \in A(k)$ with (y,x) an arc and with $x \notin A(i)$ for all $i \leq k$. One directly finds a shortest $a-b$ path from the $A(k)$. This gives an $O(m)$ algorithm.

D.A. D'Esopo (cf. the survey of Pollack and Wiebenson [1960]) proposed the following sharpening of Moore's version of the Bellman-Ford method, by indexing the vertices during the algorithm. First define index$(s) := 1$, and let $i := 1$. Then apply the following iteratively:

(8.18) Let v be the vertex with index i. For each arc (v,w) leaving v, reset $d(w) := d(v)+l(v,w)$ if it decreases $d(w)$; if w is not indexed give it the smallest unused index; if some $d(w)$ has been reset with index$(w) < i$, choose the w minimizing index(w), and let $i :=$index(w); otherwise, let $i := i+1$.

In May 1958, Hoffman and Pavley [1959b] reported, at the Western Joint Computer Conference in Los Angeles, the following computing time for finding the distances between all pairs of vertices by Moore's algorithm (with nonnegative lengths):

> It took approximately three hours to obtain the minimum paths for a network of 265 vertices on an IBM *704*.

Linear programming and transportation

Orden [1955] observed that the shortest path problem is a special case of the transshipment problem: let be given an $n \times n$ matrix $(c_{i,j})$ and a vector $g \in \mathbb{R}^n$

(8.19)
$$\begin{aligned}&\text{minimize } \textstyle\sum_{i,j} c_{i,j}x_{i,j}\\ &\text{subject to } \sum_{j=1}^{n}(x_{i,j}-x_{j,i}) = g_i \text{ for } i=1,\dots,n\\ &\text{and } x_{i,j} \geq 0 \text{ for } i,j = 1,\dots n,\end{aligned}$$

and showed that it can be reduced to a 'transportation problem', and hence to a linear programming problem. If one wants to find a shortest $1-n$ path, set $g_1 = 1$, $g_n = -1$, and $g_i = 0$ for all other i.

In a paper presented at the Summer 1955 meeting of ORSA at Los Angeles, Dantzig [1957] formulated the shortest path problem as an integer linear programming problem 'very similar to the system for the assignment problem', and similar to Orden's formulation. Dantzig observed that replacing the condition $x_{i,j} \geq 0$ by $x_{i,j} \in \{0,1\}$ does not change the minimum value. (Dantzig assumed $d_{i,j} = d_{j,i}$ for all i,j.)

He described a graphical procedure for the simplex method applied to this problem. Let T be a rooted tree on $\{1,\dots,n\}$, with root 1. For each $i = 1,\dots,n$, let u_i be equal to the length of the path from 1 to i in T. Now if $u_j \leq u_i + d_{i,j}$ for all i, j, then for each i, the $1 - i$ path in T is a shortest path. If $u_j > u_i + d_{i,j}$, replace the arc of T entering j by the arc (i, j), and iterate with the new tree.

Trivially, this process terminates (as $\sum_{j=1}^{n} u_j$ decreases at each iteration, and as there are only finitely many rooted trees). (Edmonds [1970a] showed that the method may take exponential time.) Dantzig illustrated his method by an example of sending a package from Los Angeles to Boston.

Good characterizations

Robacker [1956b] observed that the minimum length of a $P_0 - P_n$ path in a graph N is equal to the maximum number of disjoint $P_0 - P_n$ cuts:

> the maximum number of mutually disjunct cuts of N is equal to the length of the shortest chain of N from P_0 to P_n.

Gallai [1958b] noticed that if the length function $l : A \to \mathbb{Z}$ on the arcs of a digraph (V, A) gives no negative-length directed circuits, then there is a 'potential' $p : V \to \mathbb{Z}$ with $l(u, v) \geq p(v) - p(u)$ for each arc (u, v).

Case Institute of Technology 1957

In the *First Annual Report* of the project *Investigation of Model Techniques*, carried out by the Case Institute of Technology in Cleveland, Ohio for the Combat Development Department of the Army Electronic Proving Ground, Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957] describe (rudimentarily) a shortest path algorithm similar to Dijkstra's algorithm:

> (1) All the links joined to the origin, a, may be given an outward orientation. . . .
> (2) Pick out the link or links radiating from a, $a_{a\alpha}$, with the smallest delay. . . . Then it is impossible to pass from the origin to any other node in the network by any "shorter" path than $a_{a\alpha}$. Consequently, the minimal path to the general node α is $a_{a\alpha}$.
> (3) All of the other links joining α may now be directed outward. Since $a_{a\alpha}$ must necessarily be the minimal path to α, there is no advantage to be gained by directing any other links toward α. . . .
> (4) Once α has been evaluated, it is possible to evaluate immediately all other nodes in the network whose minimal values do not exceed the value of the second-smallest link radiating from the origin. Since the minimal values of these nodes are less than the values of the second-smallest, third-smallest, and all other links radiating directly from the origin, only the smallest link, $a_{a\alpha}$, can form a part of the minimal path to these nodes. Once a minimal value has been assigned to these nodes, it is possible to orient all other links except the incoming link in an outward direction.
> (5) Suppose that all those nodes whose minimal values do not exceed the value of the second-smallest link radiating from the origin have been evaluated. Now it is possible to evaluate the node on which the second-smallest link terminates. At this point, it can be observed that if conflicting directions are assigned to a link, in accordance with the rules which have been given for direction assignment, that link may be ignored. It will not be a part of the minimal path to either of the two nodes it joins. . . .
> Following these rules, it is now possible to expand from the second-smallest link as well as the smallest link so long as the value of the third-smallest link radiating from the origin is not exceeded. It is possible to proceed in this way until the entire network has been solved.

(In this quotation we have deleted sentences referring to figures.)

Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, and Seitz [1957] also described a speedup of solving the all-pairs shortest paths problem by matrix-multiplication:

> This process of multiplication may be simplified somewhat by squaring the original structure matrix to obtain a dispersion matrix which is the second power of the structure matrix; squaring the second-power matrix to obtain the fourth power of the structure matrix; and so forth.

This gives an $O(n^3 \log n)$-time all-pairs shortest paths algorithm.

Analog computing

In a reaction to the linear programming approach of Dantzig [1957] discussed above, Minty [1957] proposed an 'analog computer' for the shortest path problem:

> Build a string model of the travel network, where knots represent cities and string lengths represent distances (or costs). Seize the knot 'Los Angeles' in your left hand and the knot 'Boston' in your right and pull them apart. If the model becomes entangled, have an assistant untie and re-tie knots until the entanglement is resolved. Eventually one or more paths will stretch tight — they then are alternative shortest routes.
> Dantzig's 'shortest-route tree' can be found in this model by weighting the knots and picking up the model by the knot 'Los Angeles'.
> It is well to label the knots since after one or two uses of the model their identities are easily confused.

A similar method was proposed by Bock and Cameron [1958] (cf. Peart, Randolph, and Bartlett [1960]). The method was extended to the directed case by Klee [1964].

Rapaport and Abramson [1959] described an electric analog computer for solving the shortest path problem.

Dantzig's $O(n^2 \log n)$ algorithm

Dantzig [1958,1960] gave an $O(n^2 \log n)$ algorithm for the shortest path problem with nonnegative length function. A set X is updated throughout, together with a function $d : X \to \mathbb{Q}_+$. Initially $X = \{s\}$ and $d(s) = 0$. Then do the following iteratively:

(8.20) for each $v \in X$, let w_v be a vertex not in X with $d(w_v)$ minimal. Choose a $v \in X$ minimizing $d(v) + l(v, w_v)$. Add w_v to X and set $d(w_v) := d(v) + l(v, w_v)$.

Stop if $X = V$.

Note that throughout the iterations, the function d is only extended, and not updated. Dantzig assumed

> (a) that one can write down without effort for each node the arcs leading to other nodes in increasing order of length and (b) that it is no effort to ignore an arc of the list if it leads to a node that has been reached earlier.

Indeed, in a preprocessing the arcs can be ordered in time $O(n^2 \log n)$, and, for instance by using doubly linked lists, an arc can be deleted from the appropriate list in time $O(1)$. As each iteration can be done in time $O(n)$ (identifying a v

minimizing $d(v) + l(v, w_v)$ and deleting all arcs entering w_v from each list of arcs leaving x for $x \in X$), Dantzig's method can be performed in time $O(n^2 \log n)$.

Dantzig [1958,1960] mentioned that, beside Bellman, Moore, Ford, and himself, also D. Gale and Fulkerson proposed shortest path methods, 'in informal conversations'.

The same method as that of Dantzig (however without the observations concerning storing the outgoing arcs from any vertex in a list) was given by G.J. Minty (cf. Pollack and Wiebenson [1960]) and by Whiting and Hillier [1960].

Dijkstra's $O(n^2)$ algorithm

Dijkstra [1959] gave an $O(n^2)$ method which is slightly different from that of Dantzig [1958,1960]. Let $D = (V, A)$ be a graph and let a length function $l : A \to \mathbb{R}_+$ be given. Dijkstra's method consists of repeatedly updating a set X and a function $d : V \to \mathbb{R}_+$ as follows.

Initially, set $X = \emptyset$, $d(s) = 0$, $d(v) = \infty$ if $v \neq s$. Next move s into X. Then do the following iteratively: Let v be the vertex just moved into X;

(8.21) for each arc (v, w) with $w \notin X$, reset $d(w) := d(v) + l(v, w)$ if this would decrease $d(w)$. Choose $v' \notin X$ with minimum $d(v')$, and move v' into X.

Stop if no such v' exists.

Since each iteration can be done in time $O(n)$ and since there are at most $|V|$ iterations, the algorithm runs in time $O(n^2)$. Dijkstra states:

> The solution given above is to be preferred to the solution by L.R. FORD [*3*] as described by C. BERGE [*4*], for, irrespective of the number of branches, we need not store the data for all branches simultaneously but only those for the branches in sets I and II, and this number is always less than n. Furthermore, the amount of work to be done seems to be considerably less.

(Dijkstra's references [3] and [4] are Ford [1956] and Berge [1958b].)

Dijkstra's method is easier to implement (as an $O(n^2)$ algorithm) than Dantzig's, since we need not store the information in lists: in order to find $v' \notin X$ minimizing $d(v')$ we can just scan all vertices that are not in X.

Whiting and Hillier [1960] described the same method as Dijkstra.

Heaps

The 2-heap was introduced by Williams [1964] (describing it as an array, with subroutines to insert and extract elements of the heap), as a major improvement to the sorting algorithm 'treesort' of Floyd [1962a]. The 2-heap of Williams was next extended by Floyd [1964] to the sorting algorithm 'treesort3'.

In an erratum of 24 October 1968 to a report of the London Business School, Murchland [1967b] seems to be the first to use heaps for finding shortest paths, although he concludes to a time bound of $O(n^2 \log n)$ only — worse than Dijkstra's bound $O(n^2)$. E.L. Johnson [1972] improved Murchland's method to $O(m \log(n^2/m))$. He also considers the k-heap for arbitrary k ('k-tree').

In his Ph.D. thesis, D.B. Johnson [1973b], using a sharper analysis and k-heaps, obtains an $O((nd + m) \log_d n)$-time algorithm, implying algorithms with running time $O(m \log n)$ and $O(n^{1+\varepsilon} + m)$ (for each $\varepsilon > 0$) (published in D.B. Johnson [1977a]). Tarjan [1983] observed that taking $d := m/n$ gives $O(m \log_{m/n} n)$. Next, Fredman and Tarjan [1984,1987] showed that Fibonacci heaps give $O(m + n \log n)$.

All pairs: Roy, Warshall, Floyd, Dantzig

Based on a study of Kleene [1951,1956], McNaughton and Yamada [1960] gave a formula to calculate a 'regular expression' associated with a 'state graph' (essentially describing all paths from a given source) that is quite similar to the fast method of Roy [1959] and Warshall [1962] to compute the transitive closure $\overline{A}$ of a digraph $D = (V, A)$: Assume that the vertices are ordered $1, \ldots, n$. First set $\widetilde{A} := A$. Next, for $k = 1, \ldots, n$, add to $\widetilde{A}$ all pairs (i, j) for which both (i, k) and (k, j) belong to $\widetilde{A}$. The final $\widetilde{A}$ equals $\overline{A}$. This gives an $O(n^3)$ algorithm, which is faster than iterative matrix multiplication.

Floyd [1962b] extended this method to an algorithm to find all distances $d(i, j)$ given a length function $l : V \times V \to \mathbb{Q}_+$: First set $d(i, j) := l(i, j)$ for all i, j. Next, for $k = 1, \ldots, n$, reset, for all i, j, $d(i, j) := d(i, k) + d(k, j)$ if it decreases $d(i, j)$. This gives an $O(n^3)$ transitive closure algorithm for finding the distances between all pairs of vertices.

Dantzig [1967] proposed a variant of this method. For i, j, k with $i \leq k$ and $j \leq k$, let $d^k_{i,j}$ be the length of the shortest $i - j$ path in the graph induced by $\{1, \ldots, k\}$. Then there is an easy iterative scheme to determine the $d^k_{i,j}$ from the $d^{k-1}_{i,j}$: first $d^k_{i,k} = \min_{j<k}(d^{k-1}_{i,j} + l_{j,k})$ and $d^k_{k,i} = \min_{j<k}(l_{k,j} + d^{k-1}_{j,i})$. Next, for all $i, j < k$, $d^k_{i,j} = \min(d^{k-1}_{i,j}, d^k_{i,k} + d^k_{k,j})$.

Negative lengths

Ford and Fulkerson [1962] seem to be the first to observe that the Bellman-Ford method also works for arbitrary lengths as long as each directed circuit has nonnegative length. It is also implicit in the paper of Iri [1960].

PERT and CPM

The application of shortest path (and other) methods in the form of PERT (*Program Evaluation and Review Technique*, originally called *Program Evaluation Research Task*) started in 1958, and was reported by Malcolm, Roseboom, Clark, and Fazar [1959]. The use of the *Critical Path Method* (CPM) was described by Kelley [1957, 1961], and Kelley and Walker [1959].

The kth shortest path

Bock, Kantner, and Haynes [1957,1958] described a method to find the kth shortest path in a graph, based essentially on enumerating. Hoffman and Pavley [1959a] described an adaptation of Dantzig's tree method to obtain the kth shortest path. Bellman and Kalaba [1960] gave a method to find the kth shortest paths from a given vertex simultaneously to all other vertices. Also Pollack [1961b] described a method for the kth shortest path problem, especially suitable if k is small. A survey was given by Pollack [1961a].

Bottleneck path problems

Pollack [1960] modified the shortest path algorithm so as to obtain a path with maximum capacity (the capacity of a path is equal to the minimum of the capacities of the arcs in the path). Related work was done by Amara, Lindgren, and Pollack [1961].

Fanning out from both ends

Berge and Ghouila-Houri [1962] and Dantzig [1963] proposed to speed up Dijkstra's method by fanning out at both ends simultaneously. Berge and Ghouila-Houri [1962] proposed to stop as soon as a vertex is permanently labeled from both ends; however, one may see that this need not yield a shortest path.

Dantzig [1963] proposed to add an arc (s, v) with length $d(s, v)$ as soon as v is permanently labeled when fanning out from s, and similarly add an arc (w, t) with length $d(w, t)$ if a vertex w is labeled permanently when fanning out from t:

> The algorithm terminates whenever the fan of one of the problems reaches its terminal in the other.

Chapter 9

Disjoint paths

Having done with *shortest* paths, we now arrive at *disjoint* paths. We consider disjoint $s-t$ paths, where s and t are the same for all paths. The more general problem where we prescribe for each path a (possibly different) pair of ends, will be discussed in Part VII.
Menger's theorem equates the maximum number of disjoint $s-t$ paths to the minimum size of a cut separating s and t. There are several variants of Menger's theorem, all about equivalent: undirected, directed, vertex-disjoint, arc- or edge-disjoint. The meaning of 'cut' varies accordingly.
Next to Menger's min-max relation, we consider the algorithmic side of disjoint $s-t$ paths. This will be an extract from the related maximum flow algorithms to be discussed in the next chapter. (Maximum integer flow can be viewed as the capacitated version of disjoint $s-t$ paths.)

9.1. Menger's theorem

Menger [1927] gave a min-max theorem for the maximum number of disjoint $S-T$ paths in an undirected graph. It was observed by Grünwald [1938] (= T. Gallai) that the theorem also holds for directed graphs. We follow the proof given by Göring [2000].

Recall that a path is an $S-T$ *path* if it runs from a vertex in S to a vertex in T. A set C of vertices is called $S-T$ *disconnecting* if C intersects each $S-T$ path (C may intersect $S \cup T$).

Theorem 9.1 (Menger's theorem (directed vertex-disjoint version)). *Let $D=(V,A)$ be a digraph and let $S,T \subseteq V$. Then the maximum number of vertex-disjoint $S-T$ paths is equal to the minimum size of an $S-T$ disconnecting vertex set.*

Proof. Obviously, the maximum does not exceed the minimum. Equality is shown by induction on $|A|$, the case $A=\emptyset$ being trivial.

Let k be the minimum size of an $S-T$ disconnecting vertex set. Choose $a=(u,v) \in A$. If each $S-T$ disconnecting vertex set in $D-a$ has size at least k, then inductively there exist k vertex-disjoint $S-T$ paths in $D-a$, hence in D.

So we can assume that $D - a$ has an $S - T$ disconnecting vertex set C of size $\leq k - 1$. Then $C \cup \{u\}$ and $C \cup \{v\}$ are $S - T$ disconnecting vertex sets of D of size k.

Now each $S - (C \cup \{u\})$ disconnecting vertex set B of $D - a$ has size at least k, as it is $S - T$ disconnecting in D. Indeed, each $S - T$ path P in D intersects $C \cup \{u\}$, and hence P contains an $S - (C \cup \{u\})$ path in $D - a$. So P intersects B.

So by induction, $D - a$ contains k disjoint $S - C \cup \{u\}$ paths. Similarly, $D - a$ contains k disjoint $C \cup \{v\} - T$ paths. Any path in the first collection intersects any path in the second collection only in C, since otherwise $D - a$ contains an $S - T$ path avoiding C.

Hence, as $|C| = k - 1$, we can pairwise concatenate these paths to obtain disjoint $S - T$ paths, inserting arc a between the path ending at u and starting at v. ∎

A consequence of this theorem is a variant on *internally vertex-disjoint* $s - t$ paths, that is, $s - t$ paths having no vertex in common except for s and t. Recall that a set U of vertices is called an $s - t$ *vertex-cut* if $s, t \notin U$ and each $s - t$ path intersects U.

Corollary 9.1a (Menger's theorem (directed internally vertex-disjoint version))**.** *Let $D = (V, A)$ be a digraph and let s and t be two nonadjacent vertices of D. Then the maximum number of internally vertex-disjoint $s - t$ paths is equal to the minimum size of an $s - t$ vertex-cut.*

Proof. Let $D' := D - s - t$ and let S and T be the sets of outneighbours of s and of inneighbours of t, respectively. Then Theorem 9.1 applied to D', S, T gives the corollary. ∎

In turn, Theorem 9.1 follows from Corollary 9.1a by adding two new vertices s and t and arcs (s, v) for all $v \in S$ and (v, t) for all $v \in T$.

Also an arc-disjoint version can be derived (where paths are *arc-disjoint* is they have no arc in common). This version was first formulated by Dantzig and Fulkerson [1955,1956] for directed graphs and by Kotzig [1956] for undirected graphs.

Recall that a set C of arcs is an $s - t$ *cut* if $C = \delta^{\text{out}}(U)$ for some subset U of V with $s \in U$ and $t \notin U$.

Corollary 9.1b (Menger's theorem (directed arc-disjoint version))**.** *Let $D = (V, A)$ be a digraph and $s, t \in V$. Then the maximum number of arc-disjoint $s - t$ paths is equal to the minimum size of an $s - t$ cut.*

Proof. Let $L(D)$ be the line digraph of D and let $S := \delta_A^{\text{out}}(s)$ and $T := \delta_A^{\text{in}}(t)$. Then Theorem 9.1 for $L(D), S, T$ implies the corollary. Note that a minimum-size set of arcs intersecting each $s - t$ path necessarily is an $s - t$ cut. ∎

The internally vertex-disjoint version of Menger's theorem can be derived in turn from the arc-disjoint version: make digraph D' as follows from D: replace any vertex v by two vertices v', v'' and make an arc (v', v''); moreover, replace each arc (u, v) by (u'', v'). Then Corollary 9.1b for D', s'', t' gives Corollary 9.1a for D, s, t.

Similar theorems hold for *undirected* graphs. The undirected vertex-disjoint version follows immediately from Theorem 9.1 by replacing each undirected edge by two oppositely oriented arcs. Next, the undirected edge-disjoint version follows from the undirected vertex-disjoint version applied to the line graph (like the proof of Corollary 9.1b).

9.1a. Other proofs of Menger's theorem

The proof above of Theorem 9.1b was given by Göring [2000], which curtails the proof of Pym [1969a], which by itself is a simplification of a proof of Dirac [1966]. The basic idea (decomposition into two subproblems determined by a minimum-size cut) is due to Menger [1927], for the undirected vertex-disjoint version. (Menger's original proof contains a hole, closed by Kőnig [1931] — see Section 9.6e.)

Hajós [1934] gave a different proof for the undirected vertex-disjoint case, based on intersections and unions of sets determining a cut. Also the proofs given in Nash-Williams and Tutte [1977] are based on this. We give the first of their proofs.

Let $G = (V, E)$ be an undirected graph and let $s, t \in V$. Suppose that the minimum size of an $s - t$ vertex-cut is k. We show by induction on $|E|$ that there exist k vertex-disjoint $s - t$ paths.

The statement is trivial if each edge is incident with at least one of s and t. So we can consider an edge $e = xy$ incident with neither s nor t.

We can assume that $G - e$ has an $s - t$ vertex-cut C of size $k - 1$ — otherwise the statement follows by induction. Similarly, we can assume that G/e has an $s - t$ vertex-cut of size $k - 1$ — otherwise the statement follows by induction again. Necessarily, this cut contains the new vertex obtained by contracting e. Hence G has an $s - t$ vertex-cut C' of size k containing both x and y.

Now let C_s be the set of vertices in $C \cup C'$ that are reachable in G from s by a path with no internal vertex in $C \cup C'$. Similarly, let C_t be the set of vertices in $C \cup C'$ that are reachable in G from t by a path with no internal vertex in $C \cup C'$.

Trivially, C_s and C_t are $s - t$ vertex-cuts, and $C_s \cup C_t \subseteq C \cup C'$. Moreover, $C_s \cap C_t \subseteq C \cap C'$, since for any $v \in C_s \cap C_t$ there is an $s - t$ path P intersecting $C \cup C'$ only in v. As $x, y \in C'$, P does not traverse edge e. Hence $v \in C \cap C'$.

Therefore,

$$|C_s| + |C_t| \le |C| + |C'| \le 2k - 1, \tag{9.1}$$

contradicting the fact that C_s and C_t each have size at least k.

An augmenting path proof for the directed vertex-disjoint version was given by Grünwald [1938] (= T. Gallai), and for the directed arc-disjoint version by Ford and Fulkerson [1955,1957b] — see Section 9.2. (O'Neil [1978] gave a proof similar to that of Grünwald [1938].)

More proof ideas were given by Halin [1964,1968,1989], Hajós [1967], McCuaig [1984], and Böhme, Göring, and Harant [2001].

9.2. Path packing algorithmically

A specialization of the maximum flow algorithm of Ford and Fulkerson [1955, 1957b] (to be discussed in the next chapter) yields a polynomial-time algorithm to find a maximum number of disjoint $s-t$ paths and a minimum-size $s-t$ cut.

Define for any digraph D and any path P in D:

(9.2) $D \leftarrow P :=$ the digraph arising from D by reversing the orientation of each arc occurring in P.

Note that if P is an $s-t$ path in $D = (V, A)$, then for each $U \subseteq V$ with $s \in U$, $t \notin U$, we have

(9.3) $\delta^{\mathrm{out}}_{A'}(U) = \delta^{\mathrm{out}}_{A}(U) - 1,$

where A' is the arc set of $D \leftarrow P$.

Determine $D_0, D_1, \ldots$ as follows.

(9.4) Set $D_0 := D$. If D_k has been found and contains an $s-t$ path P, set $D_{k+1} := D_k \leftarrow P$. If D_k contains no $s-t$ path we stop.

The path P is called an *augmenting path*.

Now finding a minimum-size $s-t$ cut is easy: let U be the set of vertices reachable in the final D_k from s. Then $\delta^{\mathrm{out}}_{A}(U)$ is a minimum-size $s-t$ cut, by (9.3).

Also a maximum packing of $s-t$ paths can be derived. Indeed, the set B of arcs of D that are reversed in the final D_k contains k arc-disjoint $s-t$ paths in D. This can be seen as follows.

Let B_i be the set of arcs of D that are reversed in D_i, added with i parallel arcs from t to s. We show by induction on i that (V, B_i) is Eulerian. For $i = 0$, this is trivial. Suppose that it has been proved for i. Let P be the $s-t$ path in D_i with $D_{i+1} = D_i \leftarrow P$. Then $(V, B_i \cup AP \cup \{(t, s)\})$ is Eulerian. Since B_{i+1} arises from $B_i \cup AP \cup \{(t, s)\}$ by deleting pairs a, a^{-1} with $a \in B_i$ and $a^{-1} \in AP$, also (V, B_{i+1}) is Eulerian.

A consequence is that k arc-disjoint $s-t$ paths in B can be found in linear time.

Since an $s-t$ path in D_k can be found in time $O(m)$, and since there are at most $|A|$ arc-disjoint $s-t$ paths, one has:

Theorem 9.2. *A maximum collection of arc-disjoint $s-t$ paths and a minimum-size $s-t$ cut can be found in time $O(m^2)$.*

Proof. See above. ∎

Similarly one has for the vertex-disjoint variant:

Theorem 9.3. *A maximum collection of internally vertex-disjoint $s-t$ paths and a minimum-size $s-t$ vertex-cut can be found in time $O(nm)$.*

Proof. Apply the reduction described after Corollary 9.1b. In this case the number of iterations is at most $|V|$. ∎

One similarly derives for a fixed number k of arc-disjoint paths:

Corollary 9.3a. *Given a digraph $D = (V, A)$, $s, t \in V$, and a natural number k, we can find k arc-disjoint $s-t$ paths (if they exist) in time $O(km)$.*

Proof. Directly from the fact that the path P can be found in time $O(m)$. ∎

9.3. Speeding up by blocking path packings

The algorithm might be speeded up by selecting, at each iteration, not just *one* path P, but several arc-disjoint paths $P_1, \ldots, P_l$ in D_i at one go, and setting

(9.5) $$D_{i+1} := D_i \leftarrow P_1 \leftarrow \cdots \leftarrow P_l.$$

This might reduce the number of iterations — but of course this should be weighed against the increase in complexity of each iteration.

Such a speedup is obtained by a method of Dinits [1970] as follows. For any digraph $D = (V, A)$ and $s, t \in V$, let $\mu(D)$ denote the minimum length of an $s-t$ path in D. If no such path exists, set $\mu(D) = \infty$. If we choose the paths $P_1, \ldots, P_l$ in such a way that $\mu(D_{i+1}) > \mu(D_i)$, then the number of iterations clearly is not larger than $|V|$ (as $\mu(D_i) < |V|$ if finite).

We show that a collection $P_1, \ldots, P_l$ with the property that $\mu(D \leftarrow P_1 \leftarrow \cdots \leftarrow P_l) > \mu(D)$ indeed can be found quickly, namely in linear time.

To that end, call a collection of arc-disjoint $s-t$ paths $P_1, \ldots, P_l$ *blocking* if D contains no $s-t$ path arc-disjoint from $P_1, \ldots, P_l$. This is weaker than a maximum number of arc-disjoint paths, but a blocking collection can be found in linear time (Dinits [1970]):

Theorem 9.4. *Given an acyclic digraph $D = (V, A)$ and $s, t \in V$, a blocking collection of arc-disjoint $s-t$ paths can be found in time $O(m)$.*

Proof. With depth-first search we can find in time $O(|A'|)$ a subset A' of A and an $s-t$ path P_1 in A' such that no arc in $A' \setminus AP_1$ is contained in any $s-t$ path: scan s (cf. (6.2)) and stop as soon as t is reached; let A' be the set of arcs considered so far (as D is acyclic).

Next we find (recursively) a blocking collection $P_2, \ldots, P_k$ of arc-disjoint $s-t$ paths in the graph $D' := (V, A \setminus A')$. Then $P_1, \ldots, P_k$ is blocking in D. For suppose that D contains an $s-t$ path Q that is arc-disjoint from $P_1, \ldots, P_k$. Then $AQ \cap A' \neq \emptyset$, since $P_2, \ldots, P_k$ is blocking in D'. So AQ intersects AP_1, a contradiction. ∎

We also need the following. Let $\alpha(D)$ denote the set of arcs contained in at least one shortest $s-t$ path. Then:

Theorem 9.5. *Let $D = (V, A)$ be a digraph and let $s, t \in V$. Define $D' := (V, A \cup \alpha(D)^{-1})$. Then $\mu(D') = \mu(D)$ and $\alpha(D') = \alpha(D)$.*

Proof. It suffices to show that $\mu(D)$ and $\alpha(D)$ are invariant if we add a^{-1} to D for one arc $a \in \alpha(D)$. Suppose not. Then there is a directed $s-t$ path P in $A \cup \{a^{-1}\}$ traversing a^{-1}, of length at most $\mu(D)$. As $a \in \alpha(D)$, there is an $s-t$ path Q traversing a, of length $\mu(D)$. Hence $AP \cup AQ \setminus \{a, a^{-1}\}$ contains an $s-t$ path of length less than $\mu(D)$, a contradiction. ∎

The previous two theorems imply:

Corollary 9.5a. *Given a digraph $D = (V, A)$ and $s, t \in V$, a collection of arc-disjoint $s-t$ paths $P_1, \ldots, P_l$ with $\mu(D \leftarrow P_1 \leftarrow \cdots \leftarrow P_l) > \mu(D)$ can be found in time $O(m)$.*

Proof. Let $\widetilde{D} = (V, \alpha(D))$. (Note that $\alpha(D)$ can be identified in time $O(m)$ and that $(V, \alpha(D_f))$ is acyclic.) By Theorem 9.4, we can find in time $O(m)$ a blocking collection $P_1, \ldots, P_l$ in $\widetilde{D}$. Define:

$$D' := (V, A \cup \alpha(D)^{-1}) \text{ and } D'' := D \leftarrow P_1 \leftarrow \cdots \leftarrow P_l. \tag{9.6}$$

We show $\mu(D'') > \mu(D)$. As D'' is a subgraph of D', we have $\mu(D'') \geq \mu(D') = \mu(D)$, by Theorem 9.5. Suppose that $\mu(D'') = \mu(D')$. Then $\alpha(D'') \subseteq \alpha(D') = \alpha(D)$ (again by Theorem 9.5). Hence, as $\alpha(D'')$ contains an $s-t$ path (of length $\mu(D'')$), $\alpha(D)$ contains an $s-t$ path arc-disjoint from $P_1, \ldots, P_l$. This contradicts the fact that $P_1, \ldots, P_l$ is blocking in $\widetilde{D}$. ∎

This gives us the speedup in finding a maximum packing of $s-t$ paths:

Corollary 9.5b. *Given a digraph $D = (V, A)$ and $s, t \in V$, a maximum number of arc-disjoint $s-t$ paths and a minimum-size $s-t$ cut can be found in time $O(nm)$.*

Proof. Directly from Corollary 9.5a with the iterations (9.5). ∎

9.4. A sometimes better bound

Since $\mu(D_i)$ is at most $|V|$ (as long as it is finite), the number k of iterations is at most $|V|$. But Karzanov [1973a], Tarjan [1974e], and Even and Tarjan [1975] showed that an alternative, often tighter bound on k holds.

To see this, it is important to observe that, for each i, the set of arcs of D_i that are reversed in the final D_k (compared with D_i) forms a maximum number of arc-disjoint $s-t$ paths in D_i.

Theorem 9.6. *If $\mu(D_{i+1}) > \mu(D_i)$ for each $i < k$, then $k \leq 2|A|^{1/2}$. If moreover D is simple, then $k \leq 2|V|^{2/3}$.*

Proof. Let $p := \lfloor |A|^{1/2} \rfloor$. Then each $s-t$ path in D_p has length at least $p+1 \geq |A|^{1/2}$. Hence D_p contains at most $|A|/|A|^{1/2} = |A|^{1/2}$ arc-disjoint $s-t$ paths. Therefore $k - p \leq |A|^{1/2}$, and hence $k \leq 2|A|^{1/2}$.

If D is simple, let $p := \lfloor |V|^{2/3} \rfloor$. Then each $s-t$ path in D_p has length at least $p+1 \geq |V|^{2/3}$. Then D_p contains at most $|V|^{2/3}$ arc-disjoint $s-t$ paths. Indeed, let U_i denote the set of vertices at distance i from s in D_p. Then

$$\sum_{i=0}^{p} (|U_i| + |U_{i+1}|) \leq 2|V|. \tag{9.7}$$

Hence there is an $i \leq p$ with $|U_i|+|U_{i+1}| \leq 2|V|^{1/3}$. This implies $|U_i| \cdot |U_{i+1}| \leq \frac{1}{4}(|U_i| + |U_{i+1}|)^2 \leq |V|^{2/3}$. So D_p contains at most $|V|^{2/3}$ arc-disjoint $s-t$ paths. Therefore $k - p \leq |V|^{2/3}$, and hence $k \leq 2|V|^{2/3}$. ∎

This gives the following time bounds (Karzanov [1973a], Tarjan [1974e], Even and Tarjan [1975]):

Corollary 9.6a. *Given a digraph $D = (V, A)$ and $s, t \in V$, a maximum number of arc-disjoint $s-t$ paths and a minimum-size $s-t$ cut can be found in time $O(m^{3/2})$. If D is simple, the paths and the cut can be found also in time $O(n^{2/3}m)$.*

Proof. Directly from Corollary 9.5a and Theorem 9.6. ∎

(Related work was presented in Ahuja and Orlin [1991].)

9.5. Complexity of the vertex-disjoint case

If we are interested in *vertex*-disjoint paths, the results can be sharpened. Recall that if $D = (V, A)$ is a digraph and $s, t \in V$, then the problem of finding a maximum number of internally vertex-disjoint $s-t$ paths can be reduced to the arc-disjoint case by replacing each vertex $v \neq s, t$ by two vertices v', v'', while each arc with head v is redirected to v' and each arc with tail v is redirected from v''; moreover, an arc (v', v'') is added.

By Corollary 9.6a, this construction directly yields algorithms for vertex-disjoint paths with running time $O(m^{3/2})$ and $O(n^{2/3}m)$. But one can do better. Note that, with this construction, each of the digraphs D_i has the property that each vertex has indegree at most 1 or outdegree at most 1. Under this condition, the bound in Theorem 9.6 can be improved to $2|V|^{1/2}$:

Theorem 9.7. *If each vertex $v \neq s, t$ has indegree or outdegree equal to 1, and if $\mu(D_{i+1}) > \mu(D_i)$ for each $i \leq k$, then $k \leq 2|V|^{1/2}$.*

Proof. Let $p := \lceil |V|^{1/2} \rceil$. Then each $s-t$ path in D_p has length at least $p+1$. Let U_i be the set of vertices at distance i from s in D_i. Then $\sum_{i=1}^{p} |U_i| \leq |V|$. This implies that $|U_i| \leq |V|^{1/2}$ for some i. Hence D_i has at most $|V|^{1/2}$ arc-disjoint $s-t$ paths. So $k + 1 - p \leq |V|^{1/2}$. Hence $k \leq 2|V|^{1/2}$. ∎

This gives, similarly to Corollary 9.6a, another result of Karzanov [1973a], Tarjan [1974e], and Even and Tarjan [1975] (which can be derived also from Theorem 16.4 due to Hopcroft and Karp [1971,1973] and Karzanov [1973b], with the method of Hoffman [1960] given in Section 16.7c):

Corollary 9.7a. *Given a digraph $D = (V, A)$ and $s, t \in V$, a maximum number of internally vertex-disjoint $s - t$ paths and a minimum-size $s - t$ vertex-cut can be found in time $O(n^{1/2}m)$.*

Proof. Directly from Corollary 9.5a and Theorem 9.7. ∎

In fact one can reduce n in this bound to the minimum number $\tau(D)$ of vertices intersecting each arc of D (this bound will be used in deriving bounds for bipartite matching (Theorem 16.5)):

Theorem 9.8. *Given a digraph $D = (V, A)$ and $s, t \in V$, a maximum number of internally vertex-disjoint $s - t$ paths and a minimum-size $s - t$ vertex-cut can be found in time $O(\tau(D)^{1/2}m)$.*

Proof. Similar to Corollary 9.7a, by taking $p := \lfloor \tau(D)^{1/2} \rfloor$ in Theorem 9.7: Finding D_p takes $O(pm)$ time. Let W be a set of vertices intersecting each arc of D, of size $\tau(D)$. In D_p there are at most $2\tau(D)^{1/2}$ internally vertex-disjoint $s-t$ paths, since each $s-t$ path contains at least $p/2$ vertices in W. ∎

9.6. Further results and notes

9.6a. Complexity survey for the disjoint $s - t$ paths problem

For finding arc-disjoint $s - t$ paths we have the following survey of running time bounds ($*$ indicates an asymptotically best bound in the table):

	$O(m^2)$	Ford and Fulkerson [1955,1957b]
$*$	$O(nm)$	Dinits [1970]
$*$	$O(m^{3/2})$	Karzanov [1973a], Tarjan [1974e], Even and Tarjan [1975]
$*$	$O(n^{2/3}m)$	Karzanov [1973a], Tarjan [1974e], Even and Tarjan [1975] *D simple*

≫

continued

*	$O(k^2n)$	Nagamochi and Ibaraki [1992a] finding k arc-disjoint paths
*	$O(kn^{5/3})$	Nagamochi and Ibaraki [1992a] finding k arc-disjoint paths
*	$O(kn^2)$	Nagamochi and Ibaraki [1992a] finding k arc-disjoint paths; D *simple*
*	$O(k^{3/2}n^{3/2})$	Nagamochi and Ibaraki [1992a] finding k arc-disjoint paths; D *simple*

For *undirected* simple graphs, Goldberg and Rao [1997b,1999] gave an $O(n^{3/2}m^{1/2})$ bound, and Karger and Levine [1998] gave $(m+nk^{3/2})$ and $O(nm^{2/3}k^{1/6})$ bounds, where k is the number of paths.

For vertex-disjoint paths:

	$O(nm)$	Grünwald [1938], Ford and Fulkerson [1955, 1957b]
	$O(\sqrt{n}\,m)$	Karzanov [1973a], Tarjan [1974e], Even and Tarjan [1975]
*	$O(\sqrt{n}\,m\log_n(n^2/m))$	Feder and Motwani [1991,1995]
*	$O(k^2n)$	Nagamochi and Ibaraki [1992a] finding k vertex-disjoint paths
*	$O(kn^{3/2})$	Nagamochi and Ibaraki [1992a] finding k vertex-disjoint paths

For edge-disjoint $s-t$ paths in simple undirected *planar* graphs:

	$O(n^2\log n)$	Itai and Shiloach [1979]
	$O(n^2)$	Cheston, Probert, and Saxton [1977]
	$O(n^{3/2}\log n)$	Johnson and Venkatesan [1982]
	$O(n\log^2 n)$	Reif [1983] (minimum-size $s-t$ cut), Hassin and Johnson [1985] (edge-disjoint $s-t$ paths)
	$O(n\log n\log^* n)$	Frederickson [1983b]
	$O(n\log n)$	Frederickson [1987b]
*	$O(n)$	Weihe [1994a,1997a]

For arc-disjoint $s-t$ paths in simple *directed* planar graphs:

$O(n^{3/2}\log n)$	Johnson and Venkatesan [1982]
$O(n^{4/3}\log^2 n)$	Klein, Rao, Rauch, and Subramanian [1994], Henzinger, Klein, Rao, and Subramanian [1997]

≫

continued

	$O(n \log n)$	Weihe [1994b,1997b]
*	$O(n)$	Brandes and Wagner [1997]

For vertex-disjoint $s-t$ paths in *undirected* planar graphs:

	$O(n \log n)$	Suzuki, Akama, and Nishizeki [1990]
*	$O(n)$	Ripphausen-Lipa, Wagner, and Weihe [1993b,1997]

Orlova and Dorfman [1972] and Hadlock [1975] showed, with matching theory, that in planar undirected graphs also a *maximum*-size cut can be found in polynomial-time (Barahona [1990] gave an $O(n^{3/2} \log n)$ time bound) — see Section 29.1. Karp [1972b] showed that in general finding a maximum-size cut is NP-complete — see Section 75.1a.

9.6b. Partially disjoint paths

For any digraph $D = (V, A)$ and any $B \subseteq A$, call two paths *disjoint on* B if they have no common arc in B. One may derive from Menger's theorem a more general min-max relation for such partially disjoint paths:

Theorem 9.9. *Let $D = (V, A)$ be a digraph, $s, t \in V$, and $B \subseteq A$. Then the maximum number of $s-t$ paths such that any two are disjoint on B is equal to the minimum size of an $s-t$ cut contained in B.*

Proof. If there is no $s-t$ cut contained in B, then clearly the maximum is infinite. If B contains any $s-t$ cut, let k be its minimum size. Replace any arc $a \in A \setminus B$ by $|A|$ parallel arcs. Then by Menger's theorem there exist k arc-disjoint $s-t$ paths in the extended graph. This gives k $s-t$ paths in the original graph that are disjoint on B. ∎

The construction given in this proof can also be used algorithmically, but making $|A|$ parallel arcs takes $\Omega(m^2)$ time. However, one can prove:

Theorem 9.10. *Given a digraph $D = (V, A)$, $s, t \in V$, and $B \subseteq A$, a maximum number of $s-t$ paths such that any two are disjoint on B can be found in time $O(nm)$.*

Proof. The theorem follows from Corollary 10.11a below. ∎

9.6c. Exchange properties of disjoint paths

Disjoint paths have a number of exchange properties that imply a certain matroidal structure (cf. Section 39.4).

Let $D = (V, A)$ be a directed graph and let $X, Y \subseteq V$. Call X *linked to* Y if $|X| = |Y|$ and D has $|X|$ vertex-disjoint $X-Y$ paths. Note that the following theorem follows directly from the algorithm for finding a maximum set of disjoint paths (since any vertex in $S \cup T$, once covered by the disjoint paths, remains covered during the further iterations):

Theorem 9.11. *Let $D = (V, A)$ be a digraph, let $S, T \subseteq V$, and suppose that $X \subseteq S$ and $Y \subseteq T$ are linked. Then there exists a maximum number of vertex-disjoint $S-T$ paths covering $X \cup Y$.*

Proof. Directly from the algorithm. ∎

The following result, due to Perfect [1968], says that for two distinct maximum packings $\mathcal{P}, \mathcal{Q}$ of $S-T$ paths there exists a maximum packing of $S-T$ paths whose starting vertices are equal to those of $\mathcal{P}$ and whose end vertices are equal to those of $\mathcal{Q}$.

Theorem 9.12. *Let $D = (V, A)$ be a digraph and $S, T \subseteq V$. Let k be the maximum number of disjoint $S-T$ paths. Let $X \subseteq S$ be linked to $Y \subseteq T$, and let $X' \subseteq S$ be linked to $Y' \subseteq T$, with $|X| = |X'| = k$. Then X is linked to Y'.*

Proof. Let C be a minimum-size vertex set intersecting each $S-T$ path. So by Menger's theorem, $|C| = |X| = |Y| = |X'| = |Y'| = k$. Let $P_1', \ldots, P_k'$ be vertex-disjoint $X-Y$ paths. Similarly, let $P_1'', \ldots, P_k''$ be vertex-disjoint $X'-Y'$ paths. We may assume that, for each i, P_i' and P_i'' have a vertex in C in common. Let P_i be the path obtained by traversing P_i' until it reaches C, after which it traverses P_i''. Then $P_1, \ldots, P_k$ are vertex-disjoint $X-Y'$ paths. ∎

The previous two theorems imply:

Corollary 9.12a. *Let $D = (V, A)$ be a digraph, let X' be linked to Y', and let X'' be linked to Y''. Then there exist X and Y with $X' \subseteq X \subseteq X' \cup X''$ and $Y'' \subseteq Y \subseteq Y' \cup Y''$ such that X is linked to Y.*

Proof. Directly from Theorems 9.11 and 9.12. ∎

Other proofs of this corollary were given by Pym [1969b,1969c], Brualdi and Pym [1971], and McDiarmid [1975b].

9.6d. Further notes

Lovász, Neumann-Lara, and Plummer [1978] proved the following on the maximum number of disjoint paths of bounded length. Let $G = (V, E)$ be an undirected graph, let s and t be two distinct and nonadjacent vertices, and let $k \geq 2$. Then the minimum number of vertices ($\neq s, t$) intersecting all $s-t$ paths of length at most k is at most $\lfloor \frac{1}{2}k \rfloor$ times the maximum number of internally vertex-disjoint $s-t$ paths each of length at most k. (A counterexample to a conjecture on this raised by Lovász, Neumann-Lara, and Plummer [1978] was given by Boyles and Exoo [1982]. Related results are given by Galil and Yu [1995].)

On the other hand, when taking lower bounds on the path lengths, Montejano and Neumann-Lara [1984] showed that, in a directed graph, the minimum number of vertices ($\neq s, t$) intersecting all $s-t$ paths of length at least k is at most $3k-5$ times the maximum number of internally vertex-disjoint such paths. For $k = 3$, the factor was improved to 3 by Hager [1986] and to 2 by Mader [1989].

Egawa, Kaneko, and Matsumoto [1991] gave a version of Menger's theorem in which vertex-disjoint and edge-disjoint are mixed: Let $G = (V, E)$ be an undirected graph, let $s, t \in V$, and $k, l \in \mathbb{Z}_+$. Then G contains l disjoint edge sets, each containing k vertex-disjoint $s-t$ paths if and only if for each $U \subseteq V \setminus \{s, t\}$ there exist $l(k - |U|)$ edge-disjoint $s-t$ paths in $G - U$. Similarly, for directed graphs. The proof is by reduction to Menger's theorem using integer flow theory.

Bienstock and Diaz [1993] showed that the problem of finding a minimum-weight subset of edges intersecting all $s-t$ cuts of size at most k is polynomial-time solvable if k is fixed, while it is NP-complete if k is not fixed.

Motwani [1989] investigated the expected running time of Dinits' disjoint paths algorithm.

Extensions of Menger's theorem to the infinite case were given by P. Erdős (cf. Kőnig [1932]), Grünwald [1938] (= T. Gallai), Dirac [1960,1963,1973], Halin [1964], McDiarmid [1975b], Podewski and Steffens [1977], Aharoni [1983a,1987], and Polat [1991].

Halin [1964], Lovász [1970b], Escalante [1972], and Polat [1976] made further studies of (the lattice of) $s-t$ cuts.

9.6e. Historical notes on Menger's theorem

The topologist Karl Menger published his theorem in an article called *Zur allgemeinen Kurventheorie* (On the general theory of curves) (Menger [1927]) in the following form:

> Satz β. *Ist K ein kompakter regulär-eindimensionaler Raum, welcher zwischen den beiden endlichen Mengen P und Q n-punktig zusammenhängend ist, dann enthält K n paarweise fremde Bögen, von denen jeder einen Punkt von P und einen Punkt von Q verbindet.*[5]

It can be formulated equivalently in terms of graphs as: Let $G = (V, E)$ be an undirected graph and let $P, Q \subseteq V$. Then the maximum number of disjoint $P-Q$ paths is equal to the minimum size of a set W of vertices such that each $P-Q$ path intersects W.

The result became known as the *n-chain theorem*. Menger's interest in this question arose from his research on what he called 'curves': a *curve* is a connected compact topological space X with the property that for each $x \in X$ and each neighbourhood N of x there exists a neighbourhood $N' \subseteq N$ of x with $\mathrm{bd}(N')$ totally disconnected. (Here bd stands for boundary; a space is *totally disconnected* if each point forms an open set.)

The curve is called *regular* if for each $x \in X$ and each neighbourhood N of x there exists a neighbourhood $N' \subseteq N$ of x with $|\mathrm{bd}(N')|$ finite. The *order* of a point $x \in X$ is equal to the minimum natural number n such that for each neighbourhood N of x there exists a neighbourhood $N' \subseteq N$ of x satisfying $|\mathrm{bd}(N')| \leq n$.

According to Menger:

[5] Theorem β. *If K is a compact regularly one-dimensional space which is n-point connected between the two finite sets P and Q, then K contains n pairwise disjoint curves, each of which connects a point in P and a point in Q.*

> Eines der wichtigsten Probleme der Kurventheorie ist die Frage nach den Beziehungen zwischen der Ordnungszahl eines Punktes der regulären Kurve K und der Anzahl der im betreffenden Punkt zusammenstossenden und sonst fremden Teilbögen von K.[6]

In fact, Menger used 'Satz β' to show that if a point in a regular curve K has order n, then there exists a topological *n-leg* with p as top; that is, K contains n arcs $P_1, \ldots, P_n$ such that $P_i \cap P_j = \{p\}$ for all i, j with $i \neq j$.

The proof idea is as follows. There exists a series $N_1 \supset N_2 \supset \cdots$ of open neighbourhoods of p such that $N_1 \cap N_2 \cap \cdots = \{p\}$ and $|\mathrm{bd}(N_i)| = n$ for all $i = 1, 2, \ldots$ and such that

(9.8) $$|\mathrm{bd}(N)| \geq n \text{ for each neighbourhood } N \subseteq N_1.$$

This follows quite directly from the definition of order.

Now Menger showed that we may assume that the space $G_i := \overline{N_i} \setminus N_{i+1}$ is a (topological) graph. For each i, let $Q_i := \mathrm{bd}(N_i)$. Then (9.8) gives with Menger's theorem that there exist n disjoint paths $P_{i,1}, \ldots, P_{i,n}$ in G such that each $P_{i,j}$ runs from Q_i to Q_{i+1}. Properly connecting these paths for $i = 1, 2, \ldots$ we obtain n arcs forming the required n-leg.

It was however noticed by Kőnig [1932] that Menger's proof of 'Satz β' is incomplete. Menger applied induction on $|E|$, where E is the edge set of the graph G. Menger first claimed that one easily shows that $|E| \geq n$, and that if $|E| = n$, then G consists of n disjoint edges connecting P and Q. He stated that if $|E| > n$, then there exists a vertex $s \notin P \cup Q$, or in his words (where the 'Grad' denotes $|E|$):

> Wir nehmen also an, der irreduzibel n-punktig zusammenhängende Raum K' besitze den Grad $g(> n)$. Offenbar enthält dann K' ein punktförmiges Stück s, welches in der Menge $P + Q$ nicht enthalten ist.[7]

Indeed, as Menger showed, if such a vertex s exists one is done: If s is contained in no set W intersecting each $P - Q$ path with $|W| = n$, then we can delete s and the edges incident with s without decreasing the minimum in the theorem. If s is contained in some set W intersecting each $P - Q$ path such that $|W| = n$, then we can split G into two subgraphs G_1 and G_2 that intersect in W in such a way that $P \subseteq G_1$ and $Q \subseteq G_2$. By the induction hypothesis, there exist n disjoint $P - W$ paths in G_1 and n disjoint $W - Q$ paths in G_2. By pairwise sticking these paths together at W we obtain paths as required.

However, such a vertex s need not exist. It might be that V is the disjoint union of P and Q in such a way that each edge connects P and Q, and that there are more than n edges. In that case, G is a bipartite graph, with colour classes P and Q, and what should be shown is that G contains a matching of size n. This is a nontrivial basis of the proof.

At the meeting of 26 March 1931 of the Eötvös Loránd Matematikai és Fizikai Társulat (Loránd Eötvös Mathematical and Physical Society) in Budapest, Kőnig [1931] presented a new result that formed the missing basis for Menger's theorem:

---

[6] One of the most important problems of the theory of curves is the question of the relations between the order of a point of a regular curve K and the number of subarcs of K meeting in that point and disjoint elsewhere.

[7] Thus we assume that the irreducibly n-point-connected space K' has degree $g(> n)$. Obviously, in that case K' contains a point-shaped piece s, that is not contained in the set $P + Q$.

> Páros körüljárású graphban az éleket kimerítő szögpontok minimális száma megegyezik a páronként közös végpontot nem tartalmazó élek maximális számával.[8]

In other words, in a bipartite graph $G = (V, E)$, the maximum size of a matching is equal to the minimum number of vertices needed to cover all edges, which is Kőnig's matching theorem — see Theorem 16.2.

Kőnig did not mention in his 1931 paper that this result provided the missing element in Menger's proof, although he finishes with:

> Megemlítjük végül, hogy eredményeink szorosan összefüggnek FROBENIUSnak determinánsokra és MENGERnek graphokra vonatkozó némely vizsgálatával. E kapcsolatokra másutt fogunk kiterjeszkedni.[9]

'Elsewhere' is Kőnig [1932], in which paper he gave a full proof of Menger's theorem. The hole in Menger's original proof is discussed in a footnote:

> Der Beweis von MENGER enthält eine Lücke, da es vorausgesetzt wird (S. 102, Zeile 3–4) daß „K' ein punktförmiges Stück s enthält, welches in der Menge $P+Q$ nicht enthalten ist", während es recht wohl möglich ist, daß —mit der hier gewählten Bezeichnungsweise ausgedrückt—jeder Knotenpunkt von G zu $H_1 + H_2$ gehört. Dieser—keineswegs einfacher—Fall wurde in unserer Darstellung durch den Beweis des Satzes 13 erledigt. Die weiteren—hier folgenden—Überlegungen, die uns zum Mengerschen Satz führen werden, stimmen im Wesentlichen mit dem—sehr kurz gefaßten—Beweis von MENGER überein. In Anbetracht der Allgemeinheit und Wichtigkeit des Mengerschen Satzes wird im Folgenden auch dieser Teil ganz ausführlich und den Forderungen der *rein-kombinatorischen* Graphentheorie entsprechend dargestellt.
>
> [Zusatz bei der Korrektur, 10.V.1933] Herr MENGER hat die Freundlichkeit gehabt—nachdem ich ihm die Korrektur meiner vorliegenden Arbeit zugeschickt habe—mir mitzuteilen, daß ihm die oben beanstandete Lücke seines Beweises schon bekannt war, daß jedoch sein vor Kurzem erschienenes Buch *Kurventheorie* (Leipzig, 1932) einen vollkommen lückenlosen und rein kombinatorischen Beweis des Mengerschen Satzes (des "n-Kettensatzes") enthält. Mir blieb dieser Beweis bis jetzt unbekannt.[10]

---

[8] In an even circuit graph, the minimal number of vertices that exhaust the edges agrees with the maximal number of edges that pairwise do not contain any common end point.

[9] We finally mention that our results are closely connected to some investigations of FROBENIUS on determinants and of MENGER on graphs. We will enlarge on these connections elsewhere.

[10] The proof of MENGER contains a hole, as it is assumed (page 102, line 3–4) that 'K' contains a point-shaped piece s that is not contained in the set $P+Q$', while it is quite well possible that—expressed in the notation chosen here—every node of G belongs to $H_1 + H_2$. This—by no means simple—case is settled in our presentation by the proof of Theorem 13. The further arguments following here that will lead us to Menger's theorem, agree essentially with the—very briefly couched—proof of MENGER. In view of the generality and the importance of Menger's theorem, also this part is exhibited in the following very extensively and conforming to the progress of the *purely combinatorial* graph theory.

[Added in proof, 10 May 1933] Mr. MENGER has had the kindness—after I have sent him the galley proofs of my present work—to inform me that the hole in his proof objected above, was known to him already, but that his, recently appeared, book *Kurventheorie* (Leipzig, 1932) contains a completely holeless and purely combinatorial proof of the Menger theorem (the 'n-chain theorem'). As yet, this proof remained unknown to me.

The book *Kurventheorie* (Curve Theory) mentioned is Menger [1932b], which contains a complete proof of Menger's theorem. Menger did not refer to any hole in his original proof, but remarked:

> Über den n-Kettensatz für Graphen und die im vorangehenden zum Beweise verwendete Methode vgl. Menger (Fund. Math. *10*, 1927, S. 101 f.). Die obige detaillierte Ausarbeitung und Darstellung stammt von Nöbeling.[11]

In his book *Theorie der endlichen und unendlichen Graphen* (Theory of finite and infinite graphs), Kőnig [1936] called his theorem *ein wichtiger Satz* (an important theorem), and he emphasized the chronological order of the proofs of Menger's theorem and of Kőnig's theorem (which is implied by Menger's theorem):

> Ich habe diesen Satz 1931 ausgesprochen und bewiesen, s. Kőnig [9 und 11]. 1932 erschien dann der erste lückenlose Beweis des Mengerschen Graphensatzes, von dem in §4 die Rede sein wird und welcher als eine Verallgemeinerung dieses Satzes 13 (falls dieser *nur für endliche* Graphen formuliert wird) angesehen werden kann.[12]

([9 und 11] are Kőnig [1931] and Kőnig [1932].)

In his reminiscences on the origin of the n-arc theorem, Menger [1981] wrote:

> In the spring of 1930, I came through Budapest and met there a galaxy of Hungarian mathematicians. In particular, I enjoyed making the acquaintance of Dénes Kőnig, for I greatly admired the work on set theory of his father, the late Julius Kőnig—to this day one of the most significant contributions to the continuum problem—and I had read with interest some of Dénes' papers. Kőnig told me that he was about to finish a book that would include all that was known about graphs. I assured him that such a book would fill a great need; and I brought up my n-Arc Theorem which, having been published as a lemma in a curve-theoretical paper, had not yet come to his attention. Kőnig was greatly interested, but did not believe that the theorem was correct. "This evening," he said to me in parting, "I won't go to sleep before having constructed a counterexample." When we met again the next day he greeted me with the words, "A sleepless night!" and asked me to sketch my proof for him. He then said that he would add to his book a final section devoted to my theorem. This he did; and it is largely thanks to Kőnig's valuable book that the n-Arc Theorem has become widely known among graph theorists.

Related work

In a paper presented 7 May 1927 to the American Mathematical Society, Rutt [1927, 1929] gave the following variant of Menger's theorem, suggested by J.R. Kline. Let $G = (V, E)$ be a planar graph and let $s, t \in V$. Then the maximum number of internally vertex-disjoint $s-t$ paths is equal to the minimum number of vertices in $V \setminus \{s, t\}$ intersecting each $s-t$ path.

[11] On the n-chain theorem for graphs and the method used in the foregoing for the proof, compare Menger (Fund. Math. *10*, 1927, p. 101 ff.). The detailed elaboration and explanation above originates from Nöbeling.

[12] I have enunciated and proved this theorem in 1931, see Kőnig [9 and 11]. Next, in 1932, the first holeless proof of the Menger theorem appeared, of which will be spoken in §4 and which can be considered as a generalization of this Theorem 13 (in case this is formulated *only for finite* graphs).

In fact, the theorem follows quite easily from Menger's version of his theorem by deleting s and t and taking for P and Q the sets of neighbours of s and t respectively. (Rutt referred to Menger and gave an independent proof of the theorem.)

This construction was also observed by Knaster [1930] who showed that Menger's theorem would follow from Rutt's theorem for general (not necessarily planar) graphs. A similar theorem was published by Nöbeling [1932], using Menger's result.

A result implied by Menger's theorem was presented by Whitney [1932a] on 28 February 1931 to the American Mathematical Society: a graph is n-connected if and only if any two vertices are connected by n internally disjoint paths. While referring to the papers of Menger and Rutt, Whitney gave a direct proof. Kőnig [1932] remarked on Whitney's theorem:

> Das interessante Hauptresultat einer Abhandlung von WHITNEY [10], nämlich sein Theorem 7, folgt unmittelbar aus diesem Mengerschen Satz, jedoch, wie es scheint, nicht umgekehrt.[13]

In the 1930s, other proofs of Menger's theorem were given by Hajós [1934] and Grünwald [1938] (= T. Gallai) — the latter paper gives an essentially algorithmic proof based on augmenting paths, and it observes, in a footnote, that the theorem also holds for directed graphs:

> Die ganze Betrachtung lässt sich auch bei orientierten Graphen durchführen und liefert dann eine Verallgemeinerung des Mengerschen Satzes.[14]

The arc-disjoint version of Menger's theorem seems to be first shown by Ford and Fulkerson [1954,1956b] and Kotzig [1956] for undirected graphs and by Dantzig and Fulkerson [1955,1956] for directed graphs.

In his dissertation for the degree of Academical Doctor, Kotzig [1956] defined, for any undirected graph G and vertices u, v of G, $\sigma_G(u, v)$ to be the minimum size of a $u - v$ cut. Then he states:

> <u>Veta 35</u>. Nech G je l'ubovol'ný graf obsahujúci uzly $u \neq v$, o ktorých platí $\sigma_G(u, v) = k > 0$, potom existuje systém ciest $\{C_1, C_2, \ldots, C_k\}$ taký že každa cesta spojuje uzly u, v a žiadne dve rôzne cesty systému nemajú spoločnej hrany. Takýto systém ciest v G existuje len vtedy, keď je $\sigma_G(u, v) \geq k$.[15]

In Theorems 33 and 34 of the dissertation, methods are developed for the proof of Theorem 35. The method is to consider a minimal graph satisfying the cut condition, and next to orient it so as to make a directed graph in which each vertex w (except u and v) has indegree = outdegree, while u has outdegree k and indegree 0. This then yields the required paths.

Although the dissertation has several references to Kőnig's book, which contains the undirected vertex-disjoint version of Menger's theorem, Kotzig did not link his

[13] The interesting main result of an article of WHITNEY [10], namely his Theorem 7, follows immediately from this theorem of Menger, however, as it seems, not conversely.

[14] The whole argument lets itself carry out also for oriented graphs and then yields a generalization of Menger's theorem.

[15] <u>Theorem 35</u>. Let G be an arbitrary graph containing vertices $u \neq v$ for which $\sigma_G(u, v) = k > 0$, then there exists a system of paths $\{C_1, C_2, \ldots, C_k\}$ such that each path connects vertices u, v and no two distinct paths have an edge in common. Such a system of paths in G exists only if $\sigma_G(u, v) \geq k$.

result to that of Menger. (Kotzig [1961a] gave a proof of the directed arc-disjoint version of Menger's theorem, without reference to Menger.)

We refer to the historical notes on maximum flows in Section 10.8e for further notes on the work of Dantzig, Ford, and Fulkerson on Menger's theorem.

Chapter 10

Maximum flow

An $s-t$ flow is defined as a nonnegative real-valued function on the arcs of a digraph satisfying the 'flow conservation law' at each vertex $\neq s,t$. In this chapter we consider the problem of finding a maximum-value flow subject to a given capacity function. Basic results are Ford and Fulkerson's max-flow min-cut theorem and their augmenting path algorithm to find a maximum flow.
Each $s-t$ flow is a nonnegative linear combination of incidence vectors of $s-t$ paths and of directed circuits. Moreover, an integer flow is an integer such combination. This makes flows tightly connected to disjoint paths. Thus, maximum integer flow corresponds to a capacitated version of a maximum packing of disjoint paths, and the max-flow min-cut theorem is equivalent to Menger's theorem on disjoint paths.
Distinguishing characteristic of flow is however that it is not described by a combination of paths but by a function on the arcs. This promotes the algorithmic tractability.
In this chapter, graphs can be assumed to be simple.

10.1. Flows: concepts

Let $D=(V,A)$ be a digraph and let $s,t\in V$. A function $f:A\to\mathbb{R}$ is called a *flow from s to t*, or an *$s-t$ flow*, if:

(10.1) (i) $f(a)\geq 0$ for each $a\in A$,
(ii) $f(\delta^{\text{out}}(v))=f(\delta^{\text{in}}(v))$ for each $v\in V\setminus\{s,t\}$.

Condition (10.1)(ii) is called the *flow conservation law*: the amount of flow entering a vertex $v\neq s,t$ should be equal to the amount of flow leaving v.

The *value* of an $s-t$ flow f is, by definition:

(10.2) $\text{value}(f):=f(\delta^{\text{out}}(s))-f(\delta^{\text{in}}(s)).$

So the value is the net amount of flow leaving s. This is equal to the net amount of flow entering t (this follows from (10.5) below).

Let $c:A\to\mathbb{R}_+$ be a *capacity* function. We say that a flow f is *under c* (or *subject to c*) if

(10.3) $f(a)\leq c(a)$ for each $a\in A$.

A *maximum* $s-t$ *flow*, or just a *maximum flow*, is an $s-t$ flow under c, of maximum value. The *maximum flow problem* is to find a maximum flow.

By compactness and continuity, a maximum flow exists. It will follow from the results in this chapter (in particular, Theorem 10.4), that if the capacities are rational, then there exists a rational-valued maximum flow.

It will be convenient to make an observation on general functions $f : A \to \mathbb{R}$. For any $f : A \to \mathbb{R}$, the *excess function* is the function $\text{excess}_f : \mathcal{P}(V) \to \mathbb{R}$ defined by

(10.4) $$\text{excess}_f(U) := f(\delta^{\text{in}}(U)) - f(\delta^{\text{out}}(U))$$

for $U \subseteq V$. Set $\text{excess}_f(v) := \text{excess}_f(\{v\})$ for $v \in V$. Then:

Theorem 10.1. *Let $D = (V, A)$ be a digraph, let $f : A \to \mathbb{R}$, and let $U \subseteq V$. Then:*

(10.5) $$\text{excess}_f(U) = \sum_{v \in U} \text{excess}_f(v).$$

Proof. This follows directly by counting, for each $a \in A$, the multiplicity of $f(a)$ at both sides of (10.5). ∎

To formulate a min-max relation, define the *capacity* of a cut $\delta^{\text{out}}(U)$ by $c(\delta^{\text{out}}(U))$. Then:

Theorem 10.2. *Let $D = (V, A)$ be a digraph, $s, t \in V$, and $c : A \to \mathbb{R}_+$. Then*

(10.6) $$\text{value}(f) \leq c(\delta^{\text{out}}(U)),$$

for each $s-t$ flow $f \leq c$ and each $s-t$ cut $\delta^{\text{out}}(U)$. Equality holds in (10.6) *if and only if $f(a) = c(a)$ for each $a \in \delta^{\text{out}}(U)$ and $f(a) = 0$ for each $a \in \delta^{\text{in}}(U)$.*

Proof. Using (10.5) we have

(10.7) $$\text{value}(f) = -\text{excess}_f(s) = -\text{excess}_f(U) = f(\delta^{\text{out}}(U)) - f(\delta^{\text{in}}(U)) \leq c(\delta^{\text{out}}(U)),$$

with equality if and only if $f(\delta^{\text{out}}(U)) = c(\delta^{\text{out}}(U))$ and $f(\delta^{\text{in}}(U))= 0$. ∎

Finally, we consider a concept that turns out to be important in studying flows. Let $D = (V, A)$ be a digraph. For each $a = (u, v) \in A$, let $a^{-1} := (v, u)$. Define

(10.8) $$A^{-1} := \{a^{-1} \mid a \in A\}.$$

Fix a lower bound function $d : A \to \mathbb{R}$ and an upper bound function $c : A \to \mathbb{R}$. Then for any $f : A \to \mathbb{R}$ satisfying $d \leq f \leq c$ we define

(10.9) $$A_f := \{a \mid a \in A, f(a) < c(a)\} \cup \{a^{-1} \mid a \in A, f(a) > d(a)\}.$$

Clearly, A_f depends not only on f, but also on D, d, and c, but in the applications below D, d, and c are fixed, while f is variable. The digraph

$$D_f = (V, A_f) \tag{10.10}$$

is called the *residual graph* of f. So D_f is a subgraph of the directed graph $(V, A \cup A^{-1})$. As we shall see, the residual graph is very useful in studying flows and circulations, both theoretically and algorithmically.

In the context of flows we take $d = \mathbf{0}$. We observe:

Corollary 10.2a. *Let f be an $s-t$ flow in D with $f \leq c$. Suppose that D_f has no $s-t$ path. Define U as the set of vertices reachable in D_f from s. Then* $\text{value}(f) = c(\delta_A^{\text{out}}(U))$. *In particular, f has maximum value.*

Proof. We apply Theorem 10.2. For each $a \in \delta_A^{\text{out}}(U)$, one has $a \notin A_f$, and hence $f(a) = c(a)$. Similarly, for each $a \in \delta_A^{\text{in}}(U)$ one has $a^{-1} \notin A_f$, and hence $f(a) = 0$. So $\text{value}(f) = c(\delta_A^{\text{out}}(U))$ and f has maximum value by Theorem 10.2. ∎

Any directed path P in D_f gives an undirected path in $D = (V, A)$. Define $\chi^P \in \mathbb{R}^A$ by:

$$\chi^P(a) := \begin{cases} 1 & \text{if } P \text{ traverses } a, \\ -1 & \text{if } P \text{ traverses } a^{-1}, \\ 0 & \text{if } P \text{ traverses neither } a \text{ nor } a^{-1}, \end{cases} \tag{10.11}$$

for $a \in A$.

10.2. The max-flow min-cut theorem

The following theorem was proved by Ford and Fulkerson [1954,1956b] for the undirected case and by Dantzig and Fulkerson [1955,1956] for the directed case. (According to Robacker [1955a], the max-flow min-cut theorem was conjectured first by D.R. Fulkerson.)

Theorem 10.3 (max-flow min-cut theorem)**.** *Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let $c : A \to \mathbb{R}_+$. Then the maximum value of an $s-t$ flow subject to c is equal to the minimum capacity of an $s-t$ cut.*

Proof. Let f be an $s-t$ flow subject to c, of maximum value. By Theorem 10.2, it suffices to show that there is an $s-t$ cut $\delta^{\text{out}}(U)$ with capacity equal to $\text{value}(f)$.

Consider the residual graph D_f (for lower bound $d := \mathbf{0}$). Suppose that it contains an $s-t$ path P. Then $f' := f + \varepsilon\chi^P$ is again an $s-t$ flow subject to c, for $\varepsilon > 0$ small enough, with $\text{value}(f') = \text{value}(f) + \varepsilon$. This contradicts the maximality of $\text{value}(f)$.

So D_f contains no $s-t$ path. Let U be the set of vertices reachable in D_f from s. Then value$(f) = c(\delta^{\text{out}}(U))$ by Corollary 10.2a. ∎

This 'constructive' proof method is implied by the algorithm of Ford and Fulkerson [1955,1957b], to be discussed below.

Moreover, one has (Dantzig and Fulkerson [1955,1956])[16]:

Corollary 10.3a (integrity theorem)**.** *If c is integer, there exists an integer maximum flow.*

Proof. Directly from the proof of the max-flow min-cut theorem, where we can take $\varepsilon = 1$. ∎

10.3. Paths and flows

The following observation gives an important link between flows at one side and paths at the other side.

Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let $f : A \to \mathbb{R}_+$ be an $s-t$ flow. Then f is a nonnegative linear combination of at most $|A|$ vectors χ^P, where P is a directed $s-t$ path or a directed circuit. If f is integer, we can take the linear combination integer-scalared.

Conversely, if $P_1, \ldots, P_k$ are $s-t$ paths in D, then $f := \chi^{AP_1} + \cdots + \chi^{AP_k}$ is an integer $s-t$ flow of value k.

With this observation, Corollary 10.3a implies the arc-disjoint version of Menger's theorem (Corollary 9.1b). Conversely, Corollary 10.3a (the integrity theorem) can be derived from the arc-disjoint version of Menger's theorem by replacing each arc a by $c(a)$ parallel arcs.

10.4. Finding a maximum flow

The proof idea of the max-flow min-cut theorem can also be used algorithmically to find a maximum $s-t$ flow, as was shown by Ford and Fulkerson [1955,1957b]. Let $D = (V, A)$ be a digraph and $s, t \in V$ and let $c : A \to \mathbb{Q}_+$ be a 'capacity' function.

Initially set $f := \mathbf{0}$. Next apply the following *flow-augmenting algorithm* iteratively:

(10.12) let P be a directed $s-t$ path in D_f and reset $f := f + \varepsilon\chi^P$, where ε is as large as possible so as to maintain $\mathbf{0} \leq f \leq c$.

If no such path exists, the flow f is maximum, by Corollary 10.2a.

The path P is called a *flow-augmenting path* or an *f-augmenting path*, or just an *augmenting path*.

[16] The name 'integrity theorem' was used by Ford and Fulkerson [1962].

As for termination, we have:

Theorem 10.4. *If all capacities $c(a)$ are rational, the algorithm terminates.*

Proof. If all capacities are rational, there exists a natural number K such that $Kc(a)$ is an integer for each $a \in A$. (We can take for K the l.c.m. of the denominators of the $c(a)$.)

Then in the flow-augmenting iterations, every $f_i(a)$ and every ε is a multiple of $1/K$. So at each iteration, the flow value increases by at least $1/K$. Since the flow value cannot exceed $c(\delta^{\text{out}}(\{s\}))$, there are only finitely many iterations. ∎

If we delete the rationality condition, this theorem is not maintained — see Section 10.4a. On the other hand, in Section 10.5 we shall see that if we always choose a *shortest possible* flow-augmenting path, then the algorithm terminates in a polynomially bounded number of iterations, regardless whether the capacities are rational or not.

10.4a. Nontermination for irrational capacities

Ford and Fulkerson [1962] showed that Theorem 10.4 is not maintained if we allow arbitrary real-valued capacities. The example is as follows.

Let $D = (V, A)$ be the complete directed graph on 8 vertices, with $s, t \in V$. Let $A_0 = \{a_1, a_2, a_3\}$ consist of three disjoint arcs of D, each disjoint from s and t. Let r be the positive root of $r^2 + r - 1 = 0$; that is, $r = (-1 + \sqrt{5})/2 < 1$. Define a capacity function c on A by

(10.13) $$c(a_1) := 1,\ c(a_2) := 1,\ c(a_3) := r,$$

and $c(a)$ at least

(10.14) $$q := \frac{1}{1-r} = 1 + r + r^2 + \cdots$$

for each $a \in A \setminus A_0$. Apply the flow-augmenting algorithm iteratively as follows.

In step 0, choose, as flow-augmenting path, the $s-t$ path of length 3 traversing a_1. After this step, the flow f satisfies, for $k = 1$:

(10.15)
(i) f has value $1 + r + r^2 + \cdots + r^{k-1}$,
(ii) $\{c(a) - f(a) \mid a \in A_0\} = \{0, r^{k-1}, r^k\}$,
(iii) $f(a) \leq 1 + r + r^2 + \cdots + r^{k-1}$ for each $a \in A$.

We describe the further steps. In each step k, for $k \geq 1$, the input flow f satisfies (10.15). Choose a flow-augmenting path P in D_f that contains the arc $a \in A_0$ satisfying $c(a) - f(a) = 0$ in backward direction, and the other two arcs in A_0 in forward direction; all other arcs of P are arcs of D in forward direction. Since $r^k < r^{k-1}$, and since $(1 + r + \cdots + r^{k-1}) + r^k < q$, the flow augmentation increases the flow value by r^k. Since $r^{k-1} - r^k = r^{k+1}$, the new flow satisfies (10.15) with k replaced by $k + 1$.

We can keep iterating this, making the flow value converge to $1+r+r^2+r^3+\cdots = q$. So the algorithm does not terminate, and the flow value does not converge to the optimum value, since, trivially, the maximum flow value is more than q.

(Zwick [1995] gave the smallest directed graph (with 6 vertices and 8 arcs) for which the algorithm (with irrational capacities) need not terminate.)

10.5. A strongly polynomial bound on the number of iterations

We saw in Theorem 10.4 that the number of iterations in the maximum flow algorithm is finite, if all capacities are rational. But if we choose as our flow-augmenting path P in the auxiliary graph D_f an *arbitrary* $s-t$ path, the number of iterations yet can get quite large. For instance, in the graph in Figure 10.1 the number of iterations, at an unfavourable choice of paths, can become $2 \cdot 10^k$, so exponential in the size of the input data (which is $O(k)$).

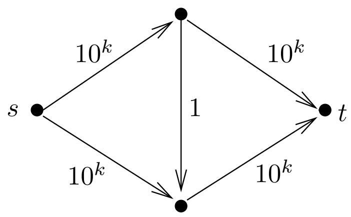


Figure 10.1

However, if we choose always a *shortest* $s-t$ path in D_f as our flow-augmenting path P (that is, with a minimum number of arcs), then the number of iterations is at most $|V| \cdot |A|$ (also if capacities are irrational). This was shown by Dinits [1970] and Edmonds and Karp [1972]. (The latter remark that this refinement 'is so simple that it is likely to be incorporated innocently into a computer implementation.')

To see this bound on the number of iterations, let again, for any digraph $D = (V, A)$ and $s, t \in V$, $\mu(D)$ denote the minimum length of an $s-t$ path. Moreover, let $\alpha(D)$ denote the set of arcs contained in at least one shortest $s-t$ path. Recall that by Theorem 9.5:

(10.16) for $D' := (V, A \cup \alpha(D)^{-1})$, one has $\mu(D') = \mu(D)$ and $\alpha(D') = \alpha(D)$.

This implies the result of Dinits [1970] and Edmonds and Karp [1972]:

Theorem 10.5. *If we choose in each iteration a shortest $s-t$ path in D_f as flow-augmenting path, the number of iterations is at most $|V| \cdot |A|$.*

Proof. If we augment flow f along a shortest $s-t$ path P in D_f, obtaining flow f', then $D_{f'}$ is a subgraph of $D' := (V, A_f \cup \alpha(D_f)^{-1})$. Hence $\mu(D_{f'}) \geq \mu(D') = \mu(D_f)$ (by (10.16)). Moreover, if $\mu(D_{f'}) = \mu(D_f)$, then $\alpha(D_{f'}) \subseteq \alpha(D') = \alpha(D_f)$ (again by (10.16)). As at least one arc in P belongs to D_f but not to $D_{f'}$, we have a strict inclusion. Since $\mu(D_f)$ increases at most $|V|$ times and, as long as $\mu(D_f)$ does not change, $\alpha(D_f)$ decreases at most $|A|$ times, we have the theorem. ∎

Since a shortest path can be found in time $O(m)$ (Theorem 6.3), this gives:

Corollary 10.5a. *A maximum flow can be found in time $O(nm^2)$.*

Proof. Directly from Theorem 10.5. ∎

10.6. Dinits' $O(n^2m)$ algorithm

Dinits [1970] observed that one can speed up the maximum flow algorithm, by not augmenting simply along *paths* in D_f, but along *flows* in D_f. The approach is similar to that of Section 9.3 for path packing.

To describe this, define a capacity function c_f on A_f by, for each $a \in A$:

(10.17) $$\begin{aligned} c_f(a) &:= c(a) - f(a) && \text{if } a \in A_f \text{ and} \\ c_f(a^{-1}) &:= f(a) && \text{if } a^{-1} \in A_f. \end{aligned}$$

Then for any flow g in D_f subject to c_f,

(10.18) $$f'(a) := f(a) + g(a) - g(a^{-1})$$

gives a flow f' in D subject to c. (We define $g(a)$ or $g(a^{-1})$ to be 0 if a or a^{-1} does not belong to A_f.)

Now we shall see that, given a flow f in D, one can find in time $O(m)$ a flow g in D_f such that the flow f' arising by (10.18) satisfies $\mu(D_{f'}) > \mu(D_f)$. It implies that there are at most n iterations.

The basis of the method is the concept of 'blocking flow'. An $s-t$ flow f is called *blocking* if for each $s-t$ flow f' with $f \leq f' \leq c$ one has $f' = f$.

Theorem 10.6. *Given an acyclic graph $D = (V, A)$, $s, t \in V$, and a capacity function $c : A \to \mathbb{Q}_+$, a blocking $s-t$ flow can be found in time $O(nm)$.*

Proof. By depth-first search we can find, in time $O(|A'|)$, a subset A' of A and an $s-t$ path P in A' such that no arc in $A' \setminus AP$ is contained in any $s-t$ path: just scan s (cf. (6.2)) until t is reached; then A' is the set of arcs considered so far.

Let f be the maximum flow that can be sent along P, and reset $c := c - f$. Delete all arcs in $A' \setminus AP$ and all arcs a with $c(a) = 0$, and recursively find a blocking $s-t$ flow f' in the new network. Then $f'+f$ is a blocking $s-t$ flow for the original data, as is easily checked.

The running time of the iteration is $O(n+t)$, where t is the number of arcs deleted. Since there are at most $|A|$ iterations and since at most $|A|$ arcs can be deleted, we have the required running time bound. ∎

Hence we have an improvement on the running time for finding a maximum flow:

Corollary 10.6a. *A maximum flow can be found in time $O(n^2m)$.*

Proof. It suffices to describe an $O(nm)$ method to find, for given flow f, a flow f' with $\mu(D_{f'}) > \mu(D_f)$.

Find a blocking flow g in $(V, \alpha(D_f))$. (Note that $\alpha(D_f)$ can be determined in $O(m)$ time.) Let $f'(a) := f(a)+g(a)-g(a^{-1})$, taking values 0 if undefined. Then $D_{f'}$ is a subgraph of $D' := (V, A_f \cup \alpha(D_f)^{-1})$, and hence by (10.16), $\mu(D_{f'}) \geq \mu(D') = \mu(D_f)$. If $\mu(D_{f'}) = \mu(D_f)$, $D_{f'}$ has a path P of length $\mu(D_f)$, which (again (10.16)) should also be a path in $\alpha(D_f)$. But then g could have been increased along this path, contradicting the fact that g is blocking in D_f. ∎

10.6a. Karzanov's $O(n^3)$ algorithm

Karzanov [1974] gave a faster algorithm to find a blocking flow, thus speeding up the maximum flow algorithm. We give the short proof of Malhotra, Kumar, and Maheshwari [1978] (see also Cherkasskiĭ [1979] and Tarjan [1984]).

Theorem 10.7. *Given an acyclic digraph $D = (V, A)$, $s, t \in V$, and a capacity function $c : A \to \mathbb{Q}_+$, a blocking $s-t$ flow can be found in time $O(n^2)$.*

Proof. First order the vertices reachable from s as $s = v_1, v_2, \ldots, v_{n-1}, v_n$ *topologically*; that is, if $(v_i, v_j) \in A$, then $i < j$. This can be done in time $O(m)$ (see Corollary 6.5b).

We describe the algorithm recursively. Consider the minimum of the values $c(\delta^{\text{in}}(v))$ for all $v \in V \setminus \{s\}$ and $c(\delta^{\text{out}}(v))$ for all $v \in V \setminus \{t\}$. Let the minimum be attained by v_i and $c(\delta^{\text{out}}(v_i))$ (without loss of generality). Define $f(a) := c(a)$ for each $a \in \delta^{\text{out}}(v_i)$ and $f(a) := 0$ for all other a.

Next for $j = i+1, \ldots, n-1$, redefine $f(a)$ for each $a \in \delta^{\text{out}}(v_j)$ such that $f(a) \leq c(a)$ and such that $f(\delta^{\text{out}}(v_j)) = f(\delta^{\text{in}}(v_j))$. By the minimality of $c(\delta^{\text{out}}(v_i))$, we can always do this, as initially $f(\delta^{\text{in}}(v_j)) \leq c(\delta^{\text{out}}(v_i)) \leq c(\delta^{\text{out}}(v_j))$. We do this in such a way that finally $f(a) \in \{0, c(a)\}$ for all but at most one a in $\delta^{\text{out}}(v_j)$.

After that, for $j = i, i-1, \ldots, 2$, redefine similarly $f(a)$ for $a \in \delta^{\text{in}}(v_j)$ such that $f(a) \leq c(a)$, $f(\delta^{\text{in}}(v_j)) = f(\delta^{\text{out}}(v_j))$, and $f(a) \in \{0, c(a)\}$ for all but at most one a in $\delta^{\text{in}}(v_j)$.

If $v_i \in \{s, t\}$ we stop, and f is a blocking flow. If $v_i \notin \{s, t\}$, set $c'(a) := c(a) - f(a)$ for each $a \in A$, and delete all arcs a with $c'(a) = 0$ and delete v_i and all arcs incident with v_i, thus obtaining the directed graph $D' = (V', A')$. Obtain (recursively) a blocking flow f' in D' subject to the capacity function c'. Define $f''(a) := f(a) + f'(a)$ for $a \in A'$ and $f''(a) = f(a)$ for $a \in A \setminus A'$. Then f'' is a blocking flow in D.

This describes the algorithm. The correctness can be seen as follows. If $v_i \in \{s,t\}$ the correctness is immediate. If $v_i \notin \{s,t\}$, suppose that f'' is not a blocking flow in D, and let P be an $s-t$ path in D with $f''(a) < c(a)$ for each arc a in P. Then each arc of P belongs to A', since $f''(a) = f(a) = c(a)$ for each $a \in A \setminus (A' \cup \delta^{\text{in}}(v_i))$. So for each arc a of P one has $c'(a) = c(a) - f(a) > f''(a) - f(a) = f'(a)$. This contradicts the fact that f' is a blocking flow in D'.

The running time of the algorithm is $O(n^2)$, since the running time of the iteration is $O(n + |A \setminus A'|)$, and since there are at most $|V|$ iterations. ∎

Theorem 10.7 improves the running time for finding a maximum flow as follows:

Corollary 10.7a. *A maximum flow can be found in time $O(n^3)$.*

Proof. Similar to the proof of Corollary 10.6a. ∎

Sharper blocking flow algorithms were found by Cherkasskiĭ [1977a] ($O(n\sqrt{m})$), Galil [1978,1980a] ($O((nm)^{2/3})$), Shiloach [1978] and Galil and Naamad [1979,1980] ($O(m \log^2 n)$), Sleator [1980] and Sleator and Tarjan [1981,1983a] ($O(m \log n)$), and Goldberg and Tarjan [1990] ($O(m \log(n^2/m))$), each yielding a maximum flow algorithm with running time bound a factor of n higher.

An alternative approach finding a maximum flow in time $O(nm \log(n^2/m))$, based on the 'push-relabel' method, was developed by Goldberg [1985,1987] and Goldberg and Tarjan [1986,1988a], and is described in the following section.

10.7. Goldberg's push-relabel method

The algorithms for the maximum flow problem described above are all based on flow augmentation. The basis is updating a flow f until D_f has no $s-t$ path. Goldberg [1985,1987] and Goldberg and Tarjan [1986,1988a] proposed a different, in a sense dual, method, the 'push-relabel' method: update a 'preflow' f, maintaining the property that D_f has no $s-t$ path, until f is a flow. (Augmenting flow methods are 'primal' as they maintain feasibility of the primal linear program, while the push-relabel method maintains feasibility of the dual linear program.)

Let $D = (V, A)$ be a digraph, $s, t \in V$, and $c : A \to \mathbb{Q}_+$. A function $f : A \to \mathbb{Q}$ is called an $s-t$ *preflow*, or just a *preflow*, if

(10.19) (i) $0 \le f(a) \le c(a)$ for each $a \in A$,
(ii) $\text{excess}_f(v) \ge 0$ for each vertex $v \neq s$.

(Preflows were introduced by Karzanov [1974]. excess_f was defined in Section 10.1.)

Condition (ii) says that at each vertex $v \neq s$, the outgoing preflow does not exceed the ingoing preflow. For any preflow f, call a vertex v *active* if

$v \neq t$ and $\text{excess}_f(v) > 0$. So f is an $s-t$ flow if and only if there are no active vertices.

The *push-relabel method* consists of keeping a pair f, p, where f is a preflow and $p : V \to \mathbb{Z}_+$ such that

(10.20) (i) if $(u,v) \in A_f$, then $p(v) \geq p(u) - 1$,
(ii) $p(s) = n$ and $p(t) = 0$.

Note that for any given f, such a function p exists if and only if D_f has no $s-t$ path. Hence, if a function p satisfying (10.20) exists and f is an $s-t$ flow, then f is an $s-t$ flow of maximum value (Corollary 10.2a).

Initially, f and p are set by:

(10.21) $f(a) := c(a)$ if $a \in \delta^{\text{out}}(s)$ and $f(a) := 0$ otherwise;
$p(v) := n$ if $v = s$ and $p(v) := 0$ otherwise.

Next, while there exist active vertices, choose an active vertex u maximizing $p(u)$, and apply the following iteratively, until u is inactive:

(10.22) choose an arc $(u,v) \in A_f$ with $p(v) = p(u) - 1$ and *push* over (u,v); if no such arc exists, *relabel* u.

Here to *push* over $(u,v) \in A_f$ means:

(10.23) if $(u,v) \in A$, reset $f(u,v) := f(u,v) + \varepsilon$, where $\varepsilon := \min\{c(u,v) - f(u,v), \text{excess}_f(u)\}$;
if $(v,u) \in A$, reset $f(v,u) := f(v,u) - \varepsilon$, where $\varepsilon := \min\{f(v,u), \text{excess}_f(u)\}$.

To *relabel* u means:

(10.24) reset $p(u) := p(u) + 1$.

Note that if A_f has no arc (u,v) with $p(v) = p(u) - 1$, then we can relabel u without violating (10.20).

This method terminates, since:

Theorem 10.8. *The number of pushes is $O(n^3)$ and the number of relabels is $O(n^2)$.*

Proof. First we show:

(10.25) throughout the process, $p(v) < 2n$ for each $v \in V$.

Indeed, if v is active, then D_f contains a $v-s$ path (since f can be decomposed as a sum of incidence vectors of $s-v$ paths, for $v \in V$, and of directed circuits). So by (10.20)(i), $p(v) - p(s) \leq \text{dist}_{D_f}(v,s) < n$. As $p(s) = n$, we have $p(v) < 2n$. This gives (10.25), which directly implies:

(10.26) the number of relabels is at most $2n^2$.

To estimate the number of pushes, call a push (10.23) *saturating* if after the push one has $f(u,v) = c(u,v)$ (if $(u,v) \in A$) or $f(v,u) = 0$ (if $(v,u) \in A$). Then:

(10.27) the number of saturating pushes is $O(nm)$.

For consider any arc $a = (u,v) \in A$. If we increase $f(a)$, then $p(v) = p(u) - 1$, while if we decrease $f(a)$, then $p(u) = p(v) - 1$. So meantime $p(v)$ should have been relabeled at least twice. As p is nondecreasing (in time), by (10.25) we have (10.27).

Finally:

(10.28) the number of nonsaturating pushes is $O(n^3)$.

Between any two relabels the function p does not change. Hence there are $O(n)$ nonsaturating pushes, as each of them makes an active vertex v maximizing $p(v)$ inactive (while possibly a vertex v' with $p(v') < p(v)$ is activated). With (10.26) this gives (10.28). ∎

There is an efficient implementation of the method:

Theorem 10.9. *The push-relabel method finds a maximum flow in time $O(n^3)$.*

Proof. We order the vertex set V as a doubly linked list, in order of increasing value $p(v)$. Moreover, for each $u \in V$ we keep the set L_u of arcs (u,v) in A_f with $p(v) = p(u) - 1$, ordered as a doubly linked list. We also keep with each vertex v the value $\text{excess}_f(v)$, and we keep linked lists of arcs of D incident with v.

Throughout the iterations, we choose an active vertex u maximizing $p(u)$, and we process u, until u becomes inactive. Between any two relabelings, this searching takes $O(n)$ time, since as long as we do not relabel, we can continue searching the list V in order. As we relabel $O(n^2)$ times, we can do the searching in $O(n^3)$ time.

Suppose that we have found an active vertex u maximizing $p(u)$. We next push over each of the arcs in L_u. So finding an arc $a = (u,v)$ for pushing takes time $O(1)$. If it is a saturating push, we can delete (u,v) from L_u in time $O(1)$. Moreover, we can update $\text{excess}_f(u)$ and $\text{excess}_f(v)$ in time $O(1)$. Therefore, as there are $O(n^3)$ pushes, they can be done in $O(n^3)$ time.

We decide to relabel u if $L_u = \emptyset$. When relabeling, updating the lists takes $O(n)$ time: When we reset $p(u)$ from i to $i+1$, then for each arc (u,v) or (v,u) of D, we add (u,v) to L_u if $p(v) = i$ and $(u,v) \in A_f$, and we remove (v,u) from L_v if $p(v) = i+1$ and $(v,u) \in A_f$; moreover, we move u to its new rank in the list V. This all takes $O(n)$ time. Therefore, as there are $O(n^2)$ relabels, they can be done in $O(n^3)$ time. ∎

Further notes on the push-relabel method. If we allow any active vertex u to be chosen for (10.22) (not requiring maximality of $p(u)$), then the bounds of

$O(n^2)$ on the number of relabels and $O(nm)$ on the number of saturating pushes are maintained, while the number of nonsaturating pushes is $O(n^2m)$.

A first-in first-out selection rule was studied by Goldberg [1985], also yielding an $O(n^3)$ algorithm. Theorem 10.9 (using the largest-label selection) is due to Goldberg and Tarjan [1986,1988a], who also showed an implementation of the push-relabel method with dynamic trees, taking $O(nm\log(n^2/m))$ time. Cheriyan and Maheshwari [1989] and Tunçel [1994] showed that the bound on the number of pushes in Theorem 10.8 can be improved to $O(n^2\sqrt{m})$, yielding an $O(n^2\sqrt{m})$ running time bound. Further improvements are given in Ahuja and Orlin [1989] and Ahuja, Orlin, and Tarjan [1989]. The worst-case behaviour of the push-relabel method was studied by Cheriyan and Maheshwari [1989].

10.8. Further results and notes

10.8a. A weakly polynomial bound

Edmonds and Karp [1972] considered the following *fattest augmenting path* rule: choose a flow-augmenting path for which the flow value increase is maximal. They showed that, if all capacities are integer, it terminates in at most $1 + m'\log\phi$ iterations, where ϕ is the maximum flow value and where m' is the maximum number of arcs in any $s-t$ cut. This gives a maximum flow algorithm of running time $O(n^2m\log nC)$, where C is the maximum capacity (assuming all capacities are integer). (For irrational capacities, Queyranne [1980] showed that the method need not terminate.)

Edmonds and Karp [1970,1972] and Dinits [1973a] introduced the idea of *capacity-scaling*, which gives the following stronger running time bound:

Theorem 10.10. *For integer capacities, a maximum flow can be found in time $O(m^2\log C)$.*

Proof. Let $L := \lceil\log_2 C\rceil+1$. For $i = L, L-1, \ldots, 0$, we can obtain a maximum flow f' for capacity function $c' := \lfloor c/2^i\rfloor$, from a maximum flow f'' for capacity function $c'' := \lfloor c/2^{i+1}\rfloor$ as follows. Observe that the maximum flow value for c' differs by at most m from that of the maximum flow value ϕ for $2c''$. For let $\delta^{\text{out}}(U)$ be a cut with $2c''(\delta^{\text{out}}(U)) = \phi$. Then $c'(\delta^{\text{out}}(U)) - \phi \leq |\delta^{\text{out}}(U)| \leq m$. So a maximum flow with respect to c' can be obtained from $2f''$ by at most m augmenting path iterations. As each augmenting path iteration can be done in $O(m)$ time, and as $\lfloor c/2^L\rfloor = 0$, we have the running time bound given. ∎

With methods similar to those used in Corollary 10.6a, the bound in Theorem 10.10 can be improved to $O(nm\log C)$, a result of Dinits [1973a] and Gabow [1985b]. To see this, observe that the proof of Theorem 10.6 also yields:

Theorem 10.11. *Given an acyclic graph $D = (V, A)$, $s, t \in V$, and a capacity function $c : A \to \mathbb{Z}_+$, an integer blocking flow f can be found in time $O(n\phi + m)$, where ϕ is the value of f.*

Proof. Consider the proof of Theorem 10.6. We do at most ϕ iterations, while each iteration takes $O(n+t)$ time, where t is the number of arcs deleted. ∎

Hence, similarly to Corollary 10.6a one has:

Corollary 10.11a. *For integer capacities, a maximum flow can be found in time $O(n(\phi + m))$, where ϕ is the maximum flow value.*

Proof. Similar to the proof of Corollary 10.6a. ∎

Therefore,

Corollary 10.11b. *For integer capacities, a maximum flow can be found in time $O(nm \log C)$.*

Proof. In the proof of Theorem 10.10, a maximum flow with respect to c' can be obtained from $2f''$ in time $O(nm)$ (by Corollary 10.11a), since the maximum flow value in the residual graph $D_{f''}$ is at most m. ∎

10.8b. Complexity survey for the maximum flow problem

Complexity survey ($*$ indicates an asymptotically best bound in the table):

$O(n^2mC)$	Dantzig [1951a] simplex method
$O(nmC)$	Ford and Fulkerson [1955,1957b] augmenting path
$O(nm^2)$	Dinits [1970], Edmonds and Karp [1972] shortest augmenting path
$O(n^2m \log nC)$	Edmonds and Karp [1972] fattest augmenting path
$O(n^2m)$	Dinits [1970] shortest augmenting path, layered network
$O(m^2 \log C)$	Edmonds and Karp [1970,1972] capacity-scaling
$O(nm \log C)$	Dinits [1973a], Gabow [1983b,1985b] capacity-scaling
$O(n^3)$	Karzanov [1974] (preflow push); cf. Malhotra, Kumar, and Maheshwari [1978], Tarjan [1984]
$O(n^2\sqrt{m})$	Cherkasskiĭ [1977a] blocking preflow with long pushes
$O(nm \log^2 n)$	Shiloach [1978], Galil and Naamad [1979,1980]
$O(n^{5/3}m^{2/3})$	Galil [1978,1980a]

≫

continued

	$O(nm \log n)$	Sleator [1980], Sleator and Tarjan [1981,1983a] dynamic trees
*	$O(nm \log(n^2/m))$	Goldberg and Tarjan [1986,1988a] push-relabel+dynamic trees
	$O(nm + n^2 \log C)$	Ahuja and Orlin [1989] push-relabel + excess scaling
	$O(nm + n^2\sqrt{\log C})$	Ahuja, Orlin, and Tarjan [1989] Ahuja-Orlin improved
*	$O(nm \log((n/m)\sqrt{\log C} + 2))$	Ahuja, Orlin, and Tarjan [1989] Ahuja-Orlin improved + dynamic trees
*	$O(n^3/\log n)$	Cheriyan, Hagerup, and Mehlhorn [1990,1996]
	$O(n(m + n^{5/3} \log n))$	Alon [1990] (derandomization of Cheriyan and Hagerup [1989,1995])
	$O(nm + n^{2+\varepsilon})$	(for each $\varepsilon > 0$) King, Rao, and Tarjan [1992]
*	$O(nm \log_{m/n} n + n^2 \log^{2+\varepsilon} n)$	(for each $\varepsilon > 0$) Phillips and Westbrook [1993,1998]
*	$O(nm \log_{\frac{m}{n \log n}} n)$	King, Rao, and Tarjan [1994]
*	$O(m^{3/2} \log(n^2/m) \log C)$	Goldberg and Rao [1997a,1998]
*	$O(n^{2/3} m \log(n^2/m) \log C)$	Goldberg and Rao [1997a,1998]

Here $C := \|c\|_\infty$ for integer capacity function c. For a complexity survey for unit capacities, see Section 9.6a.

Research problem: Is there an $O(nm)$-time maximum flow algorithm?

For the special case of *planar* undirected graphs:

	$O(n^2 \log n)$	Itai and Shiloach [1979]
	$O(n \log^2 n)$	Reif [1983] (minimum cut), Hassin and Johnson [1985] (maximum flow)
	$O(n \log n \log^* n)$	Frederickson [1983b]
*	$O(n \log n)$	Frederickson [1987b]

For *directed* planar graphs:

	$O(n^{3/2} \log n)$	Johnson and Venkatesan [1982]
	$O(n^{4/3} \log^2 n \log C)$	Klein, Rao, Rauch, and Subramanian [1994], Henzinger, Klein, Rao, and Subramanian [1997]
*	$O(n \log n)$	Weihe [1994b,1997b]

Itai and Shiloach [1979] and Hassin [1981b] showed that if s and t both are on the outer boundary, then a shortest path algorithm applied to the dual gives an $O(n \log n)$ algorithm for finding a minimum-capacity $s-t$ cut and a maximum-value $s-t$ flow, also for the directed case. This extends earlier work of Hu [1969].

Khuller, Naor, and Klein [1993] studied the lattice structure of the integer $s-t$ flows in a planar directed graph. More on planar maximum flow can be found in Khuller and Naor [1990,1994].

10.8c. An exchange property

Dinits [1973b] and Minieka [1973] observed the following analogue of Theorem 9.12:

Theorem 10.12. *Let $D = (V, A)$ be a digraph, let $s, t \in V$, let $c : A \to \mathbb{R}_+$, and let f_1 and f_2 be maximum $s-t$ flows subject to c. Then there exists a maximum $s-t$ flow f subject to c such that $f(a) = f_1(a)$ for each arc a incident with s and $f(a) = f_2(a)$ for each arc a incident with t.*

Proof. Let $\delta^{\text{out}}(U)$ be an $s-t$ cut of minimum capacity, with $U \subseteq V$ and $s \in U$,$t \notin U$. So f_1 and f_2 coincide on $\delta^{\text{out}}(U)$ and on $\delta^{\text{in}}(U)$. Define $f(a) := f_1(a)$ if a is incident with U and $f(a) := f_2(a)$ if a is incident with $V \setminus U$. This defines a maximum $s-t$ flow as required. ∎

This was also shown by Megiddo [1974], who used it to prove the following. Let $D = (V, A)$ be a directed graph, let $c : A \to \mathbb{R}_+$ be a capacity function, and let $s, t \in V$, where s is a source, and t is a sink. An $s-t$ flow $f \leq c$ is called *source-optimal* if the vector $(f(a) \mid a \in \delta^{\text{out}}(s))$ is lexicographically maximal among all $s-t$ flows subject to c (ordering the $a \in \delta^{\text{out}}(s)$ by nonincreasing value of $f(a)$). The maximum flow algorithm implies that a source-optimal $s-t$ flow is a maximum-value $s-t$ flow.

One similarly defines *sink-optimal*, and Theorem 10.12 implies that there exists an $s-t$ flow that is both source- and sink-optimal. The proof shows that this flow can be found by combining a source-optimal and a sink-optimal flow appropriately.

As Megiddo showed, a source-optimal flow can be found iteratively, by updating a flow f (starting with $f = \mathbf{0}$), by determining an arc $a \in \delta^{\text{out}}(s)$ with $f(a) = 0$, on which $f(a)$ can be increased most. Making this increase, gives the next f. Stop if no increase is possible anymore.

10.8d. Further notes

Simplex method. The maximum flow problem is a linear programming problem, and hence it can be solved with the simplex method of Dantzig [1951b] (this paper includes an anti-cycling rule based on perturbation). This was elaborated by Fulkerson and Dantzig [1955a,1955b]. A direct, combinatorial anti-cycling rule for flow problems was given by Cunningham [1976]. Goldfarb and Hao [1990] gave a pivot rule that leads to at most nm pivots, yielding an algorithm of running time $O(n^2m)$. Goldberg, Grigoriadis, and Tarjan [1991] showed that with the help of dynamic trees there is an $O(nm \log n)$ implementation. See also Gallo, Grigoriadis, and Tarjan [1989], Plotkin and Tardos [1990], Goldfarb and Hao [1991], Orlin, Plotkin, and Tardos [1993], Ahuja and Orlin [1997], Armstrong and Jin [1997], Goldfarb and Chen [1997], Tarjan [1997], Armstrong, Chen, Goldfarb, and Jin [1998], and Hochbaum [1998].

Worst-case analyses of maximum flow algorithms were given by Zadeh [1972, 1973b] (shortest augmenting path rule), Dinits [1973b] (shortest augmenting path rule), Tarjan [1974e], Even and Tarjan [1975] (Dinits $O(n^2m)$ algorithm), Baratz [1977] (Karzanov's $O(n^3)$ algorithm), Galil [1981], Cheriyan [1988] (push-relabel method), Cheriyan and Maheshwari [1989] (push-relabel method), and Martel [1989] (push-relabel method). For further analysis of maximum flow algorithms, see Tucker [1977a] and Ahuja and Orlin [1991].

Computational studies were reported by Cherkasskiĭ [1979], Glover, Klingman, Mote, and Whitman [1979,1980,1984], Hamacher [1979] (Karzanov's method), Cheung [1980], Imai [1983b], Goldfarb and Grigoriadis [1988] (Dinits' method and the simplex method), Derigs and Meier [1989] (push-relabel method), Alizadeh and Goldberg [1993] (push-relabel in parallel), Anderson and Setubal [1993] (push-relabel), Gallo and Scutellà [1993], Nguyen and Venkateswaran [1993] (push-relabel), and Cherkassky and Goldberg [1995,1997] (push-relabel method). Consult also Johnson and McGeoch [1993].

A probabilistic analysis was presented by Karp, Motwani, and Nisan [1993]. A randomized approximation algorithm for minimum $s-t$ cut was given by Benczúr and Karger [1996].

Fulkerson [1959b] gave a labeling algorithm for finding the minimum cost of capacities to be added to make an $s-t$ flow of given value possible. Wollmer [1964] studied which k arcs to remove from a capacitated digraph so as to reduce the maximum $s-t$ flow value as much as possible. McCormick [1997] studied the problem of computing 'least infeasible' flows. Akers [1960] described the effect of ΔY operations on max-flow computations.

Ponstein [1972] gave another rule guaranteeing termination of the augmenting path iterations in Ford and Fulkerson's algorithm. Karp [1972b] showed that the *maximum*-cut problem is NP-complete — see Section 75.1a. Work on flows with small capacities was reported by Fernández-Baca and Martel [1989] and Ahuja and Orlin [1991]. Decomposition algorithms for locating minimal cuts were studied by Jarvis and Tufekci [1982]. The kth best cut algorithm was given by Hamacher [1982].

The problem of determining a flow along odd paths in an undirected graph was considered by Schrijver and Seymour [1994] — see Section 29.11e.

For an in-depth survey of network flows, see Ahuja, Magnanti, and Orlin [1993]. Other surveys were given by Ford and Fulkerson [1962], Dantzig [1963], Busacker and Saaty [1965], Fulkerson [1966], Hu [1969,1982], Iri [1969], Frank and Frisch [1971], Berge [1973b], Adel'son-Vel'skiĭ, Dinits, and Karzanov [1975] (for a review, see Goldberg and Gusfield [1991]), Christofides [1975], Lawler [1976b], Bazaraa and Jarvis [1977], Minieka [1978], Even [1979], Jensen and Barnes [1980], Papadimitriou and Steiglitz [1982], Smith [1982], Chvátal [1983], Sysło, Deo, and Kowalik [1983], Tarjan [1983], Gondran and Minoux [1984], Rockafellar [1984], Tarjan [1986], Nemhauser and Wolsey [1988], Ahuja, Magnanti, and Orlin [1989,1991], Chen [1990], Cormen, Leiserson, and Rivest [1990], Goldberg, Tardos, and Tarjan [1990], Frank [1995], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], Mehlhorn and Näher [1999], and Korte and Vygen [2000].

Golden and Magnanti [1977] gave a bibliography and Slepian [1968] discussed the algebraic theory of flows.

10.8e. Historical notes on maximum flow

The problem of sending flow through a network was considered by Kantorovich [1939]. In fact, he considered multicommodity flows — see the historical notes in Section 70.13g.

The foundations for one-commodity maximum flow were laid during the period November 1954-December 1955 at RAND Corporation in Santa Monica, California. We will review the developments in a chronological order, by the date of the RAND Reports.

In their basic report *Maximal Flow through a Network* dated 19 November 1954, Ford and Fulkerson [1954,1956b] showed the max-flow min-cut theorem for undirected graphs:

> Theorem 1. (Minimal cut theorem). The maximal flow value obtainable in a network N is the minimum of $v(D)$ taken over all disconnecting sets D.

(Robacker [1955a] wrote that the max-flow min-cut theorem was conjectured first by Fulkerson.)

Ford and Fulkerson were motivated by flows and cuts in railway networks — see below. In the same report, also a simple algorithm was described for the maximum flow problem in case the graph, added with an extra edge connecting s and t, is planar.

The authors moreover observed that the maximum flow problem is a special case of a linear programming problem and that hence it can be solved by Dantzig's simplex method.

In a report of 1 January 1955 (revised 15 April 1955), Dantzig and Fulkerson [1955,1956] showed that the max-flow min-cut theorem can also be deduced from the duality theorem of linear programming (they mention that also A.J. Hoffman did this), they generalized it to the directed case, and they observed, using results of Dantzig [1951a], that if the capacities are integer, there is an integer maximum flow (the 'integrity theorem'). Hence (as they mention) Menger's theorem follows as a consequence. A simple computational method for the maximum flow problem based on the simplex method was described in a report of 1 April 1955 by Fulkerson and Dantzig [1955a,1955b].

Conversely, in a report of 26 May 1955, Robacker [1955a] derived the undirected max-flow min-cut theorem from the undirected vertex-disjoint version of Menger's theorem.

Boldyreff's heuristic

While the maximum flow algorithms found so far were derived from the simplex method, the quest for combinatorial methods remained vivid. A heuristic for the maximum flow problem, the 'flooding technique', was presented by Boldyreff [1955c, 1955b] on 3 June 1955 at the New York meeting of the Operations Research Society of America (published as a RAND Report of 5 August 1955 (Boldyreff [1955a])). The method is intuitive and the author did not claim generality (we quote from Boldyreff [1955b]):

> It has been previously assumed that a highly complex railway transportation system, too complicated to be amenable to analysis, can be represented by a much simpler model. This was accomplished by representing each complete railway operating division by a point, and by joining pairs of such points by arcs (lines) with traffic carrying capacities equal to the maximum possible volume of traffic (expressed in some convenient unit, such as trains per day) between the corresponding operating divisions.
> In this fashion, a network is obtained consisting of three sets of points — points of origin, intermediate or junction points, and the terminal points (or points of destination) — and a set of arcs of specified traffic carrying capacities, joining these points to each other.

Boldyreff's arguments for designing a heuristic procedure are formulated as follows:

> In the process of searching for the methods of solving this problem the following objectives were used as a guide:
> **1.** That the solution could be obtained quickly, even for complex networks.
> **2.** That the method could be explained easily to personnel without specialized technical training and used by them effectively.
> **3.** That the validity of the solution be subject to easy, direct verification.
> **4.** That the method would not depend on the use of high-speed computing or other specialized equipment.

Boldyreff's 'flooding technique' pushes as much flow as possible greedily through the network. If at some vertex a 'bottleneck' arises (i.e., more trains arrive than can be pushed further through the network), it is eliminated by returning the excess trains to the origin.

The method is empirical, not using backtracking, and not leading to an optimum solution in all cases:

> Whenever arbitrary decisions have to be made, ordinary common sense is used as a guide. At each step the guiding principle is to move forward the maximum possible number of trains, and to maintain the greatest flexibility for the remaining network.

Boldyreff speculates:

> In dealing with the usual railway networks a single flooding, followed by removal of bottlenecks, should lead to a maximal flow.

In the abstract of his lecture, Boldyreff [1955c] mentions:

> The mechanics of the solutions is formulated as a simple game which can be taught to a ten-year-old boy in a few minutes.

In his article, Boldyreff [1955b] gave as example the model of a real, comprehensive, railway transportation system with 41 vertices and 85 arcs:

> The total time of solving the problem is less than thirty minutes.

His closing remarks are:

> Finally there is the question of a systematic formal foundation, the comprehensive mathematical basis for empiricism and intuition, and the relation of the present techniques to other processes, such as, for instance, the multistage decision process (a suggestion of Bellman's).
> All this is reserved for the future.

Ford and Fulkerson's motivation: The Harris-Ross report

In their first report on maximum flow, *Maximal Flow through a Network*, Ford and Fulkerson [1954,1956b] mentioned that the maximum flow problem was formulated by T.E. Harris as follows:

> Consider a rail network connecting two cities by way of a number of intermediate cities, where each link of the network has a number assigned to it representing its capacity. Assuming a steady state condition, find a maximal flow from one given city to the other.

Later, in their book *Flows in Networks*, Ford and Fulkerson [1962] gave a more precise reference[17]:

> It was posed to the authors in the spring of 1955 by T.E. Harris, who, in conjunction with General F.S. Ross (Ret.), had formulated a simplified model of railway traffic flow, and pinpointed this particular problem as the central one suggested by the model [11].

Ford and Fulkerson's reference [11] here is the secret report by Harris and Ross [1955] entitled *Fundamentals of a Method for Evaluating Rail Net Capacities*, dated 24 October 1955[18], and written for the Air Force. The report was downgraded to 'unclassified' on 21 May 1999.

Unlike what Ford and Fulkerson write, the interest of Harris and Ross was not to find a maximum flow, but rather a minimum cut ('interdiction') of the Soviet railway system. We quote:

> Air power is an effective means of interdicting an enemy's rail system, and such usage is a logical and important mission for this Arm.
> As in many military operations, however, the success of interdiction depends largely on how complete, accurate, and timely is the commander's information, particularly concerning the effect of his interdiction-program efforts on the enemy's capability to move men and supplies. This information should be available at the time the results are being achieved.
> The present paper describes the fundamentals of a method intended to help the specialist who is engaged in estimating railway capabilities, so that he might more readily accomplish this purpose and thus assist the commander and his staff with greater efficiency than is possible at present.

In the Harris-Ross report, first much attention is given to modelling a railway network: taking each railway junction as a vertex would give a too refined network (for their purposes). Therefore, Harris and Ross proposed to take 'railway divisions' (organizational units based on geographical areas) as vertices, and to estimate the capacity of the connections between any two adjacent railway divisions. In an interview with Alexander [1996], Harris remembered:

> We were studying rail transportation in consultation with a retired army general, Frank Ross, who had been chief of the Army's Transportation Corps in Europe. We thought of modeling a rail system as a network. At first it didn't make sense, because there's no reason why the crossing point of two lines should be a special

[17] There seems to be some discrepancy between the date of the RAND Report of Ford and Fulkerson (19 November 1954) and the date mentioned in the quotation (spring of 1955).

[18] In their book, Ford and Fulkerson incorrectly date the Harris-Ross report 24 October 1956.

> sort of node. But Ross realized that, in the region we were studying, the "divisions" (little administrative districts) should be the nodes. The link between two adjacent nodes represents the total transportation capacity between them. This made a reasonable and manageable model for our rail system.

The Harris-Ross report stresses that specialists remain needed to make up the model (which seems always a good strategy to get new methods accepted):

> It is not the purpose that the highly specialized individual who estimates track and network capacities should be replaced by a novice with a calculating machine. Rather, it is accepted that the evaluation of track capacities remains a task for the specialist.
>
> [...]
>
> The ability to estimate with relative accuracy the capacity of single railway lines is largely an art. Specialists in this field have no authoritative text (insofar as the authors are informed) to guide their efforts, and very few individuals have either the experience or talent for this type of work. The authors assume that this job will continue to be done by the specialist.

The authors next disputed the naive belief that a railway network is just a set of disjoint through lines, and that cutting these lines would imply cutting the network:

> It is even more difficult and time-consuming to evaluate the capacity of a railway network comprising a multitude of rail lines which have widely varying characteristics. Practices among individuals engaged in this field vary considerably, but all consume a great deal of time. Most, if not all, specialists attack the problem by viewing the railway network as an aggregate of through lines.
>
> The authors contend that the foregoing practice does not portray the full flexibility of a large network. In particular it tends to gloss over the fact that even if every one of a set of independent through lines is made inoperative, there may exist alternative routings which can still move the traffic.
>
> This paper proposes a method that departs from present practices in that it views the network as an aggregate of railway operating divisions. All trackage capacities within the divisions are appraised, and these appraisals form the basis for estimating the capability of railway operating divisions to receive trains from and concurrently pass trains to each neighboring division in 24-hour periods.

Whereas experts are needed to set up the model, to solve it is routine (when having the 'work sheets'):

> The foregoing appraisal (accomplished by the expert) is then used in the preparation of comparatively simple work sheets that will enable relatively inexperienced assistants to compute the results and thus help the expert to provide specific answers to the problems, based on many assumptions, which may be propounded to him.

While Ford and Fulkerson flow-augmenting path algorithm for the maximum flow problem was not found yet, the Harris-Ross report suggests applying Boldyreff's flooding technique described above. The authors preferred this above the simplex method for maximum flow:

> The calculation would be cumbersome; and, even if it could be performed, sufficiently accurate data could not be obtained to justify such detail.

However, later in the report their assessment of the simplex method is more favourable:

> These methods do not require elaborate computations and can be performed by a relatively untrained person.

The Harris-Ross report applies Boldyreff's flooding technique to a network model of the Soviet and East European railways. For the data it refers to several secret reports of the Central Intelligence Agency (C.I.A.) on sections of the Soviet and East European railway networks. After the aggregation of railway divisions to vertices, the network has 44 vertices and 105 (undirected) edges.

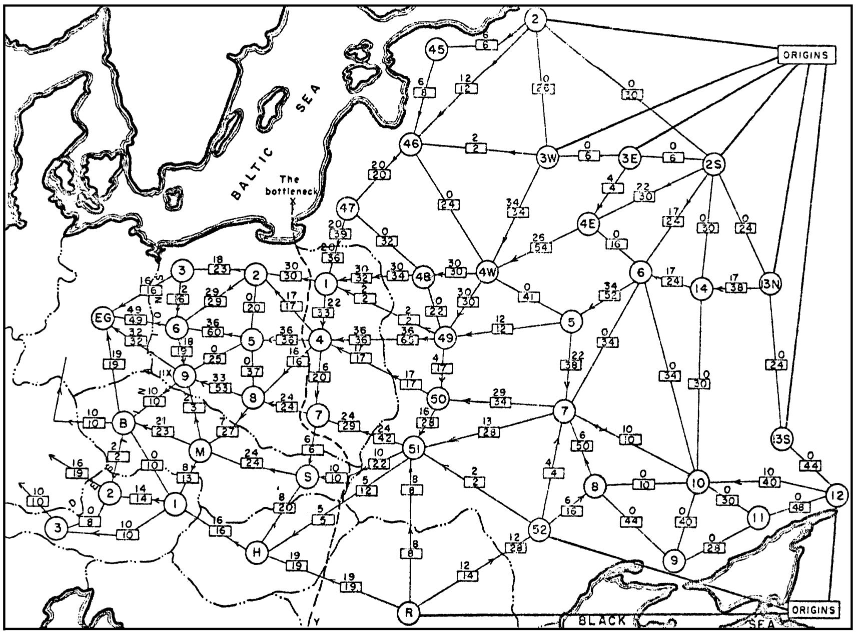


Figure 10.2

From Harris and Ross [1955]: Schematic diagram of the railway network of the Western Soviet Union and East European countries, with a maximum flow of value 163,000 tons from Russia to Eastern Europe, and a cut of capacity 163,000 tons indicated as 'The bottleneck'.

The application of the flooding technique to the problem is displayed step by step in an appendix of the report, supported by several diagrams of the railway network. (Also work sheets are provided, to allow for future changes in capacities.) It yields a flow of value 163,000 tons from sources in the Soviet Union to destinations in East European 'satellite' countries, together with a cut with a capacity of, again, 163,000 tons. So the flow value and the cut capacity are equal, hence optimum. In the report, the minimum cut is indicated as 'the bottleneck' (Figure 10.2).

Further developments

Soon after the Harris-Ross report, Ford and Fulkerson [1955,1957b] presented in a RAND Report of 29 December 1955 their 'very simple algorithm' for the maximum flow problem, based on finding 'augmenting paths' as described in Section 10.4 above. The algorithm finds in a finite number of steps a maximum flow, if all capacities have integer values. We quote:

> This problem is of course a linear programming problem, and hence may be solved by Dantzig's simplex algorithm. In fact, the simplex computation for a problem of this kind is particularly efficient, since it can be shown that the sets of equations one solves in the process are always triangular [2]. However, for the flow problem, we shall describe what appears to be a considerably more efficient algorithm; it is, moreover, readily learned by a person with no special training, and may easily be mechanized for handling large networks. We believe that problems involving more than 500 nodes and 4,000 arcs are within reach of present computing machines.

(Reference [2] is Dantzig and Fulkerson [1955].)

In the RAND Report, Ford and Fulkerson [1955] mention that Boldyreff's flooding technique might give a good starting flow, but in the final paper (Ford and Fulkerson [1957b]) this suggestion has been omitted.

An alternative proof of the max-flow min-cut theorem was given by Elias, Feinstein, and Shannon [1956] ('manuscript received by the PGIT, July 11,1956'), who claimed that the result was known by workers in communication theory:

> This theorem may appear almost obvious on physical grounds and appears to have been accepted without proof for some time by workers in communication theory. However, while the fact that this flow cannot be exceeded is indeed almost trivial, the fact that it can actually be achieved is by no means obvious. We understand that proofs of the theorem have been given by Ford and Fulkerson and Fulkerson and Dantzig. The following proof is relatively simple, and we believe different in principle.

The proof of Elias, Feinstein, and Shannon is based on a reduction technique similar to that used by Menger [1927] in proving his theorem.

Chapter 11

Circulations and transshipments

Circulations and transshipments are variants of flows. Circulations have no source or sink — so flow conservation holds in each vertex — while transshipments have several sources and sinks — so *any* nonnegative function is a transshipment (however, the problem is to find a transshipment with prescribed excess function).
Problems on circulations and transshipments can be reduced to flow problems, or can be treated with similar methods.

11.1. A useful fact on arc functions

Recall that for any digraph $D = (V, A)$ and any $f : A \to \mathbb{R}$, the *excess* function $\text{excess}_f : V \to \mathbb{R}$ is defined by

$$\text{excess}_f(v) := f(\delta^{\text{in}}(v)) - f(\delta^{\text{out}}(v)) \tag{11.1}$$

for $v \in V$.

The following theorem of Gallai [1958a,1958b] will turn out to be useful in this chapter:

Theorem 11.1. *Let $D = (V, A)$ be a digraph and let $f : A \to \mathbb{R}_+$. Then f is a nonnegative linear combination of at most $|A|$ vectors χ^P, where P is a directed path or circuit. If P is a path, it starts at a vertex v with* $\text{excess}_f(v) <$ *0 and ends at a vertex with* $\text{excess}_f(v) > 0$. *If f is integer, we can take the linear combination integer-scalared. The combination can be found in $O(nm)$ time.*

Proof. We may assume that $\text{excess}_f = \mathbf{0}$, since we can add a new vertex u, for each $v \in V$ with $\text{excess}_f(v) > 0$, an arc (v, u), and for each $v \in V$ with $\text{excess}_f(v) < 0$, an arc (u, v). Define $f(v, u) := \text{excess}_f(v)$ and $f(u, v) := -\text{excess}_f(v)$ for any new arc (u, v) or (v, u). Then the new f satisfies $\text{excess}_f = \mathbf{0}$, and a decomposition for the new f gives a decomposition for the original f.

Define $A' := \{a \mid f(a) > 0\}$. We apply induction on $|A'|$. We may assume that $A' \neq \emptyset$. Then A' contains a directed circuit, C say. Let τ be the minimum of the $f(a)$ for $a \in AC$ and let $f' := f - \tau\chi^C$. Then the theorem follows by induction.

Since we can find C in $O(n)$ time, we can find the decomposition in $O(nm)$ time. ∎

11.2. Circulations

Let $D = (V, A)$ be a digraph. A function $f : A \to \mathbb{R}$ is called a *circulation* if

(11.2) $$f(\delta^{\text{in}}(v)) = f(\delta^{\text{out}}(v))$$

for each vertex $v \in V$. So now the flow conservation law holds in *each* vertex v. By Theorem 11.1,

(11.3) each nonnegative circulation is a nonnegative linear combination of incidence vectors of directed circuits; each nonnegative integer circulation is the sum of incidence vectors of directed circuits.

Hoffman [1960] mentioned that he proved the following characterization of the existence of circulations in 1956[19]:

Theorem 11.2 (Hoffman's circulation theorem)**.** *Let $D = (V, A)$ be a digraph and let $d, c : A \to \mathbb{R}$ with $d \leq c$. Then there exists a circulation f satisfying $d \leq f \leq c$ if and only if*

(11.4) $$d(\delta^{\text{in}}(U)) \leq c(\delta^{\text{out}}(U))$$

for each subset U of V. If moreover d and c are integer, f can be taken integer.

Proof. To see necessity of (11.4), suppose that a circulation f satisfying $d \leq f \leq c$ exists. Then for each $U \subseteq V$,

(11.5) $$d(\delta^{\text{in}}(U)) \leq f(\delta^{\text{in}}(U)) = f(\delta^{\text{out}}(U)) \leq c(\delta^{\text{out}}(U)).$$

To see sufficiency, choose a function f satisfying $d \leq f \leq c$ and minimizing $\|\text{excess}_f\|_1$. Let $S := \{v \in V \mid \text{excess}_f(v) > 0\}$ and $T := \{v \in V \mid \text{excess}_f(v) < 0\}$. Suppose that $S \neq \emptyset$. Let $D_f = (V, A_f)$ be the residual graph (defined in (10.9)). If D_f contains an $S - T$ path P, we can modify f along P so as to reduce $\|\text{excess}_f\|_1$. So D_f contains no $S - T$ path. Let U be the set of vertices reachable in D_f from S. Then for each $a \in \delta_A^{\text{out}}(U)$ we have $a \notin A_f$ and hence $f(a) = c(a)$. Similarly, for each $a \in \delta_A^{\text{in}}(U)$ we have $a^{-1} \notin A_f$ and hence $f(a) = d(a)$. Therefore,

(11.6) $$d(\delta^{\text{in}}(U)) - c(\delta^{\text{out}}(U)) = f(\delta^{\text{in}}(U)) - f(\delta^{\text{out}}(U)) = \text{excess}_f(U) = \text{excess}_f(S) > 0,$$

[19] In fact, A.J. Hoffman [1960] attributes it to A.H. Hoffman, but this is a misprint (A.J. Hoffman, personal communication 1995).

contradicting (11.4). ∎

(One can derive this theorem also from the max-flow min-cut theorem, with the methods described in Section 11.6 below.)

Theorem 11.2 implies that any circulation can be rounded:

Corollary 11.2a. *Let $D = (V, A)$ be a digraph and let $f : A \to \mathbb{R}$ be a circulation. Then there exists an integer circulation f' with $\lfloor f(a) \rfloor \leq f'(a) \leq \lceil f(a) \rceil$ for each arc a.*

Proof. Take $d := \lfloor f \rfloor$ and $c := \lceil f \rceil$ in Theorem 11.2. ∎

Another consequence is:

Corollary 11.2b. *Let $D = (V, A)$ be a digraph, let $k \in \mathbb{Z}_+$ (with $k \geq 1$), and let $f : A \to \mathbb{Z}$ be a circulation. Then $f = f_1 + \cdots + f_k$ where each f_i is an integer circulation satisfying*

(11.7) $\lfloor \frac{1}{k} f \rfloor \leq f_i \leq \lceil \frac{1}{k} f \rceil$.

Proof. By induction on k. Define $d := \lfloor \frac{1}{k} f \rfloor$ and $c := \lceil \frac{1}{k} f \rceil$. It suffices to show that there exists an integer circulation f_k such that

(11.8) $d \leq f_k \leq c$ and $(k-1)d \leq f - f_k \leq (k-1)c$,

equivalently,

(11.9) $$\max\{d(a), f(a) - (k-1)c(a)\} \leq f_k(a) \leq \min\{c(a), f(a) - (k-1)d(a)\}$$

for each $a \in A$. Since these bounds are integer, by Corollary 11.2a it suffices to show that there is any circulation obeying these bounds. For that we can take $\frac{1}{k} f$. ∎

This corollary implies that the set of circulations f satisfying $d \leq f \leq c$ for some integer bounds d, c, has the integer decomposition property.

11.3. Flows with upper and lower bounds

We can derive from Corollary 11.2a that flows can be rounded similarly:

Corollary 11.2c. *Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let f be an $s-t$ flow of value $k \in \mathbb{Z}$. Then there exists an integer $s-t$ flow f' of value k with $\lfloor f(a) \rfloor \leq f'(a) \leq \lceil f(a) \rceil$ for each arc a.*

Proof. Add an arc (t, s) and define $f(t, s) := k$. We obtain a circulation to which we can apply Corollary 11.2a. Deleting the new arc, we obtain an $s-t$ flow as required. ∎

According to Berge [1958b], A.J. Hoffman showed the following on the existence of a flow obeying both an upper bound (capacity) and a lower bound (demand):

Corollary 11.2d. *Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let $d, c : A \to \mathbb{R}_+$ with $d \leq c$. Then there exists an $s-t$ flow f with $d \leq f \leq c$ if and only if*

$$(11.10) \qquad c(\delta^{\text{out}}(U)) \geq d(\delta^{\text{in}}(U))$$

for each $U \subseteq V$ not separating s and t. If moreover d and c are integer, f can be taken integer.

Proof. Necessity being direct, we show sufficiency. Identify s and t. By Theorem 11.2, there exists a circulation in the shrunk network. This gives a flow as required in the original network. ∎

(Berge [1958b] wrote that this result was shown by Hoffman with linear programming techniques, and was reduced to network theory by L.R. Ford, Jr.)

Moreover, a min-max relation for the maximum value of a flow obeying upper and lower bounds can be derived:

Corollary 11.2e. *Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let $d, c : A \to \mathbb{R}_+$ with $d \leq c$, such that there exists an $s-t$ flow f with $d \leq f \leq c$. Then the maximum value of an $s-t$ flow f with $d \leq f \leq c$ is equal to the minimum value of*

$$(11.11) \qquad c(\delta^{\text{out}}(U)) - d(\delta^{\text{in}}(U))$$

taken over $U \subseteq V$ with $s \in U$ and $t \notin U$. If d and c are integer, the maximum is attained by an integer flow f.

Proof. Let μ be the minimum value of (11.11). Add to D an arc (t, s), with $d(t, s) = c(t, s) = \mu$. Then the extended network has a circulation by Theorem 11.2. Indeed, condition (11.4) for U not separating s and t follows from (11.10). If $s \in U$, $t \notin U$, (11.4) follows from the definition of μ. If $s \notin U$, $t \in U$, then $\mu \geq \text{value}(f)$. Hence

$$(11.12) \qquad \begin{aligned} &\mu + c(\delta^{\text{out}}(U)) - d(\delta^{\text{in}}(U)) \geq \mu + f(\delta^{\text{out}}(U)) - f(\delta^{\text{in}}(U)) \\ &= \mu - \text{value}(f) \geq 0. \end{aligned}$$

Therefore, the original network has a flow as required. ∎

11.4. b-transshipments

Let $D = (V, A)$ be a digraph and let $b \in \mathbb{R}^V$. A function $f : A \to \mathbb{R}$ is called a *b-transshipment* if $\text{excess}_f = b$. (So each function $f : A \to \mathbb{R}$ is a

b-transshipment for some b. excess_f is defined in Section 10.1.) By reduction to Hoffman's circulation theorem, one may characterize the existence of a b-transshipment obeying given upper and lower bounds on the arcs:

Corollary 11.2f. *Let $D = (V, A)$ be a digraph, let $d, c : A \to \mathbb{R}$ with $d \leq c$, and let $b : V \to \mathbb{R}$ with $b(V) = 0$. Then there exists a b-transshipment f with $d \leq f \leq c$ if and only if*

(11.13) $\quad c(\delta^{\text{in}}(U)) - d(\delta^{\text{out}}(U)) \geq b(U)$

for each $U \subseteq V$. If moreover b, c, and d are integer, f can be taken integer.

Proof. The corollary can be reduced to Hoffman's circulation theorem (Theorem 11.2). Add a new vertex u, and for each $v \in V$ an arc (v, u) with $d(v, u) := c(v, u) := b(v)$. Then a function f as required exists if and only if the extended graph has a circulation f' satisfying $d \leq f' \leq c$. The condition in Hoffman's circulation theorem is equivalent to (11.13). ∎

Conversely, Hoffman's circulation theorem is the special case $b = \mathbf{0}$. The special case $d = \mathbf{0}$ is the following result of Gale [1956,1957]:

Corollary 11.2g (Gale's theorem)**.** *Let $D = (V, A)$ be a digraph and let $c : A \to \mathbb{R}$ and $b : V \to \mathbb{R}$ with $b(V) = 0$. Then there exists a b-transshipment f satisfying $\mathbf{0} \leq f \leq c$ if and only if*

(11.14) $\quad c(\delta^{\text{in}}(U)) \geq b(U)$

for each $U \subseteq V$. If moreover b and c are integer, f can be taken integer.

Proof. Take $d = \mathbf{0}$ in Corollary 11.2f. ∎

The proof of Gale is by reduction to the max-flow min-cut theorem. Conversely, the max-flow min-cut theorem follows easily: if $s, t \in V$ and ϕ is the minimum capacity of an $s - t$ cut, then set $b(s) := -\phi$, $b(t) := \phi$, and $b(v) := 0$ for each $v \neq s, t$. As (11.14) is satisfied, by Gale's theorem there exists a b-transshipment f with $\mathbf{0} \leq f \leq c$. This is an $s - t$ flow of value ϕ.

Taking $c = \boldsymbol{\infty}$ in Gale's theorem gives the following result of Rado [1943]:

Corollary 11.2h. *Let $D = (V, A)$ be a digraph and let $b : V \to \mathbb{R}$ with $b(V) = 0$. Then there exists a b-transshipment $f \geq \mathbf{0}$ if and only if $b(U) \leq 0$ for each $U \subseteq V$ with $\delta^{\text{in}}(U) = \emptyset$.*

Proof. This is Gale's theorem (Corollary 11.2g) for $c = \boldsymbol{\infty}$. ∎

11.5. Upper and lower bounds on excess_f

Instead of equality constraints on excess_f one may put upper and lower bounds b and a. This has the following characterization:

Corollary 11.2i. *Let $D = (V, A)$ be a digraph, let $d, c : A \to \mathbb{R}$ with $d \le c$, and let $a, b : V \to \mathbb{R}$ with $a \le b$. Then there exists a z-transshipment f with $d \le f \le c$ for some z with $a \le z \le b$ if and only if*

(11.15) $$c(\delta^{\text{in}}(U)) - d(\delta^{\text{out}}(U)) \ge \max\{a(U), -b(V \setminus U)\}$$

for each $U \subseteq V$. If moreover a, b, c, and d are integer, f can be taken integer.

Proof. The corollary can be reduced to Hoffman's circulation theorem: Add a new vertex u, and for each $v \in V$ an arc (v, u) with $d(v, u) := a(v)$ and $c(v, u) := b(v)$. Then a function f as required exists if and only if the extended graph has a circulation f' satisfying $d \le f' \le c$. ∎

This characterization can be formulated equivalently as:

Corollary 11.2j. *Let $D = (V, A)$ be a digraph, let $d, c : A \to \mathbb{R}$ with $d \le c$, and let $a, b : V \to \mathbb{R}$ with $a \le b$. Then there exists a z-transshipment f with $d \le f \le c$ for some z with $a \le z \le b$ if and only if there exists a z-transshipment f' with $d \le f' \le c$ for some $z \ge a$ and there exists a z-transshipment f'' with $d \le f'' \le c$ for some $z \le b$.*

Proof. Directly from Corollary 11.2i, since (11.15) can be split into a condition on a and one on b. ∎

For $d = \mathbf{0}$, Corollary 11.2i gives a result of Fulkerson [1959a]:

Corollary 11.2k. *Let $D = (V, A)$ be a digraph, let $c : A \to \mathbb{R}_+$, and let $a, b : V \to \mathbb{R}$ with $a \le b$. Then there exists a z-transshipment f satisfying $\mathbf{0} \le f \le c$, for some z with $a \le z \le b$ if and only if*

(11.16) $$c(\delta^{\text{in}}(U)) \ge \max\{a(U), -b(V \setminus U)\}$$

for each $U \subseteq V$. If moreover a, b, and c are integer, f can be taken integer.

Proof. Directly from Corollary 11.2i, taking $d = \mathbf{0}$. ∎

11.6. Finding circulations and transshipments algorithmically

Algorithmic and complexity results for circulations and transshipments follow directly from those for the maximum flow problem, by the following construction.

Let $D = (V, A)$ be a digraph and let $d, c : A \to \mathbb{Q}$ with $d \le c$. Then a circulation f satisfying $d \le f \le c$ can be found as follows. Give each arc a a new capacity

(11.17) $$c'(a) := c(a) - d(a).$$

Add two new vertices s and t. For each $v \in V$ with $\text{excess}_d(v) > 0$, add an arc (s, v) with capacity $c'(s, v) := \text{excess}_d(v)$. For each $v \in V$ with $\text{excess}_d(v) < 0$, add an arc (v, t) with capacity $c'(v, t) := -\text{excess}_d(v)$. This makes the extended graph $D' = (V', A')$.

Then D has a circulation f satisfying $d \leq f \leq c$ if and only if D' has an $s-t$ flow $f' \leq c'$ of value

$$\sum_{\substack{v \in V \\ \text{excess}_d(v) > 0}} \text{excess}_d(v) \tag{11.18}$$

(by taking $f(a) = f'(a) + d(a)$ for each $a \in A$).

This yields:

Theorem 11.3. *If a maximum flow can be found in time* $\text{MF}(n, m)$*, then a circulation can be found in time* $O(\text{MF}(n, m))$*.*

Proof. Apply the above construction. ∎

If, in addition, functions $a, b : V \to \mathbb{Q}$ are given, we can reduce the problem of finding a transshipment f satisfying $d \leq f \leq c$ and $a \leq \text{excess}_f \leq b$ to finding a circulation in a slightly larger graph — see the proof of Corollary 11.2i. This gives:

Corollary 11.3a. *If a maximum flow can be found in time* $\text{MF}(n, m)$*, then (given* a, b, d, c*) a* z*-transshipment* f *satisfying* $d \leq f \leq c$ *and* $a \leq z \leq b$ *can be found in time* $O(\text{MF}(n, m))$*.*

Proof. Reduce the problem with the construction of Corollary 11.2i to the circulation problem, and use Theorem 11.3. ∎

11.6a. Further notes

The results on flows, circulations, and transshipments extend directly to the case where also each vertex has an upper and/or lower bound on the amount of flow traversing that vertex. We can reduce this to the cases considered above by splitting any vertex v into two vertices v' and v'', adding an arc from v' to v'' with bounds equal to the vertex bounds, and replacing any arc (u, v) by (u'', v').

The results of this chapter also apply to characterizing the existence of a subgraph $D' = (V, A')$ of a given graph $D = (V, A)$, where D' has prescribed bounds on the indegrees and outdegrees (cf. Hakimi [1965]).

Chapter 12

Minimum-cost flows and circulations

Minimum-cost flows can be seen to generalize both shortest path and maximum flow. A shortest $s - t$ path can be deduced from a minimum-cost $s - t$ flow of value 1, while a maximum $s - t$ flow is a minimum-cost $s - t$ flow if we take cost -1 on arcs leaving s and 0 on all other arcs (assuming no arc enters s).
Minimum-cost flows, circulations, and transshipments are closely related, and when describing algorithms, we will choose the most suitable variant.
In this chapter, graphs can be assumed to be simple.

12.1. Minimum-cost flows and circulations

Let $D = (V, A)$ be a digraph and let $k : A \to \mathbb{R}$, called the *cost* function. For any function $f : A \to \mathbb{R}$, the *cost* of f is, by definition,

$$\text{(12.1)} \qquad \text{cost}(f) := \sum_{a \in A} k(a)f(a).$$

The *minimum-cost $s - t$ flow problem* is: given a digraph $D = (V, A)$, $s, t \in V$, a 'capacity' function $c : A \to \mathbb{Q}_+$, a 'cost' function $k : A \to \mathbb{Q}$, and a value $\phi \in \mathbb{Q}_+$, find an $s - t$ flow $f \leq c$ of value ϕ that minimizes cost(f). This problem includes the problem of finding a maximum-value $s - t$ flow that has minimum cost among all maximum-value $s - t$ flows.

Related is the *minimum-cost circulation problem:* given a digraph $D = (V, A)$, a 'demand' function $d : A \to \mathbb{Q}$, a 'capacity' function $c : A \to \mathbb{Q}$, and a 'cost' function $k : A \to \mathbb{Q}$, find a circulation f subject to $d \leq f \leq c$, minimizing cost(f).

One can easily reduce the minimum-cost flow problem to the minimum-cost circulation problem: just add an arc $a_0 := (t, s)$ with $d(a_0) := c(a_0) = \phi$ and $k(a_0) := 0$. Also, let $d(a) := 0$ for each arc $a \neq a_0$. Then a minimum-cost circulation in the extended digraph gives a minimum-cost flow of value ϕ in the original digraph.

Also the problem of finding a maximum-value $s - t$ flow can be reduced easily to a minimum-cost circulation problem in the extended digraph: now define $d(a_0) := 0$, $c(a_0) := \infty$, and $k(a_0) := -1$. Moreover, set $k(a) := 0$ for each $a \neq a_0$. Then a minimum-cost circulation gives a maximum-value $s-t$ flow.

Edmonds and Karp [1970,1972] showed that the minimum-cost circulation problem is solvable in polynomial time. Their algorithm is based on a technique called *capacity-scaling* and can be implemented to run in $O(m(m + n \log n) \log C)$ time, where $C := \|c\|_\infty$ (assuming c integer). So it is weakly polynomial-time. They raised the question of the existence of a *strongly* polynomial-time algorithm.

Tardos [1985a] answered this question positively. Her algorithm has resulted in a stream of further research on strongly polynomial-time algorithms for the minimum-cost circulation problem. It stood at the basis of the strongly polynomial-time algorithms discussed in this chapter.

12.2. Minimum-cost circulations and the residual graph D_f

It will be useful again to consider the residual graph $D_f = (V, A_f)$ of $f : A \to \mathbb{R}$ (with respect to d and c), where

(12.2) $$A_f := \{a \mid a \in A, f(a) < c(a)\} \cup \{a^{-1} \mid a \in A, f(a) > d(a)\}.$$

Here $a^{-1} := (v, u)$ if $a = (u, v)$.

We extend any cost function k to A^{-1} by defining the *cost* $k(a^{-1})$ of a^{-1} by:

(12.3) $$k(a^{-1}) := -k(a)$$

for each $a \in A$.

We also use the following notation. Any directed circuit C in D_f gives an undirected circuit in $D = (V, A)$. We define $\chi^C \in \mathbb{R}^A$ by:

(12.4) $$\chi^C(a) := \begin{cases} 1 & \text{if } C \text{ traverses } a, \\ -1 & \text{if } C \text{ traverses } a^{-1}, \\ 0 & \text{if } C \text{ traverses neither } a \text{ nor } a^{-1}, \end{cases}$$

for $a \in A$.

Given $D = (V, A)$, $d, c : A \to \mathbb{R}$, a circulation f in D is called *feasible* if $d \leq f \leq c$. The following observation is fundamental[20]:

Theorem 12.1. *Let $D = (V, A)$ be a digraph and let $d, c, k : A \to \mathbb{R}$. Let $f : A \to \mathbb{R}$ be a feasible circulation. Then f has minimum cost among all feasible circulations if and only if each directed circuit of D_f has nonnegative cost.*

[20] The idea goes back to Tolstoǐ [1930,1939] (for the transportation problem), and was observed also by Robinson [1949,1950] (for the transportation problem), Gallai [1957, 1958b], Busacker and Gowen [1960], Fulkerson [1961], and Klein [1967].

Proof. *Necessity.* Let C be a directed circuit in D_f of negative cost. Then for small enough $\varepsilon > 0$, $f' := f + \varepsilon\chi^C$ is again a circulation satisfying $d \leq f' \leq c$. Since $\text{cost}(f') < \text{cost}(f)$, f is not minimum-cost.

Sufficiency. Suppose that each directed circuit in D_f has nonnegative cost. Let f' be any feasible circulation. Then $f' - f$ is a circulation, and hence

$$f' - f = \sum_{j=1}^{m} \lambda_j \chi^{C_j} \tag{12.5}$$

for some directed circuits $C_1, \ldots, C_m$ in D_f and $\lambda_1, \ldots, \lambda_m > 0$. Hence

$$\text{cost}(f') - \text{cost}(f) = \text{cost}(f' - f) = \sum_{j=1}^{m} \lambda_j k(C_j) \geq 0. \tag{12.6}$$

So $\text{cost}(f') \geq \text{cost}(f)$. ∎

This directly implies a strong result: optimality of a given feasible circulation f can be checked in polynomial time, namely in time $O(nm)$ (with the Bellman-Ford method). It also implies the following good characterization (Gallai [1957,1958b], Ford and Fulkerson [1962]; for the simpler case of a symmetric cost function satisfying the triangle inequality it was shown by Kantorovich [1942], Kantorovich and Gavurin [1949], and Koopmans and Reiter [1951]):

Corollary 12.1a. *Let $D = (V, A)$ be a digraph, let $d, c, k : A \to \mathbb{R}$, and let f be a feasible circulation. Then f is minimum-cost if and only if there exists a function $p : V \to \mathbb{R}$ such that*

$$\begin{aligned} k(a) &\geq p(v) - p(u) \ \text{ if } f(a) < c(a), \\ k(a) &\leq p(v) - p(u) \ \text{ if } f(a) > d(a), \end{aligned} \tag{12.7}$$

for each arc $a = (u, v) \in A$.

Proof. Directly from Theorem 12.1 with Theorem 8.2. ∎

From this characterization, a min-max relation for minimum-cost circulations can be derived — see Section 12.5b. It also follows directly from the duality theorem of linear programming — see Chapter 13.

12.3. Strongly polynomial-time algorithm

Theorem 12.1 gives us a method to improve a given circulation f:

(12.8) Choose a negative-cost directed circuit C in the residual graph D_f, and reset $f := f + \tau\chi^C$ where τ is maximal subject to $d \leq f \leq c$. If no such directed circuit exists, f is a minimum-cost circulation.

It is not difficult to see that for rational data this leads to a finite algorithm.

However, if we just select circuits in an arbitrary fashion, the algorithm may take exponential time, as follows by application to the maximum flow problem given in Figure 10.1 (by adding an arc from t to s of cost -1). Zadeh [1973a,1973b] showed that several strategies of selecting a circuit do not lead to a strongly polynomial-time algorithm.

Goldberg and Tarjan [1988b,1989] were able to prove that one obtains a strongly polynomial-time algorithm if one chooses in (12.8) a directed circuit C of minimum mean cost, that is, one minimizing

(12.9) $$\frac{k(C)}{|C|}.$$

(We identify C with its set AC of arcs.) In Corollary 8.10a we saw that such a circuit can be found in time $O(nm)$.

Note that if we formulate a maximum $s-t$ flow problem as a minimum-cost circulation problem, by adding an arc (t,s) of cost -1 and capacity $+\infty$, then the minimum-mean cost cycle-cancelling algorithm reduces to the shortest augmenting path method of Dinits [1970] and Edmonds and Karp [1972] (Section 10.5).

We now prove the result of Goldberg and Tarjan (as usual, $n := |V|$, $m := |A|$):

Theorem 12.2. *Choosing a minimum-mean cost cycle C in* (12.8)*, the number of iterations is at most* $4nm^2\lceil \ln n\rceil$.

Proof. Let $f_0, f_1, \ldots$ be the circulations found. For each $i \geq 0$, define $A_i := A_{f_i}$, let ε_i be *minus* the minimum of (12.9) in (V, A_i), and let C_i be the directed circuit in A_i chosen to obtain f_{i+1} (taking circuits as arc sets). So

(12.10) $$k(C_i) = -\varepsilon_i |C_i|.$$

ε_i is the smallest value such that if we would add ε_i to the cost of each arc of A_i, then each directed circuit has nonnegative cost. So ε_i is the smallest value for which there exists a function $p_i : V \to \mathbb{Q}$ such that

(12.11) $$k(a) + \varepsilon_i \geq p_i(v) - p_i(u) \text{ for each } a = (u,v) \in A_i.$$

The proof of the theorem is based on the following two facts on the decrease of the ε_i:

(12.12) $$\text{(i) } \varepsilon_{i+1} \leq \varepsilon_i \text{ and (ii) } \varepsilon_{i+m} \leq (1 - \tfrac{1}{n})\varepsilon_i$$

(assuming in (ii) that we reach iteration $i+m$). To prove (12.12), we may assume that $i = 0$ and $p_0 = \mathbf{0}$. Then $k(a) \geq -\varepsilon_0$ for each $a \in A_0$, with equality if $a \in C_0$.

Since $A_1 \subseteq A_0 \cup C_0^{-1}$ and since each arc in C_0^{-1} has cost $\varepsilon_0 \geq 0$, we know that $k(a) \geq -\varepsilon_0$ for each $a \in A_1$. Hence $\varepsilon_1 \leq \varepsilon_0$. This proves (12.12)(i).

To prove (ii), we may assume $\varepsilon_m > 0$. We first show that at least one of the directed circuits $C_0, \ldots, C_{m-1}$ contains an arc a with $k(a) \geq 0$. Otherwise each A_h arises from A_{h-1} by deleting at least one negative-cost arc and adding only positive-cost arcs. This implies that A_m contains no negative-cost arc, and hence f_m has minimum cost; so $\varepsilon_m = 0$, contradicting our assumption.

Let h be the smallest index such that C_h contains an arc a with $k(a) \geq 0$. So all negative-cost arcs in C_h also belong to A_0, and hence have cost at least $-\varepsilon_0$. So $k(C_h) \geq -(|C_h|-1)\varepsilon_0$ and therefore $\varepsilon_h = -k(C_h)/|C_h| \leq (1-\frac{1}{n})\varepsilon_0$. This proves (12.12).

Now define

(12.13) $\quad t := 2nm\lceil \ln n \rceil.$

Then by (12.12)(ii):

(12.14) $\quad \varepsilon_t \leq (1-\dfrac{1}{n})^{2n\lceil \ln n\rceil}\varepsilon_0 < \varepsilon_0/2n,$

since $(1-\frac{1}{n})^n < e^{-1}$ and $e^{-2\ln n} = n^{-2} \leq \frac{1}{2n}$.

We finally show that for each i there exists an arc a in C_i such that $a \notin C_h$ for each $h \geq i+t$. Since $|A \cup A^{-1}| = 2m$, this implies that the number of iterations is at most $4mt$, as required.

To prove this, we may assume that $i = 0$ and that $p_t = \mathbf{0}$. As $k(C_0) = -\varepsilon_0|C_0|$, C_0 contains an arc a_0 with $k(a_0) \leq -\varepsilon_0 < -2n\varepsilon_t$. Without loss of generality, $a_0 \in A$.

Suppose that $f_h(a_0) \neq f_t(a_0)$ for some $h > t$. Since $k(a_0) < -2n\varepsilon_t \leq -\varepsilon_t$, we have that $a_0 \notin A_t$ (by (12.11)). So $f_t(a_0) = c(a_0)$, and hence $f_h(a_0) < f_t(a_0)$. Then, by (11.3) applied to $f_t - f_h$, A_h has a directed circuit C containing a_0 such that A_t contains C^{-1}. By (12.11), $-k(a) = k(a^{-1}) \geq -\varepsilon_t$ for each $a \in C$. This gives (using (12.12)(i)):

(12.15) $$k(C) = k(a_0) + k(C \setminus \{a_0\}) < -2n\varepsilon_t + (|C|-1)\varepsilon_t \leq -n\varepsilon_t \leq -n\varepsilon_h \leq -|C|\varepsilon_h,$$

contradicting the definition of ε_h. ∎

This gives for finding a minimum-cost circulation:

Corollary 12.2a. *A minimum-cost circulation can be found in $O(n^2m^3\log n)$ time. If d and c are integer, an integer minimum-cost circulation is found.*

Proof. Directly from Theorem 12.2 and Corollary 8.10a. Note that if d and c are integer and we start with an integer circulation f_0, all further circulations obtained by (12.8) are integer. ∎

So we have the theorem of Tardos [1985a]:

Corollary 12.2b. *A minimum-cost circulation can be found in strongly polynomial time. If d and c are integer, an integer circulation is found.*

Proof. Directly from Corollary 12.2a. ∎

Notes. Goldberg and Tarjan [1988b,1989] showed that, with the help of dynamic trees, the running time of the minimum-mean cost cycle-cancelling method can be improved to $O(nm \log n \min\{\log(nK), m \log n\})$, where $K := \|k\|_\infty$, assuming k to be integer.

Weintraub [1974] showed that if we take always a directed circuit C in D_f such that, by resetting f to $f + \tau\chi^C$ as in (12.8), the cost decreases most, then the number of iterations (12.8) is polynomially bounded. However, finding such a circuit is NP-complete (finding a Hamiltonian circuit in a directed graph is a special case). Weintraub [1974] also proposed a heuristic of finding a short (negative) circuit by finding a minimum-cost set of vertex-disjoint circuits in D_f (by solving an assignment problem), and choosing the shortest among them. Barahona and Tardos [1989] showed that this also leads to a (weakly) polynomial-time algorithm.

12.4. Related problems

Corollary 12.2b concerns solving the minimization problem in the following LP-duality equation, where M denotes the $V \times A$ incidence matrix of D:

(12.16) $$\min\{k^\mathsf{T} x \mid d \leq x \leq c, Mx = \mathbf{0}\}$$
$$= \max\{z_1^\mathsf{T} d - z_2^\mathsf{T} c \mid z_1, z_2 \geq \mathbf{0}, \exists y : z_1^\mathsf{T} - z_2^\mathsf{T} + y^\mathsf{T} M = k^\mathsf{T}\}.$$

It implies that also the maximization problem can be solved in strongly polynomial time:

Corollary 12.2c. *The maximization problem in* (12.16) *can be solved in strongly polynomial time. If k is integer, an integer optimum solution is found.*

Proof. Let x be an optimum solution of the minimization problem, that is, a minimum-cost circulation. Since x is extreme, the digraph D_x has no negative-cost directed circuits. Hence we can find a function ('potential') $y : V \to \mathbb{Q}$ such that $k(u,v) \geq y(v) - y(u)$ if $x(u,v) < c(u,v)$ and $k(u,v) \leq y(v) - y(u)$ if $x(u,v) > d(u,v)$, in strongly polynomial time (Theorem 8.7). If k is integer, we find an integer y.

Let z_1 and z_2 be the unique vectors with $z_1, z_2 \geq \mathbf{0}$, $z_1^\mathsf{T} - z_2^\mathsf{T} = k^\mathsf{T} - y^\mathsf{T} M$ and $z_1(a) z_2(a) = 0$ for each $a \in A$. So $z_1(a) = 0$ if $x(a) > d(a)$ and $z_2(a) = 0$ if $x(a) < c(a)$. Hence

(12.17) $$z_1^\mathsf{T} d - z_2^\mathsf{T} c = z_1^\mathsf{T} x - z_2^\mathsf{T} x = (k^\mathsf{T} - y^\mathsf{T} M)x = k^\mathsf{T} x.$$

So z_1, z_2 form an optimum solution for the maximization problem. ∎

By an easy construction, Corollary 12.2b implies that a more general problem is solvable in strongly polynomial time:

(12.18) input: a digraph $D = (V, A)$ and functions $a, b : V \to \mathbb{Q}$ and $d, c, k : A \to \mathbb{Q}$, find: a z-transshipment x with $a \le z \le b$ and $d \le x \le c$, minimizing $k^{\mathsf{T}} x$.

Corollary 12.2d. *Problem* (12.18) *is solvable in strongly polynomial time. If a, b, d, and c are integer, an integer optimum solution is found.*

Proof. Extend D by a new vertex u and arcs (v,u) for each $v \in V$. Extend d, c, and k by defining $d(v,u) := a(v)$, $c(v,u) := b(v)$ and $k(v,u) := 0$ for each $v \in V$. By Corollary 12.2b, we can find a minimum-cost circulation x with $d \le x \le c$ in the extended digraph in strongly polynomial time. It gives a b-transshipment in the original graph as required. ∎

For later reference, we derive that also the dual problem (in the LP sense) can be solved in strongly polynomial time. If M denotes $V \times A$ incidence matrix of D, problem (12.18) corresponds to the minimum in the LP-duality equation:

(12.19)
$$\begin{aligned} &\min\{k^{\mathsf{T}} x \mid d \le x \le c, a \le Mx \le b\} \\ &= \max\{y_1^{\mathsf{T}} b - y_2^{\mathsf{T}} a + z_1^{\mathsf{T}} d - z_2^{\mathsf{T}} c \mid y_1, y_2, z_1, z_2 \ge \mathbf{0}, \\ &\qquad\qquad (y_1 - y_2)^{\mathsf{T}} M + (z_1 - z_2)^{\mathsf{T}} = k^{\mathsf{T}}\}. \end{aligned}$$

Corollary 12.2e. *An optimum solution for the maximum in* (12.19) *can be found in strongly polynomial time. If k is integer, we find an integer optimum solution.*

Proof. By reduction to Corollary 12.2c, using a reduction similar to that given in the proof of Corollary 12.2d. ∎

12.4a. A dual approach

The approach above consists of keeping a feasible circulation, and throughout improving its cost. A dual approach can best be described in terms of b-transshipments: we keep a b'-transshipment f such that D_f has no negative-cost directed circuits, and improve b' until $b' = b$. This can be studied with the concept of 'extreme function'.

Let $D = (V, A)$ be a digraph and let $d, c, k : A \to \mathbb{R}$ be given, the lower bound function, the capacity function, and the cost function, respectively. Let $f : A \to \mathbb{R}$ be such that $d \le f \le c$. We call f *extreme* if $\text{cost}(f') \ge \text{cost}(f)$ for each function f' satisfying $d \le f' \le c$ and $\text{excess}_{f'} = \text{excess}_f$; in other words, setting $b := \text{excess}_f$, f is a minimum-cost b-transshipment subject to $d \le f \le c$. (excess_f is defined in Section 10.1.)

Note that the concept of extreme depends on k, d, and c. So it might be better to define a function to be extreme *with respect to* k, d, and c. However, when considering extreme functions f, the functions k, d, and c are generally fixed, or follow from the context.

Again it will be useful to consider the residual graph $D_f = (V, A_f)$ of f (with respect to d and c), where

(12.20) $A_f := \{a \mid a \in A, f(a) < c(a)\} \cup \{a^{-1} \mid a \in A, f(a) > d(a)\}.$

Here $a^{-1} := (v, u)$ if $a = (u, v)$.

We extend k to $A^{-1} := \{a^{-1} \mid a \in A\}$ by defining

(12.21) $k(a^{-1}) := -k(a)$

for each $a \in A$. We call $k(a^{-1})$ the *cost* of a^{-1}.

We also use the following notation. Any directed path P in D_f gives an undirected path in $D = (V, A)$. We define $\chi^P \in \mathbb{R}^A$ by:

(12.22) $$\chi^P(a) := \begin{cases} 1 & \text{if } P \text{ traverses } a, \\ -1 & \text{if } P \text{ traverses } a^{-1}, \\ 0 & \text{if } P \text{ traverses neither } a \text{ nor } a^{-1}, \end{cases}$$

for $a \in A$.

Theorem 12.1 for minimum-cost circulations can be directly extended to extreme functions:

Theorem 12.3. *Let $D = (V, A)$ be a digraph and let $d, c, k, f : A \to \mathbb{R}$ with $d \leq f \leq c$. Then f is extreme if and only if each directed circuit of D_f has nonnegative cost.*

Proof. Like the proof of Theorem 12.1. ∎

This implies that the optimality of a given feasible solution f of a b-transshipment problem can be checked in polynomial time, namely in time $O(nm)$ (with the Bellman-Ford method). It also implies the following good characterization (Kantorovich [1942], Gallai [1957,1958b], Ford and Fulkerson [1962]):

Corollary 12.3a. *Let $D = (V, A)$ be a digraph and let $d, c, k, f : A \to \mathbb{R}$ with $d \leq f \leq c$. Then f is extreme if and only if there exists a function $p : V \to \mathbb{R}$ such that*

(12.23) $$\begin{aligned} &k(a) \geq p(v) - p(u) \text{ if } f(a) < c(a), \\ &k(a) \leq p(v) - p(u) \text{ if } f(a) > d(a), \end{aligned}$$

for each arc $a = (u, v) \in A$.

Proof. Directly from Theorem 12.3 with Theorem 8.2. ∎

As for the algorithmic side, the following observation (Jewell [1958], Busacker and Gowen [1960], Iri [1960]) is very useful in analyzing algorithms ('This theorem may properly be regarded as the central one concerning minimal cost flows' — Ford and Fulkerson [1962]):

Theorem 12.4. *Let $D = (V, A)$ be a digraph and let $d, c, k, f : A \to \mathbb{R}$ with $d \leq f \leq c$ and with f extreme. Let P be a minimum-cost $s-t$ path in D_f, for some $s, t \in V$, and let $\varepsilon > 0$ be such that $f' := f + \varepsilon\chi^P$ satisfies $d \leq f' \leq c$. Then f' is extreme again.*

Proof. Let f'' satisfy $d \leq f'' \leq c$ and $\text{excess}_{f''} = \text{excess}_{f'}$. Then by Theorem 11.1,

$$f'' - f = \sum_{i=1}^{n} \mu_i \chi^{P_i} + \sum_{j=1}^{m} \lambda_j \chi^{C_j}, \tag{12.24}$$

where $P_1, \ldots, P_n$ are $s-t$ paths in D_f, $C_1, \ldots, C_m$ are directed circuits in D_f, $\mu_1, \ldots, \mu_n > 0$, and $\lambda_1, \ldots, \lambda_m > 0$, with $\sum_i \mu_i = \varepsilon$. Then

$$\text{cost}(f'' - f) = \sum_i \mu_i \cdot \text{cost}(P_i) + \sum_j \lambda_j \cdot \text{cost}(C_j) \geq \sum_i \mu_i \tau = \varepsilon\tau, \tag{12.25}$$

where $\tau := \text{cost}(P)$. As $\text{cost}(f' - f) = \varepsilon\tau$, we have $\text{cost}(f'' - f) \geq \text{cost}(f' - f)$, and therefore $\text{cost}(f'') \geq \text{cost}(f')$. ∎

We will refer to updating f to $f + \varepsilon\chi^P$ as in Theorem 12.4 as to *sending a flow of value ε over P*.

Also the following observation is useful in algorithms (Edmonds and Karp [1970], Tomizawa [1971]):

Theorem 12.5. *In Theorem 12.4, if p is a potential for D_f such that $p(t) - p(s) = \text{dist}_k(s,t)$, then p is also a potential for $D_{f'}$.*

Proof. Choose $a = (u,v) \in A_{f'}$. If $a \in A_f$, then $p(v) \leq p(u) + k(a)$. If $a \notin A_f$, then a^{-1} is traversed by P, and hence $p(u) = p(v) + k(a^{-1}) = p(v) - k(a)$. Therefore $p(v) \leq p(u) + k(a)$. ∎

These theorems lead to the following minimum-cost $s-t$ flow algorithm due to Ford and Fulkerson [1958b], Jewell [1958], Busacker and Gowen [1960], and Iri [1960] (an equivalent 'primal-dual' algorithm was given by Fujisawa [1959]).

Let be given $D = (V, A)$, $s, t \in V$, and $c, k : A \to \mathbb{Q}_+$, the capacity and cost function, respectively.

Algorithm for minimum-cost $s-t$ flow
Starting with $f = \mathbf{0}$ apply the following iteratively:
Iteration: Let P be an $s-t$ path in D_f minimizing $k(P)$. Reset $f := f + \varepsilon\chi^P$, where ε is maximal subject to $\mathbf{0} \leq f + \varepsilon \cdot \chi^P \leq c$.

Termination of this algorithm for rational capacities follows similarly as for the maximum flow algorithm (Theorem 10.4).

One may use the Bellman-Ford method to obtain the path P, since D_f has no negative-cost directed circuits (by Theorems 12.3 and 12.4). However, using a trick of Edmonds and Karp [1970] and Tomizawa [1971], one can use Dijkstra's algorithm, since by Theorem 12.5 we can maintain a potential that makes all lengths (= costs) nonnegative. This leads to the following theorem (where $\text{SP}_+(n,m,K)$ denotes the time needed to find a shortest path in a digraph with n vertices, m arcs, and *nonnegative* integer lengths, each at most K):

Theorem 12.6. *For $c, k : A \to \mathbb{Z}_+$ and $\phi \in \mathbb{Z}_+$, a minimum-cost $s-t$ flow $f \leq c$ of value ϕ can be found in time $O(\phi \cdot \text{SP}_+(n,m,K))$, where $K := \|k\|_\infty$.*

Proof. Note that each iteration consists of finding a shortest path in D_f. Simultaneously we can find a potential for D_f satisfying $p(t) - p(s) = \text{dist}_k(s,t)$. Since by Theorem 12.5, p is a potential also for $D_{f'}$, we can perform each iteration in time $\text{SP}_+(n,m,K)$. ∎

12.4b. A strongly polynomial-time algorithm using capacity-scaling

The algorithm given in Theorem 12.6 is not polynomial-time, but several improvements leading to a polynomial-time algorithm have been found. Orlin [1988,1993] gave the currently fastest strongly polynomial-time algorithm for minimum-cost circulation, which is based on this dual approach: while keeping an extreme b'-transshipment, it throughout improves b', until $b' = b$. This implies an algorithm for minimum-cost circulation.

Let $D = (V, A)$ be a digraph, let $k : A \to \mathbb{Q}_+$ be a cost function, and let $b : V \to \mathbb{Q}$ be such that there exists a nonnegative b-transshipment. For any $f : A \to \mathbb{Q}$ define $\mathrm{def}_f : V \to \mathbb{Q}$ by

(12.26) $\qquad \mathrm{def}_f := b - \mathrm{excess}_f.$

So $\mathrm{def}_f(v)$ is the 'deficiency' of f at v. Then $\mathrm{def}_f(V) = b(V) - \mathrm{excess}_f(V) = 0$.

The algorithm determines a sequence of functions $f_i : A \to \mathbb{Q}_+$ and rationals β_i ($i = 0, 1, 2, \ldots$). Initially, set $f_0 := \mathbf{0}$ and $\beta_0 := \|b\|_\infty$. If f_i and β_i have been found, we find f_{i+1} and β_{i+1} by the following iteration (later referred to as *iteration* i).

Let A_i be the set of arcs a with $f_i(a) > 12n\beta_i$ and let $\mathcal{K}_i$ be the collection of weak components of the digraph (V, A_i). We are going to update a function g_i starting with $g_i := f_i$.

(12.27) If there exists a component $K \in \mathcal{K}_i$ and distinct $u, v \in K$ with $|\mathrm{def}_{g_i}(u)| \geq |\mathrm{def}_{g_i}(v)| > 0$, then update g_i by sending a flow of value $|\mathrm{def}_{g_i}(v)|$ from v to u or conversely along a path in A_i, so as to make $\mathrm{def}_{g_i}(v)$ equal to 0.

(In (12.32) it is shown that this is possible, and that it does not modify D_{g_i}, hence g_i remains extreme.) We iterate (12.27), so that finally each $K \in \mathcal{K}_i$ contains at most one vertex u with $\mathrm{def}_{g_i}(u) \neq 0$.

Next do the following repeatedly, as long as there exists a $u \in V$ with $|\mathrm{def}_{g_i}(u)| > \frac{n-1}{n}\beta_i$:

(12.28) If $\mathrm{def}_{g_i}(u) > \frac{n-1}{n}\beta_i$, then there exists a $v \in V$ such that $\mathrm{def}_{g_i}(v) < -\frac{1}{n}\beta_i$ and such that u reachable from v in the residual graph D_{g_i}. Update g_i by sending a flow of value β_i along a minimum-cost $v-u$ path in D_{g_i}.
If $\mathrm{def}_{g_i}(u) < -\frac{n-1}{n}\beta_i$, proceed symmetrically.

(The existence of v in (12.28) follows from the assumption that there exists a nonnegative b-transshipment, f say, by applying Theorem 11.1 to $f - g_i$. The fact that we can send a flow of value β_i in the residual graph follows from (12.36).)

When we cannot apply (12.27) anymore, we define $f_{i+1} := g_i$. Let $T := \|\mathrm{def}_{f_{i+1}}\|_\infty$. If $T = 0$ we stop. Otherwise, define:

(12.29) $$\beta_{i+1} := \begin{cases} \frac{1}{2}\beta_i & \text{if } T \geq \frac{1}{12n}\beta_i, \\ T & \text{if } 0 < T < \frac{1}{12n}\beta_i, \end{cases}$$

and iterate.

Theorem 12.7. *The algorithm stops after at most n iterations of* (12.27) *and at most $O(n \log n)$ iterations of* (12.28).

Proof. Throughout the proof we assume $n \geq 2$. We first observe that for each i:

(12.30) $\|\text{def}_{f_{i+1}}\|_\infty \leq \frac{n-1}{n}\beta_i,$

since otherwise we could have applied (12.28) to the final $g_i(= f_{i+1})$. This implies that for each i:

(12.31) $\|\text{def}_{f_i}\|_\infty \leq 2\beta_i.$

This is direct for $i = 0$. If $\beta_{i+1} = \frac{1}{2}\beta_i$, then, by (12.30), $\|\text{def}_{f_{i+1}}\|_\infty \leq \frac{n-1}{n}\beta_i \leq \beta_i = 2\beta_{i+1}$. If $\beta_{i+1} < \frac{1}{2}\beta_i$, then $\|\text{def}_{f_{i+1}}\|_\infty = T = \beta_{i+1} \leq 2\beta_{i+1}$. This proves (12.31).

We next show, that, for any i:

(12.32) in the iterations (12.27) and (12.28), for any $a \in A_i$ the value of $g_i(a)$ remains more than $6n\beta_i$.

In each iteration (12.27), for any arc $a \in A_i$, the value of $g_i(a)$ changes by at most $\|\text{def}_{f_i}\|_\infty$, which is at most $2\beta_i$ (by (12.31)). For any fixed i, we apply (12.27) at most n times. So the value of $g_i(a)$ on any arc $a \in A_i$ changes by at most $2n\beta_i$.

In the iterations (12.28), the value of $g_i(a)$ changes by at most $4n\beta_i$. To see this, consider the sum

(12.33) $$\sum_{\substack{v \in V \\ |\text{def}_{g_i}(v)| > \frac{n-1}{n}\beta_i}} |\text{def}_{g_i}(v)|.$$

In each iteration (12.28), this sum decreases by at least $\frac{n-1}{n}\beta_i$, which is at least $\frac{1}{2}\beta_i$. On the other hand, $g_i(a)$ changes by at most β_i. Since (12.33) initially is at most $\|\text{def}_{g_i}\|_1 \leq \|\text{def}_{f_i}\|_1 \leq 2n\beta_i$, we conclude that in the iterations (12.28), $g_i(a)$ changes by at most $4n\beta_i$.

Concluding, in the iterations (12.27) and (12.28), any $g_i(a)$ changes by at most $6n\beta_i$. Since at the beginning of these iterations we have $g_i(a) > 12n\beta_i$ for $a \in A_i$, this proves (12.32).

(12.32) implies that in iteration (12.27) we can make $\text{def}_{g_i}(v)$ equal to 0. (After that it will remain 0.) Hence, iteration (12.27) can be applied at most n times in total (over all i), since each time the number of vertices v with $\text{def}_{f_i}(v) \neq 0$ drops.

(12.32) also implies:

(12.34) each f_i is extreme.

This is clearly true if $i = 0$ (since the cost function k is nonnegative). Suppose that f_i is extreme. Then also g_i is extreme initially, and remains extreme during the iterations (12.27) (since by (12.32) the residual graph D_{g_i} does not change during the iterations (12.27)). Moreover, also during the iterations (12.28) the function g_i remains extreme, since we send flow over a minimum-cost path in D_{g_i} (Theorem 12.4). This proves (12.34).

Directly from (12.32) we have, for each i:

(12.35) $A_i \subseteq A_{i+1},$

since $\beta_{i+1} \leq \frac{1}{2}\beta_i$. This implies that each set in $\mathcal{K}_i$ is contained in some set in $\mathcal{K}_{i+1}$.

Next, throughout iteration i,

(12.36) If $a \in A \setminus A_i$, then $\beta_i | g_i(a)$.

The proof is by induction on i, the case $i = 0$ being trivial (since for $i = 0$ we do not apply (12.27), as $A_0 = \emptyset$). Suppose that we know (12.36). Choose $a \in A \setminus A_{i+1}$. Then $a \in A \setminus A_i$ by (12.35). Hence $\beta_i | g_i(a)$, and so $\beta_i | f_{i+1}(a)$. If $f_{i+1}(a) > 0$ and $\beta_{i+1} < \frac{1}{2}\beta_i$, then $\beta_i > 12nT = 12n\|\text{def}_{f_{i+1}}\|_\infty$, and hence

(12.37) $f_{i+1}(a) \geq \beta_i > 12n\|\mathrm{def}_{f_{i+1}}\|_\infty = 12n\beta_{i+1}$,

contradicting the fact that a does not belong to A_{i+1}.

So $f_{i+1}(a) = 0$ or $\beta_{i+1} = \frac{1}{2}\beta_i$. This implies that $\beta_{i+1}|f_{i+1}(a)$. In iteration $i+1$, only flow packages of size β_{i+1} are sent over arc a (since $a \notin A_{i+1}$). Therefore, throughout iteration $i+1$ we have $\beta_{i+1}|g_{i+1}(a)$, which proves (12.36).

So in iteration (12.28) we indeed can send a flow of value β_i in the residual graph D_{g_i}. Since f_{i+1} is equal to the final g_i, (12.36) also implies:

(12.38) if $a \in A \setminus A_i$, then $\beta_i|f_{i+1}(a)$.

Next we come to the kernel in the proof, which gives two bounds on $b(K)$ for $K \in \mathcal{K}_i$:

Claim 1. For each i and each $K \in \mathcal{K}_i$:

(12.39) (i) $|b(K)| \leq 13n^3\beta_i$;
(ii) *suppose $i > 0$, $K \in \mathcal{K}_{i-1}$, and* (12.28) *is applied to a vertex u in K; then $|b(K)| \geq \frac{1}{n}\beta_i$.*

Proof of Claim 1. I. We first show (12.39)(i). If $i = 0$, then $|b(K)| \leq n\|b\|_\infty = n\beta_0 \leq 13n^3\beta_0$. If $i > 0$, then, by (12.31),

(12.40) $|\mathrm{def}_{f_i}(K)| \leq n\|\mathrm{def}_{f_i}\|_\infty \leq 2n\beta_i$.

Moreover, since $f_i(a) \leq 12n\beta_i$ for each $a \in \delta^{\mathrm{in}}(K)$,

(12.41) $|\mathrm{excess}_{f_i}(K)| \leq 12n\beta_i \cdot |\delta^{\mathrm{in}}(K)| \leq 12n^3\beta_i$.

Hence

(12.42) $|b(K)| \leq |\mathrm{def}_{f_i}(K)| + |\mathrm{excess}_{f_i}(K)| \leq 2n\beta_i + 12n^3\beta_i \leq 13n^3\beta_i$.

This proves (12.39)(i).

II. Next we show (12.39)(ii). Since $K \in \mathcal{K}_{i-1} \cap \mathcal{K}_i$, u is the only vertex in K with $\mathrm{def}_{f_i}(u) \neq 0$. So, in iteration i, we do not apply (12.27) to a vertex in K. Moreover, by applying (12.28), $|\mathrm{def}_{g_i}(K)|$ does not increase. This gives

(12.43) $$\frac{n-1}{n}\beta_i < |\mathrm{def}_{g_i}(K)| \leq |\mathrm{def}_{f_i}(K)| \leq \frac{n-1}{n}\beta_{i-1}.$$

The first inequality holds as we apply (12.28) to u, and the last inequality follows from (12.30).

To prove (12.39)(ii), first assume $\beta_i = \frac{1}{2}\beta_{i-1}$. Since $\mathrm{def}_{f_i}(K) = \mathrm{def}_{f_i}(u)$, we have, by (12.43),

(12.44) $$\frac{n-1}{2n}\beta_{i-1} = \frac{n-1}{n}\beta_i \leq |\mathrm{def}_{f_i}(K)| \leq \frac{n-1}{n}\beta_{i-1}.$$

So $|\mathrm{def}_{f_i}(K)|/\beta_{i-1}$ has distance at least $1/2n$ to $\mathbb{Z}$. Since $f_i(a) \equiv 0 \pmod{\beta_{i-1}}$ by (12.38), we have

(12.45) $|b(K)|/\beta_{i-1} \equiv |\mathrm{def}_{f_i}(K)|/\beta_{i-1} \pmod 1$.

Hence also $|b(K)|/\beta_{i-1}$ has distance at least $1/2n$ to $\mathbb{Z}$. So $|b(K)| \geq \frac{1}{2n}\beta_{i-1} \geq \frac{1}{n}\beta_i$, as required.

Second assume $\beta_i < \frac{1}{2}\beta_{i-1}$. Then $\beta_i = \|\mathrm{def}_{f_i}\|_\infty < \frac{1}{12n}\beta_{i-1}$. Now as $K \in \mathcal{K}_i$, we have for each $a \in \delta(K)$: $0 \leq f_i(a) < 12n\beta_i < \beta_{i-1}$, while $f_i(a) \equiv 0 \pmod{\beta_{i-1}}$ by (12.38). So $f_i(a) = 0$ for each $a \in \delta(K)$. Hence by (12.43), (12.46) $$|b(K)| = |\mathrm{def}_{f_i}(K)| \geq \frac{n-1}{n}\beta_i \geq \frac{1}{n}\beta_i,$$

which proves (12.39)(ii). *End of Proof of Claim 1*

Define $\mathcal{K}^* := \bigcup_i \mathcal{K}_i$, and consider any $K \in \mathcal{K}^*$. Let I be the set of i with $K \in \mathcal{K}_i$. Let t be the smallest element of I. Let λ_K be the number of components in $\mathcal{K}_{t-1}$ contained in K. (Set $\lambda_K := 1$ if $t = 0$.) Then in iteration t, (12.28) is applied at most $4\lambda_K$ times to a vertex u in K, since (at the start of applying (12.28))

(12.47) $$|\mathrm{def}_{g_t}(K)| = |\mathrm{def}_{f_t}(K)| \leq \lambda_K \|\mathrm{def}_{f_t}\|_\infty \leq 2\lambda_K\beta_t$$

(by (12.31)). In any further iteration $i > t$ with $i \in I$, (12.28) is applied at most twice to u (again by (12.31)).

We estimate now the number of iterations $i \in I$ in which (12.28) is applied to $u \in K$. Consider the smallest such i with $i > t$. Then:

(12.48) if $j > i + \log_2(13n^4)$, then $j \notin I$.

For suppose to the contrary that $K \in \mathcal{K}_j$. Since $\beta_i \geq 2^{j-i}\beta_j > 13n^4\beta_j$, Claim 1 gives the contradiction $b(K) \leq 13n^3\beta_j < \frac{1}{n}\beta_i \leq b(K)$. This proves (12.48).

So (12.28) is applied at most $4\lambda_K + 2\log_2(13n^4)$ times to a vertex $u \in K$, in iterations $i \in I$. Since

(12.49) $$\sum_{K \in \mathcal{K}^*} \lambda_K \leq |\mathcal{K}^*| < 2n$$

(as $\mathcal{K}^*$ is laminar — cf. Theorem 3.5), (12.28) is applied at most $8n + 4n\log_2(13n^4)$ times in total. ∎

This bound on the number of iterations gives:

Corollary 12.7a. *A minimum-cost nonnegative b-transshipment can be found in time $O(n \log n(m + n \log n))$.*

Proof. Directly from Theorem 12.7, since any iteration (12.27) or (12.28) takes $O(m + n \log n)$ time, using Fibonacci heaps (Corollary 7.7a) and maintaining a potential as in Theorem 12.5. ∎

We can derive a bound for finding a minimum-cost circulation:

Corollary 12.7b. *A minimum-cost circulation can be found in time $O(m \log n(m+ n \log n))$.*

Proof. The minimum-cost circulation problem can be reduced to the minimum-cost transshipment problem as follows. Let $D = (V, A)$, $d, c, k : A \to \mathbb{Q}$ be the input for the minimum-cost circulation problem. Define

(12.50) $$b(v) := d(\delta^{\mathrm{out}}(v)) - d(\delta^{\mathrm{in}}(v))$$

for each $v \in V$. Then any minimum-cost b-transshipment x satisfying $\mathbf{0} \leq x \leq c - d$ gives a minimum-cost circulation $x' := x + d$ satisfying $d \leq x' \leq c$. So we can assume $d = \mathbf{0}$.

Now replace each arc $a = (u, v)$ by three arcs (u, u_a), (v_a, u_a), and (v_a, v), where u_a and v_a are new vertices. This makes the digraph D' say.

Define $b(u_a) := c(a)$ and $b(v_a) := -c(a)$. Moreover, define a cost function k' on the arcs of D' by $k'(u, u_a) := k(a)$, $k'(v_a, u_a) := 0$, $k'(v_a, v) := 0$ if $k(a) \geq 0$, and $k'(u, u_a) := 0$, $k'(v_a, u_a) := -k(a)$, $k'(v_a, v) := 0$ if $k(a) < 0$. Then a minimum-cost b-transshipment $x \geq \mathbf{0}$ in D' gives a minimum-cost b-transshipment x satisfying $\mathbf{0} \leq x \leq c$ in the original digraph D.

By Theorem 12.7, a minimum-cost b-transshipment $x \geq \mathbf{0}$ in D' can be found by finding $O(n \log n)$ times a shortest path in a residual graph D'_x. While this digraph has $2m + n$ vertices, it can be reduced in $O(m)$ time to finding a shortest path in an auxiliary digraph with $O(n)$ vertices only. Hence again it takes $O(m + n \log n)$ time by using Fibonacci heaps (Corollary 7.7a) and maintaining a potential as in Theorem 12.5. ∎

12.5. Further results and notes

12.5a. Complexity survey for minimum-cost circulation

Complexity survey for minimum-cost circulation ($*$ indicates an asymptotically best bound in the table):

	$O(n^4CK)$	Ford and Fulkerson [1958b] labeling
	$O(m^3C)$	Yakovleva [1959], Minty [1960], Fulkerson [1961] out-of-kilter method
	$O(nm^2C)$	Busacker and Gowen [1960], Iri [1960] successive shortest paths
$*$	$O(nC \cdot \mathrm{SP}_+(n, m, K))$	Edmonds and Karp [1970], Tomizawa [1971] successive shortest paths with nonnegative lengths using vertex potentials
	$O(nK \cdot \mathrm{MF}(n, m, C))$	Edmonds and Karp [1972]
$*$	$O(m \log C \cdot \mathrm{SP}_+(n, m, K))$	Edmonds and Karp [1972] capacity-scaling
	$O(nm \log(nC))$	Dinits [1973a] capacity-scaling
	$O(n \log K \cdot \mathrm{MF}(n, m, C))$	Röck [1980] (cf. Bland and Jensen [1992]) cost-scaling
	$O(m^2 \log n \cdot \mathrm{MF}(n, m, C))$	Tardos [1985a]
	$O(m^2 \log n \cdot \mathrm{SP}_+(n, m, K))$	Orlin [1984a], Fujishige [1986]
	$O(n^2 \log n \cdot \mathrm{SP}_+(n, m, K))$	Galil and Tardos [1986,1988]
	$O(n^3 \log(nK))$	Goldberg and Tarjan [1987], Bertsekas and Eckstein [1988]
$*$	$O(n^{5/3}m^{2/3} \log(nK))$	Goldberg and Tarjan [1987] generalized cost-scaling

≫

continued

	$O(nm \log n \log(nK))$	Goldberg and Tarjan [1987] generalized cost-scaling; Goldberg and Tarjan [1988b,1989] minimum-mean cost cycle-cancelling
*	$O(m \log n \cdot \mathrm{SP}_+(n, m, K))$	Orlin [1988,1993]
*	$O(nm \log(n^2/m) \log(nK))$	Goldberg and Tarjan [1990] generalized cost-scaling
*	$O(nm \log \log C \log(nK))$	Ahuja, Goldberg, Orlin, and Tarjan [1992] double scaling
*	$O(n \log C(m + n \log n))$	*circulations with lower bounds only* Gabow and Tarjan [1989]
*	$O((m^{3/2}C^{1/2} + \gamma \log \gamma) \log(nK))$	Gabow and Tarjan [1989]
*	$O((nm + \gamma \log \gamma) \log(nK))$	Gabow and Tarjan [1989]

Here $K := \|k\|_\infty$, $C := \|c\|_\infty$, and $\gamma := \|c\|_1$, for integer cost function k and integer capacity function c. Moreover, $\mathrm{SP}_+(n, m, K)$ denotes the running time of any algorithm finding a shortest path in a digraph with n vertices, m arcs, and nonnegative integer length function l with $K = \|l\|_\infty$. Similarly, $\mathrm{MF}(n, m, C)$ denotes the running time of any algorithm finding a maximum flow in a digraph with n vertices, m arcs, and nonnegative integer capacity function c with $C = \|c\|_\infty$.

Complexity survey for minimum-cost nonnegative transshipment:

*	$O(n \log B \cdot \mathrm{SP}_+(n, m, K))$	Edmonds and Karp [1970,1972]
	$O(n^2 \log n \cdot \mathrm{SP}_+(n, m, K))$	Galil and Tardos [1986,1988]
*	$O(n \log n \cdot \mathrm{SP}_+(n, m, K))$	Orlin [1988,1993]

Here $B := \|b\|_\infty$ for integer b.

12.5b. Min-max relations for minimum-cost flows and circulations

From Corollary 12.1a, the following min-max equality for minimum-cost circulation can be derived. The equality also follows directly from linear programming duality and total unimodularity. Both approaches were considered by Gallai [1957,1958a, 1958b].

Theorem 12.8. *Let $D = (V, A)$ be a digraph and let $c, d, k : A \to \mathbb{R}$. Then the minimum of $\sum_{a \in A} k(a)f(a)$ taken over all circulations f in D satisfying $d \le f \le c$ is equal to the maximum value of*

$$(12.51) \qquad \sum_{a \in A} (y(a)d(a) - z(a)c(a)),$$

where $y, z : A \to \mathbb{R}_+$ are such that there exists a function $p : V \to \mathbb{R}$ with the property that

(12.52) $\quad y(a) - z(a) = k(a) - p(v) + p(u)$

for each arc $a = (u, v)$ of D.

If d and c are integer, we can take f integer. If k is integer, we can take y and z integer.

Proof. The minimum is not less than the maximum, since if f is any circulation in D with $d \leq f \leq c$ and y, z, p satisfy (12.52), then

(12.53)
$$\sum_{a \in A} k(a)f(a) = \sum_{a=(u,v) \in A} (k(a)+p(u)-p(v))f(a) = \sum_{a \in A} (y(a)-z(a))f(a) \geq \sum_{a \in A} (y(a)d(a) - z(a)c(a)).$$

To see equality, let f be a minimum-cost circulation. By Corollary 12.1a, there is a function $p : V \to \mathbb{R}$ satisfying (12.7). Define for each arc $a = (u, v)$:

(12.54)
$$y(a) := \max\{0, k(a) - p(v) + p(u)\},$$
$$z(a) := \max\{0, -k(a) + p(v) - p(u)\}.$$

So y and z satisfy (12.52). Moreover, we have by (12.7) that $y(a)(f(a) - d(a)) = 0$ and $z(a)(c(a) - f(a)) = 0$ for each arc a. Hence we have equality throughout in (12.53). ∎

We consider a special case (Gallai [1957,1958a,1958b]). Let $D = (V, A)$ be a strongly connected digraph, let $d : A \to \mathbb{Z}_+$, and let $k : A \to \mathbb{Z}_+$ be a cost function, with $k(C) \geq 0$ for each directed circuit C. Then:

(12.55) the minimum cost $\sum_{a \in A} k(a)f(a)$ of an integer circulation f in D with $f \geq d$ is equal to the maximum value of $\sum_{a \in A} d(a)y(a)$ where $y : A \to \mathbb{Z}_+$ with $y(C) = k(C)$ for each directed circuit C in D.

A consequence of this applies to the 'directed Chinese postman problem': given a strongly connected directed graph, find a shortest directed closed path traversing each arc at least once. If we take unit length, we obtain the following. The minimum number of arcs in any closed directed path traversing each arc at least once is equal to the maximum value of

(12.56)
$$|A| + \sum_{U \in \mathcal{U}} (|\delta^{\text{in}}(U)| - |\delta^{\text{out}}(U)|),$$

where $\mathcal{U}$ is a collection of subsets U of V such that the $\delta^{\text{out}}(U)$ are disjoint.

This equality follows from (12.55), by taking $d = \mathbf{1}$ and $k = \mathbf{1}$: then there is a $p : V \to \mathbb{Z}$ with $y(a) = 1 - p(v) + p(u)$ for each arc $a = (u, v)$. So $p(v) \leq p(u) + 1$ for each arc $a = (u, v)$. Taking $U_i := \{v \in V \mid p(v) \leq i\}$ for each $i \in \mathbb{Z}$ gives the required cuts $\delta^{\text{out}}(U_i)$.

12.5c. Dynamic flows

A minimum-cost flow algorithm (in disguised form) was given by Ford and Fulkerson [1958b]. They considered the following 'dynamic flow' problem. Let $D = (V, A)$ be a digraph and let $r, s \in V$ (for convenience we assume that r is a source and s is a sink of D). Let $c : A \to \mathbb{Z}_+$ be a capacity function. Moreover, let a 'traversal time' function $\tau : A \to \mathbb{Z}_+$ be given, and a 'time limit' T.

The problem now is to send a maximum amount of flow from r to s, such that, for each arc a, at each time unit at most $c(a)$ flow is sent over a; it takes $\tau(a)$ time to traverse a. All flow is sent from r at one of the times $1, 2, \ldots, T$, while it reaches s at time at most T.

More formally, for any arc $a = (u, v)$ and any $t \in \{1, 2, \ldots, T\}$, let $x(a, t)$ denote the amount of flow sent from u over a at time t, reaching v at time $t + \tau(a)$. A first constraint is:

$$(12.57) \qquad 0 \leq x(a,t) \leq c(a)$$

for each $a \in A$ and each $t \in \{1, \ldots, T\}$. Next a flow conservation law can be formulated. Flow may 'wait' at any vertex until there is capacity enough to be transmitted further. This can be described as follows:

$$(12.58) \qquad \sum_{a \in \delta^{\text{in}}(v)} \sum_{t=1}^{t'-\tau(a)} x(a,t) \geq \sum_{a \in \delta^{\text{out}}(v)} \sum_{t=1}^{t'} x(a,t)$$

for each $v \in V \setminus \{r, s\}$ and each $t' \in \{1, \ldots, T\}$. We maximize the amount of flow reaching s not later than time T; that is, we

$$(12.59) \qquad \text{maximize} \sum_{a \in \delta^{\text{in}}(s)} \sum_{t=1}^{T-\tau(a)} x(a,t).$$

Since we may assume that we do not send flow from r that will not reach s, we may assume that we have equality in (12.58) if $t' = T$.

As Ford and Fulkerson [1958b] observed, this 'dynamic flow' problem can be transformed to a 'static' flow problem as follows. Let D' be the digraph with vertices all pairs (v, t) with $v \in V$ and $t \in \{1, \ldots, T\}$, and arcs

(12.60) (i) $((u,t), (v, t+\tau(a)))$ for each $a = (u, v) \in A$ and $t \in \{1, \ldots, T-\tau(a)\}$,
(ii) $((v,t), (v, t+1))$ for each $v \in V$ and $t \in \{1, \ldots, T-1\}$.

Let any arc of type (i) have capacity $c(a)$ and let any arc of type (ii) have capacity $+\infty$. Then the maximum dynamic flow problem is equivalent to finding a maximum flow in the new network from $(r, 1)$ to (s, T).

By this construction, a maximum dynamic flow can be found by solving a maximum flow problem in the large graph D'. Ford and Fulkerson [1958b] however described an alternative way of finding a dynamic flow that has a number of advantages. First of all, no 'large' graph D' has to be constructed (and the final algorithm can be modified with the scaling method of Edmonds and Karp [1972] to a method that is polynomial also in $\log T$). Second, the solution can be represented as a relatively small number of paths over which flow is transmitted repeatedly. Finally, the method shows that at intermediate vertices hold-over of flows is not necessary (that is, all arcs of type (12.60)(ii) with $v \neq r, s$ can be deleted).

Ford and Fulkerson [1958b] showed that a solution of the dynamic flow problem can be found by solving the following problem:

$$(12.61) \qquad \text{maximize} \sum_{a \in \delta^{\text{in}}(s)} Tx(a) - \sum_{a \in A} \tau(a) x(a),$$
$$\text{where } x \text{ is an } r-s \text{ flow satisfying } \mathbf{0} \leq x \leq c.$$

This is equivalent to a minimum-cost flow problem, with cost $k(a) := \tau(a) - T$ for $a \in \delta^{\text{in}}(s)$, and $k(a) := \tau(a)$ for all other a. Note that there are arcs of negative cost (generally), and that the value of the flow is not prescribed. So by adding an arc (s,r) we obtain a minimum-cost circulation problem.

How is problem (12.61) related to the dynamic flow problem? Given an optimum solution $x : A \to \mathbb{Z}_+$ of (12.61), there exist $r-s$ paths $P_1, \ldots, P_m$ in D such that

(12.62) $$x \geq \sum_{i=1}^{m} \chi^{P_i}$$

where m is the value of x. (We identify a path P and its set of arcs.) For any path P, let $\tau(P)$ be the traversal time of P (= the sum of the traversal times of the arcs in P). Then $\tau(P_i) \leq T$ for each i, since otherwise we could replace x by $x - \chi^{P_i}$, while increasing the objective value in (12.61).

Now send, for each $i = 1, \ldots, m$, a flow of value 1 along P_i at times $1, \ldots, T - \tau(P_i)$. It is not difficult to describe this in terms of the $x(a,t)$, yielding a feasible solution for the dynamic flow problem, of value

(12.63) $$\sum_{i=1}^{m} (T - \tau(P_i)) \geq mT - \sum_{a \in A} \tau(a) x(a),$$

which is the optimum value of (12.61).

In fact, this dynamic flow is optimum. Indeed, by Theorem 12.8 (alternatively, by LP-duality and total unimodularity), the optimum value of (12.61) is equal to that of:

(12.64) $$\text{minimize} \sum_{a \in A} c(a) y(a)$$

where $y : A \to \mathbb{Z}_+$ such that there exists $p : V \to \mathbb{Z}$ satisfying:

(12.65) $p(u) - p(v) + y(a) \geq -\tau(a)$ for each $a = (u,v) \in A$,

where $p(r) = 0$ and $p(s) = T$.

Now if $x(a,t)$ is a feasible solution of the dynamic flow problem, then by (12.58), (12.57) and (12.63),

(12.66) $$\begin{aligned} &\sum_{a \in \delta^{\text{in}}(s)} \sum_{t=1}^{T-\tau(a)} x(a,t) \leq \sum_{a \in \delta^{\text{in}}(s)} \sum_{t=1}^{T-\tau(a)} x(a,t) \\ &+ \sum_{v \neq s} \Big(\sum_{a \in \delta^{\text{in}}(v)} \sum_{t=1}^{p(v)-\tau(a)} x(a,t) - \sum_{a \in \delta^{\text{out}}(v)} \sum_{t=1}^{p(v)} x(a,t) \Big) \\ &= \sum_{a=(u,v) \in A} \Big(\sum_{t=1}^{p(v)-\tau(a)} x(a,t) - \sum_{t=1}^{p(u)} x(a,t) \Big) \\ &\leq \sum_{a=(u,v) \in A} \sum_{t=p(u)+1}^{p(v)-\tau(a)} x(a,t) \leq \sum_{a=(u,v) \in A} c(a)(p(v) - \tau(a) - p(u)) \\ &\leq \sum_{a \in A} c(a) y(a). \end{aligned}$$

Therefore, the dynamic flow constructed is optimum.

Ford and Fulkerson [1958b] described a method for solving (12.61) which essentially consists of repeatedly finding a shortest $r-s$ path in the residual graph, making costs nonnegative by translating the cost with the help of the current potential p (this is 'Routine I' of Ford and Fulkerson [1958b]). In this formulation, it is a primal-dual method.

The method of Ford and Fulkerson [1958b] improves the algorithm of Ford [1956] for the dynamic flow problem. More on this and related problems can be found in Wilkinson [1971], Minieka [1973], Orlin [1983,1984b], Aronson [1989], Burkard, Dlaska, and Klinz [1993], Hoppe and Tardos [1994,1995,2000], Klinz and Woeginger [1995], Fleischer and Tardos [1998], and Fleischer [1998b,1999c,2001b,2001a].

12.5d. Further notes

The minimum-cost flow problem is a linear programming problem, and hence it can be solved with the primal simplex method or the dual simplex method. Strongly polynomial *dual* simplex algorithms for minimum-cost flow have been given by Orlin [1985] ($O(m^3)$ pivots) and Plotkin and Tardos [1990] ($O(m^2/\log n)$ pivots) (cf. Orlin, Plotkin, and Tardos [1993]). No pivot rule is known however that finds a minimum-cost flow with the *primal* simplex method in polynomial time. Partial results were found by Goldfarb and Hao [1990] and Tarjan [1991].

Further work on the primal simplex method applied to the minimum-cost flow problem is discussed by Dantzig [1963], Gassner [1964], Johnson [1966b], Grigoriadis and Walker [1968], Srinivasan and Thompson [1973], Glover, Karney, and Klingman [1974], Glover, Karney, Klingman, and Napier [1974], Glover, Klingman, and Stutz [1974], Ross, Klingman, and Napier [1975], Cunningham [1976,1979], Bradley, Brown, and Graves [1977], Gavish, Schweitzer, and Shlifer [1977], Barr, Glover, and Klingman [1978,1979], Mulvey [1978a], Kennington and Helgason [1980], Chvátal [1983], Cunningham and Klincewicz [1983], Gibby, Glover, Klingman, and Mead [1983], Grigoriadis [1986], Ahuja and Orlin [1988,1992], Goldfarb, Hao, and Kai [1990a], Tarjan [1991,1997], Eppstein [1994a,2000], Orlin [1997], and Sokkalingam, Sharma, and Ahuja [1997].

Further results on the dual simplex method applied to minimum-cost flows are given by Dantzig [1963], Helgason and Kennington [1977b], Armstrong, Klingman, and Whitman [1979], Orlin [1984a], Ikura and Nemhauser [1986], Adler and Cosares [1990], Plotkin and Tardos [1990], Orlin, Plotkin, and Tardos [1993], Eppstein [1994a,2000], and Armstrong and Jin [1997].

Further algorithmic work is presented by Briggs [1962], Pla [1971] (dual out-of-kilter), Barr, Glover, and Klingman [1974] (out-of-kilter), Hassin [1983,1992], Bertsekas [1985], Kapoor and Vaidya [1986] (interior-point method), Bertsekas and Tseng [1988] ('relaxation method'), Masuzawa, Mizuno, and Mori [1990] (interior-point method), Cohen and Megiddo [1991], Bertsekas [1992] ('auction algorithm'), Norton, Plotkin, and Tardos [1992], Wallacher and Zimmermann [1992] (combinatorial interior-point method), Ervolina and McCormick [1993a,1993b], Fujishige, Iwano, Nakano, and Tezuka [1993], McCormick and Ervolina [1994], Hadjiat and Maurras [1997], Goldfarb and Jin [1999a], McCormick and Shioura [2000a,2000b] (cycle canceling), Shigeno, Iwata, and McCormick [2000] (cycle- and cut-canceling), and Vygen [2000].

Worst-case studies are made by Zadeh [1973a,1973b,1979] (cf. Niedringhaus and Steiglitz [1978]), Adel'son-Vel'skiĭ, Dinits, and Karzanov [1975], Dinits and Karzanov [1974], Radzik and Goldberg [1991,1994] (minimum-mean cost cycle-cancelling), and Hadjiat [1998] (dual out-of-kilter).

Computational studies were presented by Glover, Karney, and Klingman [1974] (simplex method), Glover, Karney, Klingman, and Napier [1974], Harris [1976], Karney and Klingman [1976], Bradley, Brown, and Graves [1977], Helgason and Kennington [1977b] (dual simplex method), Ali, Helgason, Kennington, and Lall [1978], Mulvey [1978b] (simplex method), Armstrong, Klingman, and Whitman [1979], Monma and Segal [1982] (simplex method), Gibby, Glover, Klingman, and Mead [1983], Grigoriadis [1986] (simplex method), Ikura and Nemhauser [1986] (dual simplex method), Bertsekas and Tseng [1988], Bland and Jensen [1992], Bland, Cheriyan, Jensen, and Ladányi [1993], Fujishige, Iwano, Nakano, and Tezuka [1993], Goldberg [1993a,1997] (push-relabel and successive approximation), Goldberg and Kharitonov [1993] (push-relabel), and Resende and Veiga [1993] (interior-point). Consult also Johnson and McGeoch [1993].

Bein, Brucker, and Tamir [1985] and Hoffman [1988] considered minimum-cost circulation for series-parallel digraphs. Wagner and Wan [1993] gave a polynomial-time simplex method for the maximum k-flow problem (with a profit for every unit of flow sent, and a cost for every unit capacity added to any arc a), which can be reduced to a minimum-cost circulation problem.

Maximum flows where the cost may not exceed a given 'budget' were considered by Fulkerson [1959b] and Ahuja and Orlin [1995].

'Unsplittable' flows (with one source and several sinks, where all flow from the source to any sink follows the same path) were investigated by Kleinberg [1996, 1998], Kolliopoulos and Stein [1997,1998a,1999,2002], Srinivasan [1997], Dinitz, Garg, and Goemans [1998,1999], Skutella [2000,2002], Azar and Regev [2001], Erlebach and Hall [2002], and Kolman and Scheideler [2002].

For generalized flows (with 'gains' on arcs), see Jewell [1962], Fujisawa [1963], Eisemann [1964], Mayeda and van Valkenburg [1965], Charnes and Raike [1966], Onaga [1966,1967], Glover, Klingman, and Napier [1972], Maurras [1972], Glover and Klingman [1973], Grinold [1973], Truemper [1977], Minieka [1978], Elam, Glover, and Klingman [1979], Jensen and Barnes [1980], Gondran and Minoux [1984], Kapoor and Vaidya [1986], Bertsekas and Tseng [1988], Ruhe [1988], Vaidya [1989c], Goldberg, Tardos, and Tarjan [1990], Goldberg, Plotkin, and Tardos [1991], Cohen and Megiddo [1994], Goldfarb and Jin [1996], Tseng and Bertsekas [1996,2000], Goldfarb, Jin, and Orlin [1997], Radzik [1998], Tardos and Wayne [1998], Oldham [1999,2001], Wayne [1999,2002], Wayne and Fleischer [1999], Fleischer and Wayne [2002], and Goldfarb, Jin, and Lin [2002].

For convex costs,see Charnes and Cooper [1958], Beale [1959], Shetty [1959], Berge [1960b], Minty [1960,1961,1962], Tụy [1963,1964], Menon [1965], Hu [1966], Weintraub [1974], Jensen and Barnes [1980], Kennington and Helgason [1980], Dembo and Klincewicz [1981], Hassin [1981a], Klincewicz [1983], Ahuja, Batra, and Gupta [1984], Minoux [1984,1986], Rockafellar [1984], Florian [1986], Bertsekas, Hosein, and Tseng [1987], Katsura, Fukushima, and Ibaraki [1989], Karzanov and McCormick [1995,1997], Tseng and Bertsekas [1996,2000], and Ahuja, Hochbaum, and Orlin [1999].

Concave costs were studied by Zangwill [1968], Rothfarb and Frisch [1970], Daeninck and Smeers [1977], Jensen and Barnes [1980], Graves and Orlin [1985], and Erickson, Monma, and Veinott [1987].

The basic references on network flows are the books of Ford and Fulkerson [1962] (historical) and Ahuja, Magnanti, and Orlin [1993]. Minimum-cost flow problems are discussed also in the books of Hu [1969,1982], Iri [1969], Frank and Frisch [1971], Potts and Oliver [1972], Adel'son-Vel'skiĭ, Dinits, and Karzanov [1975] (for a review, see Goldberg and Gusfield [1991]), Christofides [1975], Lawler [1976b], Murty [1976], Bazaraa and Jarvis [1977], Minieka [1978], Jensen and Barnes [1980], Kennington and Helgason [1980], Phillips and Garcia-Diaz [1981], Swamy and Thulasiraman [1981], Papadimitriou and Steiglitz [1982], Smith [1982], Chvátal [1983], Sysło, Deo, and Kowalik [1983], Tarjan [1983], Gondran and Minoux [1984], Rockafellar [1984], Derigs [1988a], Nemhauser and Wolsey [1988], Bazaraa, Jarvis, and Sherali [1990], Chen [1990], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], and Korte and Vygen [2000].

Survey papers include Glover and Klingman [1977], Ahuja, Magnanti, and Orlin [1989,1991], Goldberg, Tardos, and Tarjan [1990], and Frank [1995]. A bibliography was given by Golden and Magnanti [1977].

The history of the minimum-cost flow, circulation, and transshipment problems is closely intertwined with that of the transportation problem — see Section 21.13e.

Chapter 13

Path and flow polyhedra and total unimodularity

A large part of the theory of paths and flows can be represented geometrically by polytopes and polyhedra, and can be studied with methods from geometry and linear programming. Theorems like Menger's theorem, the max-flow min-cut theorem, and Hoffman's circulation theorem can be derived and interpreted with elementary polyhedral tools.
This can be done with the help of the total unimodularity of the incidence matrices of directed graphs, and of the more general network matrices. It again yields proofs of basic flow theorems and implies the polynomial-time solvability of flow problems.

13.1. Path polyhedra

Let $D = (V, A)$ be a digraph and let $s, t \in V$. The $s - t$ *path polytope* $P_{s-t \text{ path}}(D)$ is the convex hull of the incidence vectors in $\mathbb{R}^A$ of $s - t$ paths in D. (We recall that paths are simple, by definition.) So $P_{s-t \text{ path}}(D)$ is a polytope in the space $\mathbb{R}^A$. Since finding a maximum-length $s - t$ path in D is NP-complete, we may not expect to have a decent description of the inequalities determining $P_{s-t \text{ path}}(D)$ (cf. Corollary 5.16a). That is, the separation problem for $P_{s-t \text{ path}}(D)$ is NP-hard.

However, if we extend the $s-t$ path polytope to its dominant, it becomes more tractable. This leads to an illuminating geometric framework, in which the (easy) max-potential min-work theorem (Theorem 7.1) and the (more difficult) max-flow min-cut theorem (Theorem 10.3) show up as polars of each other, and can be derived from each other. This duality forms a prototype for many other dual theorems and problems.

The dominant $P^{\uparrow}_{s-t \text{ path}}(D)$ of $P_{s-t \text{ path}}(D)$ is the set of vectors $x \in \mathbb{R}^A$ with $x \geq y$ for some $y \in P_{s-t \text{ path}}(D)$. So

$$P^{\uparrow}_{s-t \text{ path}}(D) = P_{s-t \text{ path}}(D) + \mathbb{R}^A_+. \tag{13.1}$$

An alternative way of describing this polyhedron is as the set of all capacity functions $c : A \to \mathbb{R}_+$ for which there exists a flow $x \leq c$ of value 1.

It is not difficult to derive from the (easy) max-potential min-work theorem that the following inequalities determine $P^{\uparrow}_{s-t \text{ path}}(D)$:

(13.2) (i) $x(a) \geq 0$ for each $a \in A$,
(ii) $x(C) \geq 1$ for each $s - t$ cut C.

Theorem 13.1. *$P^{\uparrow}_{s-t \text{ path}}(D)$ is determined by* (13.2).

Proof. Clearly, each vector in $P^{\uparrow}_{s-t \text{ path}}(D)$ satisfies (13.2). So $P^{\uparrow}_{s-t \text{ path}}(D)$ is contained in the polyhedron Q determined by (13.2). Suppose that the reverse inclusion does not hold. Then there is an $l \in \mathbb{Z}^A$ such that the minimum of $l^{\mathsf{T}}x$ over $x \in Q$ is smaller than over $x \in P^{\uparrow}_{s-t \text{ path}}(D)$. If $l \notin \mathbb{Z}^A_+$, the minimum in both cases is $-\infty$; so $l \in \mathbb{Z}^A_+$. Then the minimum over $P^{\uparrow}_{s-t \text{ path}}(D)$ is equal to the minimum length k of an $s-t$ path, taking l as length function. By Theorem 7.1, there exist $s - t$ cuts $C_1, \ldots, C_k$ such that each arc a is in at most $l(a)$ of the C_i. Hence for any $x \in Q$ one has

(13.3) $$l^{\mathsf{T}}x \geq \Big(\sum_{i=1}^{k} \chi^{C_i}\Big)^{\mathsf{T}} x = \sum_{i=1}^{k} x(C_i) \geq k,$$

by (13.2)(ii). So the minimum over Q is at least k, contradicting our assumption. ∎

So the characterization of the dominant $P^{\uparrow}_{s-t \text{ path}}(D)$ of the $s - t$ path polytope follows directly from the easy Theorem 7.1 (the max-potential min-work theorem).

Note that Theorem 13.1 is equivalent to:

Corollary 13.1a. *The polyhedron determined by* (13.2) *is integer.*

Proof. The vertices are integer, as they are incidence vectors of paths. ∎

Next, the theory of blocking polyhedra implies a similar result when we interchange 'paths' and 'cuts', thus deriving the max-flow min-cut theorem.

The *$s - t$ cut polytope* $P_{s-t \text{ cut}}(D)$ is the convex hull of the incidence vectors of $s-t$ cuts in D. Again, $P_{s-t \text{ cut}}(D)$ is a polytope in the space $\mathbb{R}^A$. Since finding a maximum-size $s-t$ cut in D is NP-complete (Theorem 75.1), we may not expect to have a decent description of inequalities determining $P_{s-t \text{ cut}}(D)$. That is, the separation problem for $P_{s-t \text{ cut}}(D)$ is NP-hard.

Again, a polyhedron that behaves more satisfactorily is the dominant $P^{\uparrow}_{s-t \text{ cut}}(D)$ of the $s - t$ cut polytope, which is the set of vectors $x \in \mathbb{R}^A$ with $x \geq y$ for some $y \in P_{s-t \text{ cut}}(D)$. That is,

(13.4) $$P^{\uparrow}_{s-t \text{ cut}}(D) = P_{s-t \text{ cut}}(D) + \mathbb{R}^A_+.$$

Now the following inequalities determine $P^{\uparrow}_{s-t \text{ cut}}(D)$:

(13.5) (i) $x(a) \geq 0$ for $a \in A$,
(ii) $x(AQ) \geq 1$ for each $s - t$ path Q.

Corollary 13.1b. $P^{\uparrow}_{s-t\text{ cut}}(D)$ *is determined by* (13.5).

Proof. Directly with the theory of blocking polyhedra (Theorem 5.8) from Theorem 13.1. ∎

Equivalently:

Corollary 13.1c. *The polyhedron determined by* (13.5) *is integer.*

Proof. The vertices are integer, as they are incidence vectors of $s-t$ cuts. ∎

The two polyhedra are connected by the blocking relation:

Corollary 13.1d. *The polyhedra* $P^{\uparrow}_{s-t\text{ path}}(D)$ *and* $P^{\uparrow}_{s-t\text{ cut}}(D)$ *form a blocking pair of polyhedra.*

Proof. Directly from the above. ∎

With linear programming duality, this theorem implies the max-flow min-cut theorem:

Corollary 13.1e (max-flow min-cut theorem)**.** *Let* $D=(V,A)$*, let* $s,t\in V$ *and let* $c:A\to\mathbb{R}_+$ *be a capacity function. Then the maximum value of an* $s-t$ *flow* $f\leq c$ *is equal to the minimum capacity of an* $s-t$ *cut.*

Proof. The minimum capacity of an $s-t$ cut is equal to the minimum of $c^{\mathsf{T}}x$ over $x\in P^{\uparrow}_{s-t\text{ cut}}(D)$ (by definition (13.4)). By Corollary 13.1b, this is equal to the minimum value μ of $c^{\mathsf{T}}x$ where x satisfies (13.5). By linear programming duality, μ is equal to the maximum value of $\sum_Q \lambda_Q$, where $\lambda_Q\geq 0$ for each $s-t$ path Q, such that

$$\sum_Q \lambda_Q \chi^{AQ}\leq c. \tag{13.6}$$

Then $f:=\sum_Q \lambda_Q\chi^{AQ}$ is an $s-t$ flow of value μ. ∎

Thus the theory of blocking polyhedra links minimum-length paths and minimum-capacity cuts.

Algorithmic duality. The duality of paths and cuts can be extended to the polynomial-time solvability of the corresponding optimization problems. Indeed, Corollary 5.14a implies that the following can be derived from the fact that $P^{\uparrow}_{s-t\text{ path}}(D)$ and $P^{\uparrow}_{s-t\text{ cut}}(D)$ form a blocking pair of polyhedra (Corollary 13.1d):

(13.7) the minimum-length path problem is polynomial-time solvable
$\iff$
the minimum-capacity cut problem is polynomial-time solvable.

(Here the length and capacity functions are restricted to be nonnegative.)

By Theorem 5.15, one can also find a dual solution, which implies:

(13.8) the minimum-capacity cut problem is polynomial-time solvable
$\iff$
the maximum flow problem is polynomial-time solvable.

The statements in (13.7) by themselves are not surprising, since the polynomial-time solvability of neither of the problems has turned out to be hard, although finding a shortest path in polynomial time is easier than finding a maximum flow in polynomial time.

However, it is good to realize that the equivalence has been derived purely from the theoretical fact that the two polyhedra form a blocking pair. In further chapters we will see more sophisticated applications of this principle.

Dual integrality. The fact that $P^{\uparrow}_{s-t \text{ path}}(D)$ and $P^{\uparrow}_{s-t \text{ cut}}(D)$ form a blocking pair of polyhedra, is equivalent to the fact that the polyhedra determined by (13.2) and (13.5) each are integer (that is, have integer vertices only). More precisely, blocking polyhedra theory tells us:

(13.9) the polyhedron determined by (13.2) is integer $\iff$ the polyhedron determined by (13.5) is integer.

In other words, minimizing any linear function over (13.2) gives an integer optimum solution if and only if minimizing any linear function over (13.5) gives an integer optimum solution. Thus there is an equivalence of the existence of integer optimum solutions between two classes of linear programming problems. What can be said about the *dual* linear programs?

There is no general theorem known that links the existence of integer optimum dual solutions of blocking pairs of polyhedra. In fact, it is not the case that if two systems $Ax \leq b$ and $A'x \leq b'$ of linear inequalities represent a blocking pair of polyhedra, then the existence of integer optimum dual solutions for one system implies the existence of integer optimum dual solutions for the other. That is, the total dual integrality of $Ax \leq b$ is not equivalent to the total dual integrality of $A'x \leq b'$ (even not if one puts strong conditions on the two systems, like A, b, A', and b' being $0, 1$).

Yet, in the special case of paths and cuts the systems *are* totally dual integral, as follows directly from theorems proved in previous chapters. (In particular, total dual integrality of (13.5) amounts to the integrity theorem for flows.)

Theorem 13.2. *The systems* (13.2) *and* (13.5) *are totally dual integral.*

Proof. Total dual integrality of (13.2) is equivalent to Theorem 7.1, and total dual integrality of (13.5) is equivalent to Corollary 10.3a. ∎

By Theorem 5.22, total dual integrality of a system $Ax \leq b$ (with b integer) implies total *primal* integrality; that is, integrality of the polyhedron determined by $Ax \leq b$. So general polyhedral theory gives the following implications:

$$\begin{array}{ccc}
\boxed{\begin{array}{l}\text{(13.2) determines an}\\ \text{integer polyhedron}\end{array}} & \Longleftrightarrow & \boxed{\begin{array}{l}\text{(13.5) determines an}\\ \text{integer polyhedron}\end{array}} \\
\Uparrow & & \Uparrow \\
\boxed{\begin{array}{l}\text{(13.2) is totally dual}\\ \text{integral}\end{array}} & & \boxed{\begin{array}{l}\text{(13.5) is totally dual}\\ \text{integral}\end{array}}
\end{array} \tag{13.10}$$

13.1a. Vertices, adjacency, and facets

Vertices of the dominant of the $s-t$ path polytope have a simple characterization:

Theorem 13.3. *A vector x is a vertex of $P^{\uparrow}_{s-t\text{ path}}$ if and only if $x = \chi^{\pi}$ for some $s-t$ path π.*

Proof. If $x = \chi^{\pi}$ for some $s-t$ path π, then x is a vertex of $P^{\uparrow}_{s-t\text{ path}}$, as for the length function l defined by $l(a) := 0$ if $a \in A\pi$ and $l(a) := 1$ otherwise, the path π is the unique shortest $s-t$ path.

Conversely, let x be a vertex. As x is integer, $x \geq \chi^{\pi}$ for some $s-t$ path π. Then χ^{π} and $2x - \chi^{\pi} = x + (x - \chi^{\pi})$ belong to $P^{\uparrow}_{s-t\text{ path}}$. As $x = (\chi^{\pi} + (2x - \chi^{\pi}))/2$, we have $x = \chi^{\pi}$. ∎

As for adjacency, one has:

Theorem 13.4. *Let π and π' be two distinct $s-t$ paths in D. Then χ^{π} and $\chi^{\pi'}$ are adjacent vertices of $P^{\uparrow}_{s-t\text{ path}}$ if and only if $A\pi \triangle A\pi'$ is an undirected circuit consisting of two internally vertex-disjoint directed paths.*

Proof. If $A\pi \triangle A\pi'$ is an undirected circuit consisting of two internally vertex-disjoint paths, define the length function l by $l(a) := 0$ if $a \in A\pi \cup A\pi'$ and $l(a) := 1$ otherwise. Then π and π' are the only two shortest $s-t$ paths.

Conversely, let χ^{π} and $\chi^{\pi'}$ be adjacent. Suppose that $A\pi \cup A\pi'$ contains an $s-t$ path π'' different from π and π'. Then $\chi^{\pi} + \chi^{\pi'} - \chi^{\pi''} = \chi^{\pi'''}$ for some $s-t$ path π''', contradicting the adjacency of χ^{π} and $\chi^{\pi'}$. This implies that $A\pi \triangle A\pi'$ is an undirected circuit consisting of two internally vertex-disjoint directed paths. ∎

Finally, for the facets we have:

Theorem 13.5. *Let C be an $s-t$ cut. Then the inequality $x(C) \geq 1$ determines a facet of $P^{\uparrow}_{s-t\text{ path}}$ if and only if C is an inclusionwise minimal $s-t$ cut.*

Proof. *Necessity.* Suppose that there is an $s-t$ cut $C' \subset C$. Then the inequalities $x \geq \mathbf{0}$ and $x(C') \geq 1$ imply $x(C) \geq 1$, and hence $x(C) \geq 1$ is not facet-inducing.

Sufficiency. If the inequality $x(C) \geq 1$ is not facet-inducing, it is a nonnegative linear combination of other inequalities in system (13.2). At least one of them is of the form $x(C') \geq 1$ for some $s-t$ cut C'. Then necessarily $C' \subset C$. ∎

Garg and Vazirani [1993,1995] characterized the vertices of and adjacency on a variant of the $s-t$ cut polytope.

13.1b. The $s-t$ connector polytope

There are a number of related polyhedra for which similar results hold. Call a subset A' of A an $s-t$ *connector* if A' contains the arc set of an $s-t$ path as a subset. The $s-t$ *connector polytope* $P_{s-t\text{ connector}}(D)$ is the convex hull of the incidence vectors of the $s-t$ connectors.

This polytope turns out to be determined by the following system of linear inequalities:

(13.11)
(i) $0 \leq x(a) \leq 1$ for each $a \in A$,
(ii) $x(C) \geq 1$ for each $s-t$ cut C.

Again, the fact that the $s-t$ connector polytope is contained in the polytope determined by (13.11) follows from the fact that χ^P satisfies (13.11) for each $s-t$ connector P.

Also in this case one has:

Theorem 13.6. *System* (13.11) *is totally dual integral.*

Proof. Directly from Theorem 13.2, using Theorem 5.23. ∎

It implies primal integrality:

Corollary 13.6a. *The $s-t$ connector polytope is equal to the solution set of* (13.11).

Proof. Directly from Theorem 13.6. ∎

The dimension of $P_{s-t\text{ connector}}(D)$ is easily determined:

Theorem 13.7. *Let A' be the set of arcs a for which there exists an $s-t$ path not traversing a. Then* $\dim P_{s-t\text{ connector}}(D) = |A'|$.

Proof. We use Theorem 5.6. Clearly, no inequality $x_a \geq 0$ is an implicit equality. Moreover, the inequality $x_a \leq 1$ is an implicit equality if and only if $a \in A \setminus A'$. For distinct arcs $a \in A \setminus A'$, these equalities are independent.

Suppose that there is a $U \subseteq V$ with $s \in U$, $t \notin U$, such that $x(\delta^{\text{out}}(U)) \geq 1$ is an implicit equality. Then $|\delta^{\text{out}}(U)| = 1$, since the all-one vector belongs to the polytope. So the arc in $\delta^{\text{out}}(U)$ belongs to $A \setminus A'$.

We conclude that the maximum number of independent implicit equalities is equal to $|A \setminus A'|$. Hence $\dim P = |A'|$. ∎

Let $D = (V, A)$ be a digraph, let $s, t \in V$, and let $k \in \mathbb{Z}_+$. By the results above, the convex hull P of the incidence vectors χ^B of those subsets B of A that contain k arc-disjoint $s - t$ paths, is determined by

$$\text{(13.12)} \quad \begin{array}{lll} \text{(i)} & 0 \leq x(a) \leq 1 & \text{for each } a \in A, \\ \text{(ii)} & x(C) \geq k & \text{for each } s-t \text{ cut } C. \end{array}$$

L.E. Trotter, Jr observed that this polytope P has the *integer decomposition property*; that is, for each $l \in \mathbb{Z}_+$, any integer vector $x \in l \cdot P$ is the sum of l integer vectors in P.

Theorem 13.8. *The polytope P determined by* (13.12) *has the integer decomposition property.*

Proof. Let $l \in \mathbb{Z}_+$ and let $x \in l \cdot P$. Then there exists an integer $s-t$ flow $f \leq x$ of value $l \cdot k$ (by the max-flow min-cut theorem). We can assume that $x = f$. As $\frac{1}{l}f$ is an $s - t$ flow of value k, by Corollary 11.2c there exists an integer $s - t$ flow f' of value k with

$$\text{(13.13)} \qquad \lfloor \frac{1}{l} f(a) \rfloor \leq f'(a) \leq \lceil \frac{1}{l} f(a) \rceil$$

for each arc a. Then f' is an integer vector in P, since $f'(a) \leq 1$ for each arc a, as $f(a) \leq l$ for each arc a. Moreover, $f - f'$ is an integer vector belonging to $(l-1) \cdot P$, as $f - f'$ is an $s - t$ flow of value $(l - 1) \cdot k$ and as $(f - f')(a) \leq l - 1$ for each arc a, since if $f(a) = l$, then $f'(a) = 1$ by (13.13). ∎

13.2. Total unimodularity

Let $D = (V, A)$ be a digraph. Recall that the $V \times A$ *incidence matrix* M of D is defined by $M_{v,a} := -1$ if a leaves v, $M_{v,a} := +1$ if a enters v, and $M_{v,a} := 0$ otherwise. So each column of M contains exactly one $+1$ and exactly one -1, while all other entries are 0. The following basic statement follows from a theorem of Poincaré [1900][21] (we follow the proofs of Chuard [1922] and Veblen and Franklin [1921]):

Theorem 13.9. *The incidence matrix M of any digraph D is totally unimodular.*

Proof. Let B be a square submatrix of M, of order k say. We prove that $\det B \in \{0, \pm 1\}$ by induction on k, the case $k = 1$ being trivial. Let $k > 1$. We distinguish three cases.

[21] Poincaré [1900] showed the total unimodularity of any $\{0, \pm 1\}$ matrix $M = (M_{i,j})$ with the property that for each k and all distinct row indices $i_1, \ldots, i_k$ and all distinct column indices $j_1, \ldots, j_k$, the product

$$M_{i_1,j_1} M_{i_1,j_2} M_{i_2,j_2} M_{i_2,j_3} \cdots M_{i_{k-1},j_{k-1}} M_{i_{k-1},j_k} M_{i_k,j_k} M_{i_k,j_1}$$

belongs to $\{0, 1\}$ if k is even and to $\{0, -1\}$ if k is odd. Incidence matrices of digraphs have this property.

Case 1: B has a column with only zeros. Then $\det B = 0$.

Case 2: B has a column with exactly one nonzero. Then we can write (up to permuting rows and columns):

$$B = \begin{pmatrix} \pm 1 & b^{\mathsf{T}} \\ \mathbf{0} & B' \end{pmatrix}, \tag{13.14}$$

for some vector b and matrix B'. Then by the induction hypothesis, $\det B' \in \{0, \pm 1\}$, and hence $\det B \in \{0, \pm 1\}$.

Case 3 : Each column of B contains two nonzeros. Then each column of B contains one $+1$ and one -1, while all other entries are 0. So each row of B adds up to 0, and hence $\det B = 0$. ∎

One can derive several results on circulations, flows, and transshipments from the total unimodularity of the incidence matrix of a digraph, like the max-flow min-cut theorem (see Section 13.2a below) and theorems characterizing the existence of a circulation or a b-transshipment (Theorem 11.2 and Corollary 11.2f). Moreover, min-max equalities for minimum-cost flow, circulation (cf. Theorem 12.8), and b-transshipment follow. We discuss some of the previous and some new results in the following sections.

13.2a. Consequences for flows

We show that the max-flow min-cut theorem can be derived from the total unimodularity of the incidence matrix of a digraph:

Corollary 13.9a (max-flow min-cut theorem). *Let $D = (V, A)$, let $s, t \in V$, and let $c : A \to \mathbb{R}_+$ be a capacity function. Then the maximum value of an $s - t$ flow $f \leq c$ is equal to the minimum capacity of an $s - t$ cut.*

Proof. Since the maximum clearly cannot exceed the minimum, it suffices to show that there exists an $s - t$ flow $x \leq c$ and an $s - t$ cut, whose capacity is not more than the value of x.

Let M be the incidence matrix of D and let M' arise from M by deleting the rows corresponding to s and t. So the condition $M'x = \mathbf{0}$ means that the flow conservation law should hold at any vertex $v \neq s, t$.

Let w be the row of M corresponding to vertex t. So for any arc a, $w_a = +1$ if a enters t, $w_a = -1$ if a leaves t, and $w_a = 0$ otherwise.

Now the maximum value of an $s - t$ flow subject to c is equal to

$$\max\{w^{\mathsf{T}} x \mid \mathbf{0} \leq x \leq c; M'x = \mathbf{0}\}. \tag{13.15}$$

By LP-duality, this is equal to

$$\min\{y^{\mathsf{T}} c \mid y \geq \mathbf{0}; \exists z : y^{\mathsf{T}} + z^{\mathsf{T}} M' \geq w^{\mathsf{T}}\}. \tag{13.16}$$

Since M' is totally unimodular by Theorem 13.9 and since w is an integer vector, minimum (13.16) is attained by *integer* vectors y and z. Extend z by defining $z_t := -1$ and $z_s := 0$. Then $y^{\mathsf{T}} + z^{\mathsf{T}} M \geq \mathbf{0}$.

Now define

(13.17) $U := \{v \in V \mid z_v \geq 0\}$.

Then U is a subset of V containing s and not containing t.

It suffices to show that

(13.18) $c(\delta^{\text{out}}(U)) \leq y^{\mathsf{T}}c$,

since $y^{\mathsf{T}}c$ is equal to the maximum flow value (13.15).

To prove (13.18), it suffices to show that

(13.19) if $a = (u, v) \in \delta^{\text{out}}(U)$, then $y_a \geq 1$.

To see this, note that $z_u \geq 0$ and $z_v \leq -1$. Moreover, $y^{\mathsf{T}} + z^{\mathsf{T}}M \geq \mathbf{0}$ implies $y_a + z_v - z_u \geq 0$. So $y_a \geq z_u - z_v \geq 1$. This proves (13.19). ∎

It follows similarly that if all capacities are integers, then there exists a maximum *integer* flow; that is, we have the integrity theorem (Corollary 10.3a).

Let $D = (V, A)$ be a digraph and $s, t \in V$. The set of all $s-t$ flows of value 1 is a polyhedron $P_{s-t \text{ flow}}(D)$, determined by:

(13.20)
$$\begin{array}{rll} \text{(i)} & x(a) \geq 0 & \text{for each } a \in A, \\ \text{(ii)} & x(\delta^{\text{in}}(v)) = x(\delta^{\text{out}}(v)) & \text{for each } v \in V \setminus \{s, t\}, \\ \text{(iii)} & x(\delta^{\text{out}}(s)) - x(\delta^{\text{in}}(s)) = 1. & \end{array}$$

The set of all $s-t$ flows of value ϕ trivially equals $\phi \cdot P_{s-t \text{ flow}}(D)$.

The total unimodularity of M gives that, for integer ϕ, the intersection of $\phi \cdot P_{s-t \text{ flow}}(D)$ with an integer box $\{x \mid \mathbf{0} \leq x \leq c\}$ is an integer polytope. In other words:

Theorem 13.10. *Let $D = (V, A)$ be a digraph, $s, t \in V$, $c : A \to \mathbb{Z}$, and $\phi \in \mathbb{Z}_+$. Then the set of $s-t$ flows $x \leq c$ of value ϕ forms an integer polytope.*

Proof. Directly from the total unimodularity of the incidence matrix of a digraph (using Theorem 5.20). ∎

In particular this gives:

Corollary 13.10a. *Let $D = (V, A)$ be a digraph, $s, t \in V$, $c : A \to \mathbb{Z}$ and $\phi \in \mathbb{Z}$. If there exists an $s-t$ flow $x \leq c$ of value ϕ, then there exists an integer such flow.*

Proof. Directly from Theorem 13.10. ∎

Notes. A relation of $P_{s-t \text{ flow}}(D)$ with the polytope $P_{s-t \text{ path}}(D)$ is that

(13.21) $P_{s-t \text{ path}}(D) \subseteq P_{s-t \text{ flow}}(D) \subseteq P^{\uparrow}_{s-t \text{ path}}(D)$.

Hence

(13.22) $P^{\uparrow}_{s-t \text{ path}}(D) = P^{\uparrow}_{s-t \text{ flow}}(D)$.

Dantzig [1963] (pp. 352–366) showed that each vertex x of $P_{s-t \text{ flow}}(D)$ is the incidence vector χ^P of some $s-t$ path P. It can be shown that, if each arc of D is in some $s-t$ path, then $P_{s-t \text{ flow}}(D)$ is the topological closure of the convex hull of the vectors $\chi^P \in \mathbb{R}^A$ where P is an $s-t$ walk and where

(13.23) $\chi^P(a) :=$ number of times P traverses a,

for $a \in A$.

For two distinct $s-t$ paths, the vertices χ^P and $\chi^{P'}$ are adjacent if and only if the symmetric difference $AP \triangle AP'$ forms an undirected circuit consisting of two internally vertex disjoint directed paths.

Saigal [1969] proved that any two vertices of $P_{s-t \text{ flow}}(D)$ are connected by a path on the 1-skeleton of $P_{s-t \text{ flow}}(D)$ with at most $|A|-1$ edges — this implies the Hirsch conjecture for this class of polyhedra. (The Hirsch conjecture (cf. Dantzig [1963,1964]) says that the 1-skeleton of a polytope in $\mathbb{R}^n$ determined by m inequalities has diameter at most $m-n$.) In fact, Saigal showed more strongly that for any two feasible bases B, B' of (13.20), there is a series of at most $|A|-1$ pivots bringing B to B'. It amounts to the following. Call a spanning tree T of D *feasible* if it contains a directed $s-t$ path. Call two feasible spanning trees T, T' *adjacent* if $|AT \setminus AT'| = 1$. Then for any two feasible spanning trees T, T' there exists a sequence $T_0, \ldots, T_k$ of feasible spanning trees such that $T_0 = T$, $T_k = T'$, $k \leq |A|-1$, and T_{i-1} and T_i adjacent for $i = 1, \ldots, k$.

Rispoli [1992] showed that if D is the complete directed graph, then for each length function l and each vertex x_0 of (13.20), there is a path $x_0, x_1, \ldots, x_d$ on the 1-skeleton of (13.20), where $l^{\mathsf{T}} x_i \leq l^{\mathsf{T}} x_{i-1}$ for $i = 1, \ldots, d$, where x_d minimizes $l^{\mathsf{T}} x$ over (13.20), and where $d \leq \frac{2}{3}(|V|-1)$.

13.2b. Consequences for circulations

Another consequence is:

Corollary 13.10b. *Let $D = (V, A)$ be a digraph and let $c, d : A \to \mathbb{Z}$. Then the set of circulations x satisfying $d \leq x \leq c$ forms an integer polytope.*

Proof. Directly from the total unimodularity of the incidence matrix of a digraph. ∎

In particular this implies:

Corollary 13.10c. *Let $D = (V, A)$ be a digraph and let $c, d : A \to \mathbb{Z}$. If there exists a circulation x satisfying $d \leq x \leq c$, then there exists an integer such circulation.*

Proof. Directly from Corollary 13.10b. ∎

Another consequence is the integer decomposition property as in Corollary 11.2b.

13.2c. Consequences for transshipments

Let $D = (V, A)$ be a digraph, $d, c \in \mathbb{R}^A_+$, and $b \in \mathbb{R}^V$. The *b-transshipment polytope* is the set of all b-transshipments x with $d \leq x \leq c$. So it is equal to

(13.24) $P := \{x \in \mathbb{R}^A \mid d \leq x \leq c, Mx = b\}$, where M is the $V \times A$ incidence matrix of D.

Again, the total unimodularity of M (Theorem 13.9) implies:

Theorem 13.11. *Let $D = (V, A)$ be a digraph, $b : V \to \mathbb{Z}$, and $c, d : A \to \mathbb{Z}$. Then the b-transshipment polytope is an integer polytope.*

Proof. Directly from the total unimodularity of the incidence matrix of a digraph. ∎

In particular this gives:

Corollary 13.11a. *Let $D = (V, A)$ be a digraph, $b : V \to \mathbb{Z}$, and $c, d : A \to \mathbb{Z}$. If there exists a b-transshipment x with $d \leq x \leq c$, then there exists an integer such b-transshipment.*

Proof. Directly from Theorem 13.11. ∎

Also Corollary 11.2f can be derived:

Theorem 13.12. *There exists a b-transshipment x satisfying $d \leq x \leq c$ if and only if $b(V) = 0$, $d \leq c$ and $c(\delta^{\mathrm{in}}(U)) - d(\delta^{\mathrm{out}}(U)) \geq b(U)$ for each $U \subseteq V$.*

Proof. Necessity being easy, we show sufficiency. If no b-transshipment as required exists, then by Farkas' lemma, there exist vectors $y \in \mathbb{R}^V$ and $z', z'' \in \mathbb{R}^A_+$ such that $y^{\mathsf{T}}M + z'^{\mathsf{T}} - z''^{\mathsf{T}} = \mathbf{0}$ and $y^{\mathsf{T}}b + z'^{\mathsf{T}}c - z''^{\mathsf{T}}d < 0$. By adding a multiple of $\mathbf{1}$ to y we can assume that $y \geq \mathbf{0}$ (since $\mathbf{1}^{\mathsf{T}}M = \mathbf{0}$ and $\mathbf{1}^{\mathsf{T}}b = 0$). Next, by scaling we can assume that $\mathbf{0} \leq y \leq \mathbf{1}$. As M is totally unimodular, we can assume moreover that y is integer. So $y = \chi^U$ for some $U \subseteq V$. Since $d \leq c$, we can assume that $z'(a) = 0$ or $z''(a) = 0$ for each $a \in A$. Hence $z' = \chi^{\delta^{\mathrm{out}}(U)}$ and $z'' = \chi^{\delta^{\mathrm{in}}(U)}$. Then $y^{\mathsf{T}}b + z'^{\mathsf{T}}c - z''^{\mathsf{T}}d < 0$ contradicts the condition for $V \setminus U$. ∎

For any digraph $D = (V, A)$ and $b \in \mathbb{R}^V$, let P_b denote the set of b-transshipments. So

(13.25) $\qquad P_b = \{x \mid Mx = b\}$

where M is the $V \times A$ incidence matrix of D. Koopmans and Reiter [1951] characterized the dimension of the transshipment space:

(13.26) $\qquad$ if P_b is nonempty, then it has dimension $|A| - |V| + k$, where k is the number of weak components of D.

(A *weak component* of a digraph is a component of the underlying undirected graph.)

To see (13.26), let $F \subseteq A$ form a spanning forest in the underlying undirected graph. So (V, F) has k weak components and contains no undirected circuit. Then $|F| = |V| - k$. Now each $x \in \mathbb{R}^{A \setminus F}$ can be extended uniquely to a b-transshipment $x \in \mathbb{R}^A$. Hence P_b has dimension $|A \setminus F| = |A| - |V| + k$.

Consider next the polyhedron

(13.27) $\qquad Q_b := \{x \in \mathbb{R}^A \mid$ there exists a nonnegative b-transshipment $f \leq x\}$

So

(13.28) $Q_b = (P_b \cap \mathbb{R}_+^A) + \mathbb{R}_+^A.$

By Gale's theorem (Corollary 11.2g), Q_b is determined by:

(13.29)
$$\begin{array}{lll} \text{(i)} & x(a) \geq 0 & \text{for each } a \in A, \\ \text{(ii)} & x(\delta^{\text{in}}(U)) \geq b(U) & \text{for each } U \subseteq V. \end{array}$$

Fulkerson and Weinberger [1975] showed that this system is TDI:

Theorem 13.13. *System* (13.29) *is TDI.*

Proof. Choose $w \in \mathbb{Z}_+^A$. We must show that the dual of minimizing $w^{\mathsf{T}}x$ over (13.29) has an integer optimum solution.

Let μ be the minimum value of $w^{\mathsf{T}}x$ over (13.29). As (13.29) determines Q_b, μ is equal to the minimum value of $w^{\mathsf{T}}x$ over $x \geq \mathbf{0}$, $Mx = b$, where M is the $V \times A$ incidence matrix of D. Since M is totally unimodular, this LP-problem has an integer optimum dual solution. That is, there exists a $y \in \mathbb{Z}^V$ such that $y^{\mathsf{T}}M \leq w^{\mathsf{T}}$ and $y^{\mathsf{T}}b = \mu$. We can assume that $y \geq \mathbf{0}$, since $\mathbf{1}^{\mathsf{T}}M = \mathbf{0}$ and $\mathbf{1}^{\mathsf{T}}b = 0$ (we can add a multiple of $\mathbf{1}$ to y). For each $i \in \mathbb{Z}_+$, let $U_i := \{v \mid y_v \geq i\}$. (So $U_i = \emptyset$ for i large enough.) Then

(13.30) $$\sum_{i=1}^{\infty} \chi^{\delta^{\text{out}}(U_i)} \leq w,$$

since for each arc $a = (u, v)$ we have $y_v - y_u \leq w(a)$, implying that the number of i such that a enters U_i is at most $\max\{0, y_v - y_u\}$, which is at most $w(a)$. So this gives a feasible integer dual solution to the problem of minimizing $w^{\mathsf{T}}x$ over (13.29). It is in fact optimum, since

(13.31) $$\sum_{i=1}^{\infty} b(U_i) = y^{\mathsf{T}}b = \mu.$$

This proves the theorem. ∎

This implies for primal integrality:

Corollary 13.13a. *If b is integer, then Q_b is integer.*

Proof. Directly from Theorem 13.13. ∎

Fulkerson and Weinberger [1975] also showed an integer decomposition theorem for nonnegative b-transshipments (it also follows directly from the total unimodularity of the incidence matrix M of D):

Theorem 13.14. *Let $D = (V, A)$, $b \in \mathbb{Z}^V$, and $k \in \mathbb{Z}_+$, with $k \geq 1$, and let $f : A \to \mathbb{Z}_+$ be a $k \cdot b$-transshipment. Then there exist b-transshipments $f_1, \ldots, f_k : A \to \mathbb{Z}_+$ such that $f = f_1 + \cdots + f_k$.*

Proof. It suffices to show that there exists a b-transshipment $g : A \to \mathbb{Z}_+$ such that $g \leq f$ — the theorem then follows by induction on k.

The existence of g follows from Gale's theorem (Corollary 11.2g), since $b(U) \leq f(\delta^{\text{in}}(U))$ for each $U \subseteq V$, as either $b(U) < 0$, or $b(U) \leq kb(U) \leq f(\delta^{\text{in}}(U))$. ∎

Theorem 13.14 implies the integer decomposition property for the polyhedron Q_b:

Corollary 13.14a. *If b is integer, the polyhedron Q_b has the integer decomposition property.*

Proof. Let $k \in \mathbb{Z}_+$ and let c be an integer vector in kQ_b. So $c \in Q_{k \cdot b}$, implying that there exists an integer $k \cdot b$-transshipment $f \leq c$. By Theorem 13.14, there exist integer b-transshipments $f_1, \ldots, f_k \geq \mathbf{0}$ with $f = f_1 + \cdots + f_k$. Define $f_1' := f_1 + (c - f)$. Then $f_1', f_2, \ldots, f_k \in Q_b$ and $c = f_1' + f_2 + \cdots + f_k$. ∎

If b is integer, we know:

(13.32) $\quad Q_b = \text{conv.hull}\{f \mid f \text{ nonnegative integer } b\text{-transshipment}\} + \mathbb{R}_+^A.$

Hence the blocking polyhedron $B(Q_b)$ of Q_b is determined by:

(13.33) (i) $x(a) \geq 0$ for each $a \in A$,
(ii) $f^{\mathsf{T}} x \geq 1$ for each nonnegative integer b-transshipment f.

Fulkerson and Weinberger [1975] derived from Corollary 13.14a that this system has the integer rounding property if b is integer:

Corollary 13.14b. *If b is integer, system* (13.33) *has the integer rounding property.*

Proof. Choose $c \in \mathbb{Z}^A$. Let

(13.34) $\quad \mu := \max\{\sum_f z_f \mid z_f \geq 0, \sum_f z_f f \leq c\}$

and let μ' be the maximum in which the z_f are restricted to nonnegative integers (here f ranges over minimal nonnegative integer b-transshipments). Let $k := \lfloor \mu \rfloor$. We must show that $\mu' = k$.

As $\mu = \min\{c^{\mathsf{T}} x \mid x \in B(Q_b)\}$, we know that $c \in \mu \cdot Q_b$. Hence, as $k \leq \mu$, $c \in kQ_b$. By Corollary 13.14a, there exist nonnegative integer b-transshipments $f_1, \ldots, f_k$ with $f_1 + \cdots + f_k \leq c$. Hence $\mu' \geq k$. Since $\mu' \leq \mu$, we have $\mu' = k$. ∎

Generally, we cannot restrict (13.33) to those f that form a vertex of Q_b while maintaining the integer rounding property, as was shown by Fulkerson and Weinberger [1975]. Trotter and Weinberger [1978] extended these results to b-transshipments with upper and lower bounds on the arcs. For related results, see Bixby, Marcotte, and Trotter [1987].

13.2d. Unions of disjoint paths and cuts

The total unimodularity of the incidence matrix of a digraph can also be used to derive min-max relations for the minimum number of arcs covered by l arc-disjoint $s - t$ paths:

Theorem 13.15. *Let $D = (V, A)$ be a digraph, $s, t \in V$, and $l \in \mathbb{Z}_+$. Then the minimum value of $|AP_1| + \cdots + |AP_l|$ where $P_1, \ldots, P_l$ are arc-disjoint $s - t$ paths is equal to the maximum value of*

$$\left|\bigcup \mathcal{C}\right| - \sum_{C \in \mathcal{C}} (|C| - l), \tag{13.35}$$

where $\mathcal{C}$ ranges over all collections of $s-t$ cuts.

Proof. The minimum in the theorem is equal to the minimum value of $\sum_{a \in A} f(a)$ subject to

(13.36)
$$\begin{array}{ll} 0 \le f(a) \le 1 & \text{for each } a \in A, \\ f(\delta^{\text{in}}(v)) - f(\delta^{\text{out}}(v)) = 0 & \text{for each } v \in V \setminus \{s,t\}, \\ f(\delta^{\text{in}}(t)) - f(\delta^{\text{out}}(t)) = l. & \end{array}$$

By LP-duality and total unimodularity of the constraint matrix, this minimum value μ is equal to the maximum value of $l \cdot p(t) - \sum_{a \in A} y(a)$, where $y \in \mathbb{Z}_+^A$ and $p \in \mathbb{Z}^V$ satisfy:

(13.37)
$$\begin{array}{l} p(s) = 0; \\ p(v) - p(u) - y(a) \le 1 \text{ for each } a = (u,v) \in A. \end{array}$$

As $\mu \ge 0$, we know $p(t) \ge 0$. Let $r := p(t)$, and for $j = 1, \ldots, r$, let $U_j := \{v \in V \mid p(v) < j\}$ and $C_j := \delta^{\text{out}}(U_j)$. Then

(13.38)
$$\begin{aligned} \sum_{j=1}^{r} |C_j| &\le \sum_{\substack{a = (u,v) \in A \\ p(v) > p(u) \\ p(v) \ge 0 \\ p(u) < r}} (p(v) - p(u)) \le \sum_{\substack{a = (u,v) \in A \\ p(v) > p(u) \\ p(v) \ge 0 \\ p(u) < r}} (1 + y(a)) \\ &\le \left| \bigcup_{j=1}^{r} C_j \right| + \sum_{a \in A} y(a) = \left| \bigcup_{j=1}^{r} C_j \right| + l \cdot r - \mu. \end{aligned}$$

So

$$\mu \le \left| \bigcup_{j=1}^{r} C_j \right| - \sum_{j=1}^{r} (|C_j| - l), \tag{13.39}$$

and we have the required min-max equality. ∎

A similar formula holds for unions of arc-disjoint $s-t$ cuts:

Theorem 13.16. *Let $D = (V,A)$ be a digraph, $s,t \in V$, and $l \in \mathbb{Z}_+$. Then the minimum value of $|C_1| + \cdots + |C_l|$ where $C_1, \ldots, C_l$ are disjoint $s-t$ cuts is equal to the maximum value of*

$$\left|\bigcup \mathcal{P}\right| - \sum_{P \in \mathcal{P}} (|AP| - l), \tag{13.40}$$

where $\mathcal{P}$ ranges over all collections of $s-t$ paths.

Proof. By total unimodularity, the minimum size of the union of l disjoint $s-t$ cuts is equal to the minimum value of $\sum_{a \in A} x(a)$ where $x \in \mathbb{R}^A$ and $p \in \mathbb{R}^V$ such that

(13.41)
$$\begin{array}{ll} 0 \le x(a) \le 1 & \text{for each } a \in A, \\ p(v) - p(u) - x(a) \le 0 & \text{for each } a = (u,v) \in A; \\ p(t) - p(s) = l. & \end{array}$$

By LP-duality and total unimodularity, this is equal to the maximum value of $l \cdot r - \sum_{a \in A} y(a)$, where $r \in \mathbb{Z}$, $y \in \mathbb{Z}_+^A$, $f \in \mathbb{Z}_+^A$ such that

(13.42)
$$
\begin{array}{ll}
f(\delta^{\text{in}}(v)) - f(\delta^{\text{out}}(v)) = 0 & \text{for each } v \in V \setminus \{s,t\}, \\
f(\delta^{\text{in}}(t)) - f(\delta^{\text{out}}(t)) = r, & \\
f(a) - y(a) \leq 1 & \text{for each } a \in A.
\end{array}
$$

As f is an $s-t$ flow of value r, it is the sum of the incidence vectors of $s-t$ paths $P_1, \ldots, P_r$, say. Then

(13.43)
$$\sum_{a \in A} y(a) \geq \sum_{j=1}^{r} |AP_j| - \Big|\bigcup_{j=1}^{r} AP_j\Big|.$$

Hence we have the required equality. ∎

One may derive similarly min-max formulas for the minimum number of vertices in l internally vertex-disjoint $s-t$ paths and for the minimum number of vertices in l disjoint $s-t$ vertex-cuts.

Minimum-cost flow methods also provide fast algorithms to find optimum unions of disjoint paths:

Theorem 13.17. *Given a digraph $D = (V, A)$, $s, t \in V$, and $l \in \mathbb{Z}_+$, a collection of arc-disjoint $s-t$ paths $P_1, \ldots, P_l$ minimizing $|AP_1| + \cdots + |AP_l|$ can be found in time $O(lm)$.*

Proof. Directly from Theorems 12.6 and 11.1 and Corollary 7.8a. ∎

Similarly for disjoint cuts:

Theorem 13.18. *Given a digraph $D = (V, A)$, $s, t \in V$, $l \in \mathbb{Z}_+$, and a length function $k : A \to \mathbb{Q}_+$, a collection of arc-disjoint $s-t$ paths $P_1, \ldots, P_l$ minimizing $k(P_1) + \cdots + k(P_l)$ can be found in time $O(l(m + n \log n))$.*

Complexity survey for finding k arc-disjoint $s-t$ paths of minimum total length ($*$ indicates an asymptotically best bound in the table):

$*$	$O(k \cdot \text{SP}(n, m, L))$	Ford and Fulkerson [1958b], Jewell [1958], Busacker and Gowen [1960], Iri [1960]
	$O(nL \cdot \text{DP}_k(n, m))$	Edmonds and Karp [1972]
$*$	$O(n \log L \cdot \text{DP}_k(n, m))$	Röck [1980] (cf. Bland and Jensen [1992])

Here $\text{DP}_k(n, m)$ denotes the time needed to find k arc-disjoint disjoint $s-t$ paths in a digraph with n vertices and m edges.

Suurballe and Tarjan [1984] described an $O(m \log_{m/n} n)$ algorithm for finding, in a digraph with nonnegative length function and fixed vertex s, for all v a pair of edge-disjoint $s-v$ paths P_v, Q_v with $\text{length}(P_v) + \text{length}(Q_v)$ minimum.

Gabow [1983b,1985b] described minimum-cost flow algorithms for networks with unit capacities. The running times are $O(m^{7/4} \log L)$ and, if D is simple, $O(n^{1/3} m^{3/2} \log L)$. For the vertex-disjoint case, he gave algorithms with running time $O(n^{3/4}m \log L)$ and $O(nm \log_{2+\frac{m}{n}} L)$. Goldberg and Tarjan [1990] gave an $O(nm \log(nL))$ algorithm for minimum-cost flow with unit capacities. More complexity results follow from the table in Section 12.5a. Disjoint $s-t$ cuts were considered by Wagner [1990] and Talluri and Wagner [1994].

13.3. Network matrices

Let $D = (V, A)$ be a digraph and let $T = (V, A')$ be a directed tree. Let C be the $A' \times A$ matrix defined as follows. Take $a' \in A'$ and $a = (u, v) \in A$ and let P be the undirected $u - v$ path in T. Define

$$(13.44) \qquad C_{a',a} := \begin{cases} +1 & \text{if } a' \text{ occurs in forward direction in } P, \\ -1 & \text{if } a' \text{ occurs in backward direction in } P, \\ \ \ 0 & \text{if } a' \text{ does not occur in } P. \end{cases}$$

Matrix C is called a *network matrix, generated by* $T = (V, A')$ and $D = (V, A)$.

Theorem 13.19. *Any submatrix of a network matrix is again a network matrix.*

Proof. Deleting column indexed by $a \in A$ corresponds to deleting a from $D = (V, A)$. Deleting the row indexed by $a' = (u, v) \in A'$ corresponds to contracting a' in the tree $T = (V, A')$ and identifying u and v in D. ∎

The following theorem is implicit in Tutte [1965a]:

Theorem 13.20. *A network matrix is totally unimodular.*

Proof. By Theorem 13.19, it suffices to show that any square network matrix C has determinant 0, 1, or -1. We prove this by induction on the size of C, the case of 1×1 matrices being trivial. We use notation as above.

Assume that $\det C \neq 0$. Let u be an end vertex of T and let a' be the arc in T incident with u. By reversing orientations, we can assume that each arc in A and A' incident with u, has u as tail. Then, by definition of C, the row indexed by a' contains only 0's and 1's.

Consider two 1's in row a'. That is, consider two columns indexed by arcs $a_1 = (u, v_1)$ and $a_2 = (u, v_2)$ in A. Subtracting column a_1 from column a_2, has the effect of resetting a_2 to (v_1, v_2). So after that, column a_2 has a 0 in position a'. Since this subtraction does not change the determinant, we can assume that there is exactly one arc in A incident with u; that is, row a' has exactly one nonzero. Then by expanding the determinant by row a', we obtain inductively that $\det C = \pm 1$. ∎

The incidence matrix of a digraph $D = (V, A)$ is a network matrix: add a new vertex u to D giving digraph $D' = (V \cup \{u\}, A)$. Let T be the directed tree on $V \cup \{u\}$ with arcs (u, v) for $v \in V$. Then the network matrix generated by T and D' is equal to the incidence matrix of D.

Notes. Recognizing whether a given matrix is a network matrix has been studied by Gould [1958], Auslander and Trent [1959,1961], Tutte [1960,1965a,1967], Tomizawa [1976b], and Fujishige [1980a] (cf. Bixby and Wagner [1988] and Section 20.1 in Schrijver [1986b]).

Seymour [1980a] showed that all totally unimodular matrices can be obtained by glueing network matrices and copies of a certain 5×5 matrix together (cf. Schrijver [1986b] and Truemper [1992]).

13.4. Cross-free and laminar families

We now show how cross-free and laminar families of sets give rise to network matrices. The results in this section will be used mainly in Part V.

A family $\mathcal{C}$ of subsets of a finite set S is called *cross-free* if for all $X, Y \in \mathcal{C}$ one has

(13.45) $X \subseteq Y$ or $Y \subseteq X$ or $X \cap Y = \emptyset$ or $X \cup Y = S$.

$\mathcal{C}$ is called *laminar* if for all $X, Y \in \mathcal{C}$ one has

(13.46) $X \subseteq Y$ or $Y \subseteq X$ or $X \cap Y = \emptyset$.

So each laminar family is cross-free.

Cross-free families could be characterized geometrically as having a 'Venn-diagram' representation on the sphere without crossing lines. If the family is laminar we have such a representation in the plane.

A laminar collection $\mathcal{C}$ can be partitioned into 'levels': the ith level consists of all sets $X \in \mathcal{C}$ such that there are $i-1$ sets $Y \in \mathcal{C}$ satisfying $Y \supset X$. Then each level consists of disjoint sets, and for each set X of level $i+1$ there is a unique set of level i containing X.

Note that if $\mathcal{C}$ is a cross-free family, then adding, for each set $X \in \mathcal{C}$, the complement $S \setminus X$ to $\mathcal{C}$ maintains cross-freeness. Moreover, for any fixed $s \in S$, the family $\{X \in \mathcal{C} \mid s \notin X\}$ is laminar.

In order to relate cross-free families and directed trees, suppose that we have a directed tree $T = (V, A)$ and a function $\pi : S \to V$, for some set S. Then the pair T, π defines a family $\mathcal{C}$ of subsets of S as follows. Define for each arc $a = (u, v)$ of T the subset X_a of S by:

(13.47) $X_a :=$ the set of vertices in the weak component of $T - a$ containing v.

So X_a is the set of $s \in S$ for which arc a 'points' in the direction of $\pi(s)$ in T.

Let $\mathcal{C}_{T,\pi}$ be the family of sets X_a; that is,

(13.48) $\mathcal{C}_{T,\pi} := \{X_a \mid a \in A\}$.

If $\mathcal{C} = \mathcal{C}_{T,\pi}$, the pair T, π is called a *tree-representation* for $\mathcal{C}$. If moreover T is a rooted tree, then T, π is called a *rooted tree-representation* for $\mathcal{C}$.

It is easy to see that $\mathcal{C}_{T,\pi}$ is cross-free. Moreover, if T is a rooted tree, then $\mathcal{C}_{T,\pi}$ is laminar. In fact, each cross-free family has a tree-representation, and each laminar family has a rooted tree-representation, as is shown by the following theorem of Edmonds and Giles [1977]:

Theorem 13.21. *A family $\mathcal{C}$ of subsets of S is cross-free if and only if $\mathcal{C}$ has a tree-representation. Moreover, $\mathcal{C}$ is laminar if and only if $\mathcal{C}$ has a rooted tree-representation.*

Proof. As sufficiency of the conditions is easy, we show necessity. We first show that each laminar family $\mathcal{C}$ of subsets of a set S has a rooted tree-representation. The proof is by induction on $|\mathcal{C}|$, the case $\mathcal{C} = \emptyset$ being trivial. If $\mathcal{C} \neq \emptyset$, choose an inclusionwise minimal $X \in \mathcal{C}$. By induction, the family $\mathcal{C}' := \mathcal{C} \setminus \{X\}$ has a rooted tree-representation $T = (V, A)$, $\pi : S \to V$.

If $X = \emptyset$, then we can add to T a new arc from any vertex to a new vertex, to obtain a rooted tree-representation T', π of $\mathcal{C}$. So we can assume that $X \neq \emptyset$.

Now $|\pi(X)| = 1$, since if $\pi(x) \neq \pi(y)$ for some $x, y \in X$, then there is an arc a of T separating $\pi(x)$ and $\pi(y)$. Hence the set $X_a \in \mathcal{C}'$ contains one of $\pi(x)$ and $\pi(y)$, say $\pi(y)$. As $\mathcal{C}$ is laminar, this implies that X_a is properly contained in X, contradicting the minimality of X.

This proves that $|\pi(X)| = 1$. Let v be the vertex of T with $\pi(X) = \{v\}$. Augment T by a new vertex w and a new arc $b = (v, w)$. Reset $\pi(z) := w$ for each $z \in X$. Then the new tree and π form a rooted tree-representation for $\mathcal{C}$. This shows that each laminar family has a rooted tree-representation.

To see that each cross-free family $\mathcal{C}$ has a tree-representation, choose $s \in S$, and let $\mathcal{G}$ be obtained from $\mathcal{C}$ by replacing any set containing s by its complement. Then $\mathcal{G}$ is laminar, and hence it has a rooted tree-representation by the foregoing. Reversing arcs in the tree if necessary, it gives a tree-representation for $\mathcal{C}$. ∎

From Theorems 13.20 and 13.21 we derive the total unimodularity of certain matrices. Let $D = (V, A)$ be a directed graph and let $\mathcal{C}$ be a family of subsets of V. Let N be the $\mathcal{C} \times A$ matrix defined by:

(13.49) $$N_{X,a} := \begin{cases} 1 & \text{if } a \text{ enters } X, \\ -1 & \text{if } a \text{ leaves } X, \\ 0 & \text{otherwise,} \end{cases}$$

for $X \in \mathcal{C}$ and $a \in A$.

Corollary 13.21a. *If $\mathcal{C}$ is cross-free, then N is a network matrix, and hence N is totally unimodular.*

Proof. Let $T = (W, B)$, $\pi : V \to W$ be a tree-representation for $\mathcal{C}$. Let $D' = (W, A')$ be the directed graph with

(13.50) $$A' := \{(\pi(u), \pi(v)) \mid (u, v) \in A\}.$$

Then N is equal to the network matrix generated by T and D' (up to identifying any arc b of T with the set X_b in $\mathcal{C}$ determined by b, and any arc (u, v) of D with the arc $(\pi(u), \pi(v))$ of D'). Hence by Theorem 13.20, N is totally unimodular. ∎

Chapter 14

Partially ordered sets and path coverings

Partially ordered sets can be considered as a special type of networks, and several optimization problems on partially ordered sets can be handled with flow techniques. Basic theorem is Dilworth's min-max relation for the maximum size of an antichain.

14.1. Partially ordered sets

A *partially ordered set* is a pair $(S, \leq)$ where S is a set and where $\leq$ is a relation on S satisfying:

(14.1) (i) $s \leq s$,
(ii) if $s \leq t$ and $t \leq s$, then $s = t$,
(iii) if $s \leq t$ and $t \leq u$, then $s \leq u$,

for all $s, t, u \in S$. We put $s < t$ if $s \leq t$ and $s \neq t$. We restrict ourselves to *finite* partially ordered sets; that is, with S finite.

A subset C of S is called a *chain* if $s \leq t$ or $t \leq s$ for all $s, t \in C$. A subset A of S is called an *antichain* if $s \not< t$ and $t \not< s$ for all $s, t \in A$. Hence if C is a chain and A is an antichain, then

(14.2) $|C \cap A| \leq 1$.

First we notice the following easy min-max relation:

Theorem 14.1. *Let $(S, \leq)$ be a partially ordered set. Then the minimum number of antichains covering S is equal to the maximum size of a chain.*

Proof. That the maximum cannot be larger than the minimum follows easily from (14.2). To see that the two numbers are equal, define for any element $s \in S$ the *height* of s as the maximum size of any chain in S with maximum s. For any $i \in \mathbb{Z}_+$, let A_i denote the set of elements of height i. Let k be the maximum height of the elements of S. Then $A_1, \ldots, A_k$ are antichains covering S, and moreover there exists a chain of size k, since there exists an element of height k. ∎

This result can also be formulated in terms of graphs. Let $D = (V, A)$ be a digraph. A subset C of A is called a *directed cut* if there exists a subset U of V such that $\emptyset \neq U \neq V$, $\delta^{\text{out}}(U) = C$, and $\delta^{\text{in}}(U) = \emptyset$. Then Vidyasankar and Younger [1975] observed:

Corollary 14.1a. *Let $D = (V, A)$ be an acyclic digraph. Then the minimum number of directed cuts covering A is equal to the maximum length of a directed path.*

Proof. Define a partial order $\leq$ on A by: $a < a'$ if there exists a directed path traversing a and a', in this order. Applying Theorem 14.1 gives the Corollary. ∎

14.2. Dilworth's decomposition theorem

Dilworth [1950] proved that Theorem 14.1 remains true after interchanging the terms 'chain' and 'antichain', which is less simple to prove:

Theorem 14.2 (Dilworth's decomposition theorem)**.** *Let $(S, \leq)$ be a partially ordered set. Then the minimum number of chains covering S is equal to the maximum size of an antichain.*

Proof. That the maximum cannot be larger than the minimum follows easily from (14.2). To see that the two numbers are equal, we apply induction on $|S|$. Let α be the maximum size of an antichain and let A be an antichain of size α. Define

(14.3) $$A^{\downarrow} := \{s \in S \mid \exists t \in A : s \leq t\},$$
$$A^{\uparrow} := \{s \in S \mid \exists t \in A : s \geq t\}.$$

Then $A^{\downarrow} \cap A^{\uparrow} = A$ and $A^{\downarrow} \cup A^{\uparrow} = S$ (otherwise we can augment A).

First assume that $A^{\downarrow} \neq S$ and $A^{\uparrow} \neq S$. Then, by induction, $A^{\downarrow}$ can be covered by α chains. Since $A \subseteq A^{\downarrow}$, each of these chains contains exactly one element in A. For each $s \in A$, let C_s denote the chain containing s. Similarly, there exist α chains C'_s (for $s \in A$) covering $A^{\uparrow}$, where C'_s contains s. Then for each $s \in A$, $C_s \cup C'_s$ forms a chain in S, and moreover these chains cover S.

So we may assume that $A^{\downarrow} = S$ or $A^{\uparrow} = S$ for each antichain A of size α. It means that each antichain A of size α is either the set of minimal elements of S or the set of maximal elements of S. Now choose a minimal element s and a maximal element t of S with $s \leq t$. Then the maximum size of an antichain in $S \setminus \{s, t\}$ is equal to $\alpha - 1$ (since each antichain in S of size α contains s or t). By induction, $S \setminus \{s, t\}$ can be covered by $\alpha - 1$ chains. Adding the chain $\{s, t\}$ yields a covering of S by α chains. ∎

Notes. This proof is due to Perles [1963]. Dilworth original proof is based on a different induction. For a proof using linear programming duality, see Dantzig and Hoffman [1956]. For a deduction of Dilworth's decomposition theorem from Kőnig's matching theorem (Theorem 16.2), see Fulkerson [1956], Ford and Fulkerson [1962] (pp. 61–64), and Mirsky and Perfect [1966]. Further proofs were given by Dilworth [1960], Tverberg [1967], and Pretzel [1979].

14.3. Path coverings

Dilworth's decomposition theorem can be formulated equivalently in terms of covering vertices of a digraph[22]:

Corollary 14.2a. *Let $D = (V, A)$ be an acyclic digraph. Then the minimum number of paths covering all vertices is equal to the maximum number of vertices no two of which belong to a directed path.*

Proof. Apply Dilworth's decomposition theorem to the partially ordered set $(V, \leq)$ where $u \leq v$ if and only if v is reachable in D from u. ∎

As for covering the arcs, we have:

Corollary 14.2b. *Let $D = (V, A)$ be an acyclic digraph. Then the minimum number of paths covering all arcs is equal to the maximum size of a directed cut.*

Proof. Apply Dilworth's decomposition theorem to the partially ordered set $(A, \leq)$ where $a < a'$ if and only if there exists a directed path traversing a and a', in this order. ∎

Similarly, for $s - t$ paths:

Corollary 14.2c. *Let $D = (V, A)$ be an acyclic digraph with exactly one source, s, and exactly one sink, t. Then the minimum number of $s - t$ paths covering A is equal to the maximum size of a directed $s - t$ cut.*

Proof. Apply Dilworth's decomposition theorem to the partially ordered set $(A, \leq)$ defined by: $a \leq a'$ if and only if there exists an $s - t$ path traversing a and a', in this order. ∎

If only a subset of the arcs has to be covered, one has more generally:

Corollary 14.2d. *Given an acyclic digraph $D = (V, A)$ and $B \subseteq A$, the minimum number of paths covering B is equal to the maximum of $|C \cap B|$ where C is a directed cut.*

[22] Gallai and Milgram [1960] claim to have found this result in 1947.

Proof. Consider the partially ordered set $(B, \leq)$ with $a < a'$ if there exists a directed path traversing a and a', in this order. Then for each chain K in $(B, \leq)$ there is a path in D covering K, and for each antichain L in $(B, \leq)$ there is a directed cut C in D with $L \subseteq C \cap B$. Hence the theorem follows from Dilworth's decomposition theorem. ∎

14.4. The weighted case

Dilworth's decomposition theorem has a self-refining nature, and implies a weighted version. Let $(S, \leq)$ be a partially ordered set. Let $\mathcal{C}$ and $\mathcal{A}$ denote the collections of chains and antichains in $(S, \leq)$, respectively. Let $w : S \to \mathbb{Z}_+$ be a 'weight' function. Then:

Theorem 14.3. *The maximum weight $w(A)$ of an antichain A is equal to the minimum size of a family of chains covering each element s exactly $w(s)$ times.*

Proof. Replace each element s of S by $w(s)$ copies, making the set S'. For any copy s' of s and t' of t, define $s' <' t'$ if and only if $s < t$. This gives the partially ordered set $(S', \leq')$. Note that the copies of one element of S form an antichain in S'.

Then the maximum weight $w(A)$ of an antichain A in S is equal to the maximum size $|A'|$ of an antichain A' in S'. By Dilworth's decomposition theorem, S' can be covered by a collection Λ of $|A'|$ chains. Replacing the elements of each chain by their originals in S, gives the required equality. ∎

In terms of digraphs this gives the following result of Gallai [1958a,1958b]:

Corollary 14.3a. *Let $D = (V, A)$ be an acyclic digraph and let S and T be subsets of V such that each vertex is on at least one $S - T$ path. Let $c \in \mathbb{Z}_+^V$. Then the minimum number k of $S - T$ paths $P_1, \ldots, P_k$ such that each vertex v is covered at least $c(v)$ times by the P_i is equal to the maximum of $c(U)$ where U is a set of vertices intersecting each $S - T$ path at most once.*

Proof. Directly from Theorem 14.3 by defining the partially ordered set $(V, \leq)$ by: $u \leq v$ if and only if there exists a $u - v$ path. ∎

Similarly, there is the following 'min-flow max-cut theorem':

Corollary 14.3b. *Let $D = (V, A)$ be an acyclic digraph with exactly one source, s, and exactly one sink, t. Let $d : A \to \mathbb{R}_+$. Then the minimum value of any $s - t$ flow f satisfying $f \geq d$ is equal to the maximum value of $d(C)$ where C is a directed $s - t$ cut. If d is integer, we can take f integer.*

Proof. Define a partial order $\leq$ on A by: $a < a'$ if there is an $s-t$ path traversing a and a' in this order. Then any chain in A is contained in some $s-t$ path and any antichain is contained in some directed $s-t$ cut. Hence this Corollary can be derived from Theorem 14.3 (using continuity, compactness, and scaling). ∎

A similar weighted variant of the easier Theorem 14.1 holds: Again, let $(S, \leq)$ be a partially ordered set. Let $w : S \to \mathbb{Z}_+$ be a 'weight' function.

Theorem 14.4. *The maximum weight $w(C)$ of any chain is equal to the minimum size of a family of antichains covering each element s exactly $w(s)$ times.*

Proof. Similar to the proof of Theorem 14.3. ∎

The following 'length-width inequality' follows similarly:

Theorem 14.5. *Let $(S, \leq)$ be a partially ordered set and let $l, w : S \to \mathbb{R}_+$. Then*

$$\max_{C \text{ chain}} l(C) \cdot \max_{A \text{ antichain}} w(A) \geq \sum_{s \in S} l(s)w(s). \tag{14.4}$$

Proof. We can assume that l and w are rational (by continuity), and hence integer. Let t be the maximum of $l(C)$ taken over all chains C.

For each $s \in S$, let $h(s)$ be the maximum of $l(C)$ taken over all chains C with maximum element s. For each $k \in \mathbb{Z}$, let A_k be the set of those elements $s \in S$ with $h(s) - l(s) < k \leq h(s)$.

Then each A_k is an antichain. Moreover, $A_k = \emptyset$ if $k > t$, and

$$\sum_{k=1}^{t} \chi^{A_k} = l. \tag{14.5}$$

Therefore,

$$\begin{aligned}\max_C l(C) \cdot \max_A w(A) = t \cdot \max_A w(A) \geq \sum_{k=1}^{t} w(A_k) = \sum_{k=1}^{t} w^{\mathsf{T}} \chi^{A_k} \\ = w^{\mathsf{T}} l,\end{aligned} \tag{14.6}$$

where C and A range over chains and antichains, respectively. ∎

14.5. The chain and antichain polytopes

Let $(S, \leq)$ be a partially ordered set. The *chain polytope* $P_{\text{chain}}(S)$ of S is the convex hull of the incidence vectors (in $\mathbb{R}^S$) of chains in S. Similarly, the *antichain polytope* $P_{\text{antichain}}(S)$ of S is the convex hull of the incidence vectors (in $\mathbb{R}^S$) of antichains in S.

These two polytopes turn out to form an antiblocking pair of polyhedra. To see this, we first show:

Corollary 14.5a. *The chain polytope of S is determined by*

$$\text{(14.7)} \qquad \begin{array}{rll} \text{(i)} & 0 \leq x_s \leq 1 & \text{for each } s \in S, \\ \text{(ii)} & x(A) \leq 1 & \text{for each antichain } A, \end{array}$$

and this system is TDI.

Proof. Directly from Theorem 14.4, by LP-duality: for any chain C, χ^C is a feasible solution of (14.7). An antichain family covering each element s precisely $w(s)$ times gives a dual feasible solution. As the minimum of $w(C)$ is equal to the maximum value of these dual feasible solutions (by Theorem 14.4), the linear program of minimizing $w^\mathsf{T} x$ over (14.7) has integer optimum primal and dual solutions. ∎

Similarly for the antichain polytope:

Theorem 14.6. *The antichain polytope of S is determined by the inequalities*

$$\text{(14.8)} \qquad \begin{array}{rll} \text{(i)} & 0 \leq x_s \leq 1 & \text{for each } s \in S, \\ \text{(ii)} & x(C) \leq 1 & \text{for each chain } C, \end{array}$$

and this system is TDI.

Proof. Similar to the previous proof, now using Theorem 14.3. ∎

This implies that the chain and antichain polytope are related by the antiblocking relation:

Corollary 14.6a. *$P^{\uparrow}_{\text{chain}}(S)$ and $P^{\uparrow}_{\text{antichain}}(S)$ form an antiblocking pair of polyhedra.*

Proof. Directly from the previous results. ∎

14.5a. Path coverings algorithmically

When studying partially ordered sets $(S, \leq)$ algorithmically, we should know how these are represented. Generally, giving all pairs (s,t) with $s \leq t$ yields a large, redundant input. It suffices to give an acyclic digraph $D = (S, A)$ such that $s \leq t$ if and only if t is reachable from s. So it is best to formulate the algorithmic results in terms of acyclic digraphs.

The strong polynomial-time solvability of the problems discussed below follow from the strong polynomial-time solvability of minimum-cost circulation. We give some better running time bounds.

Theorem 14.7. *Given an acyclic digraph $D = (V, A)$ and $B \subseteq A$, a minimum number of paths covering B can be found in time $O(nm)$.*

Proof. Add two vertices s and t to D, and, for each vertex v, make $\deg^{\text{in}}_B(v)$ parallel arcs from s to v and $\deg^{\text{out}}_B(v)$ parallel arcs from v to t. Let $D' = (V', A')$ be the extended graph. By Theorem 9.10, we can find in time $O(nm)$ a maximum collection $P_1, \ldots, P_k$ of $s-t$ paths that are disjoint on the set $A' \setminus A$ of new arcs.

We make another auxiliary graph $\widetilde{D} = (V, \widetilde{A})$ as follows. Each arc in B belongs to $\widetilde{A}$. Moreover, for each $a = (u, v) \in A$, we make r parallel arcs from u to v, where r is the number of times a is traversed by the P_i. (So if $a \in B$, there are $r+1$ parallel arcs from u to v.) This gives the acyclic graph $\widetilde{D}$. Now choose, repeatedly as long as possible, in $\widetilde{D}$ a path from a (current) source to a (current) sink and remove its arcs. This gives us a collection of paths in $\widetilde{D}$ and hence also in D, covering all arcs in B. We claim that it has minimum size.

For each i, let P_i' be the path obtained from P_i by deleting the first and last arc. For each $v \in V$, let $\sigma(v)$ be the number of P_i that start with (s, v) and let $\tau(v)$ be the number of P_i that and with (v, t).

Let $U \subseteq V$ give a minimum $s-t$ cut $\delta^{\text{out}}_{A'}(U \cup \{s\})$ in D' with $\delta^{\text{out}}_A(U) = \emptyset$. As $P_1, \ldots, P_k$ form a maximum $s-t$ path packing in D', we have for each $v \in V$:

(14.9) $\sigma(v) \le \delta^{\text{in}}_B(v)$, with equality if $v \in V \setminus U$,
$\tau(v) \le \delta^{\text{out}}_B(v)$, with equality if $v \in U$.

So

$$\text{(14.10)}\qquad \begin{aligned}\deg^{\text{in}}_{\widetilde{A}}(v) &= \deg^{\text{in}}_B(v) + \sum_{i=1}^{k} \deg^{\text{in}}_{AP_i'}(v) \ge \sigma(v) + \sum_{i=1}^{k} \deg^{\text{in}}_{AP_i'}(v) \\ &= \sum_{i=1}^{k} \deg^{\text{in}}_{AP_i}(v),\end{aligned}$$

with equality if $v \in V \setminus U$. Similarly,

$$\text{(14.11)}\qquad \begin{aligned}\deg^{\text{out}}_{\widetilde{A}}(v) &= \deg^{\text{out}}_B(v) + \sum_{i=1}^{k} \deg^{\text{out}}_{AP_i'}(v) \ge \tau(v) + \sum_{i=1}^{k} \deg^{\text{out}}_{AP_i'}(v) \\ &= \sum_{i=1}^{k} \deg^{\text{out}}_{AP_i}(v),\end{aligned}$$

with equality if $v \in U$. Hence, since $\deg^{\text{in}}_{AP_i}(v) = \deg^{\text{out}}_{AP_i}(v)$ for each $v \in V$ and each $i = 1, \ldots, k$:

(14.12) $\deg^{\text{in}}_{\widetilde{D}}(v) \ge \deg^{\text{out}}_{\widetilde{D}}(v)$ for each $v \in U$ and
$\deg^{\text{in}}_{\widetilde{D}}(v) \le \deg^{\text{out}}_{\widetilde{D}}(v)$ for each $v \in V \setminus U$.

Now deleting any source-sink path P in $\widetilde{D}$ does not invalidate (14.12). Moreover, P runs from $V \setminus U$ to U. Since none of the arcs in $\delta^{\text{in}}_A(U)$ are traversed by any P_i, we know that P should use an arc of $B \cap \delta^{\text{in}}(U)$. So the number of paths found is at most $|B \cap \delta^{\text{in}}_A(U)|$. Therefore, by Corollary 14.2d, the paths form a minimum-size collection of paths covering B. ∎

The special case where *all* arcs must be covered, is:

Corollary 14.7a. *Given an acyclic digraph $D = (V, A)$, a minimum collection of paths covering all arcs can be found in time $O(nm)$.*

Proof. Directly from the foregoing, by taking $B := A$. ∎

The theorem also applies to vertex coverings:

Corollary 14.7b. *Given an acyclic digraph $D = (V, A)$, a minimum number of paths covering all vertices can be found in time $O(nm)$.*

Proof. Introduce for each vertex v of D vertices v' and v''. Define $V' := \{v' \mid v \in V\}$, $V'' := \{v'' \mid v \in V\}$, $A' := \{(v', v'') \mid v \in V\}$, and $A'' := \{(u'', v') \mid (u, v) \in A\}$.

Then a minimum cover of A' by paths in the new graph $(V' \cup V'', A' \cup A'')$ gives a minimum cover of V by paths in the original graph. ∎

It is trivial to extend the results to $s - t$ paths:

Corollary 14.7c. *Given an acyclic digraph $D = (V, A)$ and $s, t \in V$, a minimum collection of $s - t$ paths covering all arcs can be found in time $O(nm)$.*

Proof. We may assume that each arc of D is contained in at least one $s - t$ path. But then each path can be extended to an $s - t$ path, and hence a minimum collection of paths gives a minimum collection of $s - t$ paths. ∎

One similarly has for covering the vertices:

Corollary 14.7d. *Given an acyclic digraph $D = (V, A)$ and $s, t \in V$, a minimum collection of $s - t$ paths covering all vertices can be found in time $O(nm)$.*

Proof. Similar to the proof of Corollary 14.7b. ∎

These bounds are best possible, as the size of the output is $\Omega(nm)$. As for paths covering the arcs, this can be seen by taking vertices $v_1, \ldots, v_n$, with r parallel arcs from v_1 to v_2, r parallel arcs from v_{n-1} to v_n, and one arc from v_{i-1} to v_i for each $i = 2, \ldots, n-1$. Then the number of arcs is $2r + n - 2$, while any minimum path covering of the arcs consists of r paths of length $n - 1$ each.

14.6. Unions of directed cuts and antichains

The following theorem is (in the terminology of partially ordered sets — see Corollary 14.8a) due to Greene and Kleitman [1976]. We follow the proof method of Fomin [1978] and Frank [1980a] based on minimum-cost circulations.

Theorem 14.8. *Let $D = (V, A)$ be an acyclic digraph, let $B \subseteq A$, and let $k \in \mathbb{Z}_+$. Then the maximum of $|B \cap \bigcup \mathcal{C}|$, where $\mathcal{C}$ is a collection of at most k directed cuts is equal to the minimum value of*

(14.13) $\quad |B \setminus \bigcup \mathcal{P}| + k \cdot |\mathcal{P}|,$

where $\mathcal{P}$ is a collection of directed paths.

Proof. To see max ≤ min, let $\mathcal{C}$ be a collection of at most k directed cuts and let $\mathcal{P}$ be a collection of directed paths. Then, setting $\Gamma := \bigcup \mathcal{C}$ and $\Pi := \bigcup \mathcal{P}$,

(14.14) $\quad |B \cap \Gamma| \leq |B \setminus \Pi| + |\Gamma \cap \Pi| \leq |B \setminus \Pi| + k \cdot |\mathcal{P}|.$

(Note that any directed path P intersects any directed cut in at most one edge; hence $|\Gamma \cap AP| \leq k$ for each $P \in \mathcal{P}$.)

In proving equality, we may assume that D has exactly one source, s say, and exactly one sink, t say. (Adding s to V, and all arcs (s, v) for $v \in V$ does not change the theorem. Similarly for adding a sink.)

Define for each $a \in A$, a capacity $c(a) := \infty$ and a cost $l(a) := 0$. For each arc $a = (u, v) \in B$, introduce a new arc $a' = (u, v)$ parallel to a, with $c(a) := 1$ and $l(a) := -1$. Finally, add an arc (t, s), with $c(t, s) := \infty$ and $l(t, s) := k$. This makes the digraph $\widetilde{D} = (V, \widetilde{A})$.

Let $f : \widetilde{A} \to \mathbb{Z}$ be a minimum-cost nonnegative circulation in $\widetilde{D}$ subject to c. As D_f has no negative-cost directed circuits (Theorem 12.1), there exists a function $p : V \to \mathbb{Z}$ such that for each $a = (u, v) \in A$:

(14.15) $\quad p(v) \leq p(u)$, with equality if $f(a) \geq 1$.

Moreover, for each $a = (u, v) \in B$:

(14.16)
$$p(v) \leq p(u) - 1 \text{ if } f(a') = 0,$$
$$p(v) \geq p(u) - 1 \text{ if } f(a') = 1.$$

Finally,

(14.17) $\quad p(s) \leq p(t) + k$, with equality if $f(t, s) \geq 1$.

We may assume that $p(t) = 0$. So by (14.15), $p(s) \geq 0$ and by (14.17), $p(s) \leq k$. For each $i = 1, \ldots, p(s)$, let $U_i := \{v \in V \mid p(v) \geq i\}$. Then for each i, $\delta_A^{\text{out}}(U_i)$ is a directed $s - t$ cut, since $s \in U_i$, $t \notin U_i$, and no arc in A enters U_i: if $(u, v) \in A$ with $v \in U_i$, then $p(v) \geq i$, and hence by (14.15), $p(u) \geq i$, that is $u \in U_i$.

Let $\mathcal{C}$ be the collection of these directed cuts and let $\Gamma := \bigcup \mathcal{C}$. We can decompose f as a sum of incidence vectors of directed circuits in $\widetilde{D}$. Each of these circuits contains exactly one arc (t, s). Deleting it, and identifying any a' with a (for $a \in B$) gives a path collection $\mathcal{P}$ in D. Let $\Pi := \bigcup \mathcal{P}$.

Then $B \setminus \Pi = B \cap \Gamma \setminus \Pi$. For let $a = (u, v) \in B \setminus \Pi$. Then $f(a') = 0$, and hence by (14.16), $p(v) \leq p(u) - 1$. Hence $a \in \delta^{\text{out}}(U_i)$ for $i := p(u)$. So $a \in \Gamma$.

Moreover,

(14.18)
$$\begin{aligned} k \cdot |\mathcal{P}| &= (p(s) - p(t)) f(t, s) \\ &= \sum_{a=(u,v) \in A} (p(u) - p(v)) f(a) + \sum_{a=(u,v) \in B} (p(u) - p(v)) f(a') \\ &= \sum_{a=(u,v) \in B} (p(u) - p(v)) f(a') = |B \cap \Gamma \cap \Pi|. \end{aligned}$$

Thus $|B \setminus \Pi| + k \cdot |\mathcal{P}| = |B \setminus \Pi| + |B \cap \Gamma \cap \Pi| = |B \cap \Gamma \setminus \Pi| + |B \cap \Gamma \cap \Pi| = |B \cap \Gamma|$. ∎

This implies for partially ordered sets:

Corollary 14.8a. *Let $(S, \leq)$ be a partially ordered set and let $k \in \mathbb{Z}_+$. Then the maximum size of the union of k antichains is equal to the minimum value of*

$$\sum_{C \in \mathcal{C}} \min\{k, |C|\}, \tag{14.19}$$

where $\mathcal{C}$ ranges over partitions of S into chains.

Proof. This can be reduced to Theorem 14.8, by making a digraph $D = (V, A)$ as follows. Let for each $s \in S$, s' be a copy of s, and let $V := S \cup \{s' \mid s \in S\}$. Let A consist of all pairs (s, s') with $s \in S$ and all pairs (s', t) with $s < t$. Taking $B := \{(s, s') \mid s \in S\}$ reduces Corollary 14.8a to Theorem 14.8. (For each arcs (s, s') in $B \setminus \bigcup \mathcal{P}$, we take a singleton $C = \{s\}$.) ∎

Corollary 14.8a can be stated in a slightly different form. For any partially ordered set $(S, \leq)$, any $Y \subseteq S$, and any $k \in \mathbb{Z}_+$, let

$$a_k(Y) := \max\{|Z| \bigm| Z \subseteq Y \text{ is the union of } k \text{ antichains}\}. \tag{14.20}$$

Then:

Corollary 14.8b. $a_k(S) = \min\limits_{Y \subseteq S}(|S \setminus Y| + k \cdot a_1(Y))$.

Proof. The inequality $\leq$ follows from the fact that if Z is the union of k antichains and $Y \subseteq S$, then $|Z| \leq |Z \setminus Y| + a_k(Z \cap Y) \leq |S \setminus Y| + k \cdot a_1(Y)$.

To obtain equality, let $\mathcal{C}$ be a partition of S into chains attaining the minimum in Corollary 14.8a. Let $\mathcal{C}'$ be the collection of those chains $C \in \mathcal{C}$ with $|C| \geq k$. Let Y be the union of the chains in $\mathcal{C}'$. Then (14.19) is equal to $|S \setminus Y| + k|\mathcal{C}'|$. This is at least $|S \setminus Y| + k \cdot a_1(Y)$ (as $|\mathcal{C}'| \geq a_1(Y)$). Thus we have equality. ∎

Note that the proof of Theorem 14.8 gives a polynomial-time algorithm to find a maximum union of k directed cuts or antichains. For a proof of the results in this section based on LP-duality, see Hoffman and Schwartz [1977]. For other proofs, see Saks [1979]. For extensions, see Linial [1981] and Cameron [1986].

14.6a. Common saturating collections of chains

Greene and Kleitman [1976] also showed that for each h there is a chain partition $\mathcal{C}$ of a partially ordered set $(S, \leq)$ attaining the minimum of (14.19) both for $k = h$ and for $k = h + 1$.

More generally, in terms of acyclic digraphs, there is the following result of Greene and Kleitman [1976] on the minimum in Theorem 14.8:

Theorem 14.9. *Let $D = (V, A)$ be an acyclic digraph, let $B \subseteq A$, and let $h \in \mathbb{Z}_+$. Then there is a collection $\mathcal{P}$ of directed paths attaining*

(14.21) $$\min_{\mathcal{P}}(|B \setminus \bigcup \mathcal{P}| + k \cdot |\mathcal{P}|)$$

both for $k = h$ and for $k = h + 1$.

Proof. It suffices to show that in the proof of Theorem 14.8 the minimum-cost circulation f can be chosen such that it has minimum cost simultaneously with respect to the given cost function l, and with respect to the cost function l' which is the same as l except that $l'(t, s) = k + 1$.

For choose f such that it has minimum cost with respect to l, and with $l'^{\mathsf{T}} f$ as small as possible. Suppose that f does *not* have minimum cost with respect to l'. Then D_f has a directed circuit C with $l'(C) < 0$. As $l(C) \geq 0$, C traverses (s, t). So $l(C) - l'(C) = l(s, t) - l'(s, t) = 1$, and therefore $l(C) = 0$. So $f' := f + \chi^C$ is a feasible circulation with $l^{\mathsf{T}} f' = l^{\mathsf{T}} f$ and $l'^{\mathsf{T}} f' < l'^{\mathsf{T}} f$. This contradicts our assumption. ∎

For partially ordered sets it gives (using definition (14.20)):

Corollary 14.9a. *Let $(S, \leq)$ be a partially ordered set and let $k \in \mathbb{Z}_+$. Then there exists a chain partition $\mathcal{C}$ of S such that*

(14.22) $$a_k(S) = \sum_{C \in \mathcal{C}} \min\{k, |C|\} \text{ and } a_{k+1}(S) = \sum_{C \in \mathcal{C}} \min\{k + 1, |C|\}.$$

Proof. Directly from Theorem 14.9. ∎

(For a linear programming proof and an extension, see Hoffman and Schwartz [1977]. For another proof, see Perfect [1984]. Denig [1981] showed that the common saturating chain collections determine a matroid.)

14.7. Unions of directed paths and chains

Results dual to those of the previous sections were obtained by Greene [1976] and Edmonds and Giles [1977]. They can be formulated by interchanging the terms 'chain' and 'antichain'. Again we follow the proof method of Fomin [1978] and Frank [1980a] based on minimum-cost flows.

Theorem 14.10. *Let $D = (V, A)$ be an acyclic digraph, let $B \subseteq A$, and let $k \in \mathbb{Z}_+$. Then the maximum of $|B \cap \bigcup \mathcal{P}|$, where $\mathcal{P}$ is a collection of the arc sets of at most k directed paths, is equal to the minimum value of*

(14.23) $$|B \setminus \bigcup \mathcal{C}| + k \cdot |\mathcal{C}|,$$

where $\mathcal{C}$ is a collection of directed cuts.

Proof. The inequality max $\leq$ min is shown similarly as in Theorem 14.8. In proving the theorem, we may again assume that D has only one source, s say, and only one sink, t say.

To obtain equality, we again consider the extended graph $\widetilde{D}$ as in the proof of Theorem 14.8, with capacity c and cost l, except that we delete arc (t, s).

Let $f : \widetilde{A} \to \mathbb{Z}$ be a minimum-cost $s-t$ flow in $\widetilde{D}$ of value k subject to c. As D_f has no negative-cost directed circuits, there exists a function $p : V \to \mathbb{Z}$ such that for each $a = (u, v) \in A$:

(14.24) $\quad p(v) \leq p(u)$, with equality if $f(a) \geq 1$.

Moreover, for each $a = (u, v) \in B$:

(14.25) $\quad p(v) \leq p(u) - 1$ if $f(a') = 0$,
$\quad p(v) \geq p(u) - 1$ if $f(a') = 1$.

We may assume that $p(t) = 0$. By (14.24), $p(s) \geq 0$. For each $i = 1, \ldots, p(s)$, let $U_i := \{v \in V \mid p(v) \geq i\}$. Then for each i, $\delta^{\text{out}}(U_i)$ is a directed cut, since $s \in U_i$, $t \notin U_i$, and no arc in A enters U_i: if $(u, v) \in A$ with $v \in U_i$, then $p(v) \geq i$, and hence by (14.24), $p(u) \geq i$, that is $u \in U_i$.

Let $\mathcal{C}$ be the collection of these directed cuts and let $\Gamma := \bigcup \mathcal{C}$. We can decompose f as a sum of incidence vectors of k directed paths in $\widetilde{D}$. Identifying any a' with a (for $a \in B$) this gives a collection $\mathcal{P}$ of $s-t$ paths. Let $\Pi := \bigcup_{P \in \mathcal{P}} AP$.

Then $B \setminus \Pi \subseteq \Gamma$. For let $a = (u, v) \in B \setminus \Pi$. Then $f(a') = 0$, and hence by (14.25), $p(v) \leq p(u) - 1$. Hence $a \in \delta^{\text{out}}(U_i)$ for $i = p(u)$. So $a \in \Gamma$.

Moreover,

$$
\begin{aligned}
(14.26) \qquad k \cdot |\mathcal{C}| &= k(p(s) - p(t)) \\
&= \sum_{a=(u,v) \in A} (p(u) - p(v)) f(a) + \sum_{a=(u,v) \in B} (p(u) - p(v)) f(a') \\
&= \sum_{a \in B} (p(u) - p(v)) f(a') = |B \cap \Gamma \cap \Pi|.
\end{aligned}
$$

So $|B \setminus \Gamma| + k|\mathcal{C}| = |B \setminus \Gamma| + |B \cap \Gamma \cap \Pi| = |B \cap \Pi \setminus \Gamma| + |B \cap \Gamma \cap \Pi| = |B \cap \Pi|$. ∎

This implies for partially ordered sets:

Corollary 14.10a. *Let $(S, \leq)$ be a partially ordered set and let $k \in \mathbb{Z}_+$. Then the maximum size of the union of k chains is equal to the minimum value of*

$$(14.27) \qquad \sum_{A \in \mathcal{A}} \min\{k, |A|\},$$

where $\mathcal{A}$ ranges over partitions of S into antichains.

Proof. Similar to the proof of Corollary 14.8a. ∎

Like the result in the previous section, also this theorem can be stated in a different form. For any partially ordered set $(S, \leq)$, $Y \subseteq S$ and $k \in \mathbb{Z}_+$, let

$$c_k(Y) := \max\{|Z| \mid Z \subseteq Y \text{ is the union of } k \text{ chains}\}. \tag{14.28}$$

Then:

Corollary 14.10b. $c_k(S) = \min_{Y \subseteq S}(|S \setminus Y| + k \cdot c_1(Y))$.

Proof. Similar to the proof of Corollary 14.8b. ∎

Note that the proof method gives a polynomial-time algorithm to find a maximum union of k paths or chains. A weighted version was given by Edmonds and Giles [1977]. For an extension, see Hoffman [1983].

14.7a. Common saturating collections of antichains

Similar results to those in Section 14.6a were obtained for antichain partitions by Greene [1976]. Consider the proof of Theorem 14.10. By Theorem 12.5, there exist minimum-cost flows f and f' of values k and $k + 1$ respectively and a function $p : V \to \mathbb{Z}$ that is both a potential for f and for f'.

This implies the following result of Greene [1976] on the minimum in Theorem 14.10:

Theorem 14.11. *Let $D = (V, A)$ be an acyclic digraph, let $B \subseteq A$, and let $h \in \mathbb{Z}_+$. Then there is a collection $\mathcal{C}$ of directed cuts attaining*

$$\min_{\mathcal{C}}(|B \setminus \bigcup\mathcal{C}| + k \cdot |\mathcal{C}|) \tag{14.29}$$

both for $k = h$ and for $k = h + 1$.

Proof. Directly from the foregoing observation. ∎

For partially ordered sets it gives (using definition (14.28)):

Corollary 14.11a. *Let $(S, \leq)$ be a partially ordered set and let $k \in \mathbb{Z}_+$. Then there exists an antichain partition $\mathcal{A}$ of S such that*

$$c_k(S) = \sum_{A \in \mathcal{A}} \min\{k, |A|\} \text{ and } c_{k+1}(S) = \sum_{A \in \mathcal{A}} \min\{k+1, |A|\}. \tag{14.30}$$

Proof. Directly from Theorem 14.11. ∎

More on this can be found in Perfect [1984]. For more on chain and antichain partitions, see Frank [1980a].

14.7b. Conjugacy of partitions

Greene [1976] showed that the numbers studied above give so-called 'conjugate' partitions, implying that in fact the results of Section 14.6 and those of 14.7 can be derived from each other.

Fix a partially ordered set $(S, \leq)$. For each $k = 0, 1, 2, \ldots$, let a_k be the maximum size of the union of k antichains in S and let c_k be the maximum size of the union of k chains in S.

Then Corollary 14.8a is equivalent to:

(14.31) $$a_k = |S| + \min_{p \geq 0}(kp - c_p).$$

Similarly, Corollary 14.10a is equivalent to:

(14.32) $$c_k = |S| + \min_{p \geq 0}(kp - a_p).$$

Define for each $k = 1, 2, \ldots$:

(14.33) $$\alpha_k := a_k - a_{k-1} \text{ and } \gamma_k := c_k - c_{k-1}.$$

Trivially, each α_k and γ_k is nonnegative, and both $\alpha_1, \alpha_2, \ldots$ and $\gamma_1, \gamma_2, \ldots$ are partitions of the number $|S|$. In fact:

Theorem 14.12. *$\alpha_1 \geq \alpha_2 \geq \cdots$ and $\gamma_1 \geq \gamma_2 \geq \cdots$.*

Proof. For each $k \geq 1$, one has $\alpha_k \geq \alpha_{k+1}$; equivalently

(14.34) $$a_{k+1} + a_{k-1} \leq 2a_k.$$

Indeed, by Corollary 14.8a there is a collection $\mathcal{C}$ of chains satisfying

(14.35) $$a_k = \sum_{C \in \mathcal{C}} \min\{k, |C|\}.$$

Then

(14.36) $$2a_k = \sum_{C \in \mathcal{C}} 2\min\{k, |C|\} \geq \sum_{C \in \mathcal{C}} (\min\{k-1, |C|\} + \min\{k+1, |C|\}) \\ \geq a_{k-1} + a_{k+1}.$$

The second part of the theorem is shown similarly (with Corollary 14.10a). ∎

In fact, the partitions $(\alpha_1, \alpha_2, \ldots)$ and $(\gamma_1, \gamma_2, \ldots)$ of $|S|$ are *conjugate*. To show this, we mention some of the theory of partitions of numbers.

Let $\nu_1 \geq \nu_2 \geq \cdots$ be integers forming a partition of the number n; that is, $\nu_1 + \nu_2 + \cdots = n$. (So $\nu_k = 0$ for almost all k.) The *conjugate* partition (ν_p^*) of (ν_k) is defined by:

(14.37) $$\nu_p^* := \max\{k \mid \nu_k \geq p\}$$

for $p = 1, 2, \ldots$. Then it is easy to see that $\nu_1^* \geq \nu_2^* \geq \cdots$, and that

(14.38) $$\text{for all } p, k \geq 1\text{: } p \leq \nu_k \iff k \leq \nu_p^*.$$

The conjugate partition can be interpreted in terms of the 'Young diagram'. The *Young diagram* F of (ν_k) is the collection of pairs (x, y) of natural numbers $x, y \geq 1$ satisfying $y \leq \nu_x$. So the Young diagram uniquely determines the sequence (ν_k). The number of pairs in the Young diagram is equal to n. Now the Young diagram F^* of the conjugate partition (ν_p^*) satisfies

(14.39) $F^* = \{(y,x) \mid (x,y) \in F\}$.

This follows directly from (14.38). It implies that the conjugate partition (ν_p^*) is again a partition of n, and that the conjugate of (ν_p^*) is (ν_k).

The following interprets conjugacy of partitions in terms of their partial sums. Let $\nu_1 \geq \nu_2 \geq \cdots$ and $\nu_1' \geq \nu_2' \geq \cdots$ be partitions of n. For each $k = 0, 1, \ldots$, let

(14.40) $n_k := \nu_1 + \cdots + \nu_k$ and $n_k' := \nu_1' + \cdots + \nu_k'$.

Then:

Lemma 14.13α. *(ν_k) and (ν_k') are conjugate partitions if and only if*

(14.41) $$n_p' = n + \min_{k \geq 0}(pk - n_k)$$

for each $p = 0, 1, 2, \ldots$.

Proof. First note that, for each $p = 1, 2, \ldots$,

(14.42) $$\min_{k \geq 0}(pk - n_k) \text{ is attained by } k = \nu_p^*.$$

Indeed, choose $k \geq 0$ attaining $\min_{k \geq 0}(pk - n_k)$, with k as large as possible. Then $(k+1)p - n_{k+1} > pk - n_k$, and hence $\nu_{k+1} < p$. Moreover, if $k \geq 1$, then $(k-1)p - n_{k-1} \geq pk - n_k$, and hence $\nu_k \geq p$, implying $\nu_p^* = k$. If $k = 0$, then $\nu_1 < p$, again implying $\nu_p^* = 0 = k$. This shows (14.42).

Moreover,

(14.43) $$\sum_{q=1}^{p} \nu_q^* = n + \min_{k \geq 0}(pk - n_k),$$

since

(14.44) $$\sum_{q=1}^{p} \nu_q^* = \sum_{q=1}^{p} \max\{k \mid \nu_k \geq q\} = \sum_{q=1}^{p} \sum_{\substack{k=1 \\ \nu_k \geq q}}^{\infty} 1 = \sum_{k=1}^{\infty} \min\{\nu_k, p\}$$
$$= \sum_{k=1}^{\infty} \nu_k - \sum_{k=1}^{\nu_p^*} (\nu_k - p) = n + p\nu_p^* - \sum_{k=1}^{\nu_p^*} \nu_k = n + p\nu_p^* - n_{\nu_p^*}$$
$$= n + \min_{k \geq 0}(pk - n_k),$$

by (14.42).

By (14.43), condition (14.41) is equivalent to

(14.45) $$n_p' = \sum_{q=1}^{p} \nu_q^* \text{ for } p = 0, 1, \ldots.$$

Hence it is equivalent to: $\nu_q' = \nu_q^*$ for each $q = 1, 2, \ldots$; that is to: (ν_k) and (ν_k') are conjugate. ∎

This yields the conjugacy of the α_k and γ_p:

Theorem 14.13. *(α_k) and (γ_p) are conjugate partitions of $|S|$.*

Proof. Directly from Lemma 14.13α and Corollary 14.10b. ∎

Lemma 14.13α gives the equivalence of the Corollaries 14.10b and 14.8b. For other proofs of the conjugacy of (α_k) and (γ_p), see Fomin [1978] and Frank [1980a].

14.8. Further results and notes

14.8a. The Gallai-Milgram theorem

Gallai and Milgram [1960] showed the following generalization of Dilworth's decomposition theorem. It applies to any directed graph, but generally is not a min-max relation.

Theorem 14.14 (Gallai-Milgram theorem)**.** *Let $D = (V, A)$ be a digraph and let $\alpha(D)$ be the maximum number of vertices that are pairwise nonadjacent in the underlying undirected graph. Then V can be partitioned into $\alpha(D)$ directed paths.*

Proof. For any partition Π of V into directed paths, let C_Π be the set of end vertices of the paths in Π. For any subset U of V, let $\alpha(U)$ be the maximum number of pairwise nonadjacent vertices in U. We show by induction on $|V|$ that

(14.46) for each partition Π of V into directed paths there is a partition Π' into directed paths with $C_{\Pi'} \subseteq C_\Pi$ and $|C_{\Pi'}| \leq \alpha(V)$.

This implies the theorem.

Let Π be a partition of V into directed paths. To prove (14.46), we may assume that C_Π is inclusionwise minimal among all such partitions.

If C_Π is a stable set, then (14.46) is trivial. If C_Π is not a stable set, take $u, v \in C_\Pi$ with $(v, u) \in A$. By the minimality of C_Π, the path P_u in Π ending at u consists of more than u alone, since otherwise we could extend the path ending at v by u. So P_u has a one but last vertex, w say.

Let $\widetilde{\Pi}$ be obtained from Π by deleting u from P_u. So $\widetilde{\Pi}$ is a partition of $V \setminus \{u\}$ into directed paths. By induction, there is a partition $\widetilde{\Pi}'$ of $V \setminus \{u\}$ into directed paths such that $C_{\widetilde{\Pi}'} \subseteq C_{\widetilde{\Pi}}$ and such that $|C_{\widetilde{\Pi}'}| \leq \alpha(V \setminus \{u\})$.

If one of the paths in $\widetilde{\Pi}'$ ends at w, we can extend it with u, and obtain a partition Π' of V as required. If none of the paths in $\widetilde{\Pi}'$ end at w, but one of the paths ends at v, we can extend it with u, again obtaining a partition Π' as required. If none of the paths in $\widetilde{\Pi}'$ end at v or w, then augmenting $\widetilde{\Pi}'$ by a path consisting of u alone, gives a partition Π' of V with $C_{\Pi'} \subset C_\Pi$, contradicting the minimality of C_Π. ∎

Theorem 14.14 gives no min-max relation, as is shown by a directed circuit of length $2k$: the vertices can be covered by one directed path, while there exist k pairwise nonadjacent vertices.

A consequence of Theorem 14.14 is Dilworth's decomposition theorem: for any partially ordered set $(S, \leq)$ take $V := S$ and $A := \{(x, y) \mid x < y\}$. Another consequence is the graph-theoretical result of Rédei [1934] that each tournament has a Hamiltonian path:

Corollary 14.14a (Rédei's theorem)**.** *Each tournament has a Hamiltonian path.*

Proof. This is the special case where $\alpha(D) = 1$ in the Gallai-Milgram theorem. ∎

Berge [1982b] posed the following conjecture generalizing the Gallai-Milgram theorem. Let $D = (V, A)$ be a digraph and let $k \in \mathbb{Z}_+$. Then for each path collection $\mathcal{P}$ partitioning V and minimizing

$$(14.47) \qquad \sum_{P \in \mathcal{P}} \min\{|VP|, k\},$$

there exist disjoint stable sets $C_1, \ldots, C_k$ in D such that each $P \in \mathcal{P}$ intersects $\min\{|VP|, k\}$ of them. This was proved by Saks [1986] for acyclic graphs. For extensions and related results, see Linial [1978,1981], Saks [1986], and Thomassé [2001].

14.8b. Partially ordered sets and distributive lattices

There is a strong relation between the class of partially ordered sets and the class of partially ordered sets of a special type, the *distributive lattices*. It formed the original motivation for Dilworth to study minimum chain partitions of partially ordered sets.

Let $(S, \leq)$ be a partially ordered set and let $a, b \in S$. Then an element $c \in S$ is called the *meet* of a and b if for each $s \in S$:

(14.48) $\quad s \leq c$ if and only if $s \leq a$ and $s \leq b$.

Note that if the meet of a and b exists, it is unique. Similarly, c is called the *join* of a and b if for each $s \in S$:

(14.49) $\quad s \geq c$ if and only if $s \geq a$ and $s \geq b$.

Again, if the join of a and b exists, it is unique.

S is called a *lattice* if each pair s, t of elements of S has a meet and a join; they are denoted by $s \wedge t$ and $s \vee t$, respectively. The lattice is called *distributive* if

(14.50) $\quad s \wedge (t \vee u) = (s \wedge t) \vee (s \wedge u)$ and $s \vee (t \wedge u) = (s \vee t) \wedge (s \vee u)$

for all $s, t, u \in S$. (In fact it suffices to require only one of the two equalities.)

Each partially ordered set $(S, \leq)$ gives a distributive lattice in the following way. Call a subset $I \subseteq S$ a *lower ideal*, or just an *ideal*, if $t \in I$ and $s \leq t$ implies that $s \in I$. Let $\mathcal{I}_S$ be the collection of ideals in S. Then $(\mathcal{I}_S, \subseteq)$ is a distributive lattice. This follows directly from the fact that for $I, J \in \mathcal{I}_S$ one has $I \wedge J = I \cap J$ and $I \vee J = I \cup J$, and hence (14.50) is elementary set theory. Thus

$$(14.51) \qquad (S, \leq) \to (\mathcal{I}_S, \subseteq)$$

associates a distributive lattice with any partially ordered set.

In fact, each finite distributive lattice can be obtained in this way; that is, we can reverse (14.51). For any lattice $(L, \leq)$, call an element $u \in L$ *join-irreducible* if there exist no $s, t \in L$ with $s \neq u$, $t \neq u$, and $u = s \vee t$. Let J_L be the set of join-irreducible elements in L. Trivially,

$$(14.52) \qquad (L, \leq) \to (J_L, \leq)$$

associates a partially ordered set with any distributive lattice.

Theorem 14.15. *Functions* (14.51) *and* (14.52) *are inverse to each other.*

Proof. First, let $(S \leq)$ be a partially ordered set. For each $s \in S$, let $I_s := \{t \in S \mid t \leq s\}$. Then an element I of $\mathcal{I}_S$ is join-irreducible if and only if there exists an $s \in S$ with $I = I_s$. Moreover, $s \leq t$ if and only if $I_s \subseteq I_t$. Thus we have an isomorphism of $(S, \leq)$ and $(J_{\mathcal{I}_S}, \subseteq)$.

Conversely, let $(L, \leq)$ be a distributive lattice. For each $s \in L$, let $J_s := \{t \in J_L \mid t \leq s\}$. This gives a one-to-one relation between elements of L and ideals in J_L. Indeed, if I is an ideal in J_L, let $s := \bigvee I$. Then $I = J_s$. Clearly, $I \subseteq J_s$, since $t \leq s$ for each $t \in I$. Conversely, let $u \in J_s$. Then, as L is distributive,

$$(14.53) \qquad u = u \wedge s = u \wedge \bigvee_{t \in I} t = \bigvee_{t \in I} (u \wedge t).$$

Since u is join-irreducible, $u = u \wedge t$ for some $t \in I$. Hence $u \leq t$, and therefore $u \in I$.

Moreover, for any $s, t \in L$ one has: $s \leq t$ if and only if $J_s \subseteq J_t$. So we have an isomorphism of $(L, \leq)$ and $(\mathcal{I}_{J_L}, \subseteq)$. ∎

There is moreover a one-to-one relation between ideals I in a partially ordered set $(S, \leq)$ and antichains A in S, given by:

$$(14.54) \qquad A = I^{\max} \text{ and } I = A^{\downarrow}.$$

Here, for any $Y \subseteq S$, $Y^{\max}$ denotes the set of maximal elements of Y and

$$(14.55) \qquad Y^{\downarrow} := \{s \in S \mid \exists t \in Y : s \leq t\}.$$

For each d, the set $\mathbb{Z}^d$ is a distributive lattice, under the usual order: $x \leq y$ if and only if $x_i \leq y_i$ for each $i = 1, \ldots, d$. Any finite distributive lattice $(L, \leq)$ is a sublattice of $\mathbb{Z}^d$ for some d (as will follow from the next theorem). That is, there is an injection $\phi : L \to \mathbb{Z}^d$ such that $\phi(s \wedge t) = \phi(s) \wedge \phi(t)$ and $\phi(s \vee t) = \phi(s) \vee \phi(t)$ for all $s, t \in L$.

Let $d(L)$ be the minimum number d for which L is (isomorphic to) a sublattice $\mathbb{Z}^d$. As Dilworth [1950] showed, the number $d(L)$ can be characterized with Dilworth's decomposition theorem.

To this end, an element s of a partially ordered set $(S, \leq)$ is said to *cover* $t \in S$ if $s > t$ and there is no $u \in S$ with $s > u > t$. For any $s \in S$, let $\text{cover}(s)$ be the number of elements covered by s.

Then the following result of Dilworth [1950] can be derived from Dilworth's decomposition theorem:

Theorem 14.16. *Let L be a finite distributive lattice. Then*

$$(14.56) \qquad d(L) = \max_{s \in L} \text{cover}(s).$$

Proof. We first show that $d(L) \geq \text{cover}(s)$ for each $s \in L$. Let $d := d(L)$, let $L \subset \mathbb{Z}^d$, and choose $s \in L$. Let Y be the set of elements covered by s. For each $t \in Y$, let $U_t := \{i \mid t_i < s_i\}$. Now for all $t, u \in Y$ with $t \neq u$ one has $t \vee u = s$; hence $U_t \cap U_u = \emptyset$. As $U_t \neq \emptyset$ for all $t \in Y$, we have $|Y| \leq d$.

So $d(L) \geq \max_{s \in L} \text{cover}(s)$. To see equality, by Theorem 14.15 we may assume that $L = \mathcal{I}_S$ (the set of ideals in S) for some partially ordered set $(S, \leq)$, ordered by inclusion. For any ideal I in S, $\text{cover}(I)$ is the number of inclusionwise maximal ideals $J \subset I$. Each such ideal J is equal to $I \setminus \{t\}$ for some $t \in I^{\max}$. So $\text{cover}(I) = |I^{\max}|$. Hence $\max_{s \in L} \text{cover}(s)$ is equal to the maximum antichain size in S. Let this be d, say.

By Dilworth's decomposition theorem, S can be covered by d chains, $C_1, \ldots, C_d$ say. For each j, the collection $\mathcal{I}_{C_j}$ of ideals in C_j is again a chain (ordered by inclusion). Now $I \to (I \cap C_1, \ldots, I \cap C_d)$ embeds $\mathcal{I}_S$ into the product $\mathcal{I}_{C_1} \times \cdots \times \mathcal{I}_{C_d}$. Therefore $d(\mathcal{I}_S) \leq d$. ∎

The following was noted by Dilworth [1960]. The relations (14.54) give the following partial order on the collection $\mathcal{A}_S$ of antichains in a partially ordered set $(S, \leq)$:

(14.57) $A \preceq B$ if and only if $A^{\downarrow} \subseteq B^{\downarrow}$

for $A, B \in \mathcal{A}_S$. As $(\mathcal{I}_S, \subseteq)$ is a lattice, also $(\mathcal{A}_S, \preceq)$ is a lattice.

Theorem 14.17. *Let $(S, \leq)$ be a partially ordered set and let A and B be maximum-size antichains. Then also $A \wedge B$ and $A \vee B$ are maximum-size antichains.*

Proof. One has $|A \wedge B| + |A \vee B| \geq |A| + |B|$. Indeed, $A \cup B \subseteq (A \wedge B) \cup (A \vee B)$ and $A \cap B \subseteq (A \wedge B) \cap (A \vee B)$. So

(14.58)
$$\begin{aligned}|A \wedge B| + |A \vee B| &= |(A \wedge B) \cup (A \vee B)| + |(A \wedge B) \cap (A \vee B)| \\ &\geq |A \cup B| + |A \cap B| = |A| + |B|.\end{aligned}$$

As $|A \wedge B| \leq |A| = |B|$ and $|A \vee B| \leq |A| = |B|$, we have $|A \wedge B| = |A \vee B| = |A| = |B|$. ∎

In terms of distributive lattices this gives:

Corollary 14.17a. *Let L be a finite distributive lattice. Then the elements s maximizing* $\text{cover}(s)$ *form a sublattice of L.*

Proof. We can represent L as the set $\mathcal{I}_S$ of ideals in a partially ordered set $(S, \leq)$. As one has $\text{cover}(I) = |I^{\max}|$ for any $I \in \mathcal{I}_S$, the result follows from Theorem 14.17. ∎

Theorem 14.17 led Dilworth [1960] to derive an alternative proof of Dilworth's decomposition theorem. For another proof and application of Theorem 14.17, see Freese [1974]. Theorem 14.17 was extended to maximum-size unions of k antichains by Greene and Kleitman [1976].

14.8c. Maximal chains

Let $(S, \leq)$ be a partially ordered set. Call a chain *maximal* if it is contained in no other chain. As 'complementary' to Dilworth's decomposition theorem, Greene and Kleitman [1976] observed:

Theorem 14.18. *The maximum number of disjoint maximal chains is equal to the minimum size of a set intersecting all maximal chains.*

Proof. Define a digraph $D = (S, A)$ where A consists of all pairs (s, t) where t covers s. Let U and W be the sets of minimal and maximal elements of S, respectively. So maximal chains correspond to $U-W$ paths, and the theorem follows from Menger's theorem. ∎

14.8d. Further notes

Related results were given by Bogart [1970], Saks [1986], and Behrendt [1988], extensions and algorithms by Cameron and Edmonds [1979], Frank [1980a], Linial [1981], Cameron [1982,1985,1986], and Hoffman [1983], surveys by Greene [1974b], Hoffman [1982], West [1982], and Bogart, Greene, and Kung [1990], and an introduction to and historical account of Dilworth's decomposition theorem by Dilworth [1990].

Fleiner [1997] proved the following conjecture of A. Frank: Let $(S, \leq)$ be a partially ordered set and let M be a perfect matching on S such that for any two $uu', vv' \in M$ with $u \leq v$, one has $v' \leq u'$. (This is called a *symmetric partially ordered set*.) Call a subset C of M a *symmetric chain* if S has a chain intersecting each edge in C. Then the minimum number of symmetric chains covering M is equal to the maximum value of

(14.59) $$\sum_{i=1}^{k} \lceil \tfrac{1}{2}|X_i| \rceil,$$

where $X_1, \ldots, X_k$ are disjoint subsets of M (for some k) with the properties that (i) no X_i contains a symmetric chain of size 3, and (ii) if $i \neq j$, then there exist no $x \in e \in X_i$ and $y \in f \in X_j$ with $x \leq y$.

Chapter 15

Connectivity and Gomory-Hu trees

Since a minimum $s - t$ cut can be found in polynomial time, also the connectivity of a graph can be determined in polynomial time, just by checking all pairs s, t. However, there are more economical methods, which we discuss in this chapter.

A finer description of the edge-connectivity of a graph G is given by the function $r_G(s,t)$, defined as the minimum size of a cut separating s and t. A concise description of the corresponding minimum-size cuts is given by the Gomory-Hu tree — see Section 15.4.

15.1. Vertex-, edge-, and arc-connectivity

For any undirected graph $G = (V,E)$, the *vertex-connectivity*, or just *connectivity*, of G is the minimum size of a subset U of V for which $G - U$ is not connected. A subset U of V attaining the minimum is called a *minimum vertex-cut*. If no such U exists (namely, if G is complete), then the (vertex-)connectivity is ∞.

The connectivity of G is denoted by $\kappa(G)$. If $\kappa(G) \geq k$, G is called *k-vertex-connected*, or just *k-connected*.

The following direct consequence of Menger's theorem was formulated by Whitney [1932a]:

Theorem 15.1. *An undirected graph $G = (V,E)$ is k-connected if and only if there exist k internally vertex-disjoint paths between any two nonadjacent vertices s and t.*

Proof. Directly from the vertex-disjoint version of Menger's theorem (Corollary 9.1a). ∎

Similarly, the *edge-connectivity* of G is the minimum size of a subset C of E for which $G - C$ is not connected. So it is the minimum size of any cut $\delta(U)$ with $\emptyset \neq U \neq V$. A cut C attaining the minimum size is called a *minimum cut.* The edge-connectivity of G is denoted by $\lambda(G)$. If $\lambda(G) \geq k$, G is called *k-edge connected.* Then:

Theorem 15.2. *An undirected graph $G = (V, E)$ is k-edge-connected if and only if there exist k edge-disjoint paths between any two vertices s and t.*

Proof. Directly from the edge-disjoint version of Menger's theorem (Corollary 9.1b). ∎

Similar terminology and characterizations apply to digraphs. For any digraph $D = (V, A)$ the *vertex-connectivity*, or just *connectivity*, of D is the minimum size of a subset U of V for which $D - U$ is not strongly connected. A set U attaining the minimum is called a *minimum vertex-cut.* If no such U exists (that is, if each pair (u, v) of vertices is an arc), then the (vertex-)connectivity of D is ∞.

The connectivity of D is denoted by $\kappa(D)$. If $\kappa(D) \geq k$, D is called *k-vertex-connected*, or just *k-connected.* Now one has:

Theorem 15.3. *A digraph $D = (V, A)$ is k-connected if and only if there exist k internally vertex-disjoint $s - t$ paths for any $s, t \in V$ for which there is no arc from s to t.*

Proof. Directly from the vertex-disjoint version of Menger's theorem (Corollary 9.1a). ∎

Finally, the *arc-connectivity* of D is the minimum size of a subset C of A for which $D - C$ is not strongly connected. That is, it is the minimum size of any cut $\delta^{\text{out}}(U)$ with $\emptyset \neq U \neq V$. Any cut attaining the minimum size is called a *minimum cut.* The arc-connectivity of D is denoted by $\lambda(D)$. If $\lambda(D) \geq k$, D is called *k-arc-connected* or *strongly k-connected.* (So D is 1-arc-connected if and only if D is strongly connected.) Then:

Theorem 15.4. *A digraph $D = (V, A)$ is k-arc-connected if and only if there exist k arc-disjoint paths between any two vertices s and t.*

Proof. Directly from the arc-disjoint version of Menger's theorem (Corollary 9.1b). ∎

Shiloach [1979a] observed that Edmonds' disjoint arborescences theorem (to be discussed in Chapter 53) implies a stronger characterization of k-arc-connectivity: a digraph $D = (V, A)$ is k-arc-connected if and only if for all $s_1, t_1, \ldots, s_k, t_k \in V$ there exist arc-disjoint paths $P_1, \ldots, P_k$, where P_i runs from s_i to t_i $(i = 1, \ldots, k)$ — see Corollary 53.1d.

A similar characterization does not hold for the undirected case, as is shown by the 2-edge-connected graph in Figure 15.1.

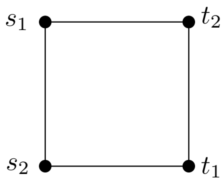


Figure 15.1

15.2. Vertex-connectivity algorithmically

It is clear that the vertex-connectivity of a directed or undirected graph can be determined in polynomial time, just by finding a minimum-size $s-t$ vertex-cut for each pair s,t of vertices. Since by Corollary 9.7a, a minimum-size $s-t$ vertex-cut can be found in $O(n^{1/2}m)$ time, this yields an $O(n^{5/2}m)$ algorithm. In fact, Podderyugin [1973] (for undirected graphs) and Even and Tarjan [1975] observed that one need not consider every pair of vertices:

Theorem 15.5. *A minimum-size vertex-cut in a digraph D can be found in $O(\kappa(D)n^{3/2}m)$ time.*

Proof. Let $D = (V, A)$ be a digraph. We may assume that D is simple. Order V arbitrarily as $v_1, \ldots, v_n$. For $i = 1, 2, \ldots$, determine, for each $v \in V$, a minimum $v_i - v$ vertex-cut $C_{v_i,v}$ and a minimum $v - v_i$ vertex-cut C_{v,v_i}. (This takes $O(n^{3/2}m)$ time by Corollary 9.7a.) At any moment, let c be the minimum size of the cuts found so far. We stop if $i > c + 1$. Then c is the vertex-connectivity of D.

Indeed let C be a minimum-size vertex-cut. Then for $i := \kappa(D) + 2$, there is a $j < i$ with $v_j \notin C$. Hence there is a vertex v such that C is a $v_j - v$ or a $v - v_j$ vertex-cut. Assume without loss of generality that C is a $v_j - v$ vertex-cut. Then $c \leq |C_{v_j,v}| = |C| = \kappa(D)$. ∎

Since $\kappa(D) \leq m/n$ if D is not complete, this implies:

Corollary 15.5a. *A minimum-size vertex-cut in a digraph can be found in $O(n^{1/2}m^2)$ time.*

Proof. Immediately from Theorem 15.5, since if D is not complete, $\kappa(D)$ is at most the minimum outdegree of D, and hence $\kappa(D) \leq m/n$. ∎

If we want to test the k-connectivity for some fixed k, we may use the following result given by Even [1975]:

Theorem 15.6. *Given a digraph D and an integer k, one can decide in $O((k + \sqrt{n})k\sqrt{n}m)$ time if D is k-connected, and if not, find a minimum cut.*

Proof. Let $V = \{1,\ldots,n\}$. Determine

(15.1) (i) for all $i,j \in \{1,\ldots,k\}$ with $(i,j) \notin A$, a minimum-size $i-j$ vertex-cut if it has size less than k;
(ii) for each $i = k+1,\ldots,n$ a minimum-size $\{1,\ldots,i-1\} - i$ vertex-cut if it has size less than k, and a minimum-size $i - \{1,\ldots,i-1\}$ vertex-cut if it has size less than k.

We claim that if we find any vertex-cut, the smallest among them is a minimum-size vertex-cut. If we find no vertex-cuts, then D is k-connected.

To see this, let U be a minimum-size vertex-cut with $|U| < k$. Suppose that each vertex-cut found has size $> |U|$. Then for all distinct $i,j \in \{1,\ldots,k\} \setminus U$ there is an $i-j$ path avoiding U. So $\{1,\ldots,k\} \setminus U$ is contained in some strong component K of $D-U$. As $D-U$ is not strongly connected, $D-U$ has a vertex not in K. Let i be the smallest index $i \notin K \cup U$. As there exist $|U|+1$ disjoint $\{1,\ldots,i-1\} - i$ paths, $D-U$ has a $j-i$ path for some $j < i$; then $j \in K$. Similarly, $D-U$ contains an $i-j'$ path for some $j' \in K$. This contradicts the fact that $i \notin K$ and K is a strong component.

This implies the theorem. Indeed, by Corollaries 9.3a and 9.7a one can find a vertex-cut as in (15.1)(i) or (ii) in time $O(\min\{k,\sqrt{n}\}m)$ time. So in total it takes $O((k^2+n)\min\{k,\sqrt{n}\}m) = O((k+\sqrt{n})k\sqrt{n}\,m)$ time. ∎

This implies:

Corollary 15.6a. *A minimum-size vertex-cut in a digraph can be found in time* $O(\max\{\frac{m^3}{n\sqrt{n}}, m^2\})$.

Proof. From Theorem 15.6 by taking $k := \lfloor m/n \rfloor$. ∎

Matula [1987] showed that Theorem 15.6 implies the following result of Galil [1980b], where $\kappa(D)$ is the vertex-connectivity of D:

Corollary 15.6b. *A minimum-size vertex-cut in a digraph D can be found in* $O((\kappa(D)+\sqrt{n})\kappa(D)\sqrt{n}\,m)$ *time.*

Proof. For $k = 2, 2^2, 2^3, \ldots$ test with the algorithm of Theorem 15.6 if D is k-connected. Stop if D is not k-connected; then the algorithm gives a minimum-size vertex-cut. Let l be such that $k = 2^l$. As $2^l \leq 2\kappa(D)$, this takes time

(15.2)
$$O\Big(\sum_{i=1}^{l}((2^i+\sqrt{n})2^i\sqrt{n}\,m)\Big) = O((4^{l+1}+2^{l+1}\sqrt{n})\sqrt{n}\,m)$$
$$= O((\kappa(D)^2+\kappa(D)\sqrt{n})\sqrt{n}\,m).$$
∎

Since vertex-cuts in an undirected graph G are equal to vertex-cuts in the digraph obtained from G by replacing each edge by two oppositely oriented arcs, the results above immediately imply similar results for vertex-cuts and vertex-connectivity in undirected graphs.

15.2a. Complexity survey for vertex-connectivity

Complexity survey for vertex-connectivity (in directed graphs, unless stated otherwise; again, * indicates an asymptotically best bound in the table):

	$O(n^2 \cdot \mathrm{VC}(n,m))$	(trivial)
	$O(kn \cdot \mathrm{VC}_k(n,m))$	Kleitman [1969]
	$O(\kappa n \cdot \mathrm{VC}(n,m))$	Podderyugin [1973], Even and Tarjan [1975]
	$O((k^2+n) \cdot \mathrm{VC}_k(n,m))$	Even [1975] (cf. Esfahanian and Hakimi [1984])
	$O((\kappa^2+n) \cdot \mathrm{VC}_\kappa(n,m))$	Galil [1980b] (cf. Matula [1987])
	$O((\kappa+\sqrt{n})\kappa^2 n^{3/2})$	*undirected* Nagamochi and Ibaraki [1992a], Cheriyan and Thurimella [1991]
	$O((\kappa+\sqrt{n})\kappa^2 n^{3/2} \log_n(n^2/m))$	*undirected* Feder and Motwani [1991, 1995]
	$O((\kappa^3+n)m)$	Henzinger, Rao, and Gabow [1996, 2000]
	$O(\kappa nm)$	Henzinger, Rao, and Gabow [1996, 2000]
	$O((\kappa^3+n)\kappa n)$	*undirected* Henzinger, Rao, and Gabow [1996,2000]
	$O(\kappa^2 n^2)$	*undirected* Henzinger, Rao, and Gabow [1996,2000]
*	$O((\kappa^{5/2}+n)m)$	Gabow [2000b]
*	$O((\kappa+n^{1/4})n^{3/4}m)$	Gabow [2000b]
*	$O((\kappa^{5/2}+n)\kappa n)$	*undirected* Gabow [2000b]
*	$O((n^{1/4}+\kappa)\kappa n^{7/4})$	*undirected* Gabow [2000b]

Here κ denotes the vertex-connectivity of the graph. Note that $\kappa \leq m/n$. If k is involved, the time bound is for determining $\min\{\kappa, k\}$. By $\mathrm{VC}(n,m)$ we denote the time needed to find the minimum size of an $s-t$ vertex-cut for fixed s,t. Moreover, $\mathrm{VC}_k(n,m)$ denotes the time needed to find the minimum size of an $s-t$ vertex-cut if this size is less than k. We refer to Sections 9.5 and 9.6a for bounds on $\mathrm{VC}(n,m)$ and $\mathrm{VC}_k(n,m)$. Note that $\mathrm{VC}_k(n,m) = O(\min\{k, \sqrt{n}\}m)$.

By the observation of Matula [1987] (cf. Corollary 15.6b above), if $\min\{k, \kappa\}$ can be determined in time $O(k^\alpha f(n,m))$ (for some $\alpha \geq 1$), then κ can be determined in time $O(\kappa^\alpha f(n,m))$.

15.2b. Finding the 2-connected components

In this section we show the result due to Paton [1971], Tarjan [1972], and Dinits, Zajtsev, and Karzanov [1974] (cf. Hopcroft and Tarjan [1973a]) that the 2-vertex-connected components of an undirected graph can be found in linear time. Hence, the 2-connectivity of an undirected graph can be tested in linear time.

Let $G = (V, E)$ be an undirected graph and let $k \in \mathbb{Z}_+$. A *k-connected component* is an inclusionwise maximal subset U of V for which $G[U]$ is k-connected. A *block* is a 2-connected component U with $|U| \geq 2$.

We note

(15.3) if U and W are two different k-connected components, then $|U \cap W| < k$.

Indeed, as $G[U \cup W]$ is not k-connected, there is a subset C of $U \cup W$ with $G[(U \cup W) \setminus C]$ disconnected and $|C| < k$. As $(U \cup W) \setminus C = (U \setminus C) \cup (W \setminus C)$ and as $G[U \setminus C]$ and $G[W \setminus C]$ are connected, it follows that $(U \setminus C) \cap (W \setminus C) = \emptyset$. Hence $C \supseteq U \cap W$, and therefore $|U \cap W| < k$.

(15.3) implies that each edge of G is contained in a unique 2-connected component. So the 2-connected components partition the edge set. One may show:

(15.4) edges e and e' are contained in the same 2-connected component if and only if G has a circuit C containing both e and e'.

Indeed, if C exists, it forms a 2-connected subgraph of G, and hence e and e' are contained in some 2-connected component. Conversely, if $e = uv$ and $e' = u'v'$ are contained in a 2-connected component H, by Menger's theorem H has two vertex-disjoint $\{u, v\} - \{u', v'\}$ paths; with e and e' these paths form a circuit C as required.

Theorem 15.7. *The collection of blocks of a graph $G = (V, E)$ can be identified in linear time.*

Proof. By Corollary 6.6a, we may assume that G is connected. Choose $s \in V$ arbitrarily. Apply depth-first search starting at s. If we orient the final tree to become a rooted tree (V, T) with root s, all further edges of G connect two vertices u, v such that T has a directed $u - v$ path P. For each such edge, make an arc (v, u'), where u' is the second vertex of P. The arcs in T and these new arcs form a directed graph denoted by $D = (V, A)$.

By adapting the depth-first search, we can find A is linear time. Indeed, while scanning s, we keep a directed path Q in T formed by the vertices whose scanning has begun but is not yet finished. If we scan v and meet an edge uv with u on Q, we can find u' and construct the arc (v, u').

Since no arc of D enters s, $\{s\}$ is a strong component. For any strong component K of D one has:

(15.5) the subgraph of T induced by K is a subtree.

Indeed, for any arc $(u, v) \in A \setminus T$ spanned by K, the $v - u$ path in T is contained in K, since it forms a directed circuit with (u, v) and since K is a strong component. This proves (15.5).

(15.5) implies that for each strong component K of D with $K \neq \{s\}$, there is a unique arc of T entering K; let u_K be its tail and define $K' := K \cup \{u_K\}$. We finally show

(15.6) $\{K' \mid K$ strong component of D, $K \neq \{s\}\}$ is equal to the collection of blocks of G.

This proves the theorem (using Theorem 6.6).

Let (t, u, v) be a directed path in T. Then

(15.7) tu and uv are contained in the same block of G if and only if D has a directed $v - u$ path.

To see this, let W be the set of vertices reachable in (V, T) from v. Then:

(15.8) D has a directed $v - u$ path $\iff$ D has an arc leaving W $\iff$ G has an edge leaving W and not containing u $\iff$ G has a $v - t$ path not traversing u $\iff$ tu and uv are contained in a circuit of G $\iff$ tu and uv are contained in the same block of G.

This proves (15.7).

(15.7) implies that for each strong component $K \subseteq V \setminus \{s\}$ of D, the set K' is contained in a block of G. Conversely, let B be a block of G. Then B induces a subtree of T. Otherwise, B contains two vertices u and v such that the undirected $u - v$ path P in T has length at least two and such that no internal vertex of P belongs to B. Let Q be a $u - v$ path in B. Then P and Q form a circuit, hence a 2-connected graph. So P is contained in B, a contradiction.

So B induces a subtree of T. Let u be its root. As $B \setminus \{u\}$ induces a connected subgraph of G, there is a unique arc in T entering $B \setminus \{u\}$. So also $B \setminus \{u\}$ induces a subtree of T. Then (15.7) implies that $B \setminus \{u\}$ is contained in a strong component of D. ∎

Corollary 15.7a. *The* 2*-connectivity of an undirected graph can be tested in linear time.*

Proof. A graph (V, E) is 2-connected if and only if V is a 2-connected component. So Theorem 15.7 implies the result. ∎

Hopcroft and Tarjan [1973b] gave a linear-time algorithm to test 3-connectivity of an undirected graph; more generally, to decompose an undirected graph into 3-connected components (cf. Miller and Ramachandran [1987,1992]). Kanevsky and Ramachandran [1987,1991] gave an $O(n^2)$ algorithm to test 4-connectivity of an undirected graph.

Finding all 2- and 3-vertex-cuts of an undirected graph has been investigated by Tarjan [1972], Hopcroft and Tarjan [1973b], Kanevsky and Ramachandran [1987, 1991], Miller and Ramachandran [1987,1992], and Kanevsky [1990a]. Kanevsky [1990b] showed that for each fixed k, the number of vertex-cuts of size k in a k-vertex-connected graph is $O(n^2)$ (cf. Kanevsky [1993]). Related results can be found in Gusfield and Naor [1990], Cohen, Di Battista, Kanevsky, and Tamassia [1993], Gabow [1993b,1995c], and Cheriyan and Thurimella [1996b,1999].

15.3. Arc- and edge-connectivity algorithmically

Denote by $\mathrm{EC}(n, m)$ the time needed to find a minimum-size $s - t$ cut for any given pair of vertices s, t. Even and Tarjan [1975] (and Podderyugin [1973] for undirected graphs) observed that one need not check all pairs of vertices to find a minimum cut:

Theorem 15.8. *A minimum-size cut in a digraph can be found in* $O(n \cdot \mathrm{EC}(n,m))$ *time.*

Proof. Choose $s \in V$. For each $t \neq s$, determine a minimum $s-t$ cut $C_{s,t}$ and a minimum $t-s$ cut $C_{t,s}$. The smallest among all these cuts is a minimum cut. ∎

Hence we have for the arc-connectivity:

Corollary 15.8a. *The arc-connectivity of a digraph can be determined in* $O(n \cdot \mathrm{EC}(n,m))$ *time.*

Proof. Directly from Theorem 15.8. ∎

As $\mathrm{EC}(n,m) = O(m^{3/2})$ by Corollary 9.6a, it follows that the arc-connectivity can be determined in time $O(nm^{3/2})$. Actually, also in time $O(m^2)$, since we need to apply the disjoint paths algorithm only until we have at most $k := \lfloor m/n \rfloor$ arc-disjoint paths, as the arc-connectivity is at most m/n (there is a $v \in V$ with $|\delta^{\mathrm{out}}(v)| \leq \lfloor m/n \rfloor$).

Moreover, again by Corollary 9.6a, $\mathrm{EC}(n,m) = O(n^{2/3}m)$ for simple digraphs, and hence the arc-connectivity of a simple directed graph can be determined in time $O(n^{5/3}m)$ (cf. Esfahanian and Hakimi [1984]).

Schnorr [1978b,1979] showed that in fact:

Theorem 15.9. *Given a digraph D and an integer k, one can decide in $O(knm)$ time if D is k-arc-connected, and if not, find a minimum cut.*

Proof. In Theorem 15.8 one needs to check only if there exist k arc-disjoint $s-t$ paths, and if not find a minimum-size $s-t$ cut. This can be done in time $O(km)$, as we saw in Corollary 9.3a. ∎

With a method of Matula [1987] this implies, where $\lambda(D)$ is the arc-connectivity of D:

Corollary 15.9a. *A minimum-size cut in a digraph D can be found in time* $O(\lambda(D)nm)$.

Proof. For $k = 2, 2^2, 2^3, \ldots$ test if D is k-arc-connected, until we find that D is not k-arc-connected, and have a minimum-size cut. With the method of Theorem 15.9 this takes time $O((2+2^2+2^3+\cdots+2^l)nm)$, with $2^l \leq 2\lambda(D)$. So $2+2^2+2^3+\cdots+2^l \leq 2^{l+1} \leq 4\lambda(D)$, and the result follows. ∎

For undirected graphs, Nagamochi and Ibaraki [1992b] showed that the edge-connectivity of an undirected graph can be determined in time $O(nm)$ (for simple graphs this bound is due to Podderyugin [1973]). We follow the shortened algorithm described by Frank [1994b] and Stoer and Wagner [1994, 1997].

Theorem 15.10. *Given an undirected graph G, a minimum cut in G can be found in time $O(nm)$.*

Proof. Let $G = (V, E)$ be a graph. For $U \subseteq V$ and $v \in V \setminus U$, let $d(U, v)$ denote the number of edges connecting U and v. Let $r(u, v)$ denote the minimum capacity of a $u - v$ cut.

Call an ordering $v_1, \ldots, v_n$ of the vertices of G a *legal order* for G if $d(\{v_1, \ldots, v_{i-1}\}, v_i) \geq d(\{v_1, \ldots, v_{i-1}\}, v_j)$ for all i, j with $1 \leq i < j \leq n$. Then:

(15.9) If $v_1, \ldots, v_n$ is a legal order for $G = (V, E)$, then $r(v_{n-1}, v_n) = d(v_n)$.

To see this, let C be any $v_{n-1} - v_n$ cut. Define $u_0 := v_1$. For $i = 1, \ldots, n-1$, define $u_i := v_j$, where j is the smallest index such that $j > i$ and C is a $v_i - v_j$ cut. Note that for each $i = 1, \ldots, n-1$ one has

$$d(\{v_1, \ldots, v_{i-1}\}, u_i) \leq d(\{v_1, \ldots, v_{i-1}\}, u_{i-1}), \tag{15.10}$$

since if $u_{i-1} \neq u_i$, then $u_{i-1} = v_i$, in which case (15.10) follows from the legality of the order.

Then we have

$$\begin{aligned} d(C) &\geq \sum_{i=1}^{n-1} d(v_i, u_i) \\ &= \sum_{i=1}^{n-1} (d(\{v_1, \ldots, v_i\}, u_i) - d(\{v_1, \ldots, v_{i-1}\}, u_i)) \\ &\geq \sum_{i=1}^{n-1} (d(\{v_1, \ldots, v_i\}, u_i) - d(\{v_1, \ldots, v_{i-1}\}, u_{i-1})) \\ &= d(\{v_1, \ldots, v_{n-1}\}, v_n) = d(v_n), \end{aligned} \tag{15.11}$$

showing (15.9).

Next one has that a legal order for a given graph G can be found in time $O(m)$. Indeed, one can find $v_1, v_2, v_3, \ldots$ successively: if $v_1, \ldots, v_{i-1}$ have been found, we need to find a $v \in V \setminus \{v_1, \ldots, v_{i-1}\}$ maximizing $d(\{v_1, \ldots, v_{i-1}\}, v)$. With a 'bucket' data structure this can be done in $O(m)$ time.[23]

[23] Suppose that we have a set V and a function $\phi : V \to \mathbb{Z}$. We can select and delete a v maximizing $\phi(v)$ in $O(1)$ time and to reset $\phi(v)$ from k to k' in $O(|k' - k|)$ time.

Concluding, in any given graph $G = (V, E)$ with $|V| \geq 2$, we can find two vertices s and t with $r(s,t) = d(s)$ in time $O(m)$. Identify s and t, and find recursively a minimum cut C in the new graph. Then $\delta(s)$ is a minimum cut separating s and t, and C is a minimum cut not separating s and t. Hence, the smallest of the two is a minimum cut. ▮

(Another correctness proof was given by Fujishige [1998].)

Theorem 15.10 extends to capacitated graphs (Nagamochi and Ibaraki [1992b]):

Theorem 15.11. *Given an undirected graph $G = (V, E)$ and a capacity function $c : E \to \mathbb{Q}_+$, a minimum-capacity cut can be found in time $O(n(m + n \log n))$.*

Proof. This can be shown in the same way as Theorem 15.10, using Fibonacci heaps for finding a legal order. ▮

15.3a. Complexity survey for arc- and edge-connectivity

Survey for arc-complexity (in uncapacitated directed graphs, unless stated otherwise):

	$O(n \cdot \mathrm{EC}(n,m))$	(trivial)
	$O(nm)$	*simple undirected* Podderyugin [1973]
	$O(n \cdot \mathrm{EC}_l(n,m))$	Schnorr [1978b,1979]
	$O(\lambda^3 n^2 + \lambda m)$	Timofeev [1982]
	$O(\lambda n^2)$	*simple undirected* Karzanov and Timofeev [1986], Matula [1987]
	$O(n \cdot \mathrm{EC}_\lambda(n,m))$	Matula [1987]
	$O(nm)$	*simple* Mansour and Schieber [1989]
	$O(\lambda^2 n^2)$	*simple* Mansour and Schieber [1989]
*	$O(n \frac{\log \delta}{\delta} \cdot \mathrm{EC}(n,m))$	*simple* N. Alon, 1988 (cf. Mansour and Schieber [1989])
	$O(nm \log_n(n^2/m))$	*undirected* Feder and Motwani [1991,1995]

≫

To this end, partition V into classes V_j, where $V_j := \{v \in V \mid \phi(v) = j\}$. Each nonempty V_j is ordered as a doubly linked list, and the nonempty V_j among them are ordered as a doubly linked list L, in increasing order of j. Then in $O(1)$ time we can choose the largest j for which V_j is nonempty, choose $v \in V_j$, delete v from V_j, and possibly delete V_j from L (if V_j has become empty). Resetting $\phi(v)$ from k to k' can be done by finding or creating $V_{k'}$ in L, which takes $O(|k' - k|)$ time.

Having this data structure, throughout let $U := V \setminus \{v_1, \ldots, v_i\}$, and for each $v \in U$ let $\phi(v) := d(\{v_1, \ldots, v_i\}, v)$. If v_{i+1} has been found, we must delete v_{i+1} from U, and reset, for each neighbour v of v_{i+1} in U, $\phi(v)$ to $d(\{v_1, \ldots, v_{i+1}\}, v)$. This gives an $O(m)$-time algorithm.

continued

	$O(nm)$	*undirected* Nagamochi and Ibaraki [1992b]
	$O(m + \lambda^2 n^2)$	*undirected* Nagamochi and Ibaraki [1992b]
	$O(m + \tilde{m}n + n^2 \log n)$	*undirected* Nagamochi and Ibaraki [1992b]
*	$O(n(m + n \log n))$	*capacitated undirected* Nagamochi and Ibaraki [1992b]
*	$O(nm \log(n^2/m))$	*capacitated* Hao and Orlin [1992,1994]
*	$O(\lambda m \log(n^2/m))$	Gabow [1991a,1995a]
*	$O(m + \lambda^2 n \log(n/\lambda))$	*undirected* Gabow [1991a,1995a]

Here λ denotes the arc- or edge-connectivity of the graph, $\tilde{m}$ the number of parallel classes of edges, and δ the minimum (out-)degree. Note that $\lambda \leq \delta \leq 2m/n$. If l is involved, the time bound is for determining $\min\{\lambda, l\}$.

EC(n, m) denotes the time needed to find the minimum size of an $s-t$ cut, for fixed s, t. Moreover, EC$_l(n, m)$ denotes the time needed to find the minimum size of an $s-t$ cut (for fixed s, t) if this size is less than l. We refer to Sections 9.4 and 9.6a for bounds on EC(n, m) and EC$_l(n, m)$.

By the observation of Matula [1987] (cf. Corollary 15.9a above), if $\min\{\lambda, l\}$ can be determined in time $O(l^\alpha f(n, m))$ (for some $\alpha \geq 1$), then λ can be determined in time $O(\lambda^\alpha f(n, m))$.

Matula [1993] gave a linear-time $2+\varepsilon$-approximative algorithm determining the edge-connectivity. (Related work was done by Henzinger [1997].)

Galil and Italiano [1991] described a linear-time method to make from a graph G a graph $\phi_k(G)$, with $m+(k-2)n$ vertices and $(2k-3)m$ edges such that: G is k-edge-connected $\iff$ $\phi_k(G)$ is k-vertex-connected. This implies, for instance, that 3-edge-connectivity can be tested in linear time (as Hopcroft and Tarjan [1973b] showed that 3-vertex-connectivity can be tested in linear time). Related work was reported by Esfahanian and Hakimi [1984] and Padberg and Rinaldi [1990a].

Karger and Stein [1993,1996] gave a randomized minimum cut algorithm for undirected graphs, with running time $O(n^2 \log^3 n)$. Karger [1996,2000] gave an improvement to $O(m \log^3 n)$.

Nagamochi, Ono, and Ibaraki [1994] report on computational experiments with the Nagamochi-Ibaraki algorithm. An experimental study of several minimum cut algorithms was presented by Chekuri, Goldberg, Karger, Levine, and Stein [1997].

15.3b. Finding the 2-edge-connected components

Let $G = (V, E)$ be an undirected graph and let $k \in \mathbb{Z}_+$. Consider the relation $\sim$ on V defined by:

$$(15.12) \qquad u \sim v \iff G \text{ has } k \text{ edge-disjoint } u-v \text{ paths.}$$

Then $\sim$ is an equivalence relation. This can be seen with Menger's theorem. If $u \sim v$ and $v \sim w$, then $u \sim w$; otherwise, there is a $u-w$ cut C of size less than k. Then C is also a $u-v$ cut or a $v-w$ cut, contradicting the fact that $u \sim v$ and $v \sim w$.

The equivalence classes are called the *k-edge-connected components* of G. So the 1-edge connected components of G coincide with the components of G, and can be found in linear time by Corollary 6.6a. Also for $k = 2$, the k-edge-connected components can be found in linear time (Karzanov [1970]; we follow the proof of Tarjan [1972]):

Theorem 15.12. *Given an undirected graph $G = (V, E)$, its 2-edge-connected components can be found in linear time.*

Proof. We may assume that G is connected, since by Corollary 6.6a, the components of G can be found in linear time.

Choose $s \in V$ arbitrarily, and consider a depth-first search tree T starting at s. Orient each edge in T away from s. For each remaining edge $e = uv$, there is a directed path in T that connects u and v. Let the path run from u to v. Then orient e from v to u. This gives the orientation D of G.

Then any edge not in T belongs to a directed circuit in D. Moreover, any edge in T that is not a cut edge, belongs to a directed circuit in D. Then the 2-edge-connected components of G coincide with the strong components of D. By Theorem 6.6, these components can be found in linear time. ∎

More on finding 2-edge-connected components can be found in Gabow [2000a].

15.4. Gomory-Hu trees

In previous sections of this chapter we have considered the problem of determining a minimum cut in a graph, where the minimum is taken over all pairs s, t. The *all-pairs minimum-size cut problem* asks for a minimum $s - t$ cut for all pairs of vertices s, t. Clearly, this can be solved in time $O(n^2\tau)$, where τ is the time needed for finding a minimum $s - t$ cut for any given s, t.

Gomory and Hu [1961] showed that for *undirected* graphs it can be done faster, and that there is a concise structure, the Gomory-Hu tree, to represent all minimum cuts. Similarly for the capacitated case.

Fix an undirected graph $G = (V, E)$ and a capacity function $c : E \to \mathbb{R}_+$. A *Gomory-Hu tree* (*for* G and c) is a tree $T = (V, F)$ such that for each edge $e = st$ of T, $\delta(U)$ is a minimum-capacity $s-t$ cut of G, where U is any of the two components of $T - e$. (Note that it is not required that T is a subgraph of G.)

Gomory and Hu [1961] showed that for each G, c there indeed exists a Gomory-Hu tree, and that it can be found by $n - 1$ minimum-cut computations.

For distinct $s, t \in V$, define $r(s, t)$ as the minimum capacity of an $s - t$ cut. The following triangle inequality holds:

(15.13) $$r(u, w) \geq \min\{r(u, v), r(v, w)\}$$

for all distinct $u, v, w \in G$. Now a Gomory-Hu tree indeed describes concisely minimum-capacity $s - t$ cuts for all s, t:

Theorem 15.13. *Let $T = (V, F)$ be a Gomory-Hu tree. Consider any $s, t \in V$, the $s-t$ path P in T, an edge $e = uv$ on P with $r(u, v)$ minimum, and any component K of $T-e$. Then $r(s, t) = r(u, v)$ and $\delta(K)$ is a minimum-capacity $s - t$ cut.*

Proof. Inductively, (15.13) gives $r(s, t) \geq r(u, v)$. Moreover, $\delta(K)$ is an $s - t$ cut, and hence $r(s, t) \leq c(\delta(K)) = r(u, v)$. ▮

To show that a Gomory-Hu tree does exist, we first prove:

Lemma 15.14α. *Let $s, t \in V$, let $\delta(U)$ be a minimum-capacity $s - t$ cut in G, and let $u, v \in U$ with $u \neq v$. Then there exists a minimum-capacity $u - v$ cut $\delta(W)$ with $W \subseteq U$.*

Proof. Consider a minimum-capacity $u - v$ cut $\delta(X)$. By symmetry we may assume that $s \in U$ (otherwise interchange s and t), $t \notin U$, $s \in X$ (otherwise replace X by $V \setminus X$), $u \in X$ (otherwise interchange u and v), and $v \notin X$. So one of the diagrams of Figure 15.2 applies.

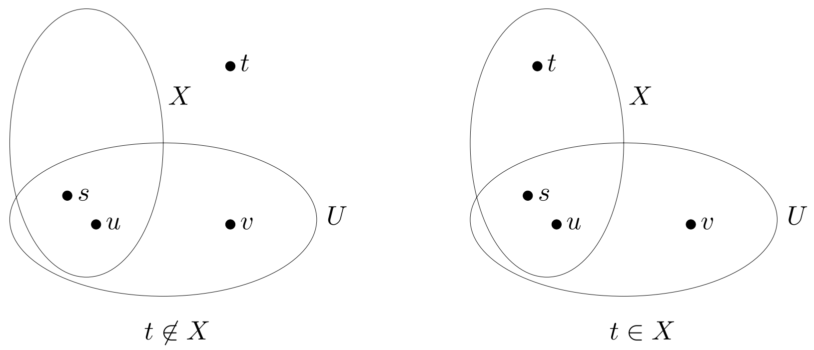


Figure 15.2

In particular, $\delta(U \cap X)$ and $\delta(U \setminus X)$ are $u-v$ cuts. If $t \notin X$, then $\delta(U \cup X)$ is an $s - t$ cut. As

(15.14) $\quad c(\delta(U \cap X)) + c(\delta(U \cup X)) \leq c(\delta(U)) + c(\delta(X))$

and

(15.15) $\quad c(\delta(U \cup X)) \geq c(\delta(U))$,

we have $c(\delta(U \cap X)) \leq c(\delta(X))$. So $\delta(U \cap X)$ is a minimum-capacity $u - v$ cut.

If $t \in X$, then $\delta(X \setminus U)$ is an $s - t$ cut. As

(15.16) $\quad c(\delta(U \setminus X)) + c(\delta(X \setminus U)) \leq c(\delta(U)) + c(\delta(X))$

and

(15.17) $\qquad c(\delta(X \setminus U)) \geq c(\delta(U)),$

we have $c(\delta(U \setminus X)) \leq c(\delta(X))$. So $\delta(U \setminus X)$ is a minimum-capacity $u - v$ cut. ∎

This lemma is used in proving the existence of Gomory-Hu trees:

Theorem 15.14. *For each graph $G = (V, E)$ and each capacity function $c : E \to \mathbb{R}_+$ there exists a Gomory-Hu tree.*

Proof. Define a *Gomory-Hu tree for* a set $R \subseteq V$ to be a pair of a tree (R, T) and a partition $(C_r \mid r \in R)$ of V such that:

(15.18) (i) $r \in C_r$ for each $r \in R$,
(ii) $\delta(U)$ is a minimum-capacity $s-t$ cut for each edge $e = st \in T$, where $U := \bigcup_{u \in K} C_u$ and K is a component of $T - e$.

We show by induction on $|R|$ that for each nonempty $R \subseteq V$ there exists a Gomory-Hu tree for R. Then for $R = V$ we have a Gomory-Hu tree.

If $|R| = 1$, (15.18) is trivial, so assume $|R| \geq 2$. Let $\delta(W)$ be a minimum-capacity cut separating at least one pair of vertices in R. Contract $V \setminus W$ to one vertex, v' say, giving graph G'. Let $R' := R \cap W$. By induction, G' has a Gomory-Hu tree (R', T'), $(C'_r \mid r \in R')$ for R'.

Similarly, contract W to one vertex, v'' say, giving graph G''. Let $R'' := R \setminus W$. By induction, G'' has a Gomory-Hu tree (R'', T''), $(C''_r \mid r \in R'')$ for R''.

Now let $r' \in R'$ be such that $v' \in C'_{r'}$. Similarly, let $r'' \in R''$ be such that $v'' \in C''_{r''}$. Let $T := T' \cup T'' \cup \{r'r''\}$, Let $C_{r'} := C'_{r'} \setminus \{v'\}$ and let $C_r := C'_r$ for all other $r \in R'$. Similarly, let $C_{r''} := C''_{r''} \setminus \{v''\}$ and let $C_r := C''_r$ for all other $r \in R''$.

Now (R, T) and the C_r form a Gomory-Hu tree for R. Indeed, for any $e \in T$ with $e \neq r'r''$, (15.18) follows from Lemma 15.14α. If $e = r'r''$, then $U = W$ and $\delta(W)$ is a minimum-capacity $r' - r''$ cut (as it is minimum-capacity over all cuts separating at least one pair of vertices in R). ∎

The method can be sharpened to give the following algorithmic result:

Theorem 15.15. *A Gomory-Hu tree can be found by $n - 1$ applications of a minimum-capacity cut algorithm.*

Proof. In the proof of Theorem 15.14, it suffices to take for $\delta(W)$ just a minimum-capacity $s - t$ cut for at least one pair $s, t \in R$. Then $\delta(W)$ is also a minimum-capacity $r' - r''$ cut. For suppose that there exists an $r' - r''$ cut $\delta(X)$ of smaller capacity. We may assume that $s \in W$ and $t \notin W$. As $\delta(W)$ is a minimum-capacity $s - t$ cut, $\delta(X)$ is not an $s - t$ cut. So it should separate

s and r' or t and r''. By symmetry, we may assume that it separates s and r'. Then it also as is a $u-v$ cut for some edge uv on the $s-r'$ path in T'. Let uv determine cut $\delta(U)$. This cut is an $s-t$ cut, and hence $c(\delta(U)) \geq c(\delta(W))$. On the other hand, $c(\delta(U)) \leq c(\delta(X))$, as $\delta(U)$ is a minimum-capacity $u-v$ cut. This contradicts our assumption that $c(\delta(X)) < c(\delta(W))$. ∎

This implies for the running time:

Corollary 15.15a. *A Gomory-Hu tree can be found in time $O(n\tau)$ time, if for any $s, t \in V$ a minimum-capacity $s-t$ cut can be found in time τ.*

Proof. Directly from Theorem 15.15. ∎

Notes. The method gives an $O(m^2)$ method to find a Gomory-Hu tree for the capacity function $c = \mathbf{1}$, since $O(m^2) = O(\sum_v d(v)m)$, and for each new vertex v a minimum cut can be found in time $O(d(v)m)$. Hao and Orlin [1992,1994] gave an $O(n^3)$-time method to find, for given graph $G = (V, E)$ and $s \in V$, all minimum-size $s-t$ cuts for all $t \neq s$ (with push-relabel). Shiloach [1979b] gave an $O(n^2m)$ algorithm to find a maximum number of edge-disjoint paths between all pairs of vertices in an undirected graph. Ahuja, Magnanti, and Orlin [1993] showed that the best directed all-pairs cut algorithm takes $\Omega(n^2)$ max-flow iterations.

For planar graphs, Hartvigsen and Mardon [1994] gave an $(n^2 \log n + m)$ algorithm to find a Gomory-Hu tree (they observed that this bound can be derived also from Frederickson [1987b]). This improves a result of Shiloach [1980a], who gave an $O(n^2(\log n)^2)$-time algorithm to find minimum-size cuts between all pairs of vertices in a planar graph.

Theorem 15.13 implies that a Gomory-Hu tree for a graph $G = (V, E)$ is a maximum-weight spanning tree in the complete graph on V, for weight function $r(u, v)$. However, not every maximum-weight spanning tree is a Gomory-Hu tree (for $G = K_{1,2}$, $c = \mathbf{1}$, only G itself is a Gomory-Hu tree, but all spanning trees on $VK_{1,2}$ have the same weight).

More on Gomory-Hu trees can be found in Elmaghraby [1964], Hu and Shing [1983], Agarwal, Mittal, and Sharma [1984], Granot and Hassin [1986], Hassin [1988], Chen [1990], Gusfield [1990], Hartvigsen and Margot [1995], Talluri [1996], Goldberg and Tsioutsiouliklis [1999,2001], and Hartvigsen [2001b]. Generalizations were given by Cheng and Hu [1990,1991,1992] and Hartvigsen [1995] (to matroids).

15.4a. Minimum-requirement spanning tree

Hu [1974] gave the following additional application of Gomory-Hu trees. Let $G = (V, E)$ be an undirected graph and let $r : E \to \mathbb{R}_+$ be a 'requirement' function (say, the number of telephone calls to be made between the end vertices of e).

We want to find a tree T on V minimizing

$$\sum_{e \in E} r(e)\mathrm{dist}_T(e), \tag{15.19}$$

where $\text{dist}_T(e)$ denotes the distance in T between the end vertices of e.

Now any Gomory-Hu tree T for G and capacity function r indeed minimizes (15.19). To see this, let for any edge f of T, $R_T(f)$ be equal to the requirement (= capacity) of the cut determined by the two components of $T - f$. Then (15.19) is equal to

(15.20) $$\sum_{f \in T} R_T(f).$$

Now T minimizes (15.20), as was shown by Adolphson and Hu [1973]. For let T' be any other spanning tree on V. Then for each $f = st \in T$ and each edge f' on the $s-t$ path in T' one has

(15.21) $$R_{T'}(f') \geq R_T(f),$$

since the components of $T - f$ determine a minimum-capacity $s - t$ cut, and since the components of $T' - f'$ determine an $s - t$ cut. Since T and T' are spanning trees, there exists a one-to-one function $\phi : T \to T'$ such that for each $f = st \in T$, $\phi(f)$ is an edge on the $s - t$ path in T'.

To see this, let u be an end vertex of T. Let $f = uv$ be the edge of T incident with u, and define $\phi(f)$ to be the first edge of the $u - v$ path in T'. Delete f and contract $\phi(f)$. Then induction gives the required function.

So (15.21) implies that (15.20) is not decreased by replacing T by T'. Hence T minimizes (15.20), and therefore also (15.19).

15.5. Further results and notes

15.5a. Ear-decomposition of undirected graphs

In Section 6.5c we characterized the strongly connected digraphs as those digraphs having an ear-decomposition. We now consider the undirected case, and we will see a correspondence between ear-decompositions and 2-(edge-)connected graphs.

Let $G = (V, E)$ be an undirected graph. An *ear* of G is a path or circuit P in G, of length ≥ 1, such that all internal vertices of P have degree 2 in G. The path may consist of a single edge — so any edge of G is an ear. A *proper ear* is an ear that is a path, that is, has two different ends.

If I is the set of internal vertices of an ear P, we say that G arises from $G - I$ by *adding ear*. An *ear-decomposition* of G is a series of graphs $G_0, G_1, \ldots, G_k$, where $G_0 = K_1$, $G_k = G$, and G_i arises from G_{i-1} by adding an ear ($i = 1, \ldots, k$). If $G_0 = K_2$ and G_i arises from G_{i-1} by adding a proper ear, it is a *proper ear-decomposition*.

Graphs with a *proper* ear-decomposition were characterized by Whitney [1932b]:

Theorem 15.16. *A graph $G = (V, E)$ with $|V| \geq 2$ has a proper ear-decomposition if and only if G is 2-vertex-connected.*

Proof. Necessity follows from the facts that K_2 is 2-vertex-connected and that 2-vertex-connectivity is maintained under adding proper ears. To see sufficiency, let G be 2-vertex-connected, and let $G' = (V', E')$ be a subgraph of G that has a proper ear-decomposition, with $|E'|$ as large as possible. Suppose that $E' \neq E$, and let $e = uv$ be an edge in $E \setminus E'$ incident with V'; say $u \in V'$. By the 2-connectivity of G, there is a path from v to V' avoiding u. Let P be a shortest such path. Then path e, P is a proper ear that can be added to G', contradicting the maximality of $|E'|$. ∎

Similarly, graphs having an ear-decomposition are characterized by being 2-edge-connected (this is implicit in Robbins [1939]):

Theorem 15.17. *A graph $G = (V, E)$ has an ear-decomposition if and only if G is 2-edge-connected.*

Proof. Necessity follows from the facts that K_1 is 2-edge-connected and that 2-edge-connectivity is maintained under adding ears. To see sufficiency, let G be 2-edge-connected, and let $G' = (V', E')$ be a subgraph of G that has an ear-decomposition, with $|E'|$ as large as possible. Suppose that $E' \neq E$, and let $e = uv$ be an edge in $E \setminus E'$ incident with V'; say $u \in V'$. Let C be a circuit in G traversing e. Let C start with $u, e, \ldots$. Let s be the first vertex in C, after e, that belongs to V'. Then subpath $P = u, e \ldots, w$ of C is an ear that can be added to G', contradicting the maximality of $|E'|$. ∎

15.5b. Further notes

Dinits, Karzanov, and Lomonosov [1976] showed that the set of all minimum-capacity cuts of an undirected graph (with positive capacities on the edges) has the "cactus structure", as follows. A *cactus* is a connected graph such that each edge belongs to at most one circuit. Let $G = (V, E)$ be a graph with a capacity function $c : E \to \mathbb{Z}_+$ such that the minimum cut capacity λ is positive. Then there exist a cactus K with $O(|V|)$ vertices and a function $\phi : V \to VK$ such that for each inclusionwise minimal cut $\delta_K(U)$ of K, the set $\delta_G(\phi^{-1}(U))$ is a cut of capacity λ, and such that each minimum-capacity cut in G can be obtained this way. Moreover, K is a tree when λ is odd. It follows that the number of minimum-capacity cuts is at most $\binom{n}{2}$ (and at most $n-1$ when λ is odd), and that the vertices of G can be ordered as $v_1, \ldots, v_n$ so that each minimum-capacity cut is of the form $\delta(\{v_i, v_{i+1}, \ldots, v_j\})$ for some $i \leq j$. Related results can be found in Picard and Queyranne [1980], Karzanov and Timofeev [1986], Gabow [1991b,1993b,1995c], Gusfield and Naor [1993], Karger and Stein [1993,1996], Nagamochi, Nishimura, and Ibaraki [1994,1997], Benczúr [1995], Henzinger and Williamson [1996], Karger [1996,2000], Fleischer [1998a,1999b], Dinitz and Vainshtein [2000], and Nagamochi, Nakao, and Ibaraki [2000].

Gusfield and Naor [1990,1991] considered the analogue of the Gomory-Hu tree for *vertex*-cuts.

A theorem of Mader [1971] implies that each k-connected graph $G = (V, E)$ contains a k-connected spanning subgraph with $O(k|V|)$ edges — similarly for k-edge-connected. This was extended by Nagamochi and Ibaraki [1992a], showing that for each k, each graph $G = (V, E)$ has a subgraph $G_k = (V, E_k)$ such that $|E_k| = O(k|V|)$ and such that for all $s, t \in V$:

(15.22) (i) $\lambda_{G_k}(s,t) \geq \min\{\lambda_G(s,t), k\}$,

(ii) $\kappa_{G_k}(s,t) \geq \min\{\kappa_G(s,t), k\}$ if $st \notin E$.

Here $\lambda_H(s,t)$ ($\kappa_H(s,t)$, respectively) denotes the maximum number of edge-disjoint (internally vertex-disjoint, respectively) $s-t$ paths in H. They also gave a linear-time algorithm finding G_k. A shorter proof and a generalization was given by Frank, Ibaraki, and Nagamochi [1993].

Frank [1995] showed that the following is implied by the existence of a Gomory-Hu tree. Let $G = (V,E)$ be an undirected graph of minimum degree k. Then there exist two distinct vertices $s,t \in V$ connected by k edge-disjoint paths. This follows by taking for s a vertex of degree 1 in the Gomory-Hu tree, and for t its neighbour in this tree. Then $\delta_E(s)$ is a minimum-size cut separating s and t.

Tamir [1994] observed that the up hull P of the incidence vectors of the nontrivial cuts of an undirected graph $G = (V,E)$ can be described as follows. Let $D = (V,A)$ be the digraph with A being the set of all ordered pairs (u,v) for adjacent $u,v \in V$. Choose $r \in V$ arbitrarily. Then P is equal to the projection to x-space of the polyhedron in the variables $x \in \mathbb{R}^E$ and $y \in \mathbb{R}^A$ determined by:

(15.23)
$$\begin{array}{rll} \text{(i)} & y(a) \geq 0 & \text{for each } a \in A, \\ \text{(ii)} & y(B) \geq 1 & \text{for each } r\text{-arborescence } B, \\ \text{(iii)} & x(e) = y(u,v) + y(v,u) & \text{for each edge } e = uv \text{ of } G. \end{array}$$

(Here an *r-arborescence* is a subset B of A such that (V,B) is a rooted tree rooted at r.) This can be shown with the help of the results to be discussed in Chapter 53. To see this, consider any $c \in \mathbb{R}^E_+$. Then the minimum value of $c^{\mathsf{T}}x$ over all x,y satisfying (15.23), is equal to the minimum value of $d^{\mathsf{T}}y$ over all x,y satisfying (15.23), where $d(u,v) := c(uv)$ for each $(u,v) \in A$. This is equal to the minimum value of $d^{\mathsf{T}}y$ over all y satisfying (i) and (ii) of (15.23). By Corollary 53.1f, below this is equal to the minimum d-weight of an r-cut in D, which is equal to the minimum c-weight of a nontrivial cut in G.

No explicit description in terms of linear inequalities is known for the up hull of the incidence vectors of nontrivial cuts. Alevras [1999] gave descriptions for small instances (up to seven vertices for undirected graphs and up to five vertices for directed graphs).

The minimum k-cut problem: 'find a partition of the vertex set of a graph into k nonempty classes such that the number of edges connecting different classes is minimized', is NP-complete if k is part of the input (there is an easy reduction from the maximum clique problem, as the problem is equivalent to maximizing the number of edges spanned by the classes in the partition). For fixed k however, it was shown to be polynomial-time solvable by Goldschmidt and Hochbaum [1988,1994]. If we prescribe certain vertices to belong to the classes, the problem is NP-complete even for $k = 3$ (Dahlhaus, Johnson, Papadimitriou, Seymour, and Yannakakis [1992, 1994]). More on this problem can be found in Hochbaum and Shmoys [1985], Lee, Park, and Kim [1989], Chopra and Rao [1991], Cunningham [1991], He [1991], Saran and Vazirani [1991,1995], Garg, Vazirani, and Yannakakis [1994], Kapoor [1996], Burlet and Goldschmidt [1997], Kamidoi, Wakabayashi, and Yoshida [1997], Călinescu, Karloff, and Rabani [1998,2000], Hartvigsen [1998b], Karzanov [1998c], Cunningham and Tang [1999], Karger, Klein, Stein, Thorup, and Young [1999], Nagamochi and Ibaraki [1999a,2000], Nagamochi, Katayama, and Ibaraki [1999, 2000], Goemans and Williamson [2001], Naor and Rabani [2001], Zhao, Nagamochi, and Ibaraki [2001], and Ravi and Sinha [2002].

Surveys on connectivity are given by Even [1979], Mader [1979], Frank [1995], and Subramanian [1995] (edge-connectivity). For the decomposition of 3-connected graphs into 4-connected graphs, see Coullard, Gardner, and Wagner [1993].

Part II

Bipartite Matching and Covering

Part II: Bipartite Matching and Covering

A second classical area of combinatorial optimization is formed by bipartite matching. The area gives rise to a number of basic problems and techniques, and has an abundance of applications in various forms of assignment and transportation.

Work of Frobenius in the 1910s on the decomposition of matrices formed the incentive to Kőnig to study matchings in bipartite graphs. An extension by Egerváry in the 1930s to weighted matchings inspired Kuhn in the 1950s to design the 'Hungarian method' for the assignment problem (which is equivalent to finding a minimum-weight perfect matching in a complete bipartite graph).

Parallel to this, Tolstoĭ, Kantorovich, Hitchcock, and Koopmans had investigated the transportation problem. It motivated Kantorovich and Dantzig to consider more general problems, culminating in the development of linear programming. It led in turn to solving the assignment problem by linear programming, and thus to a polyhedral approach.

Several variations and extensions of bipartite matching, like edge covers, factors, and transversals, can be handled similarly. Major explanation is the total unimodularity of the underlying matrices.

Bipartite matching and transportation can be considered as special cases of disjoint paths and of transshipment, studied in the previous part — just consider a bipartite graph as a directed graph, by orienting all edges from one colour class to the other. It was however observed by Hoffman and Orden that this can be turned around, and that disjoint paths and transshipment problems can be reduced to bipartite matching and transportation problems. So several results in this part on bipartite matching are matched by results in the previous part on paths and flows. Viewed this way, the present part forms a link between the previous part and the next part on *nonbipartite* matching, where the underlying matrices generally are not totally unimodular.

Chapters:

Chapter 16

Cardinality bipartite matching and vertex cover

'Cardinality matching' deals with maximum-size matchings. In this chapter we give the theorems of Frobenius on the existence of a perfect matching in a bipartite graph, and the extension by Kőnig on the maximum size of a matching in a bipartite graph. We also discuss finding a maximum-size matching in a bipartite graph algorithmically.
We start with an easy but fundamental theorem relating maximum-size matchings and M-alternating paths, that applies to any graph and that will also be important for nonbipartite matching.
In this chapter, graphs can be assumed to be simple.

16.1. M-augmenting paths

Let $G = (V, E)$ be an undirected graph. A *matching* in G is a set of disjoint edges. An important concept in finding a maximum-size matching, both in bipartite and in nonbipartite graphs, is that of an 'augmenting path' (introduced by Petersen [1891]).

Let M be a matching in a graph $G = (V, E)$. A path P in G is called *M-augmenting* if P has odd length, its ends are not covered by M, and its edges are alternatingly out of and in M.

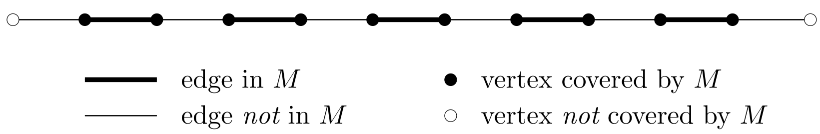


Figure 16.1
An M-augmenting path

Clearly, if P is an M-augmenting path, then

$$M' := M \triangle EP \tag{16.1}$$

is again a matching and satisfies $|M'| = |M| + 1$.[1] In fact, it is not difficult to show (Petersen [1891]):

Theorem 16.1. *Let $G = (V, E)$ be a graph and let M be a matching in G. Then either M is a matching of maximum size or there exists an M-augmenting path.*

Proof. If M is a maximum-size matching, there cannot exist an M-augmenting path P, since otherwise $M \triangle EP$ would be a larger matching.

Conversely, if M' is a matching larger than M, consider the components of the graph $G' := (V, M \cup M')$. Then G' has maximum degree two. Hence each component of G' is either a path (possibly of length 0) or a circuit. Since $|M'| > |M|$, at least one of these components should contain more edges in M' than in M. Such a component forms an M-augmenting path. ∎

So in any graph, if we have an algorithm finding an M-augmenting path for any matching M, then we can find a maximum-size matching: we iteratively find matchings $M_0, M_1, \ldots$, with $|M_i| = i$, until we have a matching M_k such that there exists no M_k-augmenting path. (Also this was observed by Petersen [1891].)

16.2. Frobenius' and Kőnig's theorems

A classical min-max relation due to Kőnig [1931] characterizes the maximum size of a matching in a bipartite graph. To this end, call a set C of vertices of a graph G a *vertex cover* if each edge of G intersects C. Define

(16.2) $\nu(G) :=$ the maximum size of a matching in G,
$\tau(G) :=$ the minimum size of a vertex cover in G.

These numbers are called the *matching number* and the *vertex cover number* of G, respectively. It is easy to see that, for any graph G,

(16.3) $\nu(G) \leq \tau(G)$,

since any two edges in any matching contain different vertices in any vertex cover. The graph K_3 has strict inequality in (16.3). However, if G is bipartite, equality holds, which is the content of Kőnig's matching theorem (Kőnig [1931]). It can be seen to be equivalent to a theorem of Frobenius [1917] (Corollary 16.2a below).

Theorem 16.2 (Kőnig's matching theorem)**.** *For any bipartite graph $G = (V, E)$ one has*

(16.4) $\nu(G) = \tau(G)$.

[1] EP denotes the set of edges in P. $\triangle$ denotes symmetric difference.

That is, the maximum size of a matching in a bipartite graph is equal to the minimum size of a vertex cover.

Proof. By (16.3) it suffices to show that $\nu(G) \geq \tau(G)$. We may assume that G has at least one edge. Then:

(16.5) G has a vertex u covered by each maximum-size matching.

To see this, let $e = uv$ be any edge of G, and suppose that there are maximum-size matchings M and N missing u and v respectively[2]. Let P be the component of $M \cup N$ containing u. So P is a path with end vertex u. Since P is not M-augmenting (as M has maximum size), P has even length, and hence does not traverse v (otherwise, P ends at v, contradicting the bipartiteness of G). So $P \cup e$ would form an N-augmenting path, a contradiction (as N has maximum size). This proves (16.5).

Now (16.5) implies that for the graph $G' := G - u$ one has $\nu(G') = \nu(G) - 1$. Moreover, by induction, G' has a vertex cover C of size $\nu(G')$. Then $C \cup \{u\}$ is a vertex cover of G of size $\nu(G') + 1 = \nu(G)$. ∎

(This proof is due to De Caen [1988]. For Kőnig's original, algorithmic proof, see the proof of Theorem 16.6. Note that also Menger's theorem implies Kőnig's matching theorem (using the construction given in the proof of Theorem 16.4 below). For a proof based on showing that any minimum bipartite graph with a given vertex cover number is a matching, see Lovász [1975d]. For another proof (of Rizzi [2000a]), see Section 16.2c. As we will see in Chapter 18, Kőnig's matching theorem also follows from the total unimodularity of the incidence matrix of a bipartite graph. (Flood [1960] and Entringer and Jackson [1969] gave proofs similar to Kőnig's proof.))

A consequence of Theorem 16.2 is a theorem of Frobenius [1917] that characterizes the existence of a perfect matching in a bipartite graph. (A matching is *perfect* if it covers all vertices.) Actually, this theorem motivated Kőnig to study matchings in graphs, and in turn it can be seen to imply Kőnig's matching theorem.

Corollary 16.2a (Frobenius' theorem)**.** *A bipartite graph $G = (V, E)$ has a perfect matching if and only if each vertex cover has size at least $\frac{1}{2}|V|$.*

Proof. Directly from Kőnig's matching theorem, since G has a perfect matching if and only if $\nu(G) \geq \frac{1}{2}|V|$. ∎

This implies an earlier theorem of Kőnig [1916] on *regular* bipartite graphs:

---

[2] M *misses* a vertex u if $u \notin \bigcup M$. Here $\bigcup M$ denotes the union of the edges in M; that is, the set of vertices covered by the edges in M.

Corollary 16.2b. *Each regular bipartite graph (of positive degree) has a perfect matching.*

Proof. Let $G = (V, E)$ be a k-regular bipartite graph. So each vertex is incident with k edges. Since $|E| = \frac{1}{2}k|V|$, we need at least $\frac{1}{2}|V|$ vertices to cover all edges. Hence Corollary 16.2a implies the existence of a perfect matching. ∎

Let A be the $V \times E$ incidence matrix of the bipartite graph $G = (V, E)$. Kőnig's matching theorem (Theorem 16.2) states that the optima in the linear programming duality equation

$$\max\{\mathbf{1}^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq \mathbf{1}\} = \min\{y^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq \mathbf{1}^{\mathsf{T}}\} \tag{16.6}$$

are attained by integer vectors x and y. This can also be derived from the total unimodularity of A — see Section 18.3.

16.2a. Frobenius' proof of his theorem

The proof method given by Frobenius [1917] of Corollary 16.2a is in terms of matrices, but can be formulated in terms of graphs as follows. Necessity of the condition being easy, we prove sufficiency. Let U and W be the colour classes of G. As both U and W are vertex covers, and hence have size at least $\frac{1}{2}|V|$, we have $|U| = |W| = \frac{1}{2}|V|$.

Choose an edge $e = \{u, w\}$ with $u \in U$ and $w \in W$. We may assume that $G - u - w$ has no perfect matching. So, inductively, $G - u - w$ has a vertex cover C' with $|C'| < |U| - 1$. Then $C := C' \cup \{u, w\}$ is a vertex cover of G, with $|C| \leq |U|$, and hence $|C| = |U|$.

Now $U \triangle C$ and $W \triangle C$ partition V (where $\triangle$ denotes symmetric difference). If both $U \triangle C$ and $W \triangle C$ are matchable[3], then G has a perfect matching. So, by symmetry, we may assume that $U \triangle C$ is not matchable. Now $U \triangle C \neq V$ as $u \notin U \triangle C$. Hence we can apply induction, giving that $G[U \triangle C]$ has a vertex cover D with $|D| < \frac{1}{2}|U \triangle C|$. Then the set $D \cup (U \cap C)$ is a vertex cover of G (since each edge of G intersects both U and C, and hence it either intersects $U \cap C$, or is contained in $U \triangle C$ and hence intersects D). However, $|D| + |U \cap C| < \frac{1}{2}|U \triangle C| + |U \cap C| = \frac{1}{2}(|U| + |C|) = \frac{1}{2}|V|$, a contradiction.

(This is essentially also the proof method of Rado [1933] and Dulmage and Halperin [1955].)

16.2b. Linear-algebraic proof of Frobenius' theorem

Frobenius [1917] was motivated by a determinant problem, namely by the following direct consequence of his theorem. Let $A = (a_{i,j})$ be an $n \times n$ matrix in which each entry $a_{i,j}$ is either 0 or a variable $x_{i,j}$ (where the variables $x_{i,j}$ are independent). Then Frobenius' theorem is equivalent to: $\det A = 0$ if and only if A has a $k \times l$ all-zero submatrix with $k + l > n$. (Earlier, Frobenius [1912] showed that for such

[3] A set T of vertices is called *matchable* if there exists a matching M with $T = \bigcup M$.

a matrix A, $\det A$ is reducible (that is, there exist nonconstant polynomials p and q with $\det A = p \cdot q$) if and only if A has a $k \times l$ all-zero submatrix with $k + l = n$ and $k, l \geq 1$.)

Edmonds [1967b] showed that the argumentation can be applied also the other way around. This gives the following linear-algebraic proof of Frobenius' theorem (implying linear-algebraic proofs also of other bipartite matching theorems).

Let $G = (V, E)$ be a bipartite graph not having a perfect matching. Let U and W be the colour classes of G. We may assume that $|U| = |W|$ (otherwise the smaller colour class is a vertex cover of size less than $\frac{1}{2}|V|$).

Make a $U \times W$ matrix $A = (a_{u,w})$, where $a_{u,w} = 0$ if u and w are not adjacent, and $a_{u,w} = x_{u,w}$ otherwise, where the $x_{u,w}$ are independent variables.

As G has no perfect matching, we know that $\det A = 0$, and hence the columns of A are linearly dependent. Let $W' \subseteq W$ be the index set of a minimal set of linearly dependent columns of A. Then there is a subset U' of U with $|U'| = |W'| - 1$ such that the $U' \times W'$ submatrix A' of A has rank $|U'|$. Hence there is a vector y such that $A'y = \mathbf{0}$ and such that each entry in y is a nonzero polynomial in those variables $x_{u,w}$ that occur in A'. Let A'' be the $U \times W'$ submatrix of A. Then $A''y = \mathbf{0}$, and hence all entries in the $(U \setminus U') \times W'$ submatrix of A are 0. Hence the rows in U' and columns in $W \setminus W'$ cover all nonzeros. As $|U'| + |W \setminus W'| < |W|$, we have Frobenius' theorem.

16.2c. Rizzi's proof of Kőnig's matching theorem

Rizzi [2000a] gave the following short proof of Kőnig's matching theorem. Let $G = (V, E)$ be a counterexample with $|V| + |E|$ minimal. Then G has a vertex u of degree at least 3. Let v be a neighbour of u. By the minimality of G, $G - v$ has a vertex cover U of size $\nu(G - v)$. Then $U \cup \{v\}$ is a vertex cover of G. As G is a counterexample, we have $|U \cup \{v\}| \geq \nu(G) + 1$, and so $\nu(G - v) = |U| \geq \nu(G)$. Therefore, G has a maximum-size matching M not covering v. Let $f \in E \setminus M$ be incident with u and not with v. Then $\nu(G - f) \geq |M| = \nu(G)$. Let W be a vertex cover of $G - f$ of size $\nu(G - f) = \nu(G)$. Then $v \notin W$, since v is not covered by M. Hence $u \in W$, as W covers edge uv of $G - f$. Therefore, W also covers f, and hence it is a vertex cover of G of size $\nu(G)$.

16.3. Maximum-size bipartite matching algorithm

We now focus on the problem of finding a maximum-size matching in a bipartite graph algorithmically. In view of Theorem 16.1, this amounts to finding an augmenting path. In the bipartite case, this can be done by finding a directed path in an auxiliary directed graph. This method is essentially due to van der Waerden [1927] and Kőnig [1931].

Matching augmenting algorithm for bipartite graphs

input: a bipartite graph $G = (V, E)$ and a matching M,
output: a matching M' satisfying $|M'| > |M|$ (if there is one).

description of the algorithm: Let G have colour classes U and W. Make a directed graph D_M by orienting each edge $e = \{u, w\}$ of G (with $u \in U, w \in W$) as follows:

(16.7) if $e \in M$, then orient e from w to u,
if $e \notin M$, then orient e from u to w.

Let U_M and W_M be the sets of vertices in U and W (respectively) missed by M.

Now an M-augmenting path (if any) can be found by finding a directed path in D_M from U_M to W_M. This gives a matching larger than M. ∎

The correctness of this algorithm is immediate. Since a directed path can be found in time $O(m)$, we can find an augmenting path in time $O(m)$. Hence we have the following result (implicit in Kuhn [1955b]):

Theorem 16.3. *A maximum-size matching in a bipartite graph can be found in time $O(nm)$.*

Proof. Note that we do at most n iterations, each of which can be done in time $O(m)$ by breadth-first search (Theorem 6.3). ∎

16.4. An $O(n^{1/2}m)$ algorithm

Hopcroft and Karp [1971,1973] and Karzanov [1973b] proved the following sharpening of Theorem 16.3, which we derive from the (equivalent) result of Karzanov [1973a], Tarjan [1974e], and Even and Tarjan [1975] on the complexity of finding a maximum number of vertex-disjoint paths (Corollary 9.7a).

Theorem 16.4. *A maximum-size matching in a bipartite graph can be found in $O(n^{1/2}m)$ time.*

Proof. Let $G = (V, E)$ be a bipartite graph, with colour classes U and W. Make a directed graph $D = (V, A)$ as follows. Orient all edges from U to W. Moreover, add a new vertex s, with arcs (s, u) for all $u \in U$, and a new vertex t, with arcs (w, t) for all $w \in W$. Then the maximum number of internally vertex-disjoint $s - t$ paths in D is equal to the maximum size of a matching in G. The result now follows from Corollary 9.7a. ∎

In fact, the factor $n^{1/2}$ can be reduced to $\nu(G)^{1/2}$ (as before, $\nu(G)$ and $\tau(G)$ denote the maximum size of a matching and the minimum size of a vertex cover, respectively):

Theorem 16.5. *A maximum-size matching in a bipartite graph G can be found in $O(\nu(G)^{1/2}m)$ time.*

Proof. Similar to the proof of Theorem 16.4, using Theorem 9.8 and the fact that $\nu(G) = \tau(G)$. ∎

Gabow and Tarjan [1988a] observed that the method of Corollary 9.7a applied to the bipartite matching problem implies that for each k one can find in time $O(km)$ a matching of size at least $\nu(G) - \frac{n}{k}$.

16.5. Finding a minimum-size vertex cover

From a maximum-size matching in a bipartite graph, one can derive a minimum-size vertex cover. The method gives an alternative proof of Kőnig's matching theorem (in fact, this is the original proof of Kőnig [1931]):

Theorem 16.6. *Given a bipartite graph G and a maximum-size matching M in G, we can find a minimum-size vertex cover in G in time $O(m)$.*

Proof. Make D_M, U_M, and W_M as in the matching-augmenting algorithm, and let R_M be the set of vertices reachable in D_M from U_M. So $R_M \cap W_M = \emptyset$. Then each edge uw in M is either contained in R_M or disjoint from R_M (that is, $u \in R_M \iff w \in R_M$). Moreover, no edge of G connects $U \cap R_M$ and $W \setminus R_M$, as no arc of D_M leaves R_M. So $C := (U \setminus R_M) \cup (W \cap R_M)$ is a vertex cover of G. Since C is disjoint from $U_M \cup W_M$ and since no edge in M is contained in C, we have $|C| \leq |M|$. Therefore, C is a minimum-size vertex cover. ∎

Hence:

Corollary 16.6a. *A minimum-size vertex cover in a bipartite graph can be found in $O(n^{1/2}m)$ time.*

Proof. Directly from Theorems 16.4 and 16.6. ∎

16.6. Matchings covering given vertices

The following theorem characterizes when one of the colour classes of a bipartite graph can be covered by a matching, and is a direct consequence of Kőnig's matching theorem (where $N(S)$ denotes the set of vertices not in S that have a neighbour in S):

Theorem 16.7. *Let $G = (V, E)$ be a bipartite graph with colour classes U and W. Then G has a matching covering U if and only if $|N(S)| \geq |S|$ for each $S \subseteq U$.*

Proof. Necessity being trivial, we show sufficiency. By Kőnig's matching theorem (Theorem 16.2) it suffices to show that each vertex cover C has $|C| \geq |U|$. This indeed is the case, since $N(U \setminus C) \subseteq C \cap W$, and hence

$$|C| = |C \cap U| + |C \cap W| \geq |C \cap U| + |N(U \setminus C)| \geq |C \cap U| + |U \setminus C| = |U|. \tag{16.8}$$

This can be extended to general subsets of V. First, Hoffman and Kuhn [1956b] and Mendelsohn and Dulmage [1958a] showed:

Theorem 16.8. *Let $G = (V, E)$ be a bipartite graph with colour classes U and W and let $R \subseteq V$. Then there exists a matching covering R if and only if there exist a matching M covering $R \cap U$ and a matching N covering $R \cap W$.*

Proof. Necessity being trivial, we show sufficiency. We may assume that G is connected, that $E = M \cup N$, and that neither M nor N covers R. This implies that there is a $u \in R \cap U$ missed by N and a $w \in R \cap W$ missed by M. So G is an even-length $u - w$ path, a contradiction, since $u \in U$ and $w \in W$.

(This theorem goes back to theorems of F. Bernstein (cf. Borel [1898] p. 103), Banach [1924], and Knaster [1927] on injective mappings between two sets.)

Theorem 16.8 implies a characterization of sets that are covered by some matching:

Corollary 16.8a. *Let $G = (V, E)$ be a bipartite graph with colour classes U and W and let $R \subseteq V$. Then there is a matching covering R if and only if $|N(S)| \geq |S|$ for each $S \subseteq R \cap U$ and for each $S \subseteq R \cap W$.*

Proof. Directly from Theorems 16.7 and 16.8.

It also gives the following exchange property:

Corollary 16.8b. *Let $G = (V, E)$ be a bipartite graph, with colour classes U and W, let M and N be maximum-size matchings, let U' be the set of vertices in U covered by M, and let W' be the set of vertices in W covered by N. Then there exists a maximum-size matching covering $U' \cup W'$.*

Proof. Directly from Theorem 16.8: the matching found is maximum-size since $|U'| = |W'| = \nu(G)$.

Notes. These results also are special cases of the exchange results on paths discussed in Section 9.6c. Perfect [1966] gave the following linear-algebraic argument for Corollary 16.8b. Make a $U \times W$ matrix A with $a_{u,w} = x_{u,w}$ if $uw \in E$ and $a_{u,w} := 0$ otherwise, where the $x_{u,w}$ are independent variables. Let U' be any maximum-size subset of U covered by some matching and let W' be any maximum-size subset of W covered by some matching. Then U' gives a maximum-size set of

linearly independent rows of A and W' gives a maximum-size set of linearly independent columns of A. Then the $U' \times W'$ submatrix of A is nonsingular, hence of nonzero determinant. It implies (by the definition of determinant) that G has a matching covering $U' \cup W'$.
(Related work includes Perfect and Pym [1966], Pym [1967], Brualdi [1969b,1971b], and Mirsky [1969].)

16.7. Further results and notes

16.7a. Complexity survey for cardinality bipartite matching

Complexity survey for cardinality bipartite matching ($*$ indicates an asymptotically best bound in the table):

	$O(nm)$	Kőnig [1931], Kuhn [1955b]
	$O(\sqrt{n}\, m)$	Hopcroft and Karp [1971,1973], Karzanov [1973a]
$*$	$\widetilde{O}(n^\omega)$	Ibarra and Moran [1981]
	$O(n^{3/2}\sqrt{\frac{m}{\log n}})$	Alt, Blum, Mehlhorn, and Paul [1991]
$*$	$O(\sqrt{n}\, m \log_n(n^2/m))$	Feder and Motwani [1991,1995]

Here ω is any real such that any two $n \times n$ matrices can be multiplied by $O(n^\omega)$ arithmetic operations (e.g. $\omega = 2.376$).

Goldberg and Kennedy [1997] described a bipartite matching algorithm based on the push-relabel method, of complexity $O(\sqrt{n}\, m \log_n(n^2/m))$. Balinski and Gonzalez [1991] gave an alternative $O(nm)$ bipartite matching algorithm (not using augmenting paths).

16.7b. Finding perfect matchings in regular bipartite graphs

By Kőnig's matching theorem, each k-regular bipartite graph has a perfect matching (if $k \geq 1$). One can use the regularity also to find quickly a perfect matching. This will be used in Chapter 20 on bipartite edge-colouring.

First we show the following result of Cole and Hopcroft [1982] (which will not be used any further in this book):

Theorem 16.9. *A perfect matching in a regular bipartite graph can be found in $O(m \log n)$ time.*

Proof. We first describe an $O(m \log n)$-time algorithm for the following problem:

(16.9) given: a k-regular bipartite graph $G = (V, E)$ with $k \geq 2$,
find: a nonempty proper subset F of E with (V, F) regular.

Let G have colour classes U and W. First let k be even. Then find an Eulerian orientation of the edges of G (this can be done in $O(m)$ time (Theorem 6.7)). Let F be the set of edges oriented from U to W.

Next let k be odd. Call a subset F of E *almost regular* if $|\deg_F(v)-\deg_F(u)| \leq 1$ for all $u, v \in V$. (Here $\deg_F(v)$ is the degree of v in the graph (V, F).)

Moreover, let $\text{odd}(F)$ and $\text{even}(F)$ denote the sets of vertices v with $\deg_F(v)$ odd and even, respectively, and let $\Delta(F)$ denote the maximum degree of the graph (V, F). We give an $O(m)$ algorithm for the following problem:

(16.10) given: an almost regular subset F of E with $\Delta(F) \geq 2$,
find: an almost regular subset F' of E with $\Delta(F') \geq 2$ and $|\text{odd}(F')| \leq \frac{1}{2}|\text{odd}(F)|$.

In time $O(m)$ we can find a subset F'' of F such that

(16.11) $\lfloor \frac{1}{2} \deg_F(v) \rfloor \leq \deg_{F''}(v) \leq \lceil \frac{1}{2} \deg_F(v) \rceil$

for each vertex v: make an Eulerian orientation in the graph obtained from (V, F) by adding edges so as to make all degrees even, and choose for F'' the subset of all edges oriented from U to W. So F'' and $F \setminus F''$ are almost regular.

We choose F'' such that

(16.12) $|\text{odd}(F'') \cap \text{odd}(F)| \leq \frac{1}{2}|\text{odd}(F)|$

(otherwise replace F'' by $F \setminus F''$). Let $2l$ be the degree of the even-degree vertices of (V, F). We consider two cases.

Case 1: l is even. Define $F' := F''$. By (16.11), F' is almost regular. Moreover, as l is even, $\text{odd}(F') \subseteq \text{odd}(F)$, implying (with (16.12)) that $|\text{odd}(F')| \leq \frac{1}{2}|\text{odd}(F)|$. Finally, $\Delta(F') \geq 2$, since otherwise $\Delta(F) \leq 3$ and hence $l = 0$, implying $\Delta(F) \leq 1$, a contradiction.

Case 2: l is odd. Define $F' := F'' \cup (E \setminus F)$. Then F' is almost regular, since each $\deg_{F'}(v)$ is either $\lfloor \frac{1}{2} \deg_F(v) \rfloor + k - \deg_F(v) = k - \lceil \frac{1}{2} \deg_F(v) \rceil$ or $\lceil \frac{1}{2} \deg_F(v) \rceil + k - \deg_F(v) = k - \lfloor \frac{1}{2} \deg_F(v) \rfloor$.

Since k is odd, one also has (by definition of F'): $\deg_{F'}(v)$ is odd $\iff$ $\deg_{F''}(v) + k - \deg_F(v)$ is odd $\iff$ $\deg_{F''}(v) \equiv \deg_F(v) \pmod 2$ $\iff$ $v \in \text{odd}(F'') \cap \text{odd}(F)$ (since $\text{even}(F) \subseteq \text{odd}(F'')$, as l is odd). So $|\text{odd}(F')| = |\text{odd}(F'') \cap \text{odd}(F)| \leq \frac{1}{2}|\text{odd}(F)|$, by (16.12).

Finally, suppose that $\Delta(F') \leq 1$. Choose $v \in \text{odd}(F) \setminus \text{odd}(F')$. So $v \in \text{even}(F')$, hence $\deg_{F'}(v) = 0$, implying $\deg_{F''}(v) = 0$ and $\deg_F(v) = k$. But then $0 = \lfloor \frac{1}{2} k \rfloor$, and so $k \leq 1$, a contradiction.

This describes the $O(m)$-time algorithm for problem (16.10). It implies that one can find an almost regular subset F of E with $\Delta(F) \geq 2$ and $\text{odd}(F) = \emptyset$ in $O(m \log n)$ time. So (V, F) is a regular subgraph of G, and we have solved (16.9).

This implies an $O(m \log n)$ algorithm for finding a perfect matching: First find a subset F of E as in (16.9). Without loss of generality, $|F| \leq \frac{1}{2}|E|$. Recursively, find a perfect matching in (V, F). The time is bounded by $O((m + \frac{1}{2}m + \frac{1}{4}m + \cdots) \log n) = O(m \log n)$. ∎

In fact, as was shown by Cole, Ost, and Schirra [2001], one can find a perfect matching in a regular bipartite graph in $O(m)$ time. To explain this algorithm, we first describe an algorithm that finds a perfect matching in a k-regular bipartite graph in $O(km)$ time (Schrijver [1999]). So for each fixed degree k one can find a perfect matching in a k-regular graph in linear time, which is also a consequence of an $O(n2^{2^{O(k)}})$-time algorithm of Cole [1982].

Theorem 16.10. *A perfect matching in a k-regular bipartite graph can be found in time $O(km)$.*

Proof. Let $G = (V, E)$ be a k-regular bipartite graph. For any function $w : E \to \mathbb{Z}_+$, define $E_w := \{e \in E \mid w_e > 0\}$.

Initially, set $w_e := 1$ for each $e \in E$. Next apply the following iteratively:

(16.13) Find a circuit C in E_w. Let $C = M \cup N$ for matchings M and N with $w(M) \geq w(N)$. Reset $w := w + \chi^M - \chi^N$.

Note that at any iteration, the equation $w(\delta(v)) = k$ is maintained for all v.

To see that the process terminates, note that at any iteration the sum

$$(16.14) \qquad \sum_{e \in E} w_e^2$$

increases by

$$(16.15) \qquad \sum_{e \in M}((w_e+1)^2 - w_e^2) + \sum_{e \in N}((w_e-1)^2 - w_e^2) = 2w(M) + |M| - 2w(N) + |N|,$$

which is at least $|M| + |N| = |C|$. Since $w_e \leq k$ for each $e \in E$, (16.14) is bounded, and hence the process terminates. We now estimate the running time.

At termination, we have that the set E_w contains no circuit, and hence is a perfect matching (since $w(\delta(v)) = k$ for each vertex v). So at termination, the sum (16.14) is equal to $\frac{1}{2}nk^2 = km$.

Now we can find a circuit C in E_w in $O(|C|)$ time on average. Indeed, keep a path P in E_w such that $w_e < k$ for each e in P. Let v be the last vertex of P. Then there is an edge $e = vu$ not occurring in P, with $0 < w_e < k$. Reset $P := P \cup \{e\}$. If P is not a path, it contains a circuit C, and we can apply (16.13) to C, after which we reset $P := P \setminus C$. We continue with P.

Concluding, as each step increases the sum (16.14) by at least $|C|$, and takes $O(|C|)$ time on average, the algorithm terminates in $O(km)$ time. ∎

The bound given in this theorem was improved to linear time *independent of* the degree, by Cole, Ost, and Schirra [2001]. Their method forms a sharpening of the method described in the proof of Theorem 16.10, utilizing the fact that when breaking a circuit, the path segments left ('chains') can be used in the further path search to extend the path by chains, rather than just edge by edge. To this end, these chains need to be supplied with some extra data structure, the 'self-adjusting binary trees', in order to avoid that we have to run through the chain to find an end of the chain where it can be attached to the path. The basic operation is the 'splay'.

The main technique of Cole, Ost, and Schirra's theorem is contained in the proof of the following theorem. For any graph $G = (V, E)$ call a ('weight') function $w : E \to \mathbb{R}$ *k-regular* if $w(\delta(v)) = k$ for all $v \in V$.

Theorem 16.11. *Given a bipartite graph $G = (V, E)$ and a k-regular $w : E \to \mathbb{Z}_+$, for some $k \geq 2$, a perfect matching in G can be found in time $O(m \log^2 k)$.*

Proof. I. *Conceptual outline.* We first give a conceptual description, as extension of the algorithm described in the previous proof. First delete all edges e with $w_e = 0$.

We keep a set F of edges such that each component of (V, F) is a path (possibly a singleton) with at most k^2 vertices, and we keep a path

(16.16) $Q = (P_0, e_1, P_1, \ldots, e_t, P_t),$

where each P_j is a (path) component of (V, F). Let v be the last vertex of Q and let $e = vu$ be an edge in $E \setminus F$ incident with v with $w_e < k$. Let P be the component of (V, F) containing u.

If u is not on Q, let R be a longest segment of P starting from u. Delete the first edge of the other segment of P (if any) from F. If $|P_t| + |R| \leq k^2$, add e to F, and reset P_t to P_t, e, R. (Here and below, $|X|$ denotes the number of vertices of a path X.) Otherwise, extend Q by e, R.

If u is on Q, then:

(16.17) split Q into a part Q_1 from the beginning to u, and a part Q_2 from u to the end;
split the circuit Q_2, e into two matchings M and N, such that $w(M) \geq w(N)$;
let α be the minimum of the weights in N;
reset $w := w + \alpha(\chi^M - \chi^N)$;
delete the edges g with $w(g) = 0$ or $w(g) = k$ (in the latter case, also delete the two ends of g);
delete the first edge of Q_2 from F if it was in F;
reset $Q := Q_1$;
iterate.

If v is incident with no edge $e \in E \setminus F$ satisfying $w_e < k$, start Q in a new vertex that is incident with an edge e with $w_e < k$. If no such vertex exists, we are done: the edges left form a perfect matching.

II. *Data structure.* In order to make profit of storing paths, we need additional data structure (based on 'self-adjusting binary trees', analyzed by Sleator and Tarjan [1983b,1985], cf. Tarjan [1983]).

We keep a collection $\mathcal{P}$ of paths (possibly singletons), each being a subpath of a component of F, such that

(16.18)
(i) each component of F itself is a path in $\mathcal{P}$;
(ii) $\mathcal{P}$ is *laminar*, that is, any two paths in $\mathcal{P}$ are vertex disjoint, or one is a subpath of the other;
(iii) any nonsingleton path $P \in \mathcal{P}$ has an edge e_P such that the two components of $P - e_P$ again belong to $\mathcal{P}$.

With any path $P \in \mathcal{P}$ we keep the following information:

(16.19)
(i) the number $|P|$ of vertices in P;
(ii) a list ends(P) of the ends of P (so ends(P) contains one or two vertices);

(iii) if P is not a singleton, the edge e_P, and a list subpaths(P) of the two components of $P - e_P$;
(iv) the smallest path parent(P) in $\mathcal{P}$ that properly contains P (null if there is no such path).

Then for each edge $e \in F$ there is a unique path $P_e \in \mathcal{P}$ traversing e such that both components of $P_e - e$ again belong to $\mathcal{P}$ (that is, $e_{P_e} = e$). We keep with any $e \in F$ the path P_e.

Call a path $P \in \mathcal{P}$ a *root* if parent$(P) =$ null. So the roots correspond to the components of the graph (V, F). Along a path $P \in \mathcal{P}$ we call edges alternatingly *odd* and *even in* P in such a way that e_P is odd.

We also store information on the current values of the w_e. Algorithmically, we only reset explicitly those w_e for which e is not in F. For $e \in F$, these values are stored implicitly, such that it takes only $O(1)$ time to update w_e for all e in a root when adding α to the odd edges and $-\alpha$ to the even edges in it. This can be done as follows.

If P is a root, we store $w(e_P)$ at P. If P has a parent Q, we store

(16.20) $$w(e_P) \pm w(e_Q)$$

at P, where $\pm$ is $-$ if e_P is odd in Q, and $+$ otherwise.

We also need the following values for any $P \in \mathcal{P}$ with $EP \neq \emptyset$:

(16.21) $$\text{minodd}(P) := \min\{w_e \mid e \text{ odd in } P\},\ \text{mineven}(P) := \min\{w_e \mid e \text{ even in } P\},$$
$$\text{diffsum}(P) := \sum(w_e \mid e \text{ odd in } P) - \sum(w_e \mid e \text{ even in } P)$$

(taking a minimum ∞ if the range is empty). When storing these data, we relate them to $w(e_P)$, again so as to make them invariant under updates. Thus we store

(16.22) $$\text{diffsum}(P) - |EP|w(e_P), \text{minodd}(P) - w(e_P), \text{mineven}(P) + w(e_P)$$

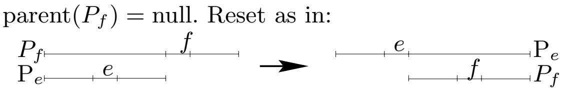
at P. So for any root P we have diffsum(P), minodd(P), and mineven(P) ready at hand, as we know $w(e_P)$.

III. *The splay.* We now describe *splaying* an edge $e \in F$. It changes the data structure so that P_e becomes a root, keeping F invariant. It modifies the tree associated with the laminar family through three generations at a time, so as to attain efficiency on average. (The adjustments make future searches more efficient.)

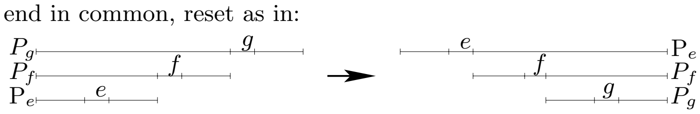
The splay is as follows. While parent$(P_e) \neq$ null, do the following:

(16.23) Let $P_f :=$ parent(P_e).

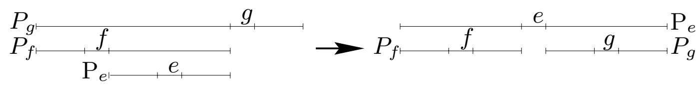
Case 1: parent$(P_f) =$ null. Reset as in:



Case 2: parent$(P_f) \neq$ null. Let $P_g :=$ parent(P_f). If P_e and P_g have an end in common, reset as in:



If P_e and P_g have no end in common, reset as in:



Note that Case 1 applies only in the last iteration of the while loop. It is straightforward to check that the data associated with the paths can be restored in $O(1)$ time at any iteration.

IV. *Running time of one splay.* To estimate the running time of a splay, define:

(16.24) $$\gamma := \sum_{P \in \mathcal{P}} \log |P|,$$

taking logarithms with base 2 (again, $|P|$ denotes the number of vertices of P).

For any splay of e one has (adding $'$ to parameters after the splay):

(16.25) the number of iterations of (16.23) is at most $\gamma - \gamma' + 3(\log |P'_e| - \log |P_e|) + 1$.

To show this, consider any iteration (16.23) (adding $'$ to parameters after the iteration).

If Case 1 applies, then

(16.26) $$\begin{aligned} &\gamma - \gamma' + 3(\log |P'_e| - \log |P_e|) + 1 \\ &= \log |P_e| + \log |P_f| - \log |P'_e| - \log |P'_f| + 3 \log |P'_e| - 3 \log |P_e| + 1 \\ &= 3 \log |P_f| - \log |P'_f| - 2 \log |P_e| + 1 \geq 1, \end{aligned}$$

since $P'_e = P_f$ and since P'_f and P_e are subpaths of P_f. If Case 2 applies, then

(16.27) $$\begin{aligned} &\gamma - \gamma' + 3(\log |P'_e| - \log |P_e|) = \log |P_e| + \log |P_f| + \log |P_g| \\ &- \log |P'_e| - \log |P'_f| - \log |P'_g| + 3 \log |P'_e| - 3 \log |P_e| \\ &= 3 \log |P_g| + \log |P_f| - \log |P'_f| - \log |P'_g| - 2 \log |P_e| \geq 1. \end{aligned}$$

The last equality follows from $P'_e = P_g$. The last inequality holds since P_e is a subpath of P_f, and P'_f, P'_g, and P_e are subpaths of P_g, and since, if the first alternative in Case 2 holds, then P_e and P'_g are vertex-disjoint (implying $2 \log |P_g| \geq \log |P_e| + \log |P'_g| + 1$), and, if the second alternative in Case 2 holds, then P'_f and P'_g are vertex-disjoint (implying $2 \log |P_g| \geq \log |P'_f| + \log |P'_g| + 1$).

(16.26) and (16.27) imply (16.25).

V. *The algorithm.* Now we use the splay to perform the conceptual operations described in the conceptual outline (proof section I above). Thus, let v be the last vertex of the current path Q (cf. (16.16)) and let $e = vu$ be an edge in $E \setminus F$ incident with u. Determine the root $P \in \mathcal{P}$ containing u (possibly by splaying an edge in F incident with u).

Case A: P is not on Q. (We keep a pointer to indicate if a root belongs to Q.) Find a root R as follows. If u is incident with no edge in F, then $R := \{u\}$. If u is incident with exactly one edge $f \in F$, splay f and let $R := P_f$. If u is incident with two edges in F, by splaying find $f \in F$ incident with u such that (after splaying f) subpaths$(P_f) = \{R, R'\}$ where $u \in$ ends(R) and $|R| > |R'|$; then delete P_f from $\mathcal{P}$, and f from F.

This determines R. If $|P_t| + |R| \leq k^2$, add e to F, let P_e be the join of P_t, e, and R, and reset P_t in Q to P_e. If $|P_t| + |R| > k^2$, extend Q by $e, P_{t+1} := R$.

Case B: P is on Q, say $P = P_j$. By (possibly) splaying, we can decide if u is at the end of P_j or not. In the former case, reset $Q := P_0, e_1, P_1, \ldots, e_j, P_j$ and let $C := e_{j+1}, P_{j+1}, \ldots, P_t, e$. In the latter case, split P_j to P'_j, f, P''_j in such a way that

$Q := P_0, e_1, P_1, \ldots, e_j, P_j'$ is the initial segment of the original Q ending at u, and let $C := f, P_j'', e_{j+1}, P_{j+1}, \ldots, P_t, e$.

Determine the difference of the sum of the w_e over the odd edges in C and that over the even edges in C. As we know diffsum(S) for any root S, this can be done in time $O(t - j + 1)$. Depending on whether this difference is positive or not, we know (implicitly) which splitting of the edges on C into matchings M and N gives $w(M) \geq w(N)$. From the values of minodd and mineven for the paths $P \in \mathcal{P}$ on C and from the values of w_e for the edges $e_{j+1}, \ldots, e_t$, e on C (and possibly f), we can find the maximum decrease α on the edges in N, and reset the parameters.

Next, for any $P \in \mathcal{P}$ on C with minodd$(P) = 0$ or mineven$(P) = 0$, determine the edges on P of weight 0, delete them after splaying, and decompose P accordingly. Delete any edge e_i on C with $w(e_i) = 0$ (similarly f).

This describes the iteration.

VI. *Running time of the algorithm.* We finally estimate the running time. In any iteration, let γ be the number of roots of $\mathcal{P}$ that are not on Q. Initially, $\gamma \leq n$. During the algorithm, γ only increases when we are in Case B and break a circuit C, in which case γ increases by at most

$$(16.28) \qquad 2\frac{L_C}{k^2} + m_C + 2,$$

where L_C is the length of C in G (that is, the number of edges e_i plus the sum of the lengths of the paths P_i in C), and where m_C is the number of edges of weight 0 deleted at the end of the iteration. Bound (16.28) uses the fact that the sizes of any two consecutive paths along C sum up to more than k^2, except possibly at the beginning and the end of the circuit, and that any edge of weight 0 can split a root into two new roots.

Now if we sum bound (16.28) over all circuits C throughout the iterations, we have

$$(16.29) \qquad \sum_C (2\frac{L_C}{k^2} + m_C + 2) = O(m),$$

since $\sum_C L_C \leq nk^2$, like in the proof of the previous theorem (note that $m_C \geq 1$ for each C, so the term 2 is absorbed by m_C). So the number of roots created throughout the Case B iterations is $O(m)$. Now at each Case A iteration, we split off a part of a root of size less than half the size of the root; the split off part can be used again by Q some time in later iterations. Hence any root can be split at most $\log k^2$ times, and therefore, the number of Case A iterations is $O(m \log k)$. In particular, the number of times we join two paths in $\mathcal{P}$ and make a new path is $O(m \log k)$.

Next consider γ as defined in (16.24). Note that at any iteration except for joins and splays, γ does not increase. At any join, γ increases by at most $\log k^2$, and hence the total increase of γ during joins is $O(m \log^2 k)$.

Now the number of splays during any Case A iteration is $O(1)$, and during any Case B iteration $O(L_C/k^2 + m_C + 1)$. Hence by (16.29), the total number of splays is $O(m \log k)$. By (16.25), each splay takes time $O(\delta + \log k)$, where δ is the decrease of γ (possibly $\delta < 0$). The sum of δ over all splays is $O(m \log^2 k)$, as this is the total increase of γ during joins. So all splays take time $O(m \log^2 k)$. As the number of splits is proportional to the number of splays, and each takes $O(1)$ time, we have the overall time bound of $O(m \log^2 k)$. ∎

This implies a linear-time perfect matching algorithm for regular bipartite graphs:

Corollary 16.11a. *A perfect matching in a regular bipartite graph can be found in linear time.*

Proof. Let $G = (V, E)$ be a k-regular bipartite graph. We keep a weight function $w : E \to \mathbb{Z}_+$, with the property that $w(\delta(v)) = k$ for each $v \in V$. Throughout the algorithm, let G_i be the subgraph of G consisting of those edges e of G with $w_e = 2^i$ (for $i = 1, \ldots$).

Initially, define a weight $w_e := 1$ for each edge e. For $i = 0, 1, \ldots, \lfloor \log_2 k \rfloor$ do the following. Perform a depth-first search in G_i. If we meet a circuit C in G_i, then split C arbitrarily into matchings M and N, reset $w := w + 2^i(\chi^M - \chi^N)$, delete the edges in N, and update G_i (that is, delete the edges of C from G_i).

As G_i has at most $m/2^i$ edges (since $w(E) = \frac{1}{2}kn = m$), and as depth-first search can be done in time linear in the number of edges, this can be done in $O(m + \frac{1}{2}m + \frac{1}{4}m + \cdots) = O(m)$ time.

For the final G and w, all weights are a power of 2 and each graph G_i has no circuits, and hence has at most $|V| - 1$ edges. So G has at most $|V| \log_2 k$ edges. As w is k-regular, by Theorem 16.11 we can find a perfect matching in G in time $O(|V| \log^3 k)$, which is linear in the number of edges of the original graph G. ∎

This result will be used in obtaining a fast edge-colouring algorithm for bipartite graphs (Section 20.9a).

Notes. Alon [2000] gave the following easy $O(m \log m)$-time method for finding a perfect matching in a regular bipartite graph $G = (V, E)$. Let k be the degree, and choose t with $2^t \geq kn$. Let $\alpha := \lfloor 2^t/k \rfloor$ and $\beta := 2^t - k\alpha$. So $\beta < k$. Let H be the graph obtained from G by replacing each edge by α parallel edges, and by adding a β-regular set F of (new) edges, consisting of $\frac{1}{2}n$ disjoint classes, each consisting of β parallel edges. So H is 2^t-regular.

Iteratively, split H into two regular graphs of equal degree (by determining an Eulerian orientation), and reset H to the graph that has a least number of edges in F.

As $|F| = \frac{1}{2}\beta n < 2^t$, after $\log_2 |F| < t$ iterations, H contains no edge in F. Hence after t iterations we have a perfect matching in H not intersecting F; that is, we have a perfect matching in G.

This gives an $O(m \log m)$-time method, provided that we do not display the graph H fully, but handle the parallel edges implicitly (by the sizes as a function of the underlying edges).

Note that $O(m \log m) = O(nk(\log k + \log n))$. An $O(nk + n \log n \log k)$-time algorithm finding a perfect matching in a k-regular bipartite graph was given by Rizzi [2002].

(Csima and Lovász [1992] described a space-efficient $O(n^2 k \log k)$-time algorithm for finding a perfect matching in a k-regular bipartite graph.)

16.7c. The equivalence of Menger's theorem and Kőnig's theorem

We have seen that Kőnig's matching theorem can be derived from Menger's theorem (by the construction given in the proof of Theorem 16.4) — in fact it forms the induction basis in Menger's proof. The interrelation however is even stronger, as was noticed by Hoffman [1960] (cf. Orden [1955], Ford and Fulkerson [1958c], Hoffman and Markowitz [1963], Ingleton and Piff [1973]): in turn Menger's theorem (in the form of Theorem 9.1) can be derived from Kőnig's matching theorem by a direct (noninductive) construction.

Let $D = (V, A)$ be a directed graph and let $S, T \subseteq V$. We may assume that $S \cap T = \emptyset$. For each $v \in V \setminus S$ introduce a vertex v' and for each $v \in V \setminus T$ introduce a vertex v''. Let E be the set of pairs $\{u', v''\}$ with $u \in V \setminus S$ and $v \in V \setminus T$ with the property that $(u, v) \in A$ or $u = v$. This makes the bipartite graph G, containing the matching

(16.30) $$M := \{\{v', v''\} \mid v \in V \setminus (S \cup T)\}.$$

For any $X \subseteq V$, let $X' := \{v' \mid v \in X\}$ and $X'' := \{v'' \mid v \in X\}$.

Now let M' be a matching in G of size $\nu(G)$. For each component of $M \triangle M'$ having more than one vertex, we may assume that it is an M-augmenting path (since any other component K has an equal number of edges in M and in M', and hence we can replace M' by $M' \triangle K$). Each M-augmenting path is an $S'' - T'$ path. Hence there exist $|M'| - |M| = \nu(G) - |V \setminus (S \cup T)|$ vertex-disjoint $S - T$ paths.

Let $U \subseteq V \setminus T$ and $W \subseteq V \setminus S$ be such that $D := U'' \cup W'$ is a vertex cover of G, with $|U| + |W| = \tau(G)$. Then

(16.31) $$C := (U \cap S) \cup (U \cap W) \cup (W \cap T)$$

intersects each $S - T$ path in D. Indeed, suppose $P = (v_0, v_1, \ldots, v_k)$ is an $S - T$ path not intersecting C. We may assume that P intersects S and T only at v_0 and v_k, respectively. Now

(16.32) $$Q := (v_0'', v_1', v_1'', \ldots, v_{k-1}', v_{k-1}'', v_k')$$

is a path in G of odd length $2k - 1$. Hence D intersects Q in at least k vertices. Therefore, $v_0'' \in D$ (hence $v_0 \in U \cap S \subseteq C$), or $v_k' \in D$ (hence $v_k \in W \cap T \subseteq C$), or $v_i', v_i'' \in D$ for some $i \in \{1, \ldots, k-1\}$ (hence $v_i \in U \cap W \subseteq C$). So C intersects each $S - T$ path in D.

As

(16.33) $$\begin{aligned}|C| &= |U \cap S| + |U \cap W| + |W \cap T| = |U \cap S| + |U| + |W| - |U \cup W| + |W \cap T| \\ &= |U| + |W| - |V \setminus (S \cup T)|\end{aligned}$$

(since $(U \cup W) \setminus (S \cup T) = V \setminus (S \cup T)$), and as $|U| + |W| = \tau(G) = \nu(G)$, we have that the size of C is at most the number of disjoint $S - T$ paths found above.

The converse construction (described by Kuhn [1956]) also applies. Let be given a bipartite graph $G = (V, E)$, with colour classes U and W, and a matching M in G. Orient each edge from U to W, and next contract all edges in M. This gives a directed graph $D = (V', A)$. Let S and T be the sets of vertices in U and W missed by M. Then the maximum number of vertex-disjoint $S - T$ paths in D is equal to $\nu(G) - |M|$.

These constructions also imply:

Theorem 16.12. *For any function $\phi(n, m)$ one has: the bipartite matching problem with n vertices and m edges is solvable in time $O(\phi(n, m))$ $\iff$ the disjoint $s - t$ paths problem with n vertices and m arcs is solvable in time $O(\phi(n, m))$.*

Proof. See above. ∎

16.7d. Equivalent formulations in terms of matrices

Frobenius [1917] proved his theorem (Corollary 16.2a) in terms of matrices, in the following form:

(16.34) Each diagonal of an $n \times n$ matrix has product 0 if and only if M has a $k \times l$ all-zero submatrix with $k + l > n$.

Similarly, Kőnig's matching can be formulated in matrix terms as follows:

(16.35) In a matrix, the maximum number of nonzero entries with no two in the same line (=row or column) is equal to the minimum number of lines that include all nonzero entries.

An equivalent form of Kőnig's theorem on the existence of a perfect matching in a regular bipartite graph (Corollary 16.2b) is:

(16.36) If in a nonnegative matrix each row and each column has the same positive sum, then it has a diagonal with positive entries.

16.7e. Equivalent formulations in terms of partitions

Bipartite graphs can be studied also as unions of two partitions of a given set. Indeed, let $G = (V, E)$ be a bipartite graph. Then the family $(\delta(v) \mid v \in V)$ is a union of two partitions of E. Since each union of two partitions arises in this way, we can formulate theorems on bipartite graphs equivalently as theorems on unions of two partitions of a set.

The following equivalent form of Frobenius' theorem (Corollary 16.2a) was given by Maak [1936]:

(16.37) Let $\mathcal{A}$ and $\mathcal{B}$ be two partitions of the finite set X. Then there is a subset Y of X intersecting each set in $\mathcal{A} \cup \mathcal{B}$ in exactly one element if and only if for each natural number k, the union of any collection of k classes of $\mathcal{A}$ intersects at least k classes of $\mathcal{B}$.

This implies the following equivalent form of Corollary 16.2b, given by van der Waerden [1927] (with short proof by Sperner [1927] — see Section 22.7d):

(16.38) Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be two partitions of a finite set X with $|A_1| = \cdots = |A_n| = |B_1| = \cdots = |B_n|$. Then there is a subset Y of X intersecting each A_i and each B_i in exactly one element.

Some of the matching results can be formulated in terms of (common) transversals. We will discuss this more extensively in Chapters 22 and 23.

16.7f. On the complexity of bipartite matching and vertex cover

In a *bipartite* graph we can derive a minimum-size vertex cover from a maximum-size matching in linear time (for general graphs this would imply NP=P) — see Theorems 16.6.

So knowing a maximum-size matching in a bipartite graph gives us a minimum-size vertex cover in linear time. The reverse, however, is unlikely, unless there would exist an algorithm to find a perfect matching in a bipartite graph in linear time. To see this, suppose that there is an algorithm $\mathcal{A}$ to derive from a minimum-size vertex cover a maximum-size matching in linear time. Now let $G = (V, E)$ be a bipartite graph in which we want to find a perfect matching. Then we may *assume* that G has a perfect matching. So we may assume by Frobenius' theorem that the colour classes U and W are minimum-size vertex covers. Then apply $\mathcal{A}$ to G and U. Then either we obtain a perfect matching if U indeed is a minimum-size vertex cover, or else (if our assumption is wrong) the algorithm gets stuck, in which case we may conclude that G has no perfect matching.

16.7g. Further notes

Extensions of Frobenius' and Kőnig's theorems to the infinite case were considered by Kőnig and Valkó [1925], Shmushkovich [1939], de Bruijn [1943], Rado [1949b], Brualdi [1971f], Aharoni [1983b,1984b], and Aharoni, Magidor, and Shore [1992].

Itai, Rodeh, and Tanimoto [1978] showed that, given a bipartite graph $G = (V, E)$, $F \subseteq E$, and $k \in \mathbb{Z}_+$, one can find a perfect matching M with $|M \cap F| \leq k$ (or decide that no such perfect matching exists) in time $O(nm)$. (This amounts to a minimum-cost flow problem.)

Karp, Vazirani, and Vazirani [1990] gave an optimal on-line bipartite matching algorithm. Motwani [1989,1994] investigated the expected running time of matching algorithms.

The following question was posed by A. Frank: Given a bipartite graph $G = (V, E)$ whose edges are coloured red and blue, and given k and l; when does there exist a matching containing k red edges and l blue edges? This problem is NP-complete, but for complete bipartite graphs it was characterized by Karzanov [1987c].

An extension of Frobenius' theorem to more general matrices than described in Section 16.2b was given by Hartfiel and Loewy [1984].

Dulmage and Mendelsohn [1958] study minimum-size vertex covers in a bipartite graph as a lattice. For maintaining perfect matchings 'in the presence of failure', see Sha and Steiglitz [1993]. Lovász [1970a] gave a generalization of Kőnig's matching theorem — see Section 60.1a. Uniqueness of a maximum-size matching in a bipartite graph was investigated by Cechlárová [1991], and related work was reported by Costa [1994]. A variant of Kőnig's matching theorem was given by de Werra [1984].

For surveys on matching algorithms, see Galil [1983,1986a,1986b]. For surveys on bipartite matching, see Woodall [1978a,1978b]. Books discussing bipartite matching include Ford and Fulkerson [1962], Ore [1962], Dantzig [1963], Christofides [1975], Lawler [1976b], Even [1979], Papadimitriou and Steiglitz [1982], Tarjan [1983], Tutte [1984], Halin [1989], Cook, Cunningham, Pulleyblank, and Schrijver

[1998], Jungnickel [1999], Mehlhorn and Näher [1999], and Korte and Vygen [2000].

16.7h. Historical notes on bipartite matching

The fundaments of matching theory in bipartite graphs were laid by Frobenius (in terms of matrices and determinants) and Kőnig. In his article *Über Matrizen aus nicht negativen Elementen* (On matrices with nonnegative elements), Frobenius [1912] investigated the decomposition of matrices:

> In §11 dehne ich die Untersuchung auf zerlegbare Matrizen aus, und in §12 zeige ich, daß eine solche nur auf eine Art in unzerlegbare Teile zerfällt werden kann. Dabei ergibt sich der merkwürdige Determinantensatz:
> I. *Die Elemente einer Determinante nten Grades seien n^2 unabhängige Veränderliche. Man setze einige derselben Null, doch so, daß die Determinante nicht identisch verschwindet. Dann bleibt sie eine irreduzible Funktion, außer wenn für einen Wert $m < n$ alle Elemente verschwinden, die m Zeilen mit $n - m$ Spalten gemeinsam haben.*[4]

Frobenius gave a combinatorial and an algebraic proof.

In a reaction to Frobenius' paper, Kőnig [1915] ('presented to Class III of the Hungarian Academy of Sciences on 16 November 1914') next gave a proof of Frobenius' result with the help of graph theory:

> A graphok alkalmazásával e tételnek egyszerű és szemléletes új bizonyitását adjuk a következőkben.[5]

He introduced a now quite standard construction of making a bipartite graph from a matrix $(a_{i,j})$: for each row index i there is a vertex A_i and for each column index j there is a vertex B_j; then vertices A_i and B_j are connected by an edge if and only if $a_{i,j} \neq 0$.

Kőnig was interested in graphs because of his interest in set theory, especially cardinal numbers (cf. footnotes in Kőnig [1916]). In proving Schröder-Bernstein type results on the equivalence of sets, graph-theoretic arguments (in particular: matchings) can be illustrative. This led Kőnig to studying graphs (in particular bipartite graphs) and its applications in other areas of mathematics.

Kőnig's work on matchings in regular bipartite graphs

Earlier, on 7 April 1914, Kőnig had presented the following theorem at the *Congrès de Philosophie mathématique* in Paris (cf. Kőnig [1923]):

> *A. Chaque graphe régulier à circuits pairs possède un facteur du premier degré.*[6]

[4] In §11, I extend the investigation to decomposable matrices, and in §12, I show that such a matrix can be decomposed in only one way into indecomposable parts. With that, the [following] curious determinant theorem comes up:
I. *Let the elements of a determinant of degree n be n^2 independent variables. One sets some of them equal to zero, but such that the determinant does not vanish identically. Then it remains an irreducible function, except when for some value $m < n$ all elements vanish that have m rows in common with $n - m$ columns.*

[5] In the following we will give a simple and clear new proof by applying graphs to this theorem.

[6] *A. Each regular graph with even circuits has a factor of the first degree.*

That is, every regular bipartite graph has a perfect matching (= factor of degree 1). As a corollary, Kőnig derived:

> *B. Chaque graphe régulier à circuits pairs est le produit de facteurs du premier degré; le nombre de ces facteurs est égal au degré du graphe.*[7]

That is, each k-regular bipartite graph is k-edge-colourable (cf. Chapter 20). Kőnig did not give a proof of the theorem in the Paris paper, but expressed the hope to give a complete proof 'at another occasion'.

This occasion came in Kőnig [1916] ('presented to Class III of the Hungarian Academy of Sciences on 15 November 1915') where next to the above mentioned Theorems A and B, Kőnig gave the following result:

> C) *Ha egy páros körüljárású graph bármelyik csúcsába legfeljebb k-számú él fut, akkor minden éléhez oly módon lehet k-számú index valamelyikét hozzárendelni, hogy ugyanabba a csúcsba futó két élhez mindenkor két különböző index legyen rendelve.*[8]

In other words, the edge-colouring number of a bipartite graph is equal to its maximum degree. Kőnig gave a proof of result C), and derived A and B. (See the proof of Theorem 20.1 below of Kőnig's proof.)

In §2 of Kőnig [1916], applications of his results to matrices and determinants are studied. First:

> D) *Ha egy nem negativ [egész számú] elemekből álló determináns minden sora és minden oszlopa ugyanazt a positiv összeget adja, akkor van a determinánsnak legalább egy el nem tünő tagja.*[9]

Next:

> E) *Ha egy determináns minden sorában és oszlopában pontosan k-számú el nem tünő elem van, akkor legalább k-számú determinánstag nem tünik el.*[10]

Third:

> F) *Ha egy n^2 mezejű quadratikus táblán kn-számú figura úgy van elhelyezve (ugyanazon a mezőn több figura is lehet), hogy minden sorban és oszlopban pontosan k-számú figura fordul elő, akkor e konfigurráczió mindig mint k-számú ugyancsak n^2 mezejű oly konfigurráczió egyesítése keletkeztethető, melyek mindegyikében egy-egy figura van minden sorban és minden oszlopban.*[11]

---

[7] *B. Each regular graph with even circuits is the product of factors of the first degree; the number of these factors is equal to the degree of the graph.*

[8] C) *If in each vertex of an even circuit graph at most k edges meet, then one can assign to each of the edges of the graph one from k indices in such a way that two edges that meet in a point always obtain different indices.*

[9] D) *If in a determinant of nonnegative [integer] numbers each row and each column yield the same positive sum, then at least one member of the determinant is different from zero.*

[10] E) *If the number of nonvanishing elements in each row and column of a determinant is exactly equal to k, then there are at least k nonvanishing determinant members.*

[11] F) *If kn pieces are placed on a quadratic board with n^2 fields (where several pieces may stand in the same field), such that each row and each column contains exactly k pieces, then this configuration always arises by joining k such configurations with also n^2 fields, in which each row and each column contains exactly one piece.*

Frobenius' theorem

Chronologically next is a paper of Frobenius [1917]. In order to give an elementary proof of his result in Frobenius [1912] quoted above, he proved the following 'Hilfssatz':

> II. *Wenn in einer Determinante nten Grades alle Elemente verschwinden, welche p $(\leq n)$ Zeilen mit $n-p+1$ Spalten gemeinsam haben, so verschwinden alle Glieder der entwickelten Determinante.*
> *Wenn alle Glieder einer Determinante nten Grades verschwinden, so verschwinden alle Elemente, welche p Zeilen mit $n-p+1$ Spalten gemeinsam haben für $p=1$ oder $2, \cdots$ oder n.*[12]

That is, if $A = (a_{i,j})$ is an $n \times n$ matrix, and if $\prod_{i=1}^{n} a_{i,j} = 0$ for each permutation π of $\{1, \ldots, n\}$, then for some p there exist p rows and $n-p+1$ columns of A such that each element that is both in one of these rows and in one of these columns, is equal to 0.

In other words, a bipartite graph $G = (V, E)$ with colour classes V_1 and V_2 satisfying $|V_1| = |V_2| = n$ has a perfect matching if and only if one cannot select p vertices in V_1 and $n-p+1$ vertices in V_2 such that no edge is connecting two of these vertices.

Frobenius noticed with respect to Kőnig's work:

> Aus dem Satze II ergibt sich auch leicht ein Ergebnis der Hrn. DÉNIS KŐNIG, *Uber Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre, Math. Ann. Bd. 77.*
> *Wenn in einer Determinante aus nicht negativen Elementen die Größen jeder Zeile und jeder Spalte dieselbe, von Null verschiedene Summe haben, so können ihre Glieder nicht sämtlich verschwinden.*[13]

Frobenius gave a short combinatorial proof of his theorem — see Section 16.2a. His proof is in terms of determinants, and he offered his opinion on graph-theoretic methods:

> Die Theorie der Graphen, mittels deren Hr. KŐNIG den obigen Satz abgeleitet hat, ist nach meiner Ansicht ein wenig geeignetes Hilfsmittel für die Entwicklung der Determinantentheorie. In diesem Falle führt sie zu einem ganz speziellen Satze von geringem Werte. Was von seinem Inhalt Wert hat, ist in dem Satze II ausgesprochen.[14]

(See Schneider [1977] for some comments.)

---

[12] II. *If in a determinant of the nth degree all elements vanish that p $(\leq n)$ rows have in common with $n-p+1$ columns, then all members of the expanded determinant vanish.*
If all members of a determinant of degree n vanish, then all elements vanish that p rows have in common with $n-p+1$ columns for $p=1$ or $2, \cdots$ or n.

[13] From Theorem II, a result of Mr DÉNIS KŐNIG, *Uber Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre, Math. Ann. Vol. 77* follows also easily.
If in a determinant of nonnegative elements the quantities of each row and of each column have the same nonzero sum, then its members cannot vanish altogether.

[14] The theory of graphs, by which Mr KŐNIG has derived the theorem above, is to my opinion of little appropriate help for the development of determinant theory. In this case it leads to a very special theorem of little value. What from its contents has value, is enunciated in Theorem II.

Equivalent formulations in terms of partitions

In October 1926, van der Waerden [1927] presented the following theorem at the *Mathematisches Seminar* in Hamburg:

> *Es seien zwei Klasseneinteilungen einer endlichen Menge $\mathcal{M}$ gegeben. Die eine soll die Menge in μ zueinander fremde Klassen $\mathcal{A}_1, \ldots, \mathcal{A}_\mu$ zu je n Elementen zerlegen, die andere ebenfalls in μ fremde Klassen $\mathcal{B}_1, \ldots, \mathcal{B}_\mu$ zu je n Elementen. Dann gibt es ein System von Elementen $x_1, \ldots, x_\mu$, derart, daß jede A-Klasse und ebenso jede B-Klasse under den x_i durch ein Element vertreten wird.*[15]

The proof of van der Waerden is based on an augmenting path argument. Moreover, van der Waerden remarked that E. Artin has communicated orally to him that the result can be sharpened to the existence of n disjoint such common transversals.

In the article, the following note is added in proof:

> **Zusatz bei der Korrektur.** Ich bemerke jetzt, daß der hier bewiesene Satz mit einem Satz von Dénes Kőnig über reguläre Graphen äquivalent ist.[16]

The article of van der Waerden is followed by an article of Sperner [1927] (presented at the *Mathematisches Seminar* in Januari 1927), which gives a 'simple proof' of van der Waerden's result — we quote the full paper in Section 22.7d.

Kőnig's matching theorem

At the meeting of 26 March 1931 of the Eötvös Loránd Matematikai és Fizikai Társulat (Loránd Eötvös Mathematical and Physical Society) in Budapest, Kőnig [1931] presented a new result that formed the basis for Menger's theorem:

> Páros körüljárású graphban az éleket kimerítő szögpontok minimális száma megegyezik a páronként közös végpontot nem tartalmazó élek maximális számával.[17]

In other words, the maximum size of a matching in a bipartite graph is equal to the minimum number of vertices needed to cover all edges. As we discussed in Section 9.6e, Kőnig's proof formed the missing basis for Menger's theorem. Kőnig also referred to the work of Frobenius (but did not notice that his theorem can be derived from Frobenius' theorem).

The proof of Kőnig [1931] is based on an augmenting path argument. A German version of it was published in Kőnig [1932] (stating that another proof was given by L. Kalmár), in which paper he described several other results as consequences of the theorem. First he derived his theorem on the existence of a perfect matching in a regular bipartite graph:

---

[15] Let be given two partitions of a finite set $\mathcal{M}$. One of them should decompose the set into μ mutually disjoint classes $\mathcal{A}_1, \ldots, \mathcal{A}_\mu$ each of n elements, the other likewise in μ disjoint classes $\mathcal{B}_1, \ldots, \mathcal{B}_\mu$ each of n elements. Then there exists a system of elements $x_1, \ldots, x_\mu$ such that each $\mathcal{A}$-class and likewise each $\mathcal{B}$-class is represented by one element among the x_i.

[16] **Note added in proof.** I now notice that the theorem proved here is equivalent to a theorem of Dénes Kőnig on regular graphs.

[17] In an even circuit graph, the minimal number of vertices that exhaust the edges agrees with the maximal number of edges that pairwise do not contain any common end point.

> Um die Tragweite dieses Satzes zu beleuchten, wollen wir noch zeigen, daß ein von mir schon vor längerer Zeit bewiesener Satz über die Faktorenzerlegung von regulären endlichen paaren Graphen aus Satz 13 unmittelbar abgeleitet werden kann.
> Der betreffende Satz lautet:
> 14. *Jeder endliche paare reguläre Graph besitzt einen Faktor ersten Grades.*[18]

In a footnote, Kőnig mentioned:

> Später wurden für diesen Satz, bzw. für seine Interpretation in der Determinantentheorie und in der Kombinatorik verschiedene Beweise gegeben, so durch Frobenius, Sainte-Laguë, van der Waerden, Sperner, Skolem, Egerváry.[19]

Another consequence is a graph-theoretic variant of the result of Frobenius [1912] on reducible determinants:

> 16. *Im (paaren) Graphen G soll jede Kante einen der Punkte von $\Pi_1 = (P_1, P_2, \ldots, P_n)$ mit einem der Punkte von $\Pi_2 = (Q_1, Q_2, \ldots, Q_n)$ verbinden $(P_i \neq Q_j)$ und diejenigen Kanten von G, die in einem Faktor ersten Grades von G enthalten sind, sollen einen nichtzusammenhängenden Graphen G^* bilden. Dann kann man $r(> 0)$ Punkte aus Π_1 und $n - r(> 0)$ Punkte aus Π_2 so auswählen, daß keine Kante von G zwei ausgewählte Punkte verbinde.*[20]

As consequences in matrix theory, Kőnig [1932] gave:

> 17. *Verschwinden sämtliche Entwicklungsglieder aller Unterdeterminanten n-ter Ordnung einer Matrix von p Zeilen und q Spalten (wo $n \leq p$, $n \leq q$ ist), so verschwinden alle Elemente, welche r Zeilen mit $(p + q - n + 1) - r$ Spalten gemeinsam haben für $r = 1$, oder $2, \ldots,$ oder p.*[21]

and

> 18. *Die Minimalzahl der Reihen (Zeilen und Spalten), welche in ihr Gesamtheit jedes nicht-verschwindende Element einer Matrix enthalten, ist gleich der Maximalzahl von nicht-verschwindenden Elementen, welche paarweise verschiedenen Zeilen und verschiedenen Spalten angehören.*[22]

Again, a footnote is added:

---

[18] To illustrate the bearing of this theorem, we want to show that a theorem, proved by me already long ago, on the factorization of regular finite bipartite graphs, can be derived immediately from Theorem 13.
The theorem referred to reads:
14. *Every finite bipartite regular graph possesses a factor of first degree.*

[19] Later, several proofs were given for this theorem, respectively for its interpretation in determinant theory and in combinatorics, so by Frobenius, Sainte-Laguë, van der Waerden, Sperner, Skolem, Egerváry.

[20] 16. *Let every edge in the (bipartite) graph G connect a vertex of $\Pi_1 = (P_1, \ldots, P_n)$ with a vertex of $\Pi_2 = (Q_1, \ldots, Q_n)$ $(P_i \neq Q_j)$, and let those edges of G that are contained in a factor of first degree form a disconnected graph G^*. Then one can choose $r(> 0)$ vertices in Π_1 and $n - r(> 0)$ vertices in Π_2 such that no edge of G connects two of the chosen vertices.*

[21] 17. *If all expansion terms of all underdeterminants of the order n of a matrix with p rows and q columns vanish (where $n \leq p$, $n \leq q$), then all entries vanish that r rows have in common with $(p + q - n + 1) - r$ columns, for $r = 1$, or $2, \ldots,$ or p.*

[22] 18. *The minimum number of lines (rows and columns) that together contain each nonvanishing entry of a matrix, is equal to the maximum number of nonvanishing entries that pairwise belong to different rows and different columns.*

> Die Sätze 17 und 18 hat der Verfasser, mit den hier gegebenen Beweisen, am 26. März 1931 in der Budapester Mathematischen und Physikalischen Gesellschaft vorgetragen, s. [6]. Hieran anschließend hat dann E. Egerváry [1] für den Satz 18 einen anderen Beweis und eine interessante Verallgemeinerung gegeben.[23]

(We note that references [6] and [1] in Kőnig's article correspond to our references Kőnig [1931] and Egerváry [1931].)

Kőnig also derived the theorems of Frobenius [1912,1917] mentioned above:

> 19. *Wenn alle Glieder einer Determinante n-ter Ordnung verschwinden, so verschwinden alle Elemente, welche r Zeilen mit $n-r+1$ Spalten gemeinsam haben, für $r = 1$ oder $2, \ldots,$ oder n.*[24]

> 20. *In einer Determinante n-ter Ordnung D seien die nichtverschwindenden Elemente unabhängige Veränderliche. Ist D eine reduzible Funktion ihrer (nichtverschwindenden) Elemente, so verschwinden alle Elemente von D, welche r Zeilen mit $n-r$ Spalten gemeinsam haben für $r = 1$ oder $2, \ldots,$ oder $n-1$.*[25]

With respect to Frobenius [1912], Kőnig noticed in a footnote:

> Dort wird dieser Satz "aus verborgenen Eigenschaften der Determinanten mit nichtnegativen Elementen" durch komplizierte Betrachtungen bewiesen. Ich gab dann in 1915 in meiner Arbeit [4] einen elementaren graphentheoretischen Beweis (welcher hier durch einen noch einfacheren ersetzt wird). In 1917 hat dann auch Frobenius [3] einen elementaren Beweis publiziert, und zwar nach dem ich ihm meinen Beweis (in deutscher Übersetzung) zugeschickt hatte. Frobenius hat es dort unterlassen, diese Tatsache, sowie überhaupt meine Arbeit [4] zu erwähnen. Jedoch zitiert er meine Arbeit [5] und zwar mit folgender Bemerkung: "Die Theorie der Graphen, mittels deren Hr. Kőnig den obigen Satz [dies ist die determinantentheoretische Interpretation von Satz 14] abgeleitet hat, ist nach meiner Ansicht ein wenig geeignetes Hilfsmittel für die Entwicklung der Determinantentheorie. In diesem Falle führt sie zu einem ganz speziellen Satz vom geringem Werte. Was von seinem Inhalt Wert hat, ist in dem Satze II [dies ist der Frobeniussche Satz 19] ausgesprochen."
> Es ist wohl natürlich, daß der Verfasser vorliegender Abhandlung diese Meinung nicht unterschreiben wird. Die Gründe, die man für oder gegen den Wert oder Unwert eines Satzes oder eine Methode anführen könnte, haben stets, mehr oder weniger, einen subjektiven Character, so daß es vom geringen wissenschaftlichen Wert wäre, wenn wir hier den Standpunkt von Frobenius zu bekämpfen versuchten. Wollte aber Frobenius seine verwerfende Kritik über die Anwendbarkeit der Graphen auf Determinantentheorie damit begründen, daß sein tatsächlich "wertvoller" Satz 19 nicht graphentheoretisch bewiesen werden kann, so ist seine Begründung—wie wir gesehen haben—sicherlich nicht stichhaltig. Der graphentheoretische Beweis, den wir für Satz 19 gegeben haben, scheint uns ein einfacher und anschaulicher Beweis zu sein, der dem *kombinatorischen* Character der Satzes in natürlicher Weise entspricht und auch zu einer bemerkenswerten Verallgemeinerung (Satz 17) führt.

[23] The author has presented Theorems 17 and 18, with the proofs given here, on 26 March 1931 to the Budapest Mathematical and Physical Society, see [6]. Following this, E. Egerváry [1] has next given another proof for Theorem 18 and an interesting generalization.

[24] 19. *When all members of a determinant of the order n vanish, then all elements vanish that have r rows in common with $n-r+1$ columns, for $r = 1$ or $2, \ldots,$ or n.*

[25] 20. *Let, in a determinant D of order n, the nonvanishing entries be independent variables. If D is a reducible function of its (nonvanishing) entries, then all entries of D vanish that have r rows in common with $n-r$ columns for $r = 1$ or $2, \ldots,$ or $n-1$.*

> Es sei noch erwähnt, daß wir oben, im §2, beim Beweis des Satzes 16 einen Gedanken von FROBENIUS benützt haben, den er bei seiner Zurückführung des Satzes 20 auf Satz 19 angewendet hat.[26]

(We note that Kőnig's quotation 'aus verborgenen Eigenschaften der Determinanten mit nichtnegativen Elementen' is from Frobenius [1917]. The references [3], [4], and [5] in Kőnig's article correspond to our references Frobenius [1917], Kőnig [1915], and Kőnig [1916], respectively.)

In terms of transversals, the theorems of Frobenius and Kőnig have been rediscovered by Hall [1935] — see the historical notes on transversals in Section 22.7d. Other developments are mentioned in Section 19.5a.

---

[26] This theorem was proved there 'from hidden properties of determinants with nonnegative elements' by complicated arguments. Next, I gave in 1915, in my work [4], an elementary, graph-theoretic proof (which was replaced here by an even simpler one). Next, in 1917, also FROBENIUS [3] has published an elementary proof, and that after I had sent him my proof (in German translation). FROBENIUS has refrained from mentioning this fact there, as well as my work [4] at all. Yet, he quotes my work [5], and that with the following remark: 'The theory of graphs, by which Mr. KŐNIG has derived the theorem above [this is the determinant-theoretic interpretation of Theorem 14], is, to my opinion, of little appropriate help for the development of determinant theory. In this case it leads to a very special theorem of little value. What from its contents has value, is expressed in Theorem II [this is Theorem 19 of Frobenius]'.

Obviously, the author of the present treatise will not subscribe to this opinion. The arguments that one can produce for or against the value or valuelessness of a theorem or a method, have always, more or less, a subjective character, so that it would be of little scientific value when we here tried to fight the point of view of FROBENIUS. But if FROBENIUS wants to base his rejecting criticism about the applicability of graphs to determinant theory on the fact that his actually 'more valuable' Theorem 19 cannot be proved graph-theoretically, then his ground is—as we have seen—certainly not solid. The graph-theoretic proof that we have given for Theorem 19 seems to us to be a simple and illustrative proof, that corresponds naturally to the *combinatorial* character of the theorem and also leads to a remarkable generalization (Theorem 17).

Let it finally be mentioned that above, in §2, in the proof of Theorem 16, we have used an idea of FROBENIUS, which he has applied at his reduction of Theorem 20 to Theorem 19.

Chapter 17

Weighted bipartite matching and the assignment problem

The methods and results of the previous chapter can be extended to handle maximum-*weight* matchings. Egerváry's theorem is the weighted version of Kőnig's matching theorem. It led Kuhn to develop the 'Hungarian method' for the assignment problem. This problem is equivalent to finding a minimum-weight perfect matching in a complete bipartite graph.

17.1. Weighted bipartite matching

For bipartite graphs, Egerváry [1931] characterized the maximum weight of a matching by the following duality relation:

Theorem 17.1 (Egerváry's theorem)**.** *Let $G = (V, E)$ be a bipartite graph and let $w : E \to \mathbb{R}_+$ be a weight function. Then the maximum weight of a matching in G is equal to the minimum value of $y(V)$, where $y : V \to \mathbb{R}_+$ is such that*

$$y_u + y_v \geq w_e \tag{17.1}$$

for each edge $e = uv$. If w is integer, we can take y integer.

Proof. The maximum is not more than the minimum, since for any matching M and any $y \in \mathbb{R}_+^V$ satisfying (17.1) for each edge $e = uv$, one has

$$w(M) \leq \sum_{e=uv \in M} (y_u + y_v) \leq \sum_{v \in V} y_v. \tag{17.2}$$

To see equality, choose a $y \in \mathbb{R}_+^V$ attaining the minimum value. Let F be the set of edges e having equality in (17.1) and let R be the set of vertices v with $y_v > 0$.

If F contains a matching M covering R, we have equality throughout in (17.2), showing that the maximum is equal to the minimum value.

So we may assume that no such matching exists. Then by Corollary 16.8a there exists a stable set $S \subseteq R$ containing no edge and such that $|N(S)| < |S|$.

Then there is an $\alpha > 0$ such that decreasing y_v by α for $v \in S$ and increasing y_v by α for $v \in N(S)$ gives a better y — a contradiction.

If w is integer we can keep y integer, by taking $\varepsilon = 1$ throughout. ∎

(This is essentially the proof method of Egerváry [1931].)

We can formulate Egerváry's theorem in combinatorial terms. Let $G = (V, E)$ be a graph and let $w \in \mathbb{Z}_+^E$. A *w-vertex cover* is a vector $y \in \mathbb{Z}_+^V$ such that

$$(17.3) \qquad y_u + y_v \geq w_e$$

for each edge $e = uv$ of G. The *size* of any vector $y \in \mathbb{R}^V$ is the sum of its components.

Corollary 17.1a. *Let $G = (V, E)$ be a bipartite graph and let $w : E \to \mathbb{Z}_+$ be a weight function. Then the maximum weight of a matching in G is equal to the minimum size of a w-vertex cover.*

Proof. The corollary is a reformulation of the integer part of Egerváry's theorem (Theorem 17.1). ∎

Let A be the $V \times E$ incidence matrix of G. Egerváry's theorem states that for $w \in \mathbb{Z}_+^V$, the optima in the linear programming duality equation

$$(17.4) \qquad \max\{w^\mathsf{T} x \mid x \geq \mathbf{0}, Ax \leq \mathbf{1}\} = \min\{y^\mathsf{T}\mathbf{1} \mid y \geq \mathbf{0}, y^\mathsf{T} A \geq w^\mathsf{T}\}$$

are attained by integer vectors x and y. This also follows from the total unimodularity of A — see Section 18.3.

17.2. The Hungarian method

We describe the *Hungarian method* for the maximum-weight matching problem. In its basic form it is due to Kuhn [1955b], based on Egerváry's proof above. Sharpenings were given by Munkres [1957] (yielding a polynomial-time method), Iri [1960], Edmonds and Karp [1970], and Tomizawa [1971].

Let $G = (V, E)$ be a bipartite graph, with colour classes U and W, and let $w : E \to \mathbb{Q}$ be a weight function.

We start with matching $M = \emptyset$. If we have found a matching M, let D_M be the directed graph obtained from G by orienting each edge e in M from W to U, with length $l_e := w_e$, and orienting each edge e not in M from U to W, with length $l_e := -w_e$. Let U_M and W_M be the set of vertices in U and W, respectively, missed by M. If there is a $U_M - W_M$ path, find a shortest such path, P say, and reset $M' := M \triangle EP$.

We iterate until no $U_M - W_M$ path exists in D_M (whence M is a maximum-*size* matching). The maximum-weight matching among the matchings found, has maximum weight among all matchings.

To see this, call a matching M *extreme* if it has maximum weight among all matchings of size $|M|$. Then, inductively:

Theorem 17.2. *Each matching M found is extreme.*

Proof. This is clearly true if $M = \emptyset$. Suppose next that M is extreme, and let P and M' be the path and matching found in the iteration. Consider any extreme matching N of size $|M|+1$. As $|N| > |M|$, $M \cup N$ has a component Q that is an M-augmenting path. As P is a shortest M-augmenting path, we know $l(Q) \geq l(P)$. As $N \triangle Q$ is a matching of size $|M|$, and as M is extreme, we have $w(N \triangle Q) \leq w(M)$. Hence $w(N) = w(N \triangle Q) - l(Q) \leq w(M) - l(P) = w(M')$. ∎

If M is extreme, then D_M has no negative-length circuit C (otherwise $M \triangle C$ is a matching of size $|M|$ and larger weight than M). So by the theorem, we can find with the Bellman-Ford method a shortest $U_M - W_M$ path in time $O(nm)$, yielding an $O(n^2 m)$ method overall (Iri [1960]).

But in fact one may apply Dijkstra's method (Edmonds and Karp [1970], Tomizawa [1971]) and obtain a better time bound:

Theorem 17.3. *The method can be performed in time $O(n(m + n \log n))$.*

Proof. Let R_M denote the set of vertices reachable in D_M from U_M. We show that along with M we can keep a potential p for the subgraph $D_M[R_M]$ of D_M induced by R_M (with respect to the length function l defined above).[27]

When $M = \emptyset$ we take $p(v) := \max\{w_e \mid e \in E\}$ if $v \in U$ and $p(v) := 0$ if $v \in W$.

Suppose next that for given extreme M we have a potential p for $D_M[R_M]$. Then define $p'(v) := \mathrm{dist}_l(U_M, v)$ for each $v \in R_M$. Note that having p, one can determine p' in $O(m + n \log n)$ time (cf. Section 8.2).

Then p' is a potential for $D_{M'}[R_{M'}]$. To see this, let P be the path in D_M with $M' = M \triangle EP$. Trivially, $U_{M'} \subseteq U_M$. Moreover, $R_{M'} \subseteq R_M$. Indeed, otherwise some arc of $D_{M'}$ leaves R_M. As no arc of D_M leaves R_M, this implies that P has an arc entering R_M. So P has an arc leaving R_M, contradicting the definition of R_M. Concluding, $R_{M'} \subseteq R_M$.

Finally consider an arc (u, v) of $D_{M'}[R_{M'}]$. If (u, v) is also an arc of D_M, then $p'(v) \leq p'(u) + l(u, v)$. If (u, v) is not an arc of D_M, then (v, u) belongs to P, and hence (as P is shortest) $p'(u) = p'(v) + l(v, u)$. So $p'(v) - p'(u) = -l(v, u) = l(u, v)$. ∎

Observe that in the Hungarian method one can stop as soon as matching M' has no larger weight than M; that is, D_M has no $U_M - W_M$ path of negative length. For let N be a matching with $w(N) > w(M)$. So $|N| > |M|$

[27] A *potential* for a digraph $D = (V, A)$ with respect to a length function $l : A \to \mathbb{R}$ is a function $p : V \to \mathbb{R}$ satisfying $p(v) - p(u) \leq l(a)$ for each arc $a = (u, v)$.

(since all matchings of size $\leq |M|$ have weight $\leq w(M)$). Choose N with $|N \triangle M|$ minimal. By similar arguments as used in the proof of Theorem 17.2, we may assume that $N \triangle M$ has $|N| - |M|$ nontrivial components, each having one more edge in N than in M. So each component gives a $U_M - W_M$ path in D_M. As none of them have negative length, we have $w(N) \leq w(M)$, a contradiction.

Hence we can reduce the factor n in the time bound:

Theorem 17.4. *In a weighted bipartite graph, a maximum-weight matching can be found in time $O(n'(m + n \log n))$, where n' is the minimum size of a maximum-weight matching.*

Proof. See above. ∎

17.3. Perfect matching and assignment problems

The methods described above also find a maximum-weight perfect matching in a bipartite graph. This follows from the fact that a maximum-weight perfect matching is an extreme matching of size $\frac{1}{2}|V|$.

By multiplying all weights by -1, this problem can be seen to be equivalent to finding a *minimum*-weight perfect matching. Hence:

Corollary 17.4a. *A minimum-weight perfect matching can be found in time $O(n(m + n \log n))$.*

Proof. Directly from the above. ∎

This in turn gives an algorithm for the *assignment problem*:

(17.5) given: a rational $n \times n$ matrix $A = (a_{i,j})$;

find: a permutation π of $\{1, \ldots, n\}$ minimizing $\displaystyle\sum_{i=1}^{n} a_{i,\pi(i)}$.

Corollary 17.4b. *The assignment problem can be solved in time $O(n^3)$.*

Proof. Take $G = K_{n,n}$ in Corollary 17.4a. ∎

The following characterization of the minimum weight of a perfect matching can be derived from Egerváry's theorem — we however give a direct proof that might be illuminating:

Theorem 17.5. *Let $G = (V, E)$ be a bipartite graph having a perfect matching and let $w : E \to \mathbb{Q}$ be a weight function. The minimum weight of a perfect matching is equal to the maximum value of $y(V)$ taken over $y : V \to \mathbb{Q}$ with*

(17.6) $\quad y_u + y_v \leq w_e$ *for each edge* $e = uv$.

If w *is integer, we can take* y *integer.*

Proof. Clearly, the minimum is not less than the maximum, since for any perfect matching M and any $y \in \mathbb{Q}^V$ satisfying (17.6) one has

$$w(M) = \sum_{e \in M} w_e \geq \sum_{v \in V} y_v = y(V). \tag{17.7}$$

To see the reverse inequality, let M be a minimum-weight perfect matching. Make a digraph $D = (V, A)$, with length function, as follows. Orient any edge e of G from one colour class, U say, to the other, W say, with length w_e. Moreover, add for each edge e in M an arc parallel to e oriented from W to U, with length $-w_e$. As M is minimum-weight, the digraph has no negative-weight directed circuits (otherwise we could make a perfect matching of smaller weight). Hence, by Theorem 8.2, there exists a function $p : V \to \mathbb{Q}$ such that $w(a) \geq p(v) - p(u)$ for each arc $a = (u, v)$ of D. Defining $y_v := -p(v)$ for $v \in U$ and $y_v := p(v)$ for $v \in W$, we obtain a function y satisfying (17.6). For each edge $e = uv$ in M, the arcs (u, v) and (v, u) form a zero-length directed circuit in D, and therefore $w_e = y_u + y_v$. This gives equality in (17.7).

If w is integer, we can take p and hence y integer. ∎

17.4. Finding a minimum-size w-vertex cover

Given a maximum-weight matching M in a bipartite graph $G = (V, E)$ with weight $w : E \to \mathbb{Z}_+$, we can find a minimum-size w-vertex cover as follows. Let U and W be the colour classes of G. As before, define $U_M := U \setminus \bigcup M$ and $W_M := W \setminus \bigcup M$.

For any edge $e = uv$, with $u \in U$, $v \in W$, make an arc $a = (u, v)$, of length $l(a) := -w_e$. If $e \in M$, make also an arc $a' = (v, u)$, of length $l(a') := w_e$. We obtain a directed graph $D = (V, A)$ without negative-length directed circuits and no negative-length directed path from $U_M \cup (W \setminus W_M)$ to $W_M \cup (U \setminus U_M)$ (otherwise we can improve M). Then we can find a potential $p : V \to \mathbb{Z}$ such that $l(a) \geq p(v) - p(u)$ for each arc $a = (u, v)$ of D and such that $p(v) = 0$ for each $v \in U_M \cup W_M$, $p(v) \geq 0$ for each $v \in U$, and $p(v) \leq 0$ for each $v \in W$. To see this, add an extra vertex r, and arcs (r, v) for each $v \in U_M \cup (W \setminus W_M)$ and (v, r) for each $v \in W_M \cup (U \setminus U_M)$. Let the new arcs have length 0. Then the extended digraph D' has no negative-length circuits. Let p be a potential for D'. By translating, we can assume $p(r) = 0$. Resetting $p(v)$ to 0 if $v \in U_M \cup W_M$ maintains that p is a potential. This gives a potential for D as described.

Now set $y_v := -p(v)$ if $v \in U$ and $y_v := p(v)$ if $v \in W$. Then y is a w-vertex cover of size $w(M)$, and hence it is a minimum-size w-vertex cover. Therefore (Iri [1960]):

Theorem 17.6. *A minimum-size w-vertex cover in a bipartite graph can be found in $O(n(m + n \log n))$ time.*

Proof. See above. ∎

17.5. Further results and notes

17.5a. Complexity survey for maximum-weight bipartite matching

Complexity survey for the maximum-weight bipartite matching ($*$ indicates an asymptotically best bound in the table):

	$O(nW \cdot \mathrm{VC}(n,m))$	Egerváry [1931] (implicitly)
	$O(2^n n^2)$	Easterfield [1946]
	$O(nW \cdot \mathrm{DC}(n,m,W))$	Robinson [1949]
	$O(n^4)$	Kuhn [1955b], Munkres [1957][28] Hungarian method
	$O(n^2 m)$	Iri [1960]
	$O(n^3)$	Dinits and Kronrod [1969]
$*$	$O(n \cdot \mathrm{SP}_+(n,m,W))$	Edmonds and Karp [1970], Tomizawa [1971]
	$O(n^{3/4} m \log W)$	Gabow [1983b,1985a,1985b]
$*$	$O(\sqrt{n}\, m \log(nW))$	Gabow and Tarjan [1988b,1989] (cf. Orlin and Ahuja [1992])
	$O(\sqrt{n}\, mW)$	Kao, Lam, Sung, and Ting [1999]
$*$	$O(\sqrt{n}\, mW \log_n(n^2/m))$	Kao, Lam, Sung, and Ting [2001]

Here $W := \|w\|_\infty$ (assuming w to be integer-valued). Moreover, $\mathrm{SP}_+(n,m,W)$ is the time needed to find a shortest path in a directed graph with n vertices and m arcs, with nonnegative integer lengths on the arcs, each at most W. Similarly, $\mathrm{DC}(n,m,W)$ is the time required to find a negative-length directed circuit in a directed graph with n vertices and m arcs, with integer lengths on the arcs, each at most W in absolute value. Moreover, $\mathrm{VC}(n,m)$ is the time required to find a minimum-size vertex cover in a bipartite graph with n vertices and m edges.

Dinits [1976] gave an algorithm for finding a minimum-weight matching in $K_{p,q}$ of size p, with time bound $O(|p|^3 + pq)$ (taking $p \le q$).

17.5b. Further notes

Simplex method. Finding a maximum-weight matching in a bipartite graph is a special case of a linear programming problem (see Chapter 18), and hence linear programming methods like the simplex method apply.

[28] Munkres showed that Kuhn's 'Hungarian method' takes $O(n^4)$ time.

Gassner [1964] studied cycling of the simplex method when applied to the assignment problem. Using the 'strongly feasible' trees of Cunningham [1976], Roohy-Laleh [1980] showed that a version of the simplex method solves the assignment problem in less than n^3 pivots (cf. Hung [1983], Orlin [1985], and Akgül [1993]; the last paper gives a method with $O(n^2)$ pivots, yielding an $O(n(m + n \log n))$ algorithm).

Balinski [1985] (cf. Goldfarb [1985]) showed that a version of the dual simplex method (the *signature method*) solves the assignment problem in strongly polynomial time ($O(n^2)$ pivots, yielding an $O(n^3)$ algorithm). More can be found in Dantzig [1963], Barr, Glover, and Klingman [1977], Balinski [1986], Ahuja and Orlin [1988,1992], Akgül [1988], Paparrizos [1988], and Akgül and Ekin [1991].

For further algorithmic studies of the assignment problem, consult Flood [1960], Kurtzberg [1962], Hoffman and Markowitz [1963], Balinski and Gomory [1964], Tabourier [1972], Carpaneto and Toth [1980a,1983,1987], Hung and Rom [1980], Karp [1980], Bertsekas [1981,1987,1992] ('auction method'), Engquist [1982], Avis [1983], Avis and Devroye [1985], Derigs [1985b,1988a], Carraresi and Sodini [1986], Derigs and Metz [1986a], Glover, Glover, and Klingman [1986], Jonker and Volgenant [1986], Kleinschmidt, Lee, and Schannath [1987], Avis and Lai [1988], Bertsekas and Eckstein [1988], Motwani [1989,1994], Kalyanasundaram and Pruhs [1991, 1993], Khuller, Mitchell, and Vazirani [1991,1994], Goldberg and Kennedy [1997] (push-relabel), and Arora, Frieze, and Kaplan [1996,2002].

For computational studies, see Silver [1960], Florian and Klein [1970], Barr, Glover, and Klingman [1977] (simplex method), Gavish, Schweitzer, and Shlifer [1977] (simplex method), Bertsekas [1981], Engquist [1982], McGinnis [1983], Lindberg and Ólafsson [1984] (simplex method), Glover, Glover, and Klingman [1986], Jonker and Volgenant [1987], Bertsekas and Eckstein [1988], and Goldberg and Kennedy [1995] (push-relabel). Consult also Johnson and McGeoch [1993].

Linear-time algorithm for weighted bipartite matching problems satisfying a quadrangle or other inequality were given by Karp and Li [1975], Buss and Yianilos [1994,1998], and Aggarwal, Bar-Noy, Khuller, Kravets, and Schieber [1995].

For generating *all* minimum-weight perfect matchings, see Fukuda and Matsui [1992]. For studies of the 'most vital' edges in a weighted bipartite graph, see Hung, Hsu, and Sung [1993].

Aráoz and Edmonds [1985] gave an example showing that iterative dual improvements in the linear programming problem dual to the assignment problem, need not converge for irrational data.

For the 'bottleneck' assignment problem, see Gross [1959] and Garfinkel [1971]. An algebraic approach to assignment problems was described by Burkard, Hahn, and Zimmermann [1977].

For surveys on matching algorithms, see Galil [1983,1986a,1986b]. Books covering the weighted bipartite matching and assignment problems include Ford and Fulkerson [1962], Dantzig [1963], Christofides [1975], Lawler [1976b], Bazaraa and Jarvis [1977], Burkard and Derigs [1980], Papadimitriou and Steiglitz [1982], Gondran and Minoux [1984], Rockafellar [1984], Derigs [1988a], Bazaraa, Jarvis, and Sherali [1990], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], Mehlhorn and Näher [1999], and Korte and Vygen [2000].

17.5c. Historical notes on weighted bipartite matching and optimum assignment

Monge: optimum assignment

The assignment problem is one of the first studied combinatorial optimization problems. It was investigated by Monge [1784], albeit camouflaged as a continuous problem, and often called a transportation problem.

Monge was motivated by transporting earth, which he considered as the discontinuous, combinatorial problem of transporting molecules:

> Lorsqu'on doit transporter des terres d'un lieu dans un autre, on a coutime de donner le nom de *Déblai* au volume des terres que l'on doit transporter, & le nom de *Remblai* à l'espace qu'elles doivent occuper après le transport.
> Le prix du transport d'une molécule étant, toutes choses d'ailleurs égales, proportionnel à son poids & à l'espace qu'on lui fait parcourir, & par conséquent le prix du transport total devant être proportionnel à la somme des produits des molécules multipliées chacune par l'espace parcouru, il s'ensuit que le déblai & le remblai étant donnés de figure & de position, il n'est pas indifférent que telle molécule du déblai soit transportée dans tel ou tel autre endroit du remblai, mais qu'il y a une certaine distribution à faire des molécules du premier dans le second, d'après laquelle la somme de ces produits sera la moindre possible, & le prix du transport total sera un *minimum.*[29]

Monge described an interesting geometric method to solve this problem. Consider a line that is tangent to both areas, and move the molecule m touched in the first area to the position x touched in the second area, and repeat, until all earth has been transported. Monge's argument that this would be optimum is simple: if molecule m would be moved to another position, then another molecule should be moved to position x, implying that the two routes traversed by these molecules cross, and that therefore a shorter assignment exists:

> Étant données sur un même plan deux aires égales $ABCD$, & $abcd$, terminées par des contours quelconques, continus ou discontinus, trouver la route que doit suivre chaque molécule M de la premiere, & le point m où elle doit arriver dans la seconde, pour que tous les points étant semblablement transportés, ils replissent exactement la seconde aire, & que la somme des produits de chaque molécule multipliée par l'espace parcouru soit un *minimum.*
> Si par un point M quelconque de la première aire, on mène une droite Bd, telle que le segment BAD soit égal au segment bad, je dis que pour satisfaire à la question, il faut que toutes les molécules du segment BAD, soient portées sur le segment bad, & que par conséquent les molécules du segment BCD soient portées

[29] When one must transport earth from one place to another, one usually gives the name of *Déblai* to the volume of earth that one must transport, & the name of *Remblai* to the space that they should occupy after the transport.

The price of the transport of one molecule being, if all the rest is equal, proportional to its weight & to the distance that one makes it covering, & hence the price of the total transport having to be proportional to the sum of the products of the molecules each multiplied by the distance covered, it follows that, the déblai & the remblai being given by figure and position, it makes difference if a certain molecule of the déblai is transported to one or to another place of the remblai, but that there is a certain distribution to make of the molecules from the first to the second, after which the sum of these products will be as little as possible, & the price of the total transport will be a *minimum.*

sur le segment égal *bcd*; car si un point K quelconque du segment BAD, étoit porté sur un point k de *bcd*, il faudroit nécessairement qu'un point égal L, pris quelque part dans BCD, fût transporté dans un certain point l de *bad*, ce qui ne pourroit pas se faire sans que les routes Kk, Ll, ne se coupassent entre leurs extrémités, & la somme des produits des molécules par les espaces parcourus ne seroit pas un *minimum.* Pareillement, si par un point M' infiniment proche du point M, on mène la droite $B'd'$, telle qu'on ait encore le segment $B'A'D'$, égal au segment $b'a'd'$, il faut pour que la question soit satisfaite, que les molécules du segment $B'A'D'$ soient transportées sur $b'a'd'$. Donc toutes les molécules de l'élément $BB'D'D$ doivent être transportées sur l'élément égal $bb'd'd$. Ainsi en divisant le déblai & le remblai en une infinité d'élémens par des droites qui coupent dans l'un & dans l'autre des segmens égaux entr'eux, chaque élément du déblai doit être porté sur l'élément correspondant du remblai.

Les droites Bd & $B'd'$ étant infiniment proches, il est indifférent dans quel ordre les molécules de l'élément $BB'D'D$ se distribuent sur l'élément $bb'd'd$; de quelque manière en effet que se fasse cette distribution, la somme des produits des molécules par les espaces parcourus, est toujours la même, mais si l'on remarque que dans la pratique il convient de débleyer premièrement les parties qui se trouvent sur le passage des autres, & de n'occuper que les dernières les parties du remblai qui sont dans le même cas; la molécule MM' ne devra se transporter que lorsque toute la partie $MM'D'D$ qui la précêde, aura été transportée en $mm'd'd$; donc dans cette hypothèse, si l'on fait $mm'd'd = MM'D'D$, le point m sera celui sur lequel le point M sera transporté.[30]

Although geometrically intuitive, the method is however not fully correct, as was noted by Appell [1928]:

---

[30] Being given, in the same plane, two equal areas $ABCD$ & *abcd*, bounded by arbitrary contours, continuous or discontinuous, find the route that every molecule M of the first should follow & the point m where it should arrive in the second, so that, all points being transported likewise, they fill precisely the second area & so that the sum of the products of each molecule multiplied by the distance covered, is *minimum.*

If one draws a straight line Bd through an arbitrary point M of the first area, such that the segment BAD is equal to the segment *bad*, I assert that, in order to satisfy the question, all molecules of the segment BAD should be carried on the segment *bad*, & hence the molecules of the segment BCD should be carried on the equal segment *bcd*; for, if an arbitrary point K of segment BAD, is carried to a point k of *bcd*, then necessarily some point L somewhere in BCD is transported to a certain point l in *bad*, which cannot be done without that the routes Kk, Ll cross each other between their end points, & the sum of the products of the molecules by the distances covered would not be a *minimum.* Likewise, if one draws a straight line $B'd'$ through a point M' infinitely close to point M, in such a way that one still has that segment $B'A'D'$ is equal to segment $b'a'd'$, then in order to satisfy the question, the molecules of segment $B'A'D'$ should be transported to $b'a'd'$. So all molecules of the element $BB'D'D$ must be transported to the equal element $bb'd'd$. Dividing the déblai & the remblai in this way into an infinity of elements by straight lines that cut in the one & in the other segments that are equal to each other, every element of the déblai must be carried to the corresponding element of the remblai.

The straight lines Bd & $B'd'$ being infinitely close, it does not matter in which order the molecules of element $BB'D'D$ are distributed on the element $bb'd'd$; indeed, in whatever manner this distribution is being made, the sum of the products of the molecules by the distances covered is always the same; but if one observes that in practice it is convenient first to dig off the parts that are in the way of others, & only at last to cover similar parts of the remblai; the molecule MM' must be transported only when the whole part $MM'D'D$ that precedes it will have been transported to $mm'd'd$; hence with this hypothesis, if one has $mm'd'd = MM'D'D$, point m will be the one to which point M will be transported.

> Il est bien facile de faire la figure de manière que les chemins suivis par les deux parcelles dont parle Monge ne se croisent pas.[31]

(cf. Taton [1951]).

Egerváry

Egerváry [1931] published a weighted version of Kőnig's theorem:

> *Ha az $\|a_{ij}\|$ n-edrendű matrix elemei adott nem negatív egész számok, úgy a*
>
> $$\lambda_i + \mu_j \geq a_{ij}, \qquad (i, j = 1, 2, ...n),$$
> *(λ_i, μ_j nem negatív egész számok)*
>
> *feltételek mellett*
>
> $$\min . \sum_{k=1}^{n} (\lambda_k + \mu_k) = \max . (a_{1\nu_1} + a_{2\nu_2} + \cdots + a_{n\nu_n}).$$
>
> *hol $\nu_1, \nu_2, ...\nu_n$ az $1, 2, ...n$ számok összes permutációit befutják.*[32]

The proof method of Egerváry is essentially algorithmic. Assume that the $a_{i,j}$ are integer. Let λ_i^*, μ_j^* attain the minimum. If there is a permutation ν of $\{1, \ldots, n\}$ with $\lambda_i^* + \mu_{\nu_i}^* = a_{i,\nu_i}$ for all i, then this permutation attains the maximum, and we have the required equality. If no such permutation exists, by Frobenius' theorem there are subsets I, J of $\{1, \ldots, n\}$ such that

(17.8) $\qquad \lambda_i^* + \mu_j^* > a_{i,j}$ for all $i \in I$, $j \in J$

and such that $|I| + |J| = n + 1$. Resetting $\lambda_i^* := \lambda_i^* - 1$ if $i \in I$ and $\mu_j^* := \mu_j^* + 1$ if $j \notin J$, would give feasible values for the λ_i and μ_j, however with their total sum being decreased. This is a contradiction.

Translated into an algorithm, it consists of applying $O(nW)$ times a cardinality bipartite matching algorithm, where W is the maximum weight. So its running time is $O(nW \cdot B(n))$, where $B(n)$ is a bound on the running time of any algorithm finding a maximum-size matching and a minimum-size vertex cover in a bipartite graph with n vertices.

This method forms the basis for the *Hungarian method* of Kuhn [1955b,1956] — see below.

---

[31] It is very easy to make the figure in such a way that the routes followed by the two particles of which Monge speaks, do not cross each other.

[32] *If the elements of the matrix $\|a_{ij}\|$ of order n are given nonnegative integers, then under the assumption*

$$\lambda_i + \mu_j \geq a_{ij}, \qquad (i, j = 1, 2, ...n),$$
(λ_i, μ_j nonnegative integers)

we have

$$\min . \sum_{k=1}^{n} (\lambda_k + \mu_k) = \max . (a_{1\nu_1} + a_{2\nu_2} + \cdots + a_{n\nu_n}).$$

where $\nu_1, \nu_2, ...\nu_n$ run over all possible permutations of the numbers $1, 2, ...n$.

The 1940s

The first algorithm for the assignment problem might have been published by Easterfield [1946], who described his motivation as follows:

> In the course of a piece of organisational research into the problems of demobilisation in the R.A.F., it seemed that it might be possible to arrange the posting of men from disbanded units into other units in such a way that they would not need to be posted again before they were demobilised; and that a study of the numbers of men in the various release groups in each unit might enable this process to be carried out with a minimum number of postings. Unfortunately the unexpected ending of the Japanese war prevented the implications of this approach from being worked out in time for effective use. The algorithm of this paper arose directly in the course of the investigation.

Easterfield seems to have worked without knowledge of the existing literature. He formulated and proved a theorem equivalent to Hall's marriage theorem (see Section 22.1a) and he described a primal-dual type method for the assignment problem from which Egerváry's result given above follows. The idea of the method can be described as follows.

Let $A = (a_{i,j})$ be an $n \times n$ matrix and let for each column index j, I_j be the set of row indices i for which $a_{i,j}$ is minimum among all entries in row i. If the collection $(I_1, \ldots, I_n)$ has a transversal, say $i_1, \ldots, i_n$ (with $i_j \in I_j$), then $i_j \to j$ is an optimum assignment.

If $(I_1, \ldots, I_n)$ has no transversal, let $\mathcal{J}$ be the collection of subsets J of $\{1, \ldots, n\}$ for which $(I_j \mid j \in J)$ has a transversal. Select an inclusionwise minimal set J that is not in $\mathcal{J}$. Then there exists an $\varepsilon > 0$ such that subtracting ε from each entry in each of the columns in J extends $\mathcal{J}$ by (at least) J. (This can be seen using Hall's condition.)

Easterfield described an implementation (including scanning all subsets in lexicographic order), that has running time $O(2^n n^2)$. (This is better than scanning all permutations, which takes time $\Omega(n!)$.) The algorithm was explained again by Easterfield [1960].

Birkhoff [1946] derived from Hall's marriage theorem that each doubly stochastic matrix is a convex combination of permutation matrices. Birkhoff's motivation was:

> Estas matrices son interesantes para la probabilidad, y los cuadrados mágicos son múltiplos escalares de estas matrices.[33]

A breakthrough in solving the assignment problem came when Dantzig [1951a] showed that the assignment problem can be formulated as a linear programming problem that automatically has an integer optimum solution. Indeed, by Birkhoff's theorem, minimizing a linear functional over the set of doubly stochastic matrices (which is a linear programming problem) gives a permutation matrix, being the optimum assignment. So the assignment problem can be solved with the simplex method.

In an address delivered on 9 September 1949 at a meeting of the American Psychological Association at Denver, Colorado, Thorndike [1950] studied the problem of the 'classification' of personnel:

[33] These matrices are interesting because of the probability, and the magic squares are scalar multiples of these matrices.

> The past decade, and particularly the war years, have witnessed a great concern about the classification of personnel and a vast expenditure of effort presumably directed towards this end.

He exhibited little trust in mathematicians:

> There are, as has been indicated, a finite number of permutations in the assignment of men to jobs. When the classification problem as formulated above was presented to a mathematician, he pointed to this fact and said that from the point of view of the mathematician there was no problem. Since the number of permutations was finite, one had only to try them all and choose the best. He dismissed the problem at that point. This is rather cold comfort to the psychologist, however, when one considers that only ten men and ten jobs mean over three and a half million permutations. Trying out all the permutations may be a mathematical solution to the problem, it is not a practical solution.

Thorndike next presented three heuristics for the assignment problem, the *Method of Divine Intuition*, the *Method of Daily Quotas*, and the *Method of Predicted Yield*.

In a RAND Report dated 5 December 1949, Robinson [1949] reported that an 'unsuccessful attempt' to solve the traveling salesman problem, led her to the following 'cycle-cancelling' method for the optimum assignment problem.

Let matrix $(a_{i,j})$ be given, and consider any permutation π. Define for all i,j a 'length' $l_{i,j}$ by: $l_{i,j} := a_{j,\pi(i)} - a_{i,\pi(i)}$ if $j \neq \pi(i)$ and $l_{i,\pi(i)} = \infty$. If there exists a negative-length directed circuit, there is a straightforward way to improve π. If there is no such circuit, then π is an optimal permutation.

This clearly is a finite method. Robinson remarked:

> I believe it would be feasible to apply it to as many as 50 points provided suitable calculating equipment is available.

The early 1950s

Von Neumann considered the complexity of the assignment problem. In a talk in the Princeton University Game Seminar on 26 October 1951, he showed that the assignment problem can be reduced to finding an optimum column strategy in a certain zero-sum two-person game, and that it can be found by a method given by Brown and von Neumann [1950]. We give first the mathematical background.

A zero-sum two-person game is given by a matrix A, the 'pay-off matrix'. The interpretation as a game is that a 'row player' chooses a row index i and a 'column player' chooses simultaneously a column index j. After that, the column player pays the row player $A_{i,j}$. The game is played repeatedly, and the question is what is the best strategy.

Let A have order $m \times n$. A *row strategy* is a vector $x \in \mathbb{R}^m_+$ satisfying $\mathbf{1}^\mathsf{T} x = 1$. Similarly, a *column strategy* is a vector $y \in \mathbb{R}^n_+$ satisfying $\mathbf{1}^\mathsf{T} y = 1$. Then

$$(17.9) \qquad \max_x \min_j (x^\mathsf{T} A)_j = \min_y \max_i (Ay)_i,$$

where x ranges over row strategies, y over column strategies, i over row indices, and j over column indices. Equality (17.9) follows from LP-duality.

It implies that the best strategy for the row player is to choose rows with distribution an optimum x in (17.9). Similarly, the best strategy for the column player is to choose columns with distribution an optimum y in (17.9). The average pay-off then is the value of (17.9).

The method of Brown [1951] to determine the optimum strategies is that each player chooses in turn the line that is best with respect to the distribution of the lines chosen by the opponent so far. It was proved by Robinson [1951] that this converges to optimum strategies. The method of Brown and von Neumann [1950] is a continuous version of this, and amounts to solving a system of linear differential equations.

Now von Neumann noted that the following reduces the assignment problem to the problem of finding an optimum column strategy. Let $C = (c_{i,j})$ be an $n \times n$ cost matrix, as input for the assignment problem. We may assume that C is positive. Consider the following pay-off matrix A, of order $2n \times n^2$, with columns indexed by ordered pairs (i,j) with $i,j = 1,\ldots,n$. The entries of A are given by: $A_{i,(i,j)} := 1/c_{i,j}$ and $A_{n+j,(i,j)} := 1/c_{i,j}$ for $i,j = 1,\ldots,n$, and $A_{k,(i,j)} := 0$ for all i,j,k with $k \neq i$ and $k \neq n+j$. Then any minimum-cost assignment, of cost γ say, yields an optimum column strategy y by: $y_{(i,j)} := c_{i,j}/\gamma$ if i is assigned to j, and $y_{(i,j)} := 0$ otherwise. Any optimum column strategy is a convex combination of strategies obtained this way from optimum assignments. So an optimum assignment can in principle be found by finding an optimum column strategy.

According to a transcript of the talk (cf. von Neumann [1951,1953]), von Neumann noted the following on the number of steps:

> It turns out that this number is a moderate power of n, i.e., considerably smaller than the "obvious" estimate n! mentioned earlier.

However, no further argumentation is given. (Related observations were given by Dulmage and Halperin [1955] and Koopmans and Beckmann [1955,1957].)

Beckmann and Koopmans [1952] studied the quadratic assignment problem, and they noted that the traveling salesman problem is a special case. In a Cowles Commission Discussion Paper of 2 April 1953, Beckmann and Koopmans [1953] mentioned applying polyhedral methods to solve the assignment problem, and noted:

> It should be added that in all the assignment problems discussed, there is, of course, the obvious brute force method of enumerating all assignments, evaluating the maximand at each of these, and selecting the assignment giving the highest value. This is too costly in most cases of practical importance, and by a method of solution we have meant a procedure that reduces the computational work to manageable proportions in a wider class of cases.

Geometric methods were proposed by Lord [1952] and Dwyer [1954] (the 'method of optimal regions') and other heuristics by Votaw and Orden [1952] and Törnqvist [1953]. A survey of developments on the assignment problem until 1955 was given by Motzkin [1956].

Computational results of the early 1950s

In a paper presented at the Symposium on Linear Inequalities and Linear Programming (14–16 June 1951 in Washington, D.C.), Votaw and Orden [1952] mentioned that solving a 10×10 transportation problem took 3 minutes on the SEAC (National Bureau of Standards Eastern Automatic Computer). However, in a later paper (submitted 1 November 1951), Votaw [1952] said that solving a 10×10 assignment problem with the simplex method on the SEAC took 20 minutes.

Moreover, in his reminiscences, Kuhn [1991] mentioned:

> The story begins in the summer of 1953 when the National Bureau of Standards and other US government agencies had gathered an outstanding group of combinatorialists and algebraists at the Institute for Numerical Analysis (INA) located on the campus of the University of California at Los Angeles. Since space was tight, I shared an office with Ted Motzkin, whose pioneering work on linear inequalities and related systems predates linear programming by more than ten years. A rather unique feature of the INA was the presence of the Standards Western Automatic Computer (SWAC), the entire memory of which consisted of 256 Williamson cathode ray tubes. The SWAC was faster but smaller than its sibling machine, the Standards Eastern Automatic Computer (SEAC), which boasted a liquid mercury memory and which had been coded to solve linear programs.
>
> During the summer, C.B. Tompkins was attempting to solve 10 by 10 assignment problems by programming the SWAC to enumerate the $10! = 3,628,800$ permutations of 10 objects. He never succeeded in this project.
>
> Thus, the 10 by 10 assignment problem is a linear program with 100 nonnegative variables and 20 equation constraints (of which only 19 are needed). In 1953, there was no machine in the world that had been programmed to solve a linear program this large!

If 'the world' includes the Eastern Coast of the U.S.A., there seems to be some discrepancy with the remarks of Votaw [1952] mentioned above.

On 23 April 1954, Gleyzal [1955] wrote that a code of his algorithm for the transportation problem, for the special case of the assignment problem with an 8×8 matrix, had just been composed for the SWAC.

Tompkins [1956] mentioned the following 'branch-and-bound' approach to the assignment problem:

> Benjamin Handy, on the suggestion of D.H. Lehmer and with advice from T.S. Motzkin [1], coded this problem for SWAC; he used exhaustive search including rejection of blocks of permutations when the first few elements of the trace led to a hopelessly low contribution. The problem worked for a problem whose matrix had 12 rows and 12 columns and was composed of random three-digit numbers. The solution in this case took three hours. Some restrictions which had been imposed concerning the types of problems to which the code should be applicable led to some inefficiencies; however, the simplex method of G.B. Dantzig [7] and various other methods of solution of this problem seem greatly superior to this method of exhaustive search;

(References [1] and [7] in this quotation are Motzkin [1956] and Dantzig [1951b].)

Kuhn, Munkres: the Hungarian method

Kuhn [1955b,1956] developed a new combinatorial procedure for solving the assignment problem. The method is based on the work of Egerváry [1931], and therefore Kuhn introduced the name *Hungarian method* for it. (According to Kuhn [1955b], the algorithm is 'latent in work of D. Kőnig and J. Egerváry'.) The method was sharpened by Munkres [1957].

In an article *On the origin of the Hungarian method*, Kuhn [1991] presented the following reminiscences on the Hungarian method, from the time starting Summer 1953:

> During this period, I was reading Kőnig's classical book on the theory of graphs and realized that the matching problem for a bipartite graph on two sets of n vertices was exactly the same as an n by n assignment problem with all $a_{ij} = 0$ or 1. More significantly, Kőnig had given a combinatorial algorithm (based on augmenting paths) that produces optimal solutions to the matching problem and its combinatorial (or linear programming) dual. In one of the several formulations given by Kőnig (p. 240, Theorem D), given an n by n matrix $A = (a_{ij})$ with all $a_{ij} = 0$ or 1, the maximum number of 1's that can be chosen with no two in the same line (horizontal row or vertical column) is equal to the minimum number of lines that contain all of the 1's. Moreover, the algorithm seemed to be 'good' in a sense that will be made precise later. The problem then was: how could the general assignment problem be reduced to the 0-1 special case?
> Reading Kőnig's book more carefully, I was struck by the following footnote (p. 238, footnote 2): "... Eine Verallgemeinerung dieser Sätze gab Egerváry, Matrixok kombinatorius tulajdonságairól (Über kombinatorische Eigenschaften von Matrizen), Matematikai és Fizikai Lapok, *38*, 1931, S. 16-28 (ungarisch mit einem deutschen Auszug) ..." This indicated that the key to the problem might be in Egerváry's paper. When I returned to Bryn Mawr College in the fall, I obtained a copy of the paper together with a large Hungarian dictionary and grammar from the Haverford College library. I then spent two weeks learning Hungarian and translated the paper [1]. As I had suspected, the paper contained a method by which a general assignment problem could be reduced to a finite number of 0-1 assignment problems.
> Using Egerváry's reduction and Kőnig's maximum matching algorithm, in the fall of 1953 I solved several 12 by 12 assignment problems (with 3-digit integers as data) by hand. Each of these examples took under two hours to solve and I was convinced that the combined algorithm was 'good'. This must have been one of the last times when pencil and paper could beat the largest and fastest electronic computer in the world.

(Reference [1] is the English translation of the paper of Egerváry [1931].)

The method described by Kuhn is a sharpening of the method of Egerváry sketched above, in two respects: (i) it gives an (augmenting path) method to find either a perfect matching or sets I and J as required, and (ii) it improves the λ_i and μ_j not by 1, but by the largest value possible.

Kuhn [1955b] described the method in terms of matrices — in terms of graphs it amounts to the following algorithm for the maximum weighted perfect matching problem in a complete bipartite graph $G = (V, E)$, with weight function $w : E \to \mathbb{Z}_+$. Let U and W be the colour classes of G. Throughout there is a function $p : V \to \mathbb{Z}$ satisfying

(17.10) $\quad p(u) + p(v) \geq w(uv)$ for each edge uv

and a matching M in the subgraph $G' = (V, E')$ of G consisting of those edges having equality in (17.10).

If M is not a perfect matching, orient each edge in M from W to U, and every other edge of G' from U to W, giving graph D'_M. Let U_M and W_M be the sets of vertices in U and W missed by M.

Kuhn [1955b] described a depth-first search to find the set R_M of vertices that are reachable by a directed path in D'_M from U_M. (In a subsequent paper, Kuhn [1956] described a breadth-first search, starting at only one vertex in U_M.)

Case 1: $R_M \cap W_M \neq \emptyset$. We have an M-augmenting path in G', by which we increase M.

Case 2: $R_M \cap U_M = \emptyset$. Determine

(17.11) $\mu := \min\{p(u) + p(v) - w(u, v) \mid u \in U \cap R_M, v \in W \setminus R_M\}$.

This number is positive, since no edge of G' connects $U \cap R_M$ and $W \setminus R_M$. Decrease $p(u)$ by μ if $u \in U \cap R_M$ and increase $p(v)$ by μ if $v \in W \cap R_M$. Then (17.10) is maintained, while the sum $\sum_{v \in V} p(v)$ decreases (as $|U \cap R_M| > |W \cap R_M|$).

After this we iterate, until we have a perfect matching M in G', which is a maximum-weight perfect matching.

Kuhn [1955b] contented himself with stating that the number of iterations is finite (since the number of iterations where Case 2 applies is finite (as $\sum_v p(v)$ is nonnegative)).

It was observed by Munkres [1957] that the method runs in strongly polynomial time, since, between any two occurrences of Case 1, the number of iterations where Case 2 applies is at most n, as at each such iteration $R_M \cap W$ increases (namely by all vertices v that attain the minimum (17.11)).

So the number of iterations is at most n^2 (since M can increase at most n times). As the (depth- or breadth-first) search takes $O(n^2)$ this gives an $O(n^4)$ algorithm.

Munkres [1957] observed also that after an occurrence of Case 2 one can continue the search of the previous iteration, since edges of G' traversed in the search from U_M, remain edges of the new graph G'. Hence between any two occurrences of Case 1, the depth-first search takes time $O(n^2)$. This still gives an $O(n^4)$ algorithm, since calculating the minimum (17.11) takes $O(n^2)$ time. (Munkres claimed that his algorithm takes $O(n^3)$ operations, but he takes 'scanning a line' (that is, considering all edges incident with a given vertex) as one operation.)

(However, all Case 2-iterations can be combined to one iteration, by finding distances from U_M, with respect to the length function w in the oriented G'. It amounts to including a Dijkstra-like labeling, yielding an $O(n^3)$ time bound. This is the method we described in Section 17.2. This principle was noticed by Edmonds and Karp [1970] and Tomizawa [1971].)

Ford and Fulkerson [1955,1957b] (cf. Ford and Fulkerson [1956c,1956d]) extended the Hungarian method to general transportation problems. They state in Ford and Fulkerson [1956c,1956d]:

> Large systems involving hundreds of equations in thousands of unknowns have been successfully solved by hand using the simplex computation. The procedure of this paper has been compared with the simplex method on a number of randomly chosen problems and has been found to take roughly half the effort for small problems. We believe that as the size of the problem increases, the advantages of the present method become even more marked.

In a footnote, the authors add as to the assignment problem:

> The largest example tried was a 20×20 optimal assignment problem. For this example, the simplex method required well over an hour, the present method about thirty minutes of hand computation.

Chapter 18

Linear programming methods and the bipartite matching polytope

The weighted matching problem for bipartite graphs discussed in the previous chapter is related to the 'matching polytope' and the 'perfect matching polytope', and can be handled with linear programming methods by the total unimodularity of the incidence matrix of a bipartite graph.
In this chapter, graphs can be assumed to be simple.

18.1. The matching and the perfect matching polytope

Let $G = (V, E)$ be a graph. The *perfect matching polytope* $P_{\text{perfect matching}}(G)$ of G is defined as the convex hull of the incidence vectors of perfect matchings in G. So $P_{\text{perfect matching}}(G)$ is a polytope in $\mathbb{R}^E$.

The perfect matching polytope is a polyhedron, and hence can be described by linear inequalities. The following are clearly valid inequalities:

$$\text{(18.1)} \qquad \begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for each edge } e, \\ \text{(ii)} & x(\delta(v)) = 1 & \text{for each vertex } v. \end{array}$$

These inequalities are generally not enough (for instance, not for K_3). However, as Birkhoff [1946] showed, for bipartite graphs they are enough:

Theorem 18.1. *If G is bipartite, the perfect matching polytope of G is determined by* (18.1).

Proof. Let x be a vertex of the polytope determined by (18.1). Let F be the set of edges e with $x_e > 0$. Suppose that F contains a circuit C. As C has even length, $EC = M \cup N$ for two disjoint matchings M and N. Then for ε close enough to 0, both $x + \varepsilon(\chi^M - \chi^N)$ and $x - \varepsilon(\chi^M - \chi^N)$ satisfy (18.1), contradicting the fact that x is a vertex of the polytope. So (V, F) is a forest, and hence by (18.1), F is a perfect matching. ∎

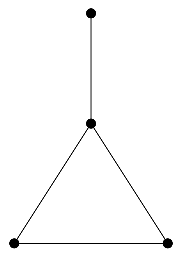


Figure 18.1

The implication cannot be reversed, as is shown by the graph in Figure 18.1.

Theorem 18.1 was shown by Birkhoff in the terminology of doubly stochastic matrices. A matrix A is called *doubly stochastic* if A is nonnegative and each row sum and each column sum equals 1. A *permutation matrix* is an integer doubly stochastic matrix (so it is $\{0,1\}$-valued, and has precisely one 1 in each row and in each column). Then:

Corollary 18.1a (Birkhoff's theorem)**.** *Each doubly stochastic matrix is a convex combination of permutation matrices.*

Proof. Directly from Theorem 18.1, by taking $G = K_{n,n}$. ∎

Theorem 18.1 also implies a characterization of the matching polytope for bipartite graphs. For any graph $G = (V, E)$, the *matching polytope* $P_{\text{matching}}(G)$ of G is the convex hull of the incidence vectors of matchings in G. So again it is a polytope in $\mathbb{R}^E$. The following are valid inequalities for the matching polytope:

(18.2)	(i)	$x_e \geq 0$	for each edge e,
	(ii)	$x(\delta(v)) \leq 1$	for each vertex v.

Then:

Corollary 18.1b. *The matching polytope of G is determined by* (18.2) *if and only if G is bipartite.*

Proof. To see necessity, suppose that G is not bipartite, and let C be an odd circuit in G. Define $x_e := \frac{1}{2}$ if $e \in C$ and $x_e := 0$ otherwise. Then x satisfies (18.2) but does not belong to the matching polytope of G.

To see sufficiency, let G be bipartite and let x satisfy (18.2). Let G' and x' be a copy of G and x, and add edges vv', where v' is the copy of $v \in V$. Define $y(vv') := 1 - x(\delta(v))$. Then x, x', y satisfy (18.1) with respect to the new graph, and hence by Theorem 18.1, it is a convex combination of incidence vectors of perfect matchings in the new graph. Hence x is a convex combination of incidence vectors of matchings in G. ∎

Notes. Birkhoff derived Corollary 18.1a from Hall's marriage theorem (Theorem 22.1), which is equivalent to Kőnig's matching theorem. (Also Dulmage and Halperin [1955] derived Birkhoff's theorem from Kőnig's matching theorem.) Other proofs were given by von Neumann [1951,1953], Dantzig [1952], Hoffman and Wielandt [1953], Koopmans and Beckmann [1955,1957], Hammersley and Mauldon [1956] (a polyhedral proof based on total unimodularity), Tompkins [1956], Mirsky [1958], and Vogel [1961]. A survey was given by Mirsky [1962]. More can be found in Johnson, Dulmage, and Mendelsohn [1960], Nishi [1979], and Brualdi [1982].

18.2. Totally unimodular matrices from bipartite graphs

In this section we show that the results on matchings discussed above can also be derived from linear programming duality with total unimodularity (Hoffman [1956b]).

Let A be the $V \times E$ incidence matrix of a graph $G = (V, E)$. The matrix A generally is not totally unimodular. E.g., if G is the complete graph K_3 on three vertices, then the determinant of A is equal to $+2$ or -2.

However, the following can be proved (necessity can also be derived directly from the total unimodularity of the incidence matrix of a directed graph (Theorem 13.9) — we give a direct proof):

Theorem 18.2. *A graph $G = (V, E)$ is bipartite if and only if its incidence matrix A is totally unimodular.*

Proof. *Sufficiency.* Assume that A is totally unimodular and G is not bipartite. Then G has a circuit of odd length, t say. The submatrix of A induced by the vertices and edges in C is a $t \times t$ matrix with exactly two ones in each row and each column. As t is odd, the determinant of this matrix is ± 2, contradicting the total unimodularity of A.

Necessity. Let G be bipartite. We show that A is totally unimodular. Let B be a square submatrix of A, of order $t \times t$ say. We show that $\det B$ equals 0 or ± 1 by induction on t. If $t = 1$, the statement is trivial. So let $t > 1$. We distinguish three cases.

Case 1: B has a column with only 0*'s.* Then $\det B$=0.

Case 2: B has a column with exactly one 1. In that case we can write (possibly after permuting rows or columns):

$$(18.3) \qquad B = \begin{pmatrix} 1 & b^{\mathsf{T}} \\ \mathbf{0} & B' \end{pmatrix},$$

for some matrix B' and vector b, where $\mathbf{0}$ denotes the all-zero vector in $\mathbb{R}^{t-1}$. By the induction hypothesis, $\det B' \in \{0, \pm 1\}$. Hence, by (18.3), $\det B \in \{0, \pm 1\}$.

Case 3. Each column of B contains exactly two 1's. Then, since G is bipartite, we can write (possibly after permuting rows):

$$B = \begin{pmatrix} B' \\ B'' \end{pmatrix}, \tag{18.4}$$

in such a way that each column of B' contains exactly one 1 and each column of B'' contains exactly one 1. So adding up all rows in B' gives the all-one vector, and also adding up all rows in B'' gives the all-one vector. The rows of B therefore are linearly dependent, and hence $\det B = 0$. ∎

18.3. Consequences of total unimodularity

Let $G = (V, E)$ be a bipartite graph and let A be its $V \times E$ incidence matrix. Consider Kőnig's matching theorem (Theorem 16.2): the maximum size of a matching in G is equal to the minimum size of a vertex cover in G. This can be derived from the total unimodularity of A as follows. By Corollary 5.20a, both optima in the LP-duality equation

$$\max\{\mathbf{1}^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq \mathbf{1}\} = \min\{y^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq \mathbf{1}^{\mathsf{T}}\} \tag{18.5}$$

have integer optimum solutions x^* and y^*. Now x^* necessarily is the incidence vector of a matching and y^* is the incidence vector of a vertex cover. So we have Kőnig's matching theorem.

One can also derive the weighted version of Kőnig's matching theorem, Egerváry's theorem (Theorem 17.1): for any weight function $w : E \to \mathbb{Z}_+$, the maximum weight of a matching in G is equal to the minimum value of $\sum_{v \in V} y_v$, where y ranges over all $y : V \to \mathbb{Z}_+$ with $y_u + y_v \geq w_e$ for each edge $e = uv$ of G. To derive this, consider the LP-duality equation

$$\max\{w^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \leq \mathbf{1}\} = \min\{y^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \geq w^{\mathsf{T}}\}. \tag{18.6}$$

By the total unimodularity of A, these optima are attained by integer x^* and y^*, and we have the theorem.

The min-max relation for minimum-weight *perfect* matching (Theorem 17.5) follows similarly.

One can also derive the characterizations of the matching polytope and perfect matching polytope of a bipartite graph (Theorem 18.1 and Corollary 18.1b) from the total unimodularity of the incidence matrix of a bipartite graph. This amounts to the fact that the polyhedra

$$\{x \mid x \geq \mathbf{0}, Ax \leq \mathbf{1}\} \tag{18.7}$$

and

$$\{x \mid x \geq \mathbf{0}, Ax = \mathbf{1}\} \tag{18.8}$$

are integer polyhedra, by the total unimodularity of A.

18.4. The vertex cover polytope

One can similarly derive, from the total unimodularity, a description of the vertex cover polytope of a bipartite graph. The *vertex cover polytope* of a graph G is the convex hull of the incidence vectors of vertex covers. It is a polytope in $\mathbb{R}^V$.

For bipartite graphs, it is determined by:

(18.9) (i) $0 \leq y_v \leq 1$ for each $v \in V$,
(ii) $y_u + y_v \geq 1$ for each $e = uv \in E$.

In fact, this characterizes bipartiteness:

Theorem 18.3. *A graph G is bipartite if and only if the vertex cover polytope of G is determined by* (18.9).

Proof. Necessity follows from the total unimodularity of the incidence matrix of A (Theorem 18.2). Sufficiency can be seen as follows. Suppose that G contains an odd circuit C. Define $y_v := \frac{1}{2}$ for each $v \in V$. Then y satisfies (18.9) but does not belong to the vertex cover polytope, as each vertex cover contains more than $\frac{1}{2}|VC|$ vertices in C. ∎

The total unimodularity of A also yields descriptions of the edge cover and stable set polytopes of a bipartite graph — see Section 19.5.

18.5. Further results and notes

18.5a. Derivation of Kőnig's matching theorem from the matching polytope

We note here that Kőnig's matching theorem quite easily follows from description (18.2) of the matching polytope of a bipartite graph.

Since the matching polytope of a bipartite graph $G = (V, E)$ is determined by (18.2), the maximum size of a matching in G is equal to the minimum value of $\sum_{v \in V} y_v$ where $y_v \geq 0$ ($v \in V$) such that $y_u + y_v \leq 1$ for each edge $e = uv$.

Now consider any vertex u with $y_u > 0$. Then by complementary slackness, each maximum-size matching covers u. That is, we have (16.5), which (as we saw) directly implies Kőnig's matching theorem, by applying induction to $G - u$.

18.5b. Dual, primal-dual, primal?

The Hungarian method is considered as the first so-called 'primal-dual' method. It maintains a feasible dual solution, and tries to build up a feasible primal solution fulfilling the complementary slackness conditions. We will show that in a certain sense the method can also be considered as just dual or just primal.

We consider the problem of finding a minimum-weight perfect matching in a bipartite graph $G = (V, E)$, with weight function $w : E \to \mathbb{Q}_+$. Let U and W be the colour classes of G, with $|U| = |W|$. The corresponding LP-duality equation is

(18.10) $$\min\{w^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax = \mathbf{1}\} = \max\{y^{\mathsf{T}}\mathbf{1} \mid y^{\mathsf{T}}A \leq w^{\mathsf{T}}\},$$

where A is the $V \times E$ incidence matrix of G.

To describe the Hungarian method as a purely dual method one can start with $y = \mathbf{0}$. So y satisfies

(18.11) $$y_u + y_v \leq w_e$$

for each edge $e = uv$ of G. Consider the subset

(18.12) $$F := \{e = uv \in E \mid y_u + y_v = w_e\}$$

of E. If F contains a perfect matching M, then M is a minimum-weight perfect matching, by complementary slackness applied to (18.10). If F contains no perfect matching, by Frobenius' theorem (Corollary 16.2a) there exist $U' \subseteq U$ and $W' \subseteq W$ such that each edge in F intersecting U' also intersects W' and such that $|W'| < |U'|$. Now we can reset

(18.13) $$y_v := \begin{cases} y_v + \alpha & \text{if } v \in U', \\ y_v - \alpha & \text{if } v \in W', \end{cases}$$

choosing α as large as possible while maintaining (18.11). That is, α is equal to the minimum of $w_e - y_u - y_v$ over all edges $e = uv \in E$ with $u \in U'$ and $v \notin W'$. So $\alpha > 0$, and hence $y^{\mathsf{T}}\mathbf{1}$ increases. After that we iterate.

Described in this way it is a purely *dual method*, since only in the last iteration we see a primal solution. In each iteration we test the existence of a perfect matching from scratch. We could, however, remember our work of the previous iteration in our search for a perfect matching in F.

To this end, we keep at any iteration a maximum-size matching M in F. Let D_M be the directed graph obtained from (V, F) by orienting each edge in M from W to U and each edge in $F \setminus M$ from U to W. Let U_M and W_M be the set of vertices in U and W, respectively, missed by M. We also keep, throughout the iterations, the set R_M of vertices reachable in D_M from U_M.

Then we can take $U' := U \cap R_M$ and $W' := W \cap R_M$. Resetting (18.13) of y increases R_M, since at least one edge connecting U' and $W \setminus W'$ is added to F, while all edges in F that were contained in $U' \cup W'$ remain in F. So after at most n iterations, R_M contains a vertex in W_M, in which case we can augment M.

Described in this way it is a *primal-dual method*. Throughout the iterations we keep a feasible dual solution y and a partially feasible primal solution M.

We could however combine all updates of y, between any two augmentations of M, by taking $l_e := w_e - y_u - y_v$ as a length function, and by determining, for each vertex v, the distance $d(v)$ from v to W_M in D_M with respect to length function l. Resetting

(18.14) $$y_v := \begin{cases} y_v + d(v) & \text{if } v \in U, \\ y_v - d(v) & \text{if } v \in W, \end{cases}$$

maintains (18.10), while the new F contains an M-augmenting path (namely, any shortest $U_M - W_M$ path in D_M). Note that this updating of y is the same as the aggregated updating of y (in (18.13)) between any two matching augmentations.

This still is a primal-dual method, since we keep sequences of vectors y and matchings M. It enables us to apply Dijkstra's method to find the distances and the shortest path, since the length function l is nonnegative. We can however do without y, at the cost of an increase in the complexity, since we then must use the Bellman-Ford method (like in our description in Section 17.2). We can use this method since D_M has no negative-length directed circuit, because M is an extreme matching (that is, a matching of minimum weight among all matchings M' with $|M'| = |M|$).

Indeed, we can define the length function l by $l_e := w_e$ if $e \in E \setminus M$ and $l_e := -w_e$ if $e \in M$. Then D_M has no negative-length directed circuits. Any shortest $U_M - W_M$ path is an M-augmenting path yielding an extreme matching M' with $|M'| = |M| + 1$.

Described in this way we have a purely *primal method*, since we keep no vector $y \in \mathbb{Q}^V$ anymore.

18.5c. Adjacency and diameter of the matching polytope

Clearly, for each perfect matching M, the incidence vector χ^M is a vertex of the perfect matching polytope. Adjacency is also easily characterized (Balinski and Russakoff [1974]):

Theorem 18.4. *Let M and N be perfect matchings in a graph $G = (V, E)$. Then χ^M and χ^N are adjacent vertices of the perfect matching polytope if and only if $M \triangle N$ is a circuit.*

Proof. To see necessity, let χ^M and χ^N be adjacent. Then $M \triangle N$ is the vertex-disjoint union of circuits $C_1, \ldots, C_k$. If $k = 1$ we are done so assume $k \geq 2$. Let $M' := M \triangle C_1$ and $N' := N \triangle C_1$. Then $\frac{1}{2}(\chi^M + \chi^N) = \frac{1}{2}(\chi^{M'} + \chi^{N'})$. This contradicts the adjacency of χ^M and χ^N.

To see sufficiency, define a weight function $w : E \to \mathbb{R}$ by $w_e := 0$ if $e \in M \cup N$ and $w_e := 1$ otherwise. Then M and N are the only two perfect matchings in G of minimum weight. Hence χ^M and χ^N are adjacent. ∎

This gives for the diameter:

Corollary 18.4a. *The perfect matching polytope of a graph $G = (V, E)$ has diameter at most $\frac{1}{2}|V|$. If G is simple, the diameter is at most $\frac{1}{4}|V|$.*

Proof. Let M and N be perfect matchings of G. Let $M \triangle N$ be the vertex-disjoint union of circuits $C_1, \ldots, C_k$. Define $M_i := M \triangle (C_1 \cup \cdots \cup C_i)$, for $i = 0, \ldots, k$. Then $M = M_0$, $N = M_k$, and M_i and M_{i+1} give adjacent vertices of the perfect matching polytope of G (by Theorem 18.4). As each C_i has at least two vertices, we have $k \leq \frac{1}{2}|V|$. If G is simple, each C_i has at least four vertices, and hence $k \leq \frac{1}{4}|V|$. ∎

For complete bipartite graphs, this bound can be strengthened. The *assignment polytope* is the perfect matching polytope of a complete bipartite graph $K_{n,n}$. So in matrix terms, it is the polytope of the $n \times n$ doubly stochastic matrices. Balinski and Russakoff [1974] showed:

Theorem 18.5. *The diameter of the assignment polytope is* 2 *(if $n \geq 4$).*

Proof. Let U and W be the two colour classes of $K_{n,n}$. Let M and N be two distinct perfect matchings in $K_{n,n}$. Assume that $M \neq N$ and that M and N are not adjacent. Let $M \triangle N$ be the vertex-disjoint union of the circuits $C_1, \ldots, C_k$. As M and N are not adjacent, $k \geq 2$. For each $i = 1, \ldots, k$, choose an edge $u_i w_i \in C_i \cap M$, with $u_i \in U$ and $w_i \in W$. Let C be the circuit

(18.15) $$C := \{u_1w_1, u_2w_1, u_2w_2, u_3w_2, \ldots, u_nw_n, u_1w_n\}$$

and let $L := M \triangle C$. As $M \triangle L = C$, L is a perfect matching adjacent to M. Now L is adjacent also to N as well, since $N \triangle L = (C_1 \cup \cdots \cup C_k) \triangle C$, which is a circuit. ∎

Naddef [1982] characterized the dimension of the perfect matching polytope of a bipartite graph (cf. Lovász and Plummer [1986]):

Theorem 18.6. *Let $G = (V, E)$ be a bipartite graph with at least one perfect matching. Then the dimension of the perfect matching polytope of G is equal to $|E_0| - |V| + k$, where E_0 is the set of edges contained in at least one perfect matching and where k is the number of components of the graph (V, E_0).*

Proof. It is easy to see that we may assume that $E_0 = E$ and that G is connected and has at least four vertices. Let T be the edge set of a spanning tree in G. So $|E \setminus T| = |E| - |V| + 1$. Now for any $x \in P_{\text{perfect matching}}(G)$, the values x_e with $e \in T$ are determined by the values x_e with $e \in E \setminus T$. Hence $\dim(P_{\text{perfect matching}}(G)) \leq |E \setminus T| = |E| - |V| + 1$.

To see the reverse inequality, choose a vector x in the relative interior of $P_{\text{perfect matching}}(G)$. So $0 < x_e < 1$ for each $e \in E$ (as each edge is contained in some perfect matching and is missed by some perfect matching). Then any small enough change of x_e for any $e \in E \setminus T$ can be corrected by changing values of $x(e')$ with $e' \in T$. Therefore $\dim(P_{\text{perfect matching}}(G)) \geq |E \setminus T|$. ∎

Rispoli [1992] showed that the 'monotonic diameter' (that is, the maximum length of a shortest path on the polytope where a given objective function is monotonically increasing) of the assignment polytope is equal to $\lfloor \frac{n}{2} \rfloor$. More can be found in Balinski and Russakoff [1974], Padberg and Rao [1974], Brualdi and Gibson [1976,1977a,1977b,1977c], Roohy-Laleh [1980], Hung [1983], Balinski [1985], and Goldfarb [1985].

18.5d. The perfect matching space of a bipartite graph

The *perfect matching space* of a graph $G = (V, E)$ is the linear hull of the incidence vectors of perfect matchings:

(18.16) $$S_{\text{perfect matching}}(G) := \text{lin.hull}\{\chi^M \mid M \text{ perfect matching in } G\}.$$

(Here lin.hull denotes linear hull.)

Note that Theorem 18.6 directly implies the dimension of the perfect matching space of a bipartite graph:

Corollary 18.6a. *Let $G = (V, E)$ be a bipartite graph with at least one perfect matching. Then the dimension of the perfect matching space of G is equal to $|E_0| -$*

$|V|+k+1$, where E_0 is the set of edges contained in at least one of perfect matching, and where k is the number of components of the graph (V, E_0).

Proof. The dimension of the perfect matching space is 1 more than the dimension of the perfect matching polytope (as $\mathbf{0}$ does not belong to the affine hull of the incidence vectors of perfect matchings). So the Corollary follows from Theorem 18.6. ∎

With the help of the description of the perfect matching polytope we can similarly describe the perfect matching space in terms of equations:

Theorem 18.7. *The perfect matching space of a bipartite graph $G = (V, E)$ is equal to the set of vectors $x \in \mathbb{R}^E$ such that*

(18.17) (i) $x_e = 0$ if e is contained in no perfect matching,
(ii) $x(\delta(u)) = x(\delta(v))$ for all $u, v \in V$.

Proof. (18.17) clearly is a necessary condition for each vector x in the perfect matching space. To see sufficiency, let $x \in \mathbb{R}^E$ satisfy (18.17). We can assume that G has at least one perfect matching.

By adding sufficiently many incidence vectors of perfect matchings to x, we can achieve that $x_e \geq 0$ for all $e \in E$. By scaling we can achieve that $x(\delta(v)) = 1$ for each $v \in V$. Then x belongs to the perfect matching polytope of G, and hence to the perfect matching space. ∎

This theorem has as direct consequence a characterization of the linear space orthogonal to the perfect matching space:

Corollary 18.7a. *Let $G = (V, E)$ be a bipartite graph and let $w \in \mathbb{R}^E$. Then $w(M) = 0$ for each perfect matching M if and only if there exists a vector $b \in \mathbb{R}^V$ with $b(V) = 0$ such that $w_e = b_u + b_v$ for each edge $e = uv$ contained in at least one perfect matching.*

Proof. Directly by orthogonality from Theorem 18.7. ∎

18.5e. Up and down hull of the perfect matching polytope

Fulkerson [1970b] studied the up hull of the perfect matching polytope of a graph $G = (V, E)$, that is,

(18.18) $$P^{\uparrow}_{\text{perfect matching}}(G) = P_{\text{perfect matching}}(G) + \mathbb{R}^E_+.$$

Any x in this polyhedron satisfies:

(18.19) (i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(E[S]) \geq |S| - \frac{1}{2}|V|$ for each $S \subseteq V$.

Here $E[S]$ denotes the set of edges spanned by S. Inequality (18.19)(ii) follows from the fact that any perfect matching M has at most $|V \setminus S|$ edges not contained in S, and hence at least $\frac{1}{2}|V| - |V \setminus S| = |S| - \frac{1}{2}|V|$ edges contained in S.

Fulkerson [1970b] showed that for bipartite graphs these inequalities are enough to characterize polyhedron (18.18):

Theorem 18.8. *If G is bipartite, then $P^{\uparrow}_{\text{perfect matching}}(G)$ is determined by* (18.19).

Proof. Let U and W be the colour classes of G. Let $x \in \mathbb{R}^E$ satisfy (18.19). Note that this implies that $|U| = |W| = \frac{1}{2}|V|$, for if (say) $|U| > \frac{1}{2}|V|$, then (18.19) implies that $0 = x(E[U]) \geq |U| - \frac{1}{2}|V| > 0$, a contradiction.

We must show that there exists a vector y such that $\mathbf{0} \leq y \leq x$ and such that $y(\delta(v)) = 1$ for each $v \in V$. This can be shown quite directly with flow theory, for instance with Gale's theorem (Corollary 11.2g): Make a directed graph by orienting each edge from U to W. Then by Gale's theorem (taking $b(v) := -1$ if $v \in U$ and $b(v) := 1$ if $v \in W$), it suffices to show that $|W'| - |U'| \leq x(\delta^{\text{in}}(U' \cup W'))$ for each $U' \subseteq U$ and $W' \subseteq W$. Let $S := (U \setminus U') \cup W'$. Then $\delta^{\text{in}}(U' \cup W') = E[S]$ and $|W'| - |U'| = |S| - \frac{1}{2}|V|$, giving the required inequality. ∎

(Fulkerson [1970b] derived Theorem 18.8 from an earlier result in Fulkerson [1964b], which is Corollary 20.9a below. Related results were given by O'Neil [1971,1975], Cruse [1975], and Houck and Pittenger [1979].)

Note that the theorem gives also a characterization of the convex hull of the incidence vectors of edge sets containing a perfect matching in a bipartite graph:

Corollary 18.8a. *Let $G = (V, E)$ be a bipartite graph. Then the convex hull of the incidence vectors of edge sets containing a perfect matching is determined by* (18.19) *together with $x_e \leq 1$ for each $e \in E$.*

Proof. Directly from Theorem 18.8. ∎

One can similarly characterize the convex hull of the incidence vectors of subsets of perfect matchings in a bipartite graph. Consider:

(18.20)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(E[S]) \leq |S| - \frac{1}{2}|V|$ for each vertex cover S.

Theorem 18.9. *The convex hull of the incidence vectors of subsets of perfect matchings in a bipartite graph is determined by* (18.20).

Proof. Similar to the proof of Theorem 18.8. ∎

(Alternative proofs of Theorems 18.8 and 18.9 were given by Cunningham and Green-Krótki [1986].)

See Section 20.6a for more results on $P^{\uparrow}_{\text{perfect matching}}(G)$.

18.5f. Matchings of given size

Let $G = (V, E)$ be a graph and let $k, l \in \mathbb{Z}_+$ with $k \leq l$. It is easy to derive from the description of the matching polytope, a description of the convex hull of incidence vectors of matchings M satisfying $k \leq |M| \leq l$. To this end we show:

Theorem 18.10. *Let $G = (V, E)$ be an undirected graph and let $x \in P_{\text{matching}}(G)$. Then x is a convex combination of incidence vectors of matchings M satisfying*

$$(18.21) \qquad \lfloor \mathbf{1}^{\mathsf{T}} x \rfloor \leq |M| \leq \lceil \mathbf{1}^{\mathsf{T}} x \rceil.$$

Proof. Write $x = \sum_M \lambda_M \chi^M$, where M ranges over all matchings in G and where $\lambda_M \geq 0$ with $\sum_M \lambda_M = 1$. Assume that we have chosen the λ_M such that

$$(18.22) \qquad \sum_M \lambda_M |M|^2$$

is as small as possible. We show that if M and N are matchings with $\lambda_M > 0$ and $\lambda_N > 0$, then $\big||M| - |N|\big| \leq 1$. This implies the theorem.

Suppose that $|M| \geq |N| + 2$. Let P be a component of $M \cup N$ having more elements in M than in N. Let $M' := M \triangle EP$ and $N' := N \triangle EP$. Then $\chi^{M'} + \chi^{N'} = \chi^M + \chi^N$ and $|M'|^2 + |N'|^2 < |M|^2 + |N|^2$. So decreasing λ_M and λ_N by ε, and increasing $\lambda_{M'}$ and $\lambda_{N'}$ by ε, where $\varepsilon := \min\{\lambda_M, \lambda_N\}$, would decrease sum (18.22), contradicting our assumption. ∎

This implies that certain slices of the matching polytope are again integer polytopes:

Corollary 18.10a. *Let $G = (V, E)$ be an undirected graph and let $k, l \in \mathbb{Z}_+$ with $k \leq l$. Then the convex hull of the incidence vectors of matchings M satisfying $k \leq |M| \leq l$ is equal to the set of those vectors x in the matching polytope of G satisfying $k \leq \mathbf{1}^{\mathsf{T}} x \leq l$.*

Proof. Directly from Theorem 18.10. ∎

A special case is the following result of Mendelsohn and Dulmage [1958b]. Call a matrix a *subpermutation matrix* if it is a $\{0, 1\}$-valued matrix with at most one 1 in each row and in each column. Then:

Corollary 18.10b. *A matrix M belongs to the convex hull of the subpermutation matrices of rank r if and only if M is nonnegative, each row and column sum is at most 1, and the sum of the entries in M is equal to r.*

Proof. Directly from Theorem 18.10. ∎

18.5g. Stable matchings

Let $G = (V, E)$ be a graph and let for each $v \in V$, $\leq_v$ be a total order on $\delta(v)$. Put $e \preceq f$ if e and f have a vertex v in common with $e \leq_v f$. Call a set M of edges *stable* if for each $e \in E$ there exists an $f \in M$ with $e \preceq f$.

In general, stable matchings need not exist (e.g., generally not for K_3). However, Gale and Shapley [1962] showed that if G is bipartite, they do exist:

Theorem 18.11 (Gale-Shapley theorem)**.** *If G is bipartite, then there exists a stable matching.*

Proof. Let U and W be the colour classes of G. For each edge $e = uw$ with $u \in U$ and $w \in W$, let $\phi(e)$ be the height of e in $(\delta(w), \leq_w)$. (The *height* of e is the maximum size of a chain with maximum e.) Choose a matching M in G such that for each edge $e = uw$ of G, with $u \in U$ and $w \in W$,

(18.23) if $f \leq_u e$ for some $f \in M$, then $e \leq_w g$ for some $g \in M$,

and such that $\sum_{e \in M} \phi(e)$ is as large as possible. (Such a matching exists, since $M = \emptyset$ satisfies (18.23).) We show that M is stable.

Choose $e = uw \in E$ with $u \in U$ and $w \in W$ and suppose that there is no $e' \in M$ with $e \preceq e'$. Choose e largest in $\leq_u$ with this property. Then by (18.23) there is no $f \in M$ with $f \leq_u e$; and moreover, there is no $f \in M$ with $e \leq_u f$. Hence u is missed by M.

Since also there is no $g \in M$ with $e \leq_w g$, we can remove any edge in M incident with w and add e to M, so as to obtain a matching satisfying (18.23) with larger $\sum_{e \in M} \phi(e)$, a contradiction. ∎

This proof also gives a polynomial-time algorithm to find a stable matching[34].

The following fact was shown by McVitie and Wilson [1970]:

Theorem 18.12. *Each two stable matchings cover the same set of vertices.*

Proof. Let M and N be two stable matchings, and suppose that there exists a vertex v covered by M but not by N. Let P be the path component of $M \cup N$ starting at v. Denote $P = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$ with $v = v_0$. As v_0 is missed by N, $e_1 <_{v_1} e_2$. As M and N are stable, if $e_{i-1} <_{v_{i-1}} e_i$, then $e_i <_{v_i} e_{i+1}$ for each $i < k$. So $e_{k-1} <_{v_{k-1}} e_k$. However, as v_k is missed by M or N, $e_k <_{v_{k-1}} e_{k-1}$. So we have a contradiction. ∎

In particular:

Corollary 18.12a. *All stable matchings have the same size.*

Proof. Directly from Theorem 18.12. ∎

In order to find a maximum-*weight* stable matching, we consider the *stable matching polytope* $P_{\text{stable matching}}(G)$ of G, which is defined as the convex hull of the incidence vectors of the stable matchings. Vande Vate [1989] (also Rothblum [1992]) characterized the inequalities determining the stable matching polytope if G is bipartite. In that case it suffices to add the following inequalities to the system defining the matching polytope:

(18.24) $$\sum_{f \succeq e} x(f) \geq 1 \text{ for each } e \in E.$$

Theorem 18.13. *If G is bipartite, then $x \in P_{\text{stable matching}}(G)$ if and only if $x \in P_{\text{matching}}(G)$ and x satisfies* (18.24).

[34] It was noted by Roth [1984] that this algorithm is in fact in use in practice since 1951 in the U.S., to match hospitals and medical students (cf. Roth and Sotomayor [1990]).

Proof. Necessity is easy, since the incidence vector of any stable matching satisfies (18.24). To see sufficiency, let x be a vertex of the polytope of all vectors in $P_{\text{matching}}(G)$ satisfying (18.24). Define E^+ to be the set of edges e with $x_e > 0$, and V^+ the set of vertices covered by E^+. For each $v \in V^+$, let e_v be the maximum element of $(\delta(v) \cap E^+, \leq_v)$.

We first show that for each $v \in V^+$, with say $e_v = vv'$,

(18.25) e_v is the minimum element in $(\delta(v') \cap E^+, \leq_{v'})$ and that $x(\delta(v')) = 1$.

Indeed, (18.24) implies (writing $e := e_v$):

(18.26) $$1 \leq \sum_{f \succeq e} x(f) = \sum_{f \geq_{v'} e} x(f) = x(\delta(v')) - \sum_{f <_{v'} e} x(f) \leq 1 - \sum_{f <_{v'} e} x(f).$$

Hence we have equality throughout in (18.26). This implies that $x(f) = 0$ for each $f <_{v'} e$ and that $x(\delta(v')) = 1$. This proves (18.25).

It follows that for each $v' \in V^+$ there is exactly one $v \in V^+$ with $e_v = vv'$. Now let U and W be the colour classes of G. The sets $M := \{e_v \mid v \in U \cap V^+\}$ and $N := \{e_v \mid v \in W \cap V^+\}$ are matchings covering V^+. Consider the vector $x' = x + \varepsilon\chi^M - \varepsilon\chi^N$, with ε close enough to 0 (positive or negative). It is easy to see that x' again belongs to the matching polytope. To see that x' satisfies (18.24) for ε close enough to 0, let e be an edge of G attaining equality in (18.24). We show that $e \preceq f$ for exactly one $f \in M$. If $e \in M$, this is trivial, so assume that $e \notin M$. Let $e = uw$ with $u \in U$ and $w \in W$. Then

(18.27) $$\text{there is an } f \in M \text{ with } e <_u f \iff \sum_{f >_u e} x(f) > 0 \iff \sum_{g \geq_w e} x(g) < 1$$
$$\iff \text{ there is no } g \in M \text{ with } e <_w g.$$

Similarly, $e \preceq f$ for exactly one $f \in N$. Concluding,

(18.28) $$\sum_{f \succeq e} x'(f) = \sum_{f \succeq e} x(f) = 1$$

if ε is close enough to 0. So x' again satisfies (18.24). Since x is a vertex, we have $\chi^M = \chi^N$, that is, $M = N$. So $E^+ = M$, and hence $x = \chi^M$, and therefore x is $\{0,1\}$-valued. ∎

As for algorithms, this theorem directly implies:

Corollary 18.13a. *A maximum-weight stable matching can be found in polynomial time.*

Proof. This follows from the fact that Theorem 18.13 transforms the problem to a linear programming problem. ∎

For surveys and further results, see Wilson [1972a], Knuth [1976], Itoga [1978, 1981], Roth [1982], Gale and Sotomayor [1985], Irving [1985], Gusfield [1987b,1988], Irving, Leather, and Gusfield [1987], Blair [1988], Gusfield and Irving [1989], Ng [1989], Knuth, Motwani, and Pittel [1990a,1990b], Ng and Hirschberg [1990], Ronn [1990], Roth and Sotomayor [1990], Khuller, Mitchell, and Vazirani [1991,1994], Tan [1991], Feder [1992], Roth, Rothblum, and Vande Vate [1993], Abeledo and Rothblum [1994], Feder, Megiddo, and Plotkin [1994,2000], Subramanian [1994], Abeledo and Blum [1996], Balinski and Ratier [1997], Teo and Sethuraman [1997, 1998], Teo, Sethuraman, and Tan [1999], Fleiner [2001a], and Aharoni and Fleiner [2002].

18.5h. Further notes

Perfect and Mirsky [1965] characterized which patterns can occur as the support of a doubly stochastic matrix. It is equivalent to characterizing matching-covered bipartite graphs (that is, bipartite graphs in which each edge belongs to at least one perfect matching).

Frank and Karzanov [1992] gave a polynomial-time combinatorial algorithm to determine the Euclidean distance of the perfect matching polytope of a bipartite graph to the origin.

Chapter 19

Bipartite edge cover and stable set

While matchings cover each vertex *at most* once, edge covers are required to cover each vertex *at least* once. Most edge cover results can be proved similarly to matching results, but in fact, they often can be reduced to matching results, by a method of Gallai.

In this chapter, graphs can be assumed to be simple.

19.1. Matchings, edge covers, and Gallai's theorem

Let $G = (V, E)$ be a graph. An *edge cover* is a subset F of E such that for each vertex v there exists an edge $e \in F$ satisfying $v \in e$. Note that an edge cover can exist only if G has no isolated vertices.

A *stable set* is a subset S of V such that no two vertices in S are adjacent. So for any $U \subseteq V$:

(19.1) $\quad S$ is a stable set $\iff$ $V \setminus S$ is a vertex cover.

Define:

(19.2) $\quad \alpha(G) :=$ the maximum size of a stable set in G,
$\rho(G) :=$ the minimum size of an edge cover in G.

These numbers are called the *stable set number* and the *edge cover number*, respectively.

It is not difficult to show that:

(19.3) $\quad \alpha(G) \leq \rho(G)$.

The triangle K_3 shows that strict inequality is possible. Recall that for the matching number $\nu(G)$ and the vertex cover number $\tau(G)$ we have

(19.4) $\quad \nu(G) \leq \tau(G)$.

In fact, equality in one of the relations (19.3) and (19.4) implies equality in the other, as Gallai [1959a] proved the following[35]:

[35] Gallai mentioned that he had formulated and proved this theorem in 1932 (cf. also Erdős [1982]), and that to his knowledge also D. Kőnig had known this theorem.

Theorem 19.1 (Gallai's theorem)**.** *For any graph $G = (V, E)$ without isolated vertices one has*

(19.5) $$\alpha(G) + \tau(G) = |V| = \nu(G) + \rho(G).$$

Proof. The first equality follows directly from (19.1).

To see the second equality, let M be a maximum-size matching and let U be the set of vertices missed by M. For each vertex $v \in U$, choose an edge e_v containing v. Then $F = M \cup \{e_v \mid v \in U\}$ is an edge cover of size

(19.6) $$|F| = |M| + |U| = |M| + (|V| - 2|M|) = |V| - |M| = |V| - \nu(G).$$

So $\rho(G) \leq |V| - \nu(G)$.

To see the reverse inequality, let F be a minimum-size edge cover. Let M be an inclusionwise maximal matching contained in F. Let U be the set of vertices missed by M. Since U spans no edge in F, we have $|U| \leq |F \setminus M|$. Hence $|V| - 2|M| = |U| \leq |F \setminus M| = |F| - |M|$. This implies $\nu(G) \geq |M| \geq |V| - |F| = |V| - \rho(G)$. ∎

This proof method implies the following theorem (observed by Gallai [1959a] and Norman and Rabin [1959]):

Theorem 19.2. *Let $G = (V, E)$ be a graph without isolated vertices. Then every maximum-size matching is contained in a minimum-size edge cover, and every minimum-size edge cover contains a maximum-size matching.*

Proof. See above. ∎

Moreover, there is the following complexity result, observed by Norman and Rabin [1959]:

Theorem 19.3. *Let $G = (V, E)$ be an undirected graph with n vertices and m edges. If we have a maximum-size matching in G, we can find a minimum-size edge cover in time $O(m)$, and vice versa.*

Proof. See the proof of Gallai's theorem (Theorem 19.1). ∎

This gives:

Corollary 19.3a. *A minimum-size edge cover and a maximum-size stable set in a bipartite graph can be found in time $O(n^{1/2}m)$.*

Proof. By Theorems 16.4 and 19.3 and Corollary 16.6a. ∎

Short proof of Gallai's theorem. For any partition Π of V into edges and singletons, let $f(\Pi)$ be the number of edges in Π. So $f(\Pi) + |\Pi| = |V|$. Then $\nu(G)$ is equal to the maximum of $f(\Pi)$ over all such partitions, and $\rho(G)$ is equal to the minimum of $|\Pi|$ over all such partitions. Hence $\nu(G) + \rho(G) = |V|$.

19.2. The Kőnig-Rado edge cover theorem

Combination of Theorems 19.1 and 16.2 yields the following theorem, which Gallai [1958a,1958b] attributes to oral communication from D. Kőnig in 1932. In a different but equivalent form it was stated by Rado [1933] — see Section 19.5a. (Hoffman [1956b] called it a 'well-known theorem'.)

Theorem 19.4 (Kőnig-Rado edge cover theorem)**.** *For any bipartite graph $G = (V, E)$ without isolated vertices one has*

(19.7) $\qquad \alpha(G) = \rho(G).$

That is, the maximum size of a stable set in a bipartite graph is equal to the minimum size of an edge cover.

Proof. Directly from Theorems 19.1 and 16.2, as $\alpha(G) = |V| - \tau(G) = |V| - \nu(G) = \rho(G)$. ∎

By representing a bipartite graph as a partially ordered set, the Kőnig-Rado edge cover theorem can be derived also from Dilworth's decomposition theorem (Theorem 14.2).

19.3. Finding a minimum-weight edge cover

There is a straightforward reduction of the minimum-weight edge cover problem to the minimum-weight perfect matching problem. Indeed, let $G = (V, E)$ be a graph without isolated vertices, and let $w : E \to \mathbb{Q}_+$. Let $G' = (V', E')$ be the graph obtained from G by adding a disjoint copy $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ of G, and adding for each vertex v of G an edge $v\tilde{v}$ connecting v with its copy $\tilde{v}$. Let w' be the weight function on E' defined by:

(19.8) $\qquad w'(e) := w'(\tilde{e}) := w(e)$ for each $e \in E$ (where $\tilde{e}$ is the copy of e);
$\qquad w'(v\tilde{v}) := 2\mu(v)$ for each $v \in V$, where $\mu(v)$ is the minimum weight of the edges of G incident with v.

Then a minimum-weight perfect matching M in G' yields a minimum-weight edge cover F in G: replace any edge $v\tilde{v}$ in M by an edge e_v of minimum weight of G incident with v, and delete all edges in $M \cap \widetilde{E}$. Then $w(F) = \frac{1}{2}w'(M)$. Conversely, any edge cover F' of G gives by a reverse construction a perfect matching M' in G' with $w'(M') \leq 2w(F')$. Hence $w(F) = \frac{1}{2}w'(M) \leq \frac{1}{2}w'(M') \leq w(F')$. So F is a minimum-weight edge cover in G.

Note that if G is bipartite, then also G' is bipartite. Hence:

Corollary 19.4a. *A minimum-weight edge cover in a bipartite graph can be found in time $O(n(m + n\log n))$.*

Proof. From the above, using Theorem 17.3. ∎

19.4. Bipartite edge covers and total unimodularity

Similarly to Kőnig's matching theorem, also the Kőnig-Rado edge cover theorem (Theorem 19.4) can be derived from the total unimodularity of the $V \times E$ incidence matrix of a bipartite graph $G = (V, E)$. This follows by considering the LP-duality equation

(19.9) $$\min\{\mathbf{1}^{\mathsf{T}} x \mid x \geq \mathbf{0}, Ax \geq \mathbf{1}\} = \max\{y^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, y^{\mathsf{T}} A \leq \mathbf{1}^{\mathsf{T}}\}.$$

More generally, we can derive the analogue of Egerváry's theorem:

Theorem 19.5. *Let $G = (V, E)$ be a bipartite graph and let $w : E \to \mathbb{R}_+$ be a weight function on E. Then the minimum weight of an edge cover in G is equal to the maximum value of $y(V)$, where y ranges over all functions $y : V \to \mathbb{R}_+$ with $y_u + y_v \leq w_e$ for each edge $e = uv$ of G. If w is integer, we can restrict y to be integer.*

Proof. Again, let A be the $V \times E$ incidence matrix of G. Then the statement is equivalent to the statement that the minimum in

(19.10) $$\min\{w^{\mathsf{T}} x \mid x \geq \mathbf{0}, Ax \geq \mathbf{1}\} = \max\{y^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, y^{\mathsf{T}} A \leq w^{\mathsf{T}}\}$$

has an integer optimum solution x. This fact follows from the total unimodularity of A. If w is integer, we can take also y integer. ∎

The integer part of this theorem can be formulated as follows. For any graph $G = (V, E)$ and $w \in \mathbb{Z}_+^E$, a *w-stable set* is a function $y \in \mathbb{Z}_+^V$ with $y_u + y_v \leq w_e$ for each edge $e = uv$. So if $w = \mathbf{1}$ and G has no isolated vertices, w-stable sets coincide with the incidence vectors of stable sets.

The *size* of a vector $y \in \mathbb{R}^V$ is equal to $y(V)$. Then:

Corollary 19.5a. *Let $G = (V, E)$ be a bipartite graph and let $w : E \to \mathbb{Z}_+$ be a weight function on E. Then the minimum weight of an edge cover in G is equal to the maximum size of a w-stable set.*

Proof. Directly from Theorem 19.5. ∎

19.5. The edge cover and stable set polytope

Like in Sections 18.3 and 18.4, the total unimodularity of the incidence matrix of a bipartite graph yields descriptions of the edge cover and the stable set polytope for bipartite graphs.

The *edge cover polytope* $P_{\text{edge cover}}(G)$ of a graph is the convex hull of the incidence vectors of the edge covers in G. For any graph, each vector x in $P_{\text{edge cover}}(G)$ satisfies:

(19.11)
(i) $0 \leq x_e \leq 1$ for each $e \in E$,
(ii) $x(\delta(v)) \geq 1$ for each $v \in V$.

Theorem 19.6. *If G is bipartite, the edge cover polytope is determined by* (19.11).

Proof. Directly from the total unimodularity of the constraint matrix in (19.11). ∎

This implication cannot be turned around, as is shown by the graph in Figure 18.1.

The *stable set polytope* $P_{\text{stable set}}(G)$ of a graph $G = (V, E)$ is the convex hull of the incidence vectors of the stable sets in G. For any graph G, each vector x in $P_{\text{stable set}}(G)$ satisfies:

$$\text{(19.12)} \qquad \begin{array}{rll} \text{(i)} & 0 \leq x_v \leq 1 & \text{for each } v \in V, \\ \text{(ii)} & x_u + x_v \leq 1 & \text{for each edge } e = uv \in E. \end{array}$$

Theorem 19.7. *The stable set polytope is determined by* (19.12) *if and only if G is bipartite.*

Proof. Sufficiency follows from the total unimodularity of the incidence matrix of a bipartite graph. Necessity follows from the fact that if C is an odd circuit in G, then defining $x_v := \frac{1}{2}$ for each $v \in V$, we obtain a vector x satisfying (19.12) but not belonging to the stable set polytope of G, since any stable set intersects C in at most $\frac{1}{2}|VC| - \frac{1}{2}$ vertices. ∎

In fact, there is an easy direct proof of sufficiency in Theorem 19.7. Let x satisfy (19.12) and let U and W be the colour classes of G. For any $\lambda \in [0, 1]$, define

$$\text{(19.13)} \qquad S_\lambda := \{u \in U \mid x_u > \lambda\} \cup \{w \in W \mid x_w > 1 - \lambda\}.$$

Then S_λ is a stable set, and

$$\text{(19.14)} \qquad x = \int_0^1 \chi^{S_\lambda} d\lambda.$$

This describes x as a convex combination of incidence vectors of stable sets.

19.5a. Some historical notes on bipartite edge covers

Gallai [1958a,1958b,1959a] wrote that the edge cover theorem (Theorem 19.4) was orally communicated to him by Kőnig in 1932. In the latter paper, Gallai also mentioned that he found Theorem 19.1 in 1932, and that, to his knowledge, also D. Kőnig knew this theorem. Together with Theorem 16.2 of Kőnig [1931] it implies Theorem 19.4.

The oldest written version of Theorem 19.4 seems to be the paper of Rado [1933] entitled *Bemerkungen zur Kombinatorik im Anschluß an Untersuchungen von Herrn D. Kőnig*[36]. The investigations referred to in the title are those of Kőnig [1916] on matchings in *regular* bipartite graphs.

[36] Remarks on combinatorics in connection to investigations of Mr D. Kőnig.

Rado formulated the edge cover theorem in terms of partitions:

> *Es seien $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$ endlich viele nicht leere, paarweise elementenfremde Mengen. Ebenso seien $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$ endlich viele nicht leere, paarweise elementenfremde Mengen. Alle Mengen $\mathcal{A}_\mu$* und *$\mathcal{B}_\nu$ seien Teilmengen einer Menge $\mathcal{M}$. Unter dieser Annahme gilt: Dann und nur dann sind die Mengen*
>
> (26) $\qquad \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m, \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$
>
> *durch k Elemente von $\mathcal{M}$ zu repräsentieren, wenn es unter den Mengen* (26) *keine $k+1$ zu einander fremde Mengen gibt.*[37]

The proof of Rado is based on a decomposition similar to that used by Frobenius (see Section 16.2a). The equivalence with Theorem 19.4 follows with the construction described in Section 16.7e. (A theorem similar to Rado's was published by Kreweras [1946].)

[37] *Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$ be finitely many nonempty, pairwise disjoint sets. Similarly, let $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$ be finitely many nonempty, pairwise disjoint sets. All sets $\mathcal{A}_\mu$* and *$\mathcal{B}_\nu$ are subsets of a set $\mathcal{M}$. Under this condition the following holds: The sets*

(26) $\qquad \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m, \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$

can be represented by k elements of $\mathcal{M}$, if and only if there are no $k+1$ disjoint sets among the sets (26).

Chapter 20

Bipartite edge-colouring

Edge-colouring means partitioning the edge set into matchings. While for general graphs, finding a minimum edge-colouring is NP-complete, another fundamental theorem of Kőnig gives a min-max relation for bipartite edge-colouring, and his proof method yields a polynomial-time algorithm. Also the capacitated case and the 'dual' problem of partitioning the edge set into edge covers are tractable for bipartite graphs.

20.1. Edge-colourings of bipartite graphs

For any graph $G = (V, E)$, an *edge-colouring* or *k-edge-colouring* is a partition $\Pi = (M_1, \ldots, M_k)$ of the edge set E into matchings. Each of the M_i is called a *colour*. If $e \in M_i$ we say that e *has* colour i.

The *edge-colouring number* $\chi(G)$ of G is the minimum number of colours in an edge-colouring of G.

Let $\Delta(G)$ denote the maximum degree of (the vertices of) G. Clearly,

(20.1) $\qquad \chi(G) \geq \Delta(G)$,

since at each vertex v, the edges incident with v should have different colours. The triangle K_3 has strict inequality in (20.1). Kőnig [1916] showed that for bipartite graphs the two numbers are equal:

Theorem 20.1 (Kőnig's edge-colouring theorem)**.** *For any bipartite graph $G = (V, E)$,*

(20.2) $\qquad \chi(G) = \Delta(G)$.

That is, the edge-colouring number of a bipartite graph is equal to its maximum degree.

Proof. Let $M_1, \ldots, M_{\Delta(G)}$ be a collection of disjoint matchings covering a maximum number of edges. If all edges are covered, we are done. So suppose that edge $e = uv$, say, is not covered. Then (since $\deg(u) \leq \Delta(G)$) some M_i misses u and (similarly) some M_j misses v. If $i = j$ we can extend M_i to $M_i \cup \{e\}$. If $i \neq j$, $M_i \cup M_j \cup \{e\}$ makes a bipartite graph of maximum degree at most two. Hence there exist matchings M and N with $M_i \cup M_j \cup \{e\} = M \cup N$.

So replacing M_i and M_j by M and N, increases the number of edges covered, contradicting our assumption. ∎

This proof, due to Kőnig [1916] (using a simplification of Skolem [1927]), also gives a polynomial-time algorithm to find a $\Delta(G)$-edge-colouring with $\Delta(G)$ colours. In fact, if G is simple, it gives an $O(nm)$ algorithm for edge-colouring. This bound can be achieved also for bipartite multigraphs using an appropriate data-structure — see Section 20.9a.

20.1a. Edge-colouring regular bipartite graphs

Kőnig's edge-colouring theorem is directly equivalent to the special case of regular bipartite graphs (since any bipartite graph of maximum degree Δ is a subgraph of a Δ-regular bipartite graph (Kőnig [1932])). Rizzi [1997,1998] gave the following very elegant short argument for the k-edge-colourability of k-regular bipartite graphs. (A similar proof in terms of common transversals of two partitions of a set into equally sized classes was given by Sperner [1927] — see Section 22.7d.)

Let G be a counterexample with fewest edges. So G has no perfect matching. Choose an edge $e = uv$. Then we can extend the graph $G - u - v$ to a k-regular bipartite graph H by adding at most $k - 1$ new edges. As H has fewer edges than G, H has a k-edge-colouring. Since less than k new edges have been added, there is a colour M that uses none of the new edges. Then $M \cup \{e\}$ is a perfect matching in G, a contradiction.

20.2. The capacitated case

Egerváry [1931] observed that the following capacitated version directly follows from Kőnig's edge-colouring theorem:

Corollary 20.1a. *Let $G = (V, E)$ be a bipartite graph and let $c : E \to \mathbb{Z}_+$ be a capacity function. Then the minimum size of a family of matchings such that each edge e is in at least c_e of them is equal to the maximum of $c(\delta(v))$ over all $v \in V$.*

Proof. Directly from Kőnig's edge-colouring theorem, by replacing each edge e by c_e parallel edges. ∎

This reduction being easy, it might not be satisfactory algorithmically. It would not yield a polynomial-time reduction for the following problem:

(20.3) given: a bipartite graph $G = (V, E)$ and a capacity function $c : E \to \mathbb{Z}_+$;
find: matchings $M_1, \ldots, M_k$ and nonnegative integers $\lambda_1, \ldots, \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i \chi^{M_i} = c$ and such that $\sum_{i=1}^{k} \lambda_i$ is minimized.

However, there is an easy strongly polynomial-time algorithm for this problem: Let F be the subset of edges e of G with $c_e > 0$. Find a matching M in F covering all vertices v of G that maximize $c(\delta(v))$. Let $\lambda := \min\{c_e \mid e \in M\}$, and replace c by $c - \lambda\chi^M$. Next iterate this.

Since in each iteration the number of edges e with $c_e > 0$ decreases, there are at most $|E|$ iterations. Since a matching covering a given set R of vertices can be found in time $O(|R||E|)$, this gives an $O(nm^2)$ algorithm. However, by starting in each iteration with the matching left from the previous iteration, one can do better (Gonzalez and Sahni [1976]):

Theorem 20.2. *Problem* (20.3) *can be solved in time* $O(m^2)$.

Proof. We may assume that $c(\delta(v))$ is equal for all v, by duplicating G and connecting each vertex with its copy, giving the new edges appropriate capacities. We can also assume that $c_e > 0$ for each edge e.

First we find a perfect matching in G, which can be done in time $O(nm)$, since we can apply $O(n)$ matching-augmenting iterations to find a perfect matching.

In any further iteration, let M be the matching obtained in the previous iteration. Suppose that after resetting c, there exist α edges e in M with $c_e = 0$. Delete these edges. Then in α matching-augmenting steps we can obtain a perfect matching M' in the new graph. So the iteration takes $O(\alpha m)$ time. Since over all iterations the α add up to $|E|$, we have the time bound $O(m^2)$. ∎

20.3. Edge-colouring polyhedrally

Polyhedrally, edge-colouring can be studied with the help of the 'substar polytope' of an undirected graph $G = (V, E)$. Call a set F of edges of G a *substar* if $F \subseteq \delta(v)$ for some $v \in V$. The *substar polytope* $P_{\text{substar}}(G)$ of G is the convex hull of the incidence vectors of substars. So it is a polytope in $\mathbb{R}^E$.

Each vector x in the substar polytope trivially satisfies

(20.4)	(i)	$x_e \geq 0$	for each $e \in E$,
	(ii)	$x(M) \leq 1$	for each matching M.

The following is direct from the description of the bipartite matching polytope (Corollary 18.1b) with the theory of antiblocking polyhedra:

Theorem 20.3. *The substar polytope of a bipartite graph is determined by* (20.4).

Proof. By Corollary 18.1b, the matching polytope is the antiblocking polyhedron of the substar polytope. Hence the substar polytope is the antiblocking polyhedron of the matching polytope (cf. Section 5.9), which is the content of the theorem. ∎

What Kőnig's edge-colouring theorem adds to it is:

Theorem 20.4. *System* (20.4) *is TDI.*

Proof. This is equivalent to Corollary 20.1a. ∎

Note that Kőnig's edge-colouring theorem also can be derived easily from the characterization of the matching polytope. For any bipartite graph $G = (V, E)$, the vector $\Delta(G)^{-1} \cdot \mathbf{1}$ belongs to the matching polytope (where $\mathbf{1}$ is the all-one vector in $\mathbb{R}^E$), and hence it is a convex combination of matchings. Each of these matchings should cover each maximum-degree vertex. So there exists a matching M covering all maximum-degree vertices. Hence $\Delta(G - M) = \Delta(G) - 1$, and we can apply induction.

Also, the integer decomposition property of the matching polytope is equivalent to Kőnig's edge-colouring theorem. (The integer decomposition property follows from the total unimodularity of the incidence matrix of G.)

20.4. Packing edge covers

A theorem 'dual' to Kőnig's edge-colouring theorem was shown by Gupta [1967,1978]. The edge-colouring number $\chi(G)$ of a graph G is the minimum number of matchings needed to cover the edges of a G. Dually, one can define the *edge cover packing number* $\xi(G)$ of a graph by:

(20.5) $\quad \xi(G) :=$ the maximum number of disjoint edge covers in G.

So, in terms of colours, $\xi(G)$ is the maximum number of colours that can be used in colouring the edges of G in such a way that at each vertex all colours occur. Hence, if $\delta(G)$ denotes the minimum degree of G, then

(20.6) $\quad \xi(G) \leq \delta(G)$.

The triangle K_3 again is an example having strict inequality. For bipartite graphs however Gupta [1967,1978] showed:

Theorem 20.5. *For any bipartite graph $G = (V, E)$:*

(20.7) $\quad \xi(G) = \delta(G)$.

That is, the maximum number of disjoint edge covers is equal to the minimum degree.

Proof. We give a reduction to Kőnig's edge-colouring theorem (Theorem 20.1).

One may derive from G a bipartite graph H, each vertex of which has degree $\delta(G)$ or 1, by repeated application of the following procedure:

(20.8) for any vertex v of degree larger than $\delta(G)$, add a new vertex u, and replace one of the edges incident with v, $\{v, w\}$ say, by $\{u, w\}$.

So there is a one-to-one correspondence between the edges of the final graph H and the edges of G. Since H has maximum degree $\delta(G)$, by Theorem 20.1 the edges of H can be coloured with $\delta(G)$ colours such that no two edges of the same colour intersect. So at any vertex of H of degree $\delta(G)$, all colours occur. This gives a colouring of the edges of G with $\delta(G)$ colours such that at any vertex of G all colours occur. ∎

Gupta [1974,1978] gave the following common generalization of Theorems 20.1 on edge-colouring and 20.5 on disjoint edge covers:

Theorem 20.6. *Let $G = (V, E)$ be a bipartite graph and let $k \in \mathbb{Z}_+$. Then E can be partitioned into classes $E_1, \ldots, E_k$ such that each vertex v is covered by at least* $\min\{k, \deg_G(v)\}$ *of the E_i.*

Proof. Like in the proof of Theorem 20.5, split off edges from vertices of degree larger than k, until each vertex has degree at most k. Applying Kőnig's edge-colouring theorem to the final graph yields a partitioning of the original edge set as required. ∎

Call a set F of edges of a graph $G = (V, E)$ a *superstar* if $F \supseteq \delta(v)$ for some $v \in V$. The *superstar polytope* $P_{\text{superstar}}(G)$ of G is the convex hull of the incidence vectors of superstars in G. Consider

$$\text{(20.9)} \qquad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq 1 & \text{for each } e \in E, \\ \text{(ii)} & x(F) \geq 1 & \text{for each edge cover } F. \end{array}$$

Theorem 20.7. *If G is bipartite, system* (20.9) *determines the superstar polytope and is TDI.*

Proof. With the theory of blocking polyhedra, Theorem 19.6 implies that the superstar polytope is determined by (20.9). Total dual integrality of (20.9) is equivalent to the capacitated version of Theorem 20.5. ∎

20.5. Balanced colours

McDiarmid [1972] and de Werra [1970,1972] showed the following generalization of Kőnig's edge-colouring theorem (in fact, it is a special case of a theorem of Folkman and Fulkerson [1969] (see Theorem 20.10 below), and also it is a consequence of the result in Dulmage and Mendelsohn [1969]):

Theorem 20.8. *Let $G = (V, E)$ be a bipartite graph and let $k \geq \Delta(G)$. Then E can be partitioned into matchings $M_1, \ldots, M_k$ such that*

(20.10) $\lfloor |E|/k \rfloor \leq |M_i| \leq \lceil |E|/k \rceil$

for each $i = 1, \ldots, k$.

Proof. As $k \geq \Delta(G)$, by Kőnig's edge-colouring theorem, E can be partitioned into matchings $M_1, \ldots, M_k$ (possibly empty). Choose $M_1, \ldots, M_k$ such that

(20.11) $$\sum_{i=1}^{k} |M_i|^2$$

is minimized.

Suppose that (20.10) is violated. Then there exist M_i and M_j with $|M_i| \geq |M_j|+2$. Then $M_i \cup M_j$ has at least one component K containing more edges in M_i than in M_j. Let $M_i' := M_i \triangle K$ and $M_j' := M_j \triangle K$. Then $|M_i'|^2 + |M_j'|^2 = (|M_i|-1)^2 + (|M_j|+1)^2 = |M_i|^2 + |M_j|^2 - 2|M_i| + 2|M_j| + 2 < |M_i|^2 + |M_j|^2$. So replacing M_i and M_j by M_i' and M_j' decreases the sum (20.11), contradicting our minimality assumption. ∎

Related results can be found in Dulmage and Mendelsohn [1969], Folkman and Fulkerson [1969], Brualdi [1971b], and de Werra [1971,1976].

20.6. Packing perfect matchings

Packing perfect matchings seems less directly reducible to partitioning into matchings or edge covers. It can be handled with the following more general result of Folkman and Fulkerson [1969] on packing matchings of a fixed size p, which is proved by reduction to Menger's theorem:

Theorem 20.9. *Let $G = (V, E)$ be a bipartite graph and let $k, p \in \mathbb{Z}_+$. Then there exist k disjoint matchings of size p if and only if each subset X of V spans at least $k(p + |X| - |V|)$ edges.*

Proof. To see necessity, let $X \subseteq V$ and consider a matching M in G of size p. Since at most $|V| - |X|$ edges in M intersect $V \setminus X$, at least $|M| - (|V| - |X|) = p + |X| - |V|$ edges of M are spanned by X. So k disjoint matchings of size p have at least $k(p + |X| - |V|)$ edges spanned by X.

To see sufficiency, let U and W be the colour classes of G. Orient all edges from U to W. Moreover, add vertices s and t, and, for each $u \in U$, add k parallel arcs from s to u, and, for each $w \in W$, add k parallel arcs from w to t. Let D be the directed graph arising.

We show with Menger's theorem that D contains kp arc-disjoint $s - t$ paths. Consider any $s - t$ cut $\delta^{\text{out}}(Y)$, with $s \in Y$, $t \notin Y$. Let $X := (U \cap Y) \cup (W \setminus Y)$. Then

(20.12) $$|\delta^{\text{out}}(Y)| = k|U \setminus Y| + k|W \cap Y| + |E[X]| = k(|V| - |X|) + |E[X]|,$$

where $E[X]$ is the set of edges spanned by X. As $|E[X]| \geq k(p + |X| - |V|)$, it follows that $|\delta^{\text{out}}(Y)| \geq kp$.

So D contains kp arc-disjoint $s-t$ paths. The edges of G that belong to these paths form a subgraph of G with kp edges, of maximum degree at most k. So by Theorem 20.8, G has k disjoint matchings of size p. ∎

This implies the following theorem of Fulkerson [1964b] on the maximum number of disjoint perfect matchings (in fact equivalent to a result of Ore [1956], see Corollary 20.9b below):

Corollary 20.9a. *Let $G = (V, E)$ be a bipartite graph and let $k \in \mathbb{Z}_+$. Then G has k disjoint perfect matchings if and only if each subset X of V spans at least $k(|X| - \frac{1}{2}|V|)$ edges.*

Proof. Directly by taking $p := \frac{1}{2}|V|$ in Theorem 20.9. ∎

(Lebensold [1977] and Murty [1978] gave other proofs of this corollary.)

Note that, by Kőnig's edge-colouring theorem, a bipartite graph $G = (V, E)$ has k disjoint perfect matchings if and only if G has a k-factor. (A *k-factor* is a subset F of E with the graph (V, F) k-regular.)

So Corollary 20.9a is equivalent to the following result of Ore [1956]:

Corollary 20.9b. *A bipartite graph $G = (V, E)$ has a k-factor if and only if each subset X of V spans at least $k(|X| - \frac{1}{2}|V|)$ edges.*

Proof. Directly from Corollary 20.9a. ∎

20.6a. Polyhedral interpretation

We can interpret these results polyhedrally. In Theorem 18.8 we saw that for any bipartite graph $G = (V, E)$, the up hull of the perfect matching polytope of G,

(20.13) $$P^{\uparrow}_{\text{perfect matching}}(G) = P_{\text{perfect matching}}(G) + \mathbb{R}^E_+$$

is determined by the inequalities

(20.14)
$$\begin{array}{lll} \text{(i)} & x_e \geq 0 & \text{for each } e \in E, \\ \text{(ii)} & x(E[S]) \geq |S| - \frac{1}{2}|V| & \text{for each } S \subseteq V. \end{array}$$

Then Corollary 20.9a implies that for each $k \in \mathbb{Z}_+$, each integer vector $w \in k \cdot P^{\uparrow}_{\text{perfect matching}}(G)$ is the sum of k vectors in $P^{\uparrow}_{\text{perfect matching}}(G)$. In other words:

Corollary 20.9c. *$P^{\uparrow}_{\text{perfect matching}}(G)$ has the integer decomposition property.*

Proof. From Corollary 20.9a, by replacing each edge by $w(e)$ parallel edges. ∎

We can view this also in terms of the blocking polyhedron of $P^{\uparrow}_{\text{perfect matching}}(G)$, which is the polyhedron Q determined by

(20.15) (i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(M) \geq 1$ for each perfect matching M.

Since $P^{\uparrow}_{\text{perfect matching}}(G)$ is determined by (20.14), the theory of blocking polyhedra gives that Q is equal to the up hull of the convex hull of the vectors

(20.16) $$\frac{1}{|S|-\frac{1}{2}|V|}\chi^{E[S]}$$

where $S \subseteq V$ with $|S| > \frac{1}{2}|V|$.

So the minimum value of $\mathbf{1}^{\mathsf{T}}x$ over Q is equal to

(20.17) $$\min\{\frac{|E[S]|}{|S|-\frac{1}{2}|V|} \mid S \subseteq V, |S| > \tfrac{1}{2}|V|\}.$$

By LP-duality, this is equal to the maximum value of $\sum_M \lambda_M$, where M ranges over perfect matchings and where $\lambda_M \geq 0$ such that $\sum_M \lambda_M \chi^M \leq \mathbf{1}$. So Corollary 20.9a states: the maximum number of disjoint perfect matchings in a bipartite graph is equal to

(20.18) $$\lfloor \max\{\sum_M \lambda_M \mid \lambda_M \geq 0, \sum_M \lambda_M \chi^M \leq \mathbf{1}\}\rfloor.$$

As we can directly extend this to a weighted version, one has:

Corollary 20.9d. *System* (20.15) *has the integer rounding property.*

Proof. See above. ∎

20.6b. Extensions

The results of Sections 20.5 and 20.6 can be extended as follows, as was shown by Folkman and Fulkerson [1969]. It is based on the following theorem:

Theorem 20.10. *Let $G = (V, E)$ be a bipartite graph, let $k \geq \Delta(G)$, and let $p \geq |E|/k$. Then G has a k-edge-colouring in which l colours have size p if and only if G has l disjoint matchings of size p.*

Proof. Necessity being trivial, we show sufficiency. Let G have l disjoint matchings of size p. We must show that there exist l disjoint matchings of size p such that at each vertex v at most $k - l$ edges incident with v are in none of these matchings (since then the edges not contained in the matchings can be properly coloured by $k - l$ colours).

That is, by Theorem 20.8 it suffices to show that there exists a subset F of E such that

(20.19) (i) $\deg_F(v) \leq l$ and $\deg_{E\setminus F}(v) \leq k - l$ for each vertex v;
(ii) $|F| = lp$.

Let F be any subset of E satisfying (20.19)(i), with $|F| \leq lp$, and with $|F|$ as large as possible. Such an F exists, since by Theorem 20.8 we can k-edge-colour G such that each colour has size at most $\lceil |E|/k \rceil \leq p$. Any l of the colours gives F as required.

If $|F| = lp$ we are done, so assume that $|F| < lp$. Since G has l disjoint matchings of size p, E has a subset F' of size lp with $\deg_{F'}(v) \leq l$ for each vertex v. Choose F' with $F' \setminus F$ as small as possible.

Consider an orientation D of the graph $(V, F\triangle F')$, where each edge in $F \setminus F'$ is oriented from colour class U (say) to colour class W (say), and where each edge in $F' \setminus F$ is oriented from W to U. If D contains a directed circuit C, we can reduce $F' \setminus F$, by replacing F' by $F'\triangle C$. So D is acyclic, and hence we can partition $F\triangle F'$ into directed paths, where each path starts at a vertex v with $\deg^{\text{out}}_D(v) > \deg^{\text{in}}_D(v)$ and ends at a vertex v with $\deg^{\text{in}}_D(v) > \deg^{\text{out}}_D(v)$. As $|F'| > |F|$, at least one of these paths, P say, has more edges in F' than in F. Now replacing F by $F\triangle EP$ does not violate (20.19)(i), since $\deg_{F\triangle EP}(v) = \deg_F(v) + 1 \leq \deg_{F'}(v) \leq l$ if v is an end of P and $\deg_{F\triangle EP}(v) = \deg_F(v)$ for any other vertex v. As this increases $|F|$, it contradicts our maximality assumption. ∎

This implies the following result of Folkman and Fulkerson [1969], generalizing Theorems 20.8 and 20.9 (by taking $p_2 = 1$):

Corollary 20.10a. *Let $G = (V, E)$ be a bipartite graph and let $k_1, k_2, p_1, p_2 \in \mathbb{Z}_+$ be such that $k_1 + k_2 \geq \Delta(G)$, $k_1p_1 + k_2p_2 = |E|$, and $p_1 \geq p_2$. Then E can be partitioned into k_1 matchings of size p_1 and k_2 matchings of size p_2 if and only if each subset X of V spans at least $k_1(p_1 + |X| - |V|)$ edges.*

Proof. Necessity being easy, we prove sufficiency. By Theorem 20.9, G has k_1 disjoint matchings of size p_1. Let $k := k_1 + k_2$. Since $p_1 \geq p_2$, we have $p_1 \geq (p_1k_1 + p_2k_2)/k = |E|/k$. Hence, by Theorem 20.10, G has k_1 disjoint matchings of size p_1, such that the uncovered edges form a subgraph of maximum degree at most k_2. As this subgraph has $|E| - p_1k_1 = p_2k_2$ edges, by Theorem 20.8 we can split its edge set into k_2 matchings of size p_2. ∎

These results relate to simple b-matchings — see Corollary 21.29a.

20.7. Covering by perfect matchings

A series of results similar to those in Section 20.6 can be derived for covering by perfect matchings and for the down hull of the perfect matching polytope. Brualdi [1979] showed the covering analogue of Corollary 20.9a:

Theorem 20.11. *Let $G = (V, E)$ be a bipartite graph and let $k \in \mathbb{Z}_+$. Then E can be covered by k perfect matchings if and only if any vertex cover X spans at most $k(|X| - \frac{1}{2}|V|)$ edges.*

Proof. *Necessity.* Let G be covered by k perfect matchings and let X be a vertex cover. Each perfect matching contains $|V \setminus X|$ edges not spanned by X, and hence $\frac{1}{2}|V| - |V \setminus X| = |X| - \frac{1}{2}|V|$ edges spanned by X. This proves necessity.

Sufficiency. Assume that the condition holds. This implies that both colour classes of G have size $\frac{1}{2}|V|$, since each of them is a vertex cover X

spanning no edge, implying $|X| \geq \frac{1}{2}|V|$. It also implies that the maximum degree of G is at most k, since for each vertex v the set $U \cup \{v\}$ (where U is the colour class of G not containing v) spans at most k edges.

For each vertex v, let $b_v := k - \deg(v)$. Split each vertex v into b_v vertices, and replace any edge uv by $b_u b_v$ edges connecting the b_u copies of u with the b_v copies of v. This yields the bipartite graph H, with $k|V| - 2|E|$ vertices.

Now H has a perfect matching, as follows from Frobenius' theorem: if Y is a vertex cover in H, then the set X of vertices v of G for which all copies in H belong to Y, is a vertex cover in G. Now by the condition, X spans at most $k(|X| - \frac{1}{2}|V|)$ edges of G. Hence

(20.20) $$|Y| \geq \sum_{v \in X} (k - \deg(v)) = k|X| - |E| - |E[X]| \geq \tfrac{1}{2}k|V| - |E|.$$

So Y is not smaller than half the number of vertices of H. Therefore, by Frobenius' theorem, H has a perfect matching M.

For each edge e of G, add parallel edges to e as often as a copy of e occurs in M. We obtain a k-regular bipartite graph G'. By Kőnig's edge-colouring theorem, the edges of G' can be partitioned into k perfect matchings. This gives k perfect matchings in G covering E. ∎

(This proof method in fact consists of showing that G has a perfect b-matching — see Chapter 21.)

The result is equivalent to characterizing bipartite graphs that are k-regularizable. A graph $G = (V, E)$ is *k-regularizable* if we can replace each edge by a positive number of parallel edges so as to obtain a k-regular graph. Then:

Corollary 20.11a. *Let $G = (V, E)$ be a bipartite graph and let $k \in \mathbb{Z}_+$. Then G is k-regularizable if and only if any vertex cover X spans at most $k(|X| - \frac{1}{2}|V|)$ edges.*

Proof. Directly from Theorem 20.11. ∎

20.7a. Polyhedral interpretation

Again we can interpret Theorem 20.11 polyhedrally. In Theorem 18.9 we saw that for a bipartite graph $G = (V, E)$, the down hull of the perfect matching polytope of G,

(20.21) $$P^{\downarrow}_{\text{perfect matching}}(G) = (P_{\text{perfect matching}}(G) - \mathbb{R}^E_+) \cap \mathbb{R}^E_+$$

is determined by the inequalities

(20.22)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(E[S]) \leq |S| - \frac{1}{2}|V|$ for each vertex cover S.

Then Theorem 20.11 implies that for each $k \in \mathbb{Z}_+$, each integer vector $w \in k \cdot P^{\downarrow}_{\text{perfect matching}}(G)$ is a sum of k integer vectors in $P^{\downarrow}_{\text{perfect matching}}(G)$. That is:

Corollary 20.11b. $P^{\downarrow}_{\text{perfect matching}}(G)$ *has the integer decomposition property.*

Proof. See above. ∎

We can view this result also in terms of the antiblocking polyhedron of $P^{\downarrow}_{\text{perfect matching}}(G)$, which is the polyhedron Q determined by

(20.23) (i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(M) \leq 1$ for each perfect matching M.

By the theory of antiblocking polyhedra, Q is equal to the down hull of the convex hull of the vectors

(20.24) $$\frac{1}{|S|-\frac{1}{2}|V|}\chi^{E[S]}$$

where S is a vertex cover with $|S| > \frac{1}{2}|V|$.

So the maximum value of $\mathbf{1}^{\mathsf{T}}x$ over Q is equal to

(20.25) $$\max\{\frac{|E[S]|}{|S|-\frac{1}{2}|V|} \mid S \text{ vertex cover, } |S| > \tfrac{1}{2}|V|\}.$$

By LP-duality, this is equal to the minimum value of $\sum_M \lambda_M$, where M ranges over perfect matchings and where $\lambda_M \geq 0$ with $\sum_M \lambda_M \chi^M \geq \mathbf{1}$. So Theorem 20.11 states: the minimum number of perfect matchings needed to cover all edges in a bipartite graph is equal to

(20.26) $$\lceil \min\{\sum_M \lambda_M | \lambda_M \geq 0, \sum_M \lambda_M \chi^M \geq \mathbf{1}\}\rceil.$$

As we can directly extend this to a weighted version, one has:

Corollary 20.11c. *The polyhedron determined by* (20.23) *has the integer rounding property.*

Proof. See above. ∎

20.8. The perfect matching lattice of a bipartite graph

The *perfect matching lattice* (often briefly the *matching lattice*) of a graph $G = (V, E)$ is the lattice generated by the incidence vectors of perfect matchings in G; that is,

(20.27) $$L_{\text{perfect matching}}(G) := \text{lattice}\{\chi^M \mid M \text{ perfect matching in } G\}.$$

With the help of Kőnig's edge-colouring theorem, it is not difficult to characterize the perfect matching lattice of a bipartite graph (cf. Lovász [1985]). Recall that the *perfect matching space* of a graph G is the linear hull of the incidence vectors of the perfect matchings in G (cf. Section 18.5d).

Theorem 20.12. *The perfect matching lattice of a bipartite graph $G = (V, E)$ is equal to the set of integer vectors in the perfect matching space of G.*

Proof. Obviously, each vector in the perfect matching lattice is integer and belongs to the perfect matching space. To see the reverse inclusion, let x be an integer vector in the perfect matching space. So $x_e = 0$ for each edge covered by no perfect matching, and $x(\delta(u)) = x(\delta(v))$ for all $u, v \in V$. By adding to x incidence vectors of perfect matchings, we can assume that $x_e \geq 0$ for all $e \in E$.

Replace any edge e by x_e parallel copies. We obtain a k-regular bipartite graph H, with $k := x(\delta(v))$ for any $v \in V$. Hence, by Kőnig's edge-colouring theorem, H is k-edge-colourable. As each colour is a perfect matching in H, we can decompose x as a sum of k incidence vectors of perfect matchings in G. So x belongs to the perfect matching lattice of G. ∎

This gives a characterization of the perfect matching lattice for matching-covered bipartite graphs (which will be used in the characterization of the perfect matching lattice of an arbitrary graph in Chapter 38). A graph is called *matching-covered* if each edge belongs to a perfect matching.

Corollary 20.12a. *Let $G = (V, E)$ be a matching-covered bipartite graph and let $x \in \mathbb{Z}^E$ be such that $x(\delta(u)) = x(\delta(v))$ for any two vertices u and v. Then x belongs to the perfect matching lattice of G.*

Proof. Directly from Theorems 20.12 and 18.7. ∎

By lattice duality theory, Theorem 20.12 is equivalent to the following.

Corollary 20.12b. *Let $G = (V, E)$ be a bipartite graph and let $w \in \mathbb{R}^E$ be a weight function. Then each perfect matching has integer weight if and only if there exists a vector $b \in \mathbb{R}^V$ with $b(V) = 0$ and with $w_e - b_u - b_v$ integer for each edge $e = uv$ covered by at least one perfect matching.*

Proof. Sufficiency is easy, since if such a b exists, then, for each perfect matching M,

$$w(M) = b(V) + \sum_{e=uv \in M} (w_e - b_u - b_v) = \sum_{e=uv \in M} (w_e - b_u - b_v) \tag{20.28}$$

is an integer.

To see necessity, suppose that $w(M)$ is integer for each perfect matching M. Then (by definition of dual lattice) w belongs to the dual lattice of the perfect matching lattice. Theorem 20.12 implies that the dual lattice is the sum of $\mathbb{Z}^E$ and the linear space orthogonal to the perfect matching space. So $w = w' + w''$, where $w' \in \mathbb{Z}^E$ and w'' is orthogonal to the perfect matching space; that is, $w''(M) = 0$ for each perfect matching M. By Corollary 18.7a, there exists a vector $b \in \mathbb{R}^V$ with $b(V) = 0$ and with $w''_e = b_u + b_v$ for each edge $e = uv$ covered by at least one perfect matching. This is equivalent to the present Corollary. ∎

20.9. Further results and notes

20.9a. Some further edge-colouring algorithms

As mentioned, it is easy to implement an $O(nm)$-time algorithm for finding a $\Delta(G)$-edge-colouring in a simple bipartite graph G. Such an algorithm also exists if G has multiple edges:

Theorem 20.13. *The edges of a bipartite graph G can be coloured with $\Delta(G)$ colours in $O(nm)$ time.*

Proof. Let $\Delta := \Delta(G)$. We update a collection of disjoint matchings $M_1, \ldots, M_\Delta$ (the colours), each stored as a doubly linked list. For each edge e, we keep the i for which $e \in M_i$ ($i = 0$ if e is in no M_i). Initially we set $M_i := \emptyset$ for $i := 1, \ldots, \Delta$. We also store the colour classes U and W as lists.

The algorithm runs along all pairs of vertices $u \in U$ and $w \in W$. Fixing $u \in U$ and $w \in W$, make a list L of edges e connecting u and w (taking $O(\deg(u))$ time, by scanning $\delta(u)$); define $d(u, w) := |L|$; make a list I of $d(u, w)$ indices i for which M_i misses u (taking $O(\deg(u))$ time, by scanning $\delta(u)$); make a list J of $d(u, w)$ indices j for which M_j misses w (taking $O(\deg(w))$ time, by scanning $\delta(w)$); next, while there is an edge e_0 in L:

(20.29) choose $i \in I$ and $j \in J$;
if $i = j$, insert e_0 in M_i, delete e_0 from L, and delete i from I and J;
if $i \neq j$, make for each $v \in V$ a list T_v of edges in $M_i \cup M_j$ incident with v (taking $O(n)$ time, by scanning M_i and M_j);
identify the path component P in $M_i \cup M_j$ starting at u (taking $O(n)$ time, using the T_v);
for each edge e on P, if e is in M_i move e to M_j and if e is in M_j we move e to M_i (taking $O(n)$ time);
insert e_0 in M_j, delete e_0 from L, delete i from I, and delete j from J.

Fixing u and w, the preprocessing takes $O(\deg(u) + \deg(w))$ time, and each of the $d(u, w)$ iterations takes $O(n)$ time. As $\sum_{u \in U} \sum_{w \in W} (\deg(u) + \deg(w) + nd(u, w)) = 2nm$, we obtain an algorithm as required. ∎

From their linear-time perfect matching algorithm for regular bipartite graphs, Cole, Ost, and Schirra [2001] derived (using an idea of Gabow [1976c]):

Theorem 20.14. *A k-regular bipartite graph $G = (V, E)$ can be k-edge-coloured in time $O(m \log k)$.*

Proof. We describe a recursive algorithm, the case $k = 1$ being the basis.

If k is even, find an Eulerian orientation of G, let G' be the $\frac{1}{2}k$-regular graph consisting of all edges oriented from one colour class of G to the other, let G'' be the $\frac{1}{2}k$-regular graph consisting of the remaining edges, and recursively $\frac{1}{2}k$-edge-colour G' and G''. This gives a k-edge-colouring of G.

If k is odd and ≥ 3, find a perfect matching M in G, and recursively $(k-1)$-edge-colour $G - M$. With M, this gives a k-edge-colouring of G.

We show that the running time is $O(m \log k)$. The recursive step takes time $O(m)$, since finding an Eulerian orientation or finding a perfect matching takes $O(m)$ time (Corollary 16.11a). Moreover, in one or two recursive steps, the graph is split into two graphs with half the number of edges. Since $m \log_2 k = m + 2(\frac{1}{2}m \log_2(\frac{1}{2}k))$, the result follows. ∎

Corollary 20.14a. *The edges of a bipartite graph G can be coloured with $\Delta(G)$ colours in $O(m \log \Delta(G))$ time.*

Proof. Let $k := \Delta(G)$. First iteratively merge any two vertices in the same colour class of G if each of them has degree at most $\frac{1}{2}k$. The final graph H will have at most two vertices of degree at most $\frac{1}{2}k$, and moreover, $\Delta(H) = k$ and any k-edge-colouring of H yields a k-edge-colouring of G. Next make a copy H' of H, and join each vertex v of H by $k - \deg_H(v)$ parallel edges with its copy v' in H' (where $\deg_H(v)$ is the degree of v in H). This gives the k-regular bipartite graph G', with $|EG'| = O(|EG|)$.

By Theorem 20.14, we can find a k-edge-colouring of G' in $O(m \log k)$ time. This gives a k-edge-colouring of H and hence a k-edge-colouring of G. ∎

20.9b. Complexity survey for bipartite edge-colouring

	$O(nm)$	Kőnig [1916]
	$O(\sqrt{n}\, m\Delta)$	Hopcroft and Karp [1971,1973] (cf. Gabow and Kariv [1978])
*	$O(\tilde{m}^2)$	Gonzalez and Sahni [1976]
	$O(\sqrt{n}\, m \log \Delta)$	Gabow [1976c]
	$O(m\sqrt{n \log n})$	Gabow and Kariv [1978]
	$O(m\Delta \log n)$	Gabow and Kariv [1978]
	$O((m + n^2) \log \Delta)$	Gabow and Kariv [1978,1982]
	$O(m(\log n)^2 \log \Delta)$	Lev, Pippenger, and Valiant [1981]
	$O(m(\log m)^2)$	Gabow and Kariv [1982]
	$O(m \log m)$	Cole and Hopcroft [1982]
*	$O(n\tilde{m} \log \mu)$	Gabow and Kariv [1982]
	$O((m + n \log n \log^2 \Delta) \log \Delta)$	Cole and Hopcroft [1982]
	$O((m + n \log n \log \Delta) \log \Delta)$	Cole [1982]
	$O(n2^{2^{O(\Delta)}})$	Cole [1982]
	$O((m + n \log n) \log \Delta)$	R. Cole and K. Ost (cf. Ost [1995]), Kapoor and Rizzi [2000]
	$O(m\Delta)$	Schrijver [1999]
	$O(m \log \Delta + n \log n \log \Delta)$	Rizzi [2002]
*	$O(m \log \Delta)$	Cole, Ost, and Schirra [2001]

Here $\tilde{m}$ denotes the number of parallel classes of edges, μ the maximum size of a parallel class, and Δ the maximum degree. As before, $*$ indicates an asymptotically best bound in the table.

Kapoor and Rizzi [2000] showed that a bipartite graph of maximum degree Δ can be Δ-edge-coloured in time $T + O(m \log \Delta)$, where T is the time needed to find a perfect matching in a k-regular bipartite graph with m edges and $k \leq \Delta$. (So this is applied only once!)

20.9c. List-edge-colouring

An interesting extension of Kőnig's edge-colouring theorem was shown by Galvin [1995], which was the 'list-edge-colouring conjecture' for bipartite graphs (cf. Alon [1993], Häggkvist and Chetwynd [1992]). It implies the conjecture of J. Dinitz (1979) that the list-edge-colouring number of the complete bipartite graph $K_{n,n}$ equals n. (This is in fact a special case of the conjecture, formulated by V.G. Vizing in 1975, that the list-edge-colouring number of any graph is equal to its edge-colouring number (see Häggkvist and Chetwynd [1992]).) The proof of Galvin is based on the Gale-Shapley theorem on stable matchings (Theorem 18.11).

Let $G = (V, E)$ be a graph. Then G is *k-list-edge-colourable* if for each choice of finite sets L_e for $e \in E$ with $|L_e| = k$, we can choose $l_e \in L_e$ for $e \in E$ such that $l_e \neq l_f$ if e and f are incident. The smallest k for which G is k-list-edge-colourable is called the *list-edge-colouring number* of G.

Trivially, the list-edge-colouring number of G is at least the edge-colouring number of G, and hence at least the maximum degree $\Delta(G)$ of G. Galvin [1995] showed:

Theorem 20.15. *The list-edge-colouring number of a bipartite graph is equal to its maximum degree.*

Proof. Let $G = (V, E)$ be a bipartite graph, with colour classes U and W, and with maximum degree $k := \Delta(G)$. The theorem follows by applying the following statement to any $\Delta(G)$-edge-colouring $\phi : E \to \{1, \ldots, \Delta(G)\}$ of G.

(20.30) Let $\phi : E \to \mathbb{Z}$ be such that $\phi(e) \neq \phi(f)$ if e and f are incident. For each $e = uw \in E$ with $u \in U$ and $w \in W$, let L_e be a finite set satisfying

$$|L_e| > |\{f \in \delta(u) \mid \phi(f) < \phi(e)\}| + |\{f \in \delta(w) \mid \phi(f) > \phi(e)\}|.$$

Then there exist $l_e \in L_e$ ($e \in E$) such that $l_e \neq l_f$ if e and f are incident.

So it suffices to prove (20.30), which is done by induction on $|E|$. Choose $p \in \bigcup L_e$ and let $F := \{e \in E \mid p \in L_e\}$. Define for each $v \in V$ a total order $<_v$ on $\delta_F(v)$ by:

(20.31)
$$e \leq_v f \iff \phi(e) \geq \phi(f), \text{ if } v \in U,$$
$$e \leq_v f \iff \phi(e) \leq \phi(f), \text{ if } v \in W,$$

for $e, f \in \delta_F(v)$. By the Gale-Shapley theorem (Theorem 18.11), F contains a stable matching M. So M is a matching such that for each $e \in F$ there is an $f \in M$ with $e \leq_v f$ for some $v \in e$. Hence for each edge $e = uw \in F \setminus M$, with $u \in U$ and

$w \in W$: $\exists f \in M \cap \delta(u) : \phi(f) < \phi(e)$ or $\exists f \in M \cap \delta(w) : \phi(f) > \phi(e)$. So removing M from E and resetting $L_e := L_e \setminus \{p\}$ for each $e \in F \setminus M$, we can apply induction. ∎

(The proof by Slivnik [1996] is similar.) An extension of Galvin's theorem was given by Borodin, Kostochka, and Woodall [1997].

20.9d. Further notes

Edge-colouring relates to timetabling — see Appleby, Blake, and Newman [1960], Gotlieb [1963], Broder [1964], Cole [1964], Csima and Gotlieb [1964], Barraclough [1965], Duncan [1965], Almond [1966], Lions [1966b,1966a,1967], Welsh and Powell [1967], Yule [1967], Dempster [1968,1971], Wood [1968], de Werra [1970,1972], and McDiarmid [1972].

However, most practical timetabling problems require more than just bipartite edge-colouring, and are NP-complete. It is NP-complete to decide if a given partial edge-colouring in a bipartite graph can be extended to a minimum edge-colouring (Even, Itai, and Shamir [1975,1976]). This corresponds to a timetabling problem with 'time windows'. Moreover, the 3-dimensional analogue is NP-complete (Karp [1972b]): given three disjoint sets R, S, and T and a family $\mathcal{F}$ of triples $\{r, s, t\}$ with $r \in R$, $s \in S$, and $t \in T$, colour the sets in $\mathcal{F}$ with a minimum number of colours in such a way that sets of the same colour are disjoint.

Analogues of Kőnig's edge-colouring theorem, in terms of odd paths packing and covering, were given by de Werra [1986,1987]. The edge-colouring number of *almost bipartite graphs* (graphs which have a vertex whose deletion makes the graph bipartite) was characterized by Eggan and Plantholt [1986] and Reed [1999b].

Kőnig [1916] also proved an infinite extension of Theorem 20.1. We refer to Section 16.7h for some historical notes on the fundamental paper Kőnig [1916].

Sainte-Laguë [1923] mentioned (without proof and without reference to Kőnig's work) the result that each k-regular bipartite graph is k-edge-colourable.

Chapter 21

Bipartite b-matchings and transportation

The total unimodularity of the incidence matrix of a bipartite graph leads to general min-max relations, for b-matchings, b-edge covers, w-vertex covers, w-stable sets, and b-factors. The weighted versions of these problems relate to the classical transportation problem.

In this chapter, graphs can be assumed to be simple.

21.1. b-matchings and w-vertex covers

Let $G = (V, E)$ be a graph, with $V \times E$ incidence matrix A. We introduce the concepts of b-matching and w-vertex cover, which will turn out to be dual.

For $b : V \to \mathbb{Z}_+$, a *b-matching* is a function $x : E \to \mathbb{Z}_+$ such that for each vertex v of G:

(21.1) $\quad x(\delta(v)) \leq b_v,$

where $\delta(v)$ is the set of edges incident with v. In other words, x is a b-matching if and only if x is an integer vector satisfying $x \geq \mathbf{0}$, $Ax \leq b$. So if $b = \mathbf{1}$, then b-matchings are precisely the incidence vectors of matchings.

For $w : E \to \mathbb{Z}_+$, a *w-vertex cover* is a function $y : V \to \mathbb{Z}_+$ such that for each edge $e = uv$ of G:

(21.2) $\quad y_u + y_v \geq w_e.$

In other words, y is a w-vertex cover if and only if y is an integer vector satisfying $y \geq \mathbf{0}$, $y^{\mathsf{T}} A \geq w^{\mathsf{T}}$. So if $w = \mathbf{1}$, then $\{0, 1\}$-valued w-vertex covers are precisely the incidence vectors of vertex covers.

b-matchings and w-vertex covers are related by the following LP-duality equation:

(21.3) $\quad \max\{w^{\mathsf{T}} x \mid x \geq \mathbf{0}, Ax \leq b\} = \min\{y^{\mathsf{T}} b \mid y \geq \mathbf{0}, y^{\mathsf{T}} A \geq w^{\mathsf{T}}\}.$

Since A is totally unimodular (Theorem 18.2), both optima are attained by integer vectors. In other words (where the *w-weight* of a vector x equals $w^{\mathsf{T}} x$ and the *b-weight* of a vector y equals $y^{\mathsf{T}} b$):

Theorem 21.1. *Let $G = (V, E)$ be a bipartite graph and let $b : V \to \mathbb{Z}_+$ and $w : E \to \mathbb{Z}_+$. Then the maximum w-weight of a b-matching is equal to the minimum b-weight of a w-vertex cover.*

Proof. See above. ∎

Taking $b = \mathbf{1}$, we obtain Corollary 17.1a. For $w = \mathbf{1}$, we get the following min-max relation for maximum-size b-matching (again, the sum of the entries in a vector is called its *size*):

Corollary 21.1a. *Let $G = (V, E)$ be a bipartite graph and let $b : V \to \mathbb{Z}_+$. Then the maximum size of a b-matching is equal to the minimum b-weight of a vertex cover.*

Proof. This is the special case $w = \mathbf{1}$ of Theorem 21.1. ∎

An alternative way of proving this is by derivation from Kőnig's matching theorem: Split each vertex v into b_v copies, and replace each edge uv by $b_u b_v$ edges connecting the b_u copies of u with the b_v copies of v. (This construction is due to Tutte [1954b].)

Corollary 21.1a implies a characterization of the existence of a perfect b-matching. A b-matching is called *perfect* if equality holds in (21.1) for each vertex v. So a b-matching is perfect if and only if it has size $\frac{1}{2}b(V)$. Hence:

Corollary 21.1b. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$. Then there exists a perfect b-matching if and only if $b(C) \geq \frac{1}{2}b(V)$ for each vertex cover C.*

Proof. Directly from Corollary 21.1a. ∎

21.2. The b-matching polytope and the w-vertex cover polyhedron

The total unimodularity of the incidence matrix also implies characterizations of the corresponding polyhedra.

The *b-matching polytope* is the convex hull of the b-matchings. For bipartite graphs it is determined by:

(21.4)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(\delta(v)) \leq b_v$ for each $v \in V$.

Theorem 21.2. *The b-matching polytope of a bipartite graph $G = (V, E)$ is determined by* (21.4).

Proof. Directly from the facts that system (21.4) amounts to $x \geq \mathbf{0}, Ax \leq b$ and that A is totally unimodular, where A is the $V \times E$ incidence matrix of G. By Theorem 5.20, the vertices of the polytope $\{x \geq \mathbf{0} \mid Ax \leq b\}$ are integer, hence they are b-matchings. ∎

This generalizes the sufficiency part of Corollary 18.1b.

Similarly, the *w-vertex cover polyhedron*, being the convex hull of the w-vertex covers, is, for bipartite graphs, determined by:

(21.5)
(i) $y_v \geq 0$ for each $v \in V$,
(ii) $y_u + y_v \geq w_e$ for each $e = uv \in E$.

Theorem 21.3. *The w-vertex cover polyhedron of a bipartite graph is determined by* (21.5).

Proof. Directly from the total unimodularity of the incidence matrix of a bipartite graph. ∎

This generalizes the necessity part in Theorem 18.3.

21.3. Simple b-matchings and b-factors

In the context of b-matchings, call a vector x *simple* if it is $\{0,1\}$-valued. So a simple b-matching is the incidence vector of a set F of edges with $\deg_F(v) \leq b_v$ for each vertex v. We will identify the vector and the subset.

To characterize the maximum size of a simple b-matching, let, for any $X \subseteq V$, $E[X]$ denote the set of edges spanned by X.

Theorem 21.4. *The maximum size of a simple b-matching in a bipartite graph $G = (V, E)$ is equal to the minimum value of $b(V \setminus X) + |E[X]|$ taken over $X \subseteq V$.*

Proof. This can be reduced to the nonsimple case by replacing each edge uv by a path of length 3 connecting u and v (thus introducing two new vertices for each edge), and extending b by defining $b(s) := 1$ for each new vertex s. Then the maximum size of a simple b-matching in the original graph is equal to the maximum size of a b-matching in the new graph minus $|E|$, and we can apply Corollary 21.1a. ∎

(This construction is due to Tutte [1954b].)

The theorem can also be derived from the fact that both optima in the LP-duality equation:

(21.6)
$$\max\{\mathbf{1}^{\mathsf{T}}x \mid \mathbf{0} \leq x \leq \mathbf{1}, Ax \leq b\}$$
$$= \min\{y^{\mathsf{T}}b + z^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, z \geq \mathbf{0}, y^{\mathsf{T}}A + z^{\mathsf{T}} \geq \mathbf{1}^{\mathsf{T}}\}$$

have integer optimum solutions, since A (the incidence matrix of G) is totally unimodular.

Theorem 21.4 implies the following result of Ore [1956] (who formulated it in terms of directed graphs). A *b-factor* is a simple perfect b-matching. So it is a subset F of E with $\deg_F(v) = b_v$ for each $v \in V$ (again identifying a subset of E with its incidence vector in $\mathbb{R}^E$).

Corollary 21.4a. *Let $G = (V, E)$ be a bipartite graph and let $b : V \to \mathbb{Z}_+$. Then G has a b-factor if and only if each subset X of V spans at least $b(X) - \frac{1}{2}b(V)$ edges.*

Proof. Directly from Theorem 21.4. ∎

If b is equal to a constant k, Theorem 21.4 amounts to (with the help of Kőnig's edge-colouring theorem):

Corollary 21.4b. *Let $G = (V, E)$ be a bipartite graph and let $k \in \mathbb{Z}_+$. Then the maximum size of the union of k matchings is equal to the minimum value of $k|V \setminus X| + |E[X]|$ taken over $X \subseteq V$.*

Proof. Apply Theorem 21.4 to $b_v := k$ for all $v \in V$. We obtain a formula for the maximum size of a subset F of E with $\deg_F(v) \leq k$ for all $v \in V$. By Theorem 20.1, this is the union of k matchings. ∎

A *k-factor* in a graph $G = (V, E)$ is a subset F of E with $\deg_F(v) = k$ for each $v \in V$. Then:

Corollary 21.4c. *Let $G = (V, E)$ be a bipartite graph and let $k \in \mathbb{Z}_+$. Then G has a k-factor if and only if each subset X of V spans at least $k(|X| - \frac{1}{2}|V|)$ edges.*

Proof. Directly from Corollary 21.4a. ∎

From this one can derive the result of Fulkerson [1964b] (Corollary 20.9a) that a bipartite graph has k disjoint perfect matchings if and only if each subset X of V spans at least $k(|X| - \frac{1}{2}|V|)$ edges.

By the total unimodularity of the incidence matrix of bipartite graphs, the *simple b-matching polytope* (the convex hull of the simple b-matchings) of a bipartite graph $G = (V, E)$ is determined by:

(21.7)
$$\begin{aligned} 0 \leq x_e &\leq 1 && \text{for each } e \in E, \\ x(\delta(v)) &\leq b_v && \text{for each } v \in V. \end{aligned}$$

Similarly, the following min-max relation for maximum-weight simple b-matching follows (Vogel [1963]):

Theorem 21.5. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$ and $w \in \mathbb{Z}_+^E$. Then the maximum weight $w^{\mathsf{T}}x$ of a simple b-matching x is equal to the minimum value of*

(21.8) $$\sum_{v \in V} y_v b_v + \sum_{e \in E} z_e$$

where $y \in \mathbb{Z}_+^V$ *and* $z \in \mathbb{Z}_+^E$ *with* $y_u + y_v + z_e \geq w_e$ *for each edge* $e = uv$.

Proof. Directly from the LP-duality equation

(21.9) $$\max\{w^{\mathsf{T}} x \mid \mathbf{0} \leq x \leq \mathbf{1}, Ax \leq b\} \\ = \min\{y^{\mathsf{T}} b + z^{\mathsf{T}} \mathbf{1} \mid y \geq \mathbf{0}, z \geq \mathbf{0}, y^{\mathsf{T}} A + z^{\mathsf{T}} \geq w^{\mathsf{T}}\}$$

(where A is the $V \times E$ incidence matrix of G), using the total unimodularity of A. ∎

Moreover:

Theorem 21.6. *Let* $G = (V, E)$ *be a bipartite graph and let* $b \in \mathbb{Z}_+^V$ *and* $w \in \mathbb{Z}_+^E$. *Then the minimum weight* $w^{\mathsf{T}} x$ *of a* b*-factor* x *is equal to the maximum value of*

(21.10) $$\sum_{v \in V} y_v b_v + \sum_{e \in E} z_e$$

where $y \in \mathbb{Z}^V$ *and* $z \in \mathbb{Z}_+^E$ *with* $y_u + y_v - z_e \leq w_e$ *for each edge* $e = uv$.

Proof. Directly from the LP-duality equation

(21.11) $$\min\{w^{\mathsf{T}} x \mid \mathbf{0} \leq x \leq \mathbf{1}, Ax = b\} \\ = \max\{y^{\mathsf{T}} b - z^{\mathsf{T}} \mathbf{1} \mid z \geq \mathbf{0}, y^{\mathsf{T}} A - z^{\mathsf{T}} \leq w^{\mathsf{T}}\}$$

(where A is the $V \times E$ incidence matrix of G), using the total unimodularity of A. ∎

Notes. Hartvigsen [1999] gave a characterization of the convex hull of square-free simple 2-matching in a bipartite graph. (A 2-*matching* is a b-matching with $b = \mathbf{2}$. A simple 2-matching is *square-free* if it contains no circuit of length 4.) It implies that a maximum-weight square-free 2-matching in a bipartite graph can be found in strongly polynomial time.

21.4. Capacitated b-matchings

If we require that a b-matching x satisfies $x \leq c$ for some 'capacity' function $c : E \to \mathbb{Z}_+$, we speak of a *capacitated* b-matching. So simple b-matchings correspond to capacitated b-matchings for $c = \mathbf{1}$.

Theorem 21.7. *Let* $G = (V, E)$ *be a bipartite graph and let* $b \in \mathbb{Z}_+^V$ *and* $c \in \mathbb{Z}_+^E$. *Then the maximum size of a* b*-matching* $x \leq c$ *is equal to*

(21.12) $$\min_{X \subseteq V} b(V \setminus X) + c(E[X]).$$

Proof. The proof is similar to that of Theorem 21.4. Now we define $b(s) := c_e$ if s is a new vertex on the path connecting the end vertices of e. ∎

Alternatively, we can reduce this theorem to Theorem 21.4, by replacing each edge e by c_e parallel edges, or we can use total unimodularity similarly to (21.6).

Again we have the perfect case as direct consequence:

Corollary 21.7a. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$. Then there exists a perfect b-matching $x \le c$ if and only if*

(21.13) $$c(E[X]) \ge b(X) - \tfrac{1}{2}b(V)$$

for each $X \subseteq V$.

Proof. Directly from Theorem 21.7. ∎

Again, by the total unimodularity of the incidence matrix of bipartite graphs, the *c-capacitated b-matching polytope* (the convex hull of the b-matchings $x \le c$) of a bipartite graph $G = (V, E)$ is determined by:

(21.14) $$\begin{array}{ll} 0 \le x_e \le c_e & \text{for each } e \in E, \\ x(\delta(v)) \le b_v & \text{for each } v \in V. \end{array}$$

Similarly, the following min-max relation for maximum-weight capacitated b-matching follows:

Theorem 21.8. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$ and $w, c \in \mathbb{Z}_+^E$. Then the maximum weight $w^\mathsf{T}x$ of a b-matching $x \le c$ is equal to the minimum value of*

(21.15) $$\sum_{v \in V} y_v b_v + \sum_{e \in E} z_e c_e$$

where $y \in \mathbb{Z}_+^V$ and $z \in \mathbb{Z}_+^E$ satisfy $y_u + y_v + z_e \ge w_e$ for each edge $e = uv$.

Proof. Directly from the LP-duality equation

(21.16) $$\begin{aligned} &\max\{w^\mathsf{T}x \mid \mathbf{0} \le x \le c, Ax \le b\} \\ &= \min\{y^\mathsf{T}b + z^\mathsf{T}c \mid y \ge \mathbf{0}, z \ge \mathbf{0}, y^\mathsf{T}A + z^\mathsf{T} \ge w^\mathsf{T}\} \end{aligned}$$

(where A is the $V \times E$ incidence matrix of G), using the total unimodularity of A. ∎

21.5. Bipartite *b*-matching and *w*-vertex cover algorithmically

Algorithmically, optimization problems on b-matchings and w-vertex covers in bipartite graphs can be reduced to minimum-cost flow problems, and hence can be solved in strongly polynomial time.

Theorem 21.9. *Given a bipartite graph $G = (V, E)$, $b : V \to \mathbb{Z}_+$, $c : E \to \mathbb{Z}_+$, and $w : E \to \mathbb{Q}$, a b-matching $x \leq c$ maximizing $w^\mathsf{T} x$ can be found in strongly polynomial time. Similarly, a perfect b-matching $x \leq c$ minimizing $w^\mathsf{T} x$ can be found in strongly polynomial time.*

Proof. Let S and T be the colour classes of G, and orient the edges of G from S to T, giving the digraph D. Then b-matchings in G correspond to integer z-transshipments in D with $0 \leq z(v) \leq b(v)$ if $v \in T$ and $-b(v) \leq z(v) \leq 0$ if $v \in S$. Perfect b-matchings correspond to integer b'-transshipments, where $b'(v) := -b(v)$ if $v \in S$ and $b'(v) := b(v)$ if $v \in T$. Hence this theorem follows from Corollary 12.2d. ∎

Wagner [1958] (cf. Dantzig [1955]) observed that the capacitated version of the minimum-weight perfect b-matching problem can be reduced to the uncapacitated version, by a construction similar to that used in proving Theorem 21.4.

One similarly has for w-vertex covers:

Theorem 21.10. *Given a bipartite graph $G = (V, E)$, $b : V \to \mathbb{Q}_+$, $c : V \to \mathbb{Z}_+$, and $w : E \to \mathbb{Z}_+$, a w-vertex cover $y \leq c$ minimizing $y^\mathsf{T} b$ can be found in strongly polynomial time.*

Proof. By reduction to Corollary 12.2e. ∎

Although these results suggest a symmetry between matchings and vertex covers, we mention here that the nonbipartite version of Theorem 21.9 holds true (Section 32.4), but that finding a maximum-size stable set in a nonbipartite graph is NP-complete (see Section 64.2).

21.6. Transportation

The minimum-weight perfect b-matching problem is close to the classical transportation problem. Given a bipartite graph $G = (V, E)$ and a vector $b \in \mathbb{R}^V_+$, a *b-transportation* is a vector $x \in \mathbb{R}^E_+$ with

(21.17) $$x(\delta(v)) = b_v$$

for each $v \in V$. So a b-transportation is a fractional version of a perfect b-matching. Integer b-transportations are exactly the perfect b-matchings.

The following characterization of the existence of a b-transportation was shown (in a much more general form) by Rado [1943] — compare Corollary 21.1b:

Theorem 21.11. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{R}^V_+$. Then there exists a b-transportation if and only if $b(C) \geq \frac{1}{2} b(V)$ for each vertex cover C.*

Proof. Since the inequalities $b(C) \geq \frac{1}{2}b(V)$ (for vertex covers C), define a rational polyhedral cone, we can assume that b is rational, and hence, by scaling, that b is integer. Then the theorem follows from Corollary 21.1b. ∎

Note that, trivially, there exists a b-transportation if and only if b belongs to the convex cone in $\mathbb{R}^V$ generated by the incidence vectors of the edges of G. So Theorem 21.11 characterizes this cone.

A negative cycle criterion follows directly from the corresponding criterion for transshipments. For any b-transportation x in a bipartite graph $G = (V, E)$ and any cost function $c : E \to \mathbb{R}$, make the directed graph $D_x = (V, A)$ as follows. Let U and W be the colour classes of G. For each edge $e = uv$ of G, with $u \in U$ and $v \in W$, let A have an arc (u, v) of cost c_e, and, if $x_e > 0$, an arc (v, u).of cost $-c_e$. Then (Tolstoǐ [1930]):

Theorem 21.12. *x is a minimum-cost b-transportation if and only if D_x has no negative-cost directed circuits.*

Proof. Directly from Theorem 12.3. ∎

Transportations in a complete bipartite graph can be formulated in terms of matrices. Fixing vectors $a \in \mathbb{R}^m_+$ and $b \in \mathbb{R}^n_+$, an $m \times n$ matrix $X = (x_{i,j})$ is called a *transportation* if

(21.18)
$$\begin{array}{lll} \text{(i)} & x_{i,j} \geq 0 & i = 1, \ldots, m; j = 1, \ldots, n, \\ \text{(ii)} & \displaystyle\sum_{j=1}^{n} x_{i,j} = a_i & i = 1, \ldots, m, \\ \text{(iii)} & \displaystyle\sum_{i=1}^{m} x_{i,j} = b_j & j = 1, \ldots, n. \end{array}$$

Clearly, a transportation exists if and only if $\sum_i a_i = \sum_j b_j$.

Given an $m \times n$ 'cost' matrix $C = (c_{i,j})$, the cost of a transportation $X = (x_{i,j})$ is defined as $\sum_{i,j} c_{i,j}x_{i,j}$. Then the *transportation problem* (also called the *Hitchcock-Koopmans transportation problem*) is:

(21.19) given: vectors $a \in \mathbb{Q}^m_+$, $b \in \mathbb{Q}^n_+$ and an $m \times n$ 'cost' matrix $C = (c_{i,j})$,
find: a minimum-cost transportation.

So it is equivalent to solving the LP problem of minimizing $\sum_{i,j} c_{i,j}x_{i,j}$ over (21.18). The transportation problem formed a major impulse to introduce linear programming. Hitchcock [1941] and Dantzig [1951a] showed that the simplex method applies to the transportation problem.

The transportation problem is also a special case of the minimum-cost b-transshipment problem, and hence can be solved with the methods of Chapter 12. In particular, it is solvable in strongly polynomial time.

Linear programming also yields a min-max relation, originally due to Hitchcock [1941] (also implicit in Kantorovich [1939]):

Theorem 21.13 (Hitchcock's theorem)**.** *The minimum cost of a transportation is equal to the maximum value of* $y^{\mathsf{T}}a + z^{\mathsf{T}}b$*, where* $y \in \mathbb{R}^m$ *and* $z \in \mathbb{R}^n$ *such that* $y_i + z_j \leq c_{i,j}$ *for all* i, j*.*

Proof. This is LP-duality. ∎

(Hitchcock [1941] gave a direct proof.)

The transportation problem differs from the minimum-weight perfect b-matching problem in having a complete bipartite graph $K_{m,n}$ as underlying bipartite graph and in not requiring integrality of the output. This last however is not a restriction, as Dantzig [1951a] showed:

Theorem 21.14. *If* a *and* b *are integer, the transportation problem has an integer optimum solution* x*.*

Proof. Directly from the total unimodularity of the matrix underlying the system (21.18), which is the incidence matrix of the complete bipartite graph $K_{m,n}$. ∎

For a different proof, see the proof of Corollary 21.15a below.

Notes. Ford and Fulkerson [1955,1957b], Gleyzal [1955], Munkres [1957], and Egerváry [1958] described primal-dual methods for the transportation problem, and Ford and Fulkerson [1956a,1957a] extended it to the capacitated version.

If the a_i and b_j are small integers, the transportation problem can be reduced to the assignment problem, by 'splitting' each i into a_i or b_i copies. (This observation is due to Egerváry [1958], and in a different context to Tutte [1954b].)

21.6a. Reduction of transshipment to transportation

It is direct to transform a transportation problem to a transshipment problem. Orden [1955] observed a reverse reduction (similar to the reduction described in Section 16.7c). Indeed, let input $D = (V, A)$, $b \in \mathbb{R}^V$ and $k \in \mathbb{R}^A$ for the transshipment problem be given. Split each vertex v into two vertices v', v'' and replace each arc (u, v) by an arc (u', v''), with cost $k(u, v)$. Moreover, add arcs (v', v''), each with cost 0. Let $N := \sum_{v \in V} |b(v)|$. Define $b'(v') := -N$ and $b'(v'') := b(v) + N$. Then a minimum-cost b'-transshipment in the new structure gives a minimum-cost b-transshipment in the original structure. Since the new graph is bipartite with all edges oriented from one colour class to the other, we have a reduction to the transportation problem.

(Orden [1955] also gave an alternative reduction of the transshipment problem to the transportation problem. Let A' be the set of pairs (u, v) with $b_u < 0$ and $b_v > 0$ and with v is reachable in D from U. For each $(u, v) \in A'$, let $k'(u, v)$ be the length of a shortest $u - v$ path in D, taking k as length function. Then the (bipartite) transshipment problem for $D' := (V, A')$, b, and k' is equivalent to the original transshipment problem.)

Fulkerson [1960] gave the following reduction of the *capacitated* transshipment problem to the uncapacitated transportation problem. Let be given directed graph

$D = (V, A)$, $b \in \mathbb{R}^V$, a 'capacity' function $c \in \mathbb{R}^A$, and a 'cost' function $k \in \mathbb{R}^A$. Define $V' := V \cup A$ and $E' := \{\{a, v\} \mid a = (v, u) \text{ or } a = (u, v)\}$. Define $w(\{a, v\}) := k(a)$ if v is head of a, and $:= 0$ if v is tail of a. Let $b'(a) := c(a)$ and $b'(v) := b(v) + c(\delta^{\text{out}}(v))$. Then a minimum-cost b-transshipment subject to c corresponds to a minimum-cost b'-transportation. (More can be found in Wagner [1958].)

21.6b. The transportation polytope

Given $a \in \mathbb{R}^m_+$ and $b \in \mathbb{R}^n_+$, the *transportation polytope* is the set of all matrices $X = (x_{i,j})$ in $\mathbb{R}^{m \times n}$ satisfying (21.18). The transportation polytope was first studied by Hitchcock [1941]. The following result is due to Dantzig [1951a].

Theorem 21.15. *Let $X = (x_{i,j})$ belong to the transportation polytope. Then X is a vertex of the transportation polytope if and only if the set $F := \{ij \mid x_{i,j} > 0\}$ forms a forest in the complete bipartite graph $K_{m,n}$.*

Proof. If F contains a circuit $C = (i_0, j_1, i_1, j_2, i_2, \ldots, j_k, i_k)$, with $i_k = i_0$, define $Y = (y_{i,j})$ by: $y_{i,j} := 1$ if $(i, j) = (i_h, j_h)$ for some $h = 1, \ldots, k$, $y_{i,j} := -1$ if $(i, j) = (i_{h-1}, j_h)$ for some $h = 1, \ldots, k$, and $y_{i,j} := 0$ for all other (i, j). Then $X + \varepsilon Y$ belongs to the transportation polytope for any ε close enough to 0 (positive or negative), and hence X is not a vertex of the transportation polytope.

Conversely, if X is not a vertex of the transportation polytope, there exists a nonzero matrix $Y = (y_{i,j})$ such that $X + \varepsilon Y$ is in the transportation polytope for any ε close enough to 0 (positive or negative). Then Y satisfies $\sum_{j=1}^n y_{i,j} = 0$ for $i = 1, \ldots, m$ and $\sum_{i=1}^m y_{i,j} = 0$ for $j = 1, \ldots, n$. Since Y is nonzero, the set $F' := \{ij \mid y_{i,j} \neq 0\}$ contains a circuit. Since $F' \subseteq F$, it implies that F contains a circuit. ∎

This gives:

Corollary 21.15a. *If a and b are integer vectors, the transportation polytope is an integer polyhedron.*

Proof. By Theorem 21.15, for any vertex $X = (x_{i,j})$ of the transportation polytope, the set of pairs (i, j) with $x_{i,j}$ not an integer is a forest. Hence, if it is nonempty, this forest has an end edge, say (i, j). Assume without loss of generality that i has degree 1 in this forest. Then $x_{i,j}$ is equal to a_i minus $\sum_{j' \neq j} x_{i,j'}$, which is an integer as a_i and each of the $x_{i,j'}$ $(j' \neq j)$ is an integer. ∎

The dimension of the transportation polytope is easy to determine (Koopmans and Reiter [1951], Dulmage and Mendelsohn [1962], Klee and Witzgall [1968]):

Theorem 21.16. *If $a > 0$ and $b > 0$, the dimension of the transportation polytope is equal to $(m - 1)(n - 1)$.*

Proof. Let $X = (x_{i,j})$ be a vector in the relative interior of the transportation polytope. So $x_{i,j} > 0$ for all i, j. For each (i, j) with $i \in \{1, \ldots, m - 1\}$ and $j \in \{1, \ldots, n - 1\}$, we can correct any small perturbation of $x_{i,j}$ by a unique change of the $x_{i,n}$ and $x_{m,j}$. So the dimension of the transportation polytope is $(m-1)(n-1)$. ∎

Notes. Balinski [1974] (cf. Balinski and Rispoli [1993]) showed the Hirsch conjecture for some classes of transportation polytopes. For counting and estimating the number of vertices of transportation polytopes, see Simonnard and Hadley [1959], Demuth [1961], Wintgen [1964], Szwarc and Wintgen [1965], Klee and Witzgall [1968], Bolker [1972], and Ahrens [1981]. For counting facets, see Klee and Witzgall [1968].

Given $C = (c_{i,j}) \in \mathbb{R}^{m\times n}$, the *dual transportation polyhedron* is the set of all vectors $(u; v) \in \mathbb{R}^m \times \mathbb{R}^n$ satisfying[38]:

(21.20) $$\begin{aligned} &u_1 = 0\\ &u_i + v_j \geq c_{i,j} \quad i = 1,\ldots,m; j = 1,\ldots,n. \end{aligned}$$

(The condition $u_1 = 0$ is added for normalization.) It is easy to see that the dimension of the dual transportation polyhedron is $m + n - 1$, and that $(u; v)$ satisfying (21.20) is a vertex of the dual transportation polyhedron if and only if the graph with vertex set $\{p_1,\ldots,p_m,q_1,\ldots,q_n\}$ and edge set $\{\{p_i,q_j\} \mid u_i + v_j = c_{i,j}\}$ is connected.

Balinski [1984] showed with the 'signature method' that the diameter of the dual transportation polyhedron is at most $(m-1)(n-1)$, thus proving the Hirsch conjecture for this class of polyhedra.

Balinski and Russakoff [1984] characterized vertices and higher-dimensional faces of dual transportation polyhedra. More can be found in Zhu [1963], Balinski [1983], and Kleinschmidt, Lee, and Schannath [1987].

21.7. b-edge covers and w-stable sets

Exchanging $\leq$ and $\geq$ appropriately in the definitions of b-matchings and w-vertex covers gives the b-edge covers and the w-stable sets. These concepts again turn out to be each others dual.

Let $G = (V, E)$ be a graph, with $V \times E$ incidence matrix A. For $b : V \to \mathbb{Z}_+$, a *b-edge cover* is a function $x : E \to \mathbb{Z}_+$ such that for each vertex v of G:

(21.21) $$x(\delta(v)) \geq b_v.$$

In other words, x is a b-edge cover if and only if x is an integer vector satisfying $x \geq \mathbf{0}$, $Ax \geq b$. So if $b = \mathbf{1}$, then $\{0,1\}$-valued b-edge covers are precisely the incidence vectors of edge covers.

For $w : E \to \mathbb{Z}_+$, a *w-stable set* is a function $y : V \to \mathbb{Z}_+$ such that for each edge $e = uv$ of G:

(21.22) $$y_u + y_v \leq w_e.$$

[38] We write $(u; v)$ for $\begin{pmatrix} u \\ v \end{pmatrix}$.

In other words, y is a w-stable set if and only if y is an integer vector satisfying $y \geq \mathbf{0}$, $y^{\mathsf{T}}A \leq w^{\mathsf{T}}$. So if $w = \mathbf{1}$, then $\{0,1\}$-valued w-stable sets are precisely the incidence vectors of stable sets.

In this case, b-edge covers and w-stable sets are related by the following LP-duality equation:

(21.23) $$\min\{w^{\mathsf{T}}x \mid x \geq \mathbf{0}, Ax \geq b\} = \max\{y^{\mathsf{T}}b \mid y \geq \mathbf{0}, y^{\mathsf{T}}A \leq w^{\mathsf{T}}\}.$$

Since A is totally unimodular (Theorem 18.2), both optima are attained by integer vectors. This gives (where the *w-weight* of a vector x equals $w^{\mathsf{T}}x$ and the *b-weight* of a vector y equals $y^{\mathsf{T}}b$):

Theorem 21.17. *Let $G = (V, E)$ be a bipartite graph and let $b : V \to \mathbb{Z}_+$ and $w : E \to \mathbb{Z}_+$. Then the minimum w-weight $w^{\mathsf{T}}x$ of a b-edge cover x is equal to the maximum b-weight of a w-stable set.*

Proof. See above. ∎

Taking $b = \mathbf{1}$, we obtain Corollary 19.5a. For $w = \mathbf{1}$, we get a min-max relation for minimum-size b-edge cover:

Corollary 21.17a. *Let $G = (V, E)$ be a bipartite graph and let $b : V \to \mathbb{Z}_+$. Then the minimum size of a b-edge cover is equal to the maximum b-weight of a stable set.*

Proof. This is the special case $w = \mathbf{1}$ of Theorem 21.17. ∎

Again, an alternative way of proving this is by derivation from the Kőnig-Rado edge cover theorem (Theorem 19.4): Split each vertex v into b_v copies, replace each edge uv by $b_u b_v$ edges connecting the b_u copies of u with the b_v copies of v.

21.8. The b-edge cover and the w-stable set polyhedron

The total unimodularity of the incidence matrix of a bipartite graph also gives descriptions of the corresponding polyhedra.

The *b-edge cover polyhedron* is the convex hull of the b-edge covers. For bipartite graphs it is determined by:

(21.24)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(\delta(v)) \geq b_v$ for each $v \in V$.

Theorem 21.18. *The b-edge cover polyhedron of a bipartite graph $G = (V, E)$ is determined by* (21.24).

Proof. Directly from the facts that system (21.24) amounts to $x \geq \mathbf{0}, Ax \geq b$ and that A is totally unimodular. ∎

This extends Theorem 19.6 on the edge cover polytope.

Similarly, the *w-stable set polyhedron*, being the convex hull of the w-stable sets, is, for bipartite graphs, determined by:

(21.25) (i) $y_v \geq 0$ for each $v \in V$,
(ii) $y_u + y_v \leq w_e$ for each $e = uv \in E$.

Theorem 21.19. *The w-stable set polyhedron of a bipartite graph is determined by* (21.25)*.*

Proof. Directly from the total unimodularity of the incidence matrix of a bipartite graph. ∎

This generalizes the necessity part of Theorem 19.7.

21.9. Simple b-edge covers

Again, call a vector x *simple* if it is $\{0,1\}$-valued. Then a simple b-edge cover corresponds to a set F of edges with $\deg_F(v) \geq b_v$ for each $v \in V$. We will identify the vector and the set. Note that a simple b-edge cover can exist only if $b_v \leq \deg(v)$ for each vertex v.

It is easy to derive the following min-max relation for simple b-edge covers from Theorem 21.4 on the maximum size of a simple b-matching ($E[X]$ denote the set of edges spanned by X):

Theorem 21.20. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$ with $b_v \leq \deg(v)$ for each vertex v. Then the minimum size of a simple b-edge cover in G is equal to the maximum value of $b(X) - |E[X]|$ taken over $X \subseteq V$.*

Proof. Define $b'(v) := \deg(v) - b(v)$ for each vertex v. Then a subset F of E is a simple b-edge cover if and only if $E \setminus F$ is a simple b'-matching. By Theorem 21.4, the maximum size of a simple b'-matching is equal to the minimum value of $b'(V \setminus X) + |E[X]|$ taken over $X \subseteq V$. Hence the minimum size of a simple b-edge cover is equal to the maximum value of

(21.26)
$$\begin{aligned}
&|E| - b'(V \setminus X) - |E[X]| = |E| - \sum_{v \in V \setminus X} (\deg(v) - b(v)) - |E[X]| \\
&= |E| - 2|E[V \setminus X]| - |\delta(X)| + b(V \setminus X) - |E[X]| \\
&= b(V \setminus X) - |E[V \setminus X]|,
\end{aligned}$$

taken over $X \subseteq V$. ∎

Alternatively, the theorem follows from the fact that both optima in the LP-duality equation (where A is the $V \times E$ incidence matrix of G):

(21.27)
$$\begin{aligned}
&\min\{\mathbf{1}^\mathsf{T} x \mid \mathbf{0} \leq x \leq \mathbf{1}, Ax \geq b\} \\
&= \max\{y^\mathsf{T} b - z^\mathsf{T} \mathbf{1} \mid y \geq \mathbf{0}, z \geq \mathbf{0}, y^\mathsf{T} A - z^\mathsf{T} \leq \mathbf{1}^\mathsf{T}\}
\end{aligned}$$

have integer optimum solutions, since A is totally unimodular.

If b is equal to a constant k, Theorem 21.20 amounts to (with the help of the edge cover variant of Kőnig's edge-colouring theorem (Theorem 20.5)):

Corollary 21.20a. *Let $G = (V, E)$ be a bipartite graph and let $k \in \mathbb{Z}_+$. Then the minimum size of the union of k disjoint edge covers is equal to the maximum value of $k|X| - |E[X]|$ taken over $X \subseteq V$.*

Proof. Apply Theorem 21.20 to $b_v := k$ for all $v \in V$. We obtain a formula for the maximum size of a subset F of E with $\deg_F(v) \geq k$ for all $v \in V$. By Theorem 20.5, F is the union of k disjoint edge covers. ∎

By the total unimodularity of the incidence matrix of bipartite graphs, the *simple b-edge cover polytope* (the convex hull of the simple b-edge covers) of a bipartite graph $G = (V, E)$ is determined by:

(21.28)
$$\begin{array}{ll} 0 \leq x_e \leq 1 & \text{for each } e \in E, \\ x(\delta(v)) \geq b_v & \text{for each } v \in V. \end{array}$$

LP-duality also gives a min-max formula for the minimum weight of simple b-edge covers:

Theorem 21.21. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$ and $w \in \mathbb{Z}_+^E$. Then the minimum weight $w^{\mathsf{T}}x$ of a simple b-edge cover x is equal to the maximum value of*

(21.29)
$$\sum_{v \in V} y_v b_v - \sum_{e \in E} z_e$$

where $y \in \mathbb{Z}_+^V$ and $z \in \mathbb{Z}_+^E$ with $y_u + y_v - z_e \leq w_e$ for each edge $e = uv$.

Proof. Directly from the LP-duality equation

(21.30)
$$\begin{aligned} &\min\{w^{\mathsf{T}}x \mid \mathbf{0} \leq x \leq \mathbf{1}, Ax \geq b\} \\ &= \max\{y^{\mathsf{T}}b - z^{\mathsf{T}}\mathbf{1} \mid y \geq \mathbf{0}, z \geq \mathbf{0}, y^{\mathsf{T}}A - z^{\mathsf{T}} \leq w^{\mathsf{T}}\} \end{aligned}$$

(where A is the $V \times E$ incidence matrix of G), using the total unimodularity of A. ∎

21.10. Capacitated b-edge covers

If we require that a b-edge cover x satisfies $x \leq c$ for some 'capacity' function $c : E \to \mathbb{Z}_+$, we speak of a *capacitated* b-edge cover. So simple b-edge covers correspond to capacitated b-edge covers with $c = \mathbf{1}$.

Theorem 21.22. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$ with $c(\delta(v)) \geq b_v$ for each $v \in V$. Then the minimum size of a b-edge cover $x \leq c$ is equal to*

(21.31) $$\max_{X \subseteq V} b(X) - c(E[X]).$$

Proof. The proof is similar to that of Theorem 21.20. ∎

Alternatively, we can reduce this theorem to Theorem 21.20, by replacing each edge e by c_e parallel edges, or we can use total unimodularity similarly to (21.27).

Theorem 21.23. *Let $G = (V, E)$ be a bipartite graph and let $b \in \mathbb{Z}_+^V$ and $c, w \in \mathbb{Z}_+^E$. Then the minimum weight $w^\mathsf{T} x$ of a b-edge cover $x \leq c$ is equal to the maximum value of*

(21.32) $$\sum_{v \in V} y_v b_v - \sum_{e \in E} z_e c_e$$

where $y \in \mathbb{Z}_+^V$ and $z \in \mathbb{Z}_+^E$ with $y_u + y_v - z_e \leq w_e$ for each edge $e = uv$.

Proof. Directly from the LP-duality equation

(21.33) $$\begin{aligned} &\min\{w^\mathsf{T} x \mid \mathbf{0} \leq x \leq c, Ax \geq b\} \\ &= \max\{y^\mathsf{T} b - z^\mathsf{T} c \mid y \geq \mathbf{0}, z \geq \mathbf{0}, y^\mathsf{T} A - z^\mathsf{T} \leq w^\mathsf{T}\} \end{aligned}$$

(where A is the $V \times E$ incidence matrix of G), using the total unimodularity of A. ∎

By the total unimodularity of the incidence matrix of G, the convex hull of b-edge covers $x \leq c$ of a bipartite graph G is determined by the inequalities

(21.34) (i) $0 \leq x_e \leq c_e$ for each $e \in E$,
(ii) $x(\delta(v)) \geq b_v$ for each $v \in V$.

21.11. Relations between b-matchings and b-edge covers

Like for matchings and edge covers, there is also a close relation between maximum-size b-matchings and minimum-size b-edge covers, as was shown by Gallai [1959a]. This gives a connection between Corollaries 21.1a and 21.17a.

Let $G = (V, E)$ be an undirected graph without isolated vertices, and let $b \in \mathbb{Z}_+^V$. Define:

(21.35) $\nu_b(G) :=$ the maximum size of a b-matching,
$\rho_b(G) :=$ the minimum size of a b-edge cover.

Theorem 21.24. *Let $G = (V, E)$ be an undirected graph without isolated vertices, and let $b \in \mathbb{Z}_+^V$. Then*

(21.36) $$\nu_b(G) + \rho_b(G) = b(V).$$

Proof. This can be reduced to Gallai's theorem (Theorem 19.1), by splitting each vertex v into b_v copies, and replacing each edge $e = uv$ by $b_u b_v$ edges connecting the b_u copies of u with the b_v copies of v. ∎

A direct proof of the previous theorem is given in the proof of the following theorem, also due to Gallai [1959a]:

Theorem 21.25. *Let $G = (V, E)$ be an undirected graph and let $b \in \mathbb{Z}_+^V$. Then for each maximum-size b-matching x there is a minimum-size b-edge cover y with $x \leq y$. Conversely, for each minimum-size b-edge cover y there is a maximum-size b-matching x with $x \leq y$.*

Proof. Let x be a maximum-size b-matching. For each vertex v of G, increase the value of x on some edge incident with v, by $b_v - x(\delta(v))$. We obtain a b-edge cover y satisfying

$$y(E) = x(E) + \sum_{v \in V} (b_v - x(\delta(v))) = b(V) - x(E). \tag{21.37}$$

Conversely, let y be a minimum-size b-edge cover. For each vertex v of G, decrease the value of y on edges incident with v, by a total amount of $y(\delta(v)) - b_v$ (as long as $y \geq \mathbf{0}$). We obtain a b-matching x satisfying

$$x(E) \geq y(E) - \sum_{v \in V} (y(\delta(v)) - b_v) = b(V) - y(E). \tag{21.38}$$

(21.37) and (21.38) imply that the y (x, respectively) obtained from x (y, respectively) is optimum, thus showing the theorem, and also showing (21.36). ∎

In a bipartite graph, a minimum-size b-edge cover and a maximum-weight stable set can be found in strongly polynomial time, by reduction to Theorem 21.9:

Corollary 21.25a. *Given a bipartite graph $G = (V, E)$ and $b \in \mathbb{Z}_+^V$, a minimum-size b-edge cover and a maximum b-weight stable set can be found in strongly polynomial time.*

Proof. Since stable sets are exactly the complements of vertex covers, finding a maximum b-weight stable sets is directly reduced to finding a minimum b-weight vertex cover. The construction given in the proof of Theorem 21.25 implies that a maximum-size b-matching gives a minimum-size b-edge cover in polynomial time. So Theorem 21.9 gives the present corollary. ∎

Moreover, for the weighted case:

Theorem 21.26. *A minimum-weight capacitated b-edge cover in a bipartite graph can be found in strongly polynomial time.*

Proof. Directly from Corollary 12.2d, by orienting the edges from one colour class to the other. ∎

21.12. Upper and lower bounds

We finally consider upper *and* lower bounds. That is, for a graph $G = (V, E)$ and $a, b \in \mathbb{R}^V$ and $d, c \in \mathbb{R}^E$, we consider vectors $x \in \mathbb{R}^E$ satisfying:

$$\text{(21.39)} \quad \begin{array}{lll} \text{(i)} & d_e \leq x_e \leq c_e & \text{for each } e \in E, \\ \text{(ii)} & a_v \leq x(\delta(v)) \leq b_v & \text{for each } v \in V, \end{array}$$

If integer, x is both a b-matching and an a-edge cover.

The optimization problem can be reduced again to minimum-cost circulation, and hence:

Theorem 21.27. *Given $w : E \to \mathbb{Q}$, an integer vector x maximizing $w^\mathsf{T} x$ over* (21.39) *can be found in strongly polynomial time.*

Proof. This is a special case of Corollary 12.2d, by orienting the edges of G from one colour class to the other. ∎

Corresponding min-max and polyhedral characterizations directly follow from LP-duality and the total unimodularity of the incidence matrix of G. We formulate them for existence and optimum size of solutions of (21.39).

The following was formulated by Kellerer [1964]:

Theorem 21.28. *Let $G = (V, E)$ be a bipartite graph and let $a, b \in \mathbb{Z}^V$ and $d, c \in \mathbb{Z}^E$ with $a \leq b$ and $d \leq c$. Then there exists an $x \in \mathbb{Z}^E$ satisfying* (21.39) *if and only if for each $X \subseteq V$ one has*

$$\text{(21.40)} \quad \begin{array}{l} c(E[X]) - d(E[V \setminus X]) \\ \geq \max\{a(S \cap X) - b(T \setminus X), a(T \cap X) - b(S \setminus X)\}, \end{array}$$

where S and T are the colour classes of G.

Proof. From Corollary 11.2i, by orienting all edges from S to T and taking $U := (S \setminus X) \cup (T \cap X)$. ∎

This theorem has several special cases. For $d = \mathbf{0}$ it implies the following result due to Fulkerson [1959a] (a generalization of Theorem 16.8):

Corollary 21.28a. *Let $G = (V, E)$ be a bipartite graph with colour classes S and T, let $a, b \in \mathbb{Z}_+^V$ with $a \leq b$, and let $c \in \mathbb{Z}_+^E$. Then there is a vector $x \leq c$ that is both a b-matching and an a-edge cover if and only if there exist $y \in \mathbb{Z}_+^E$ and $z \in \mathbb{Z}_+^E$ with $y \leq c$ and $z \leq c$, such that*

$$\text{(21.41)} \quad \begin{array}{l} y(\delta(v)) \leq b_v \text{ and } z(\delta(v)) \geq a_v \text{ for each } v \in S \text{ and} \\ y(\delta(v)) \geq a_v \text{ and } z(\delta(v)) \leq b_v \text{ for each } v \in T. \end{array}$$

Proof. Note that (21.40) can be decomposed into two inequalities, one involving $a|S$ and $b|T$ only, the other involving $a|T$ and $b|S$ only[39]. This gives the present corollary. ∎

The special case $d = \mathbf{0}$, $c = \mathbf{1}$ is:

Corollary 21.28b. *Let $G = (V, E)$ be a bipartite graph with colour classes S and T and let $a, b \in \mathbb{Z}_+^V$ with $a \leq b$. Then E has a subset F that is both a b-matching and an a-edge cover if and only if E has subsets F' and F'' such that F' contains at least a_v edges covering v if $v \in S$ and at most b_v edges covering v if $v \in T$, and F'' contains at least a_v edges covering v if $v \in T$ and at most b_v edges covering v if $v \in S$.*

Proof. Directly from Corollary 21.28a by taking $c = \mathbf{1}$. ∎

A min-max relation for such vectors can be derived from Hoffman's circulation theorem (Theorem 11.2):

Theorem 21.29. *Let $G = (V, E)$ be a bipartite graph and let $a, b \in \mathbb{Z}^V$ and $d, c \in \mathbb{Z}^E$, such that there exists an $x \in \mathbb{Z}^E$ satisfying* (21.39). *Then the minimum size of such a vector x is equal to*

(21.42) $$\max_{Z \subseteq V}(a(Z) - c(E[Z]) + d(E[V \setminus Z])),$$

while the maximum size of such a vector x is equal to

(21.43) $$\min_{Z \subseteq V}(c(E[V \setminus Z]) - d(E[Z]) + b(Z)).$$

For each integer value τ between (21.42) *and* (21.43) *there exists such a vector x of size τ.*

Proof. Choose $\tau \in \mathbb{Z}$. Make a directed graph $D = (V, A)$ as follows.

Let S and T be the colour classes of G. Orient each edge of G from S to T. Add new vertices s and t. For each $v \in S$, make an arc from s to v, with $d(s, v) := a_v$ and $c(s, v) := b_v$. For each $v \in T$, make an arc from v to t, with $d(v, t) := a_v$ and $c(v, t) := b_v$. Finally, make an arc from t to s with $d(t, s) := c(t, s) := \tau$.

It suffices to show that D has a circulation x satisfying $d \leq x \leq c$ if and only if τ is between (21.42) and (21.43). We do this by using Hoffman's circulation theorem. Choose a subset X of the vertex set of D. Consider Hoffman's condition:

(21.44) $$d(\delta^{\text{in}}(X)) \leq c(\delta^{\text{out}}(X)).$$

Since by assumption some vector x satisfying (21.39) exists, (21.44) holds if $s, t \in X$ or $s, t \notin X$ (as ignoring the bounds on (t, s) there is a circulation).

If $s \in X$ and $t \notin X$, we have

[39] $f|X$ denotes the restriction of a function f to a set X.

(21.45) $$d(\delta^{\text{in}}(X)) = \tau + d(E[(S \setminus X) \cup (T \cap X)])$$

and

(21.46) $$c(\delta^{\text{out}}(X)) = b(S \setminus X) + c(E[(S \cap X) \cup (T \setminus X)]) + b(T \cap X).$$

Hence (21.44) for such X is equivalent to

(21.47) $$\tau \le b(Z) + c(E[V \setminus Z]) - d(E[Z])$$

for all $Z \subseteq V$ (take $Z = (S \setminus X) \cup (T \cap X)$). That is, to τ being at most (21.43).

If $t \in X$ and $s \notin X$, we have

(21.48) $$d(\delta^{\text{in}}(X)) = a(S \cap X) + d(E[(S \setminus X) \cup (T \cap X)]) + a(T \setminus X)$$

and

(21.49) $$c(\delta^{\text{out}}(X)) = \tau + c(E[(S \cap X) \cup (T \setminus X)]).$$

Hence (21.44) for such X is equivalent to

(21.50) $$\tau \ge a(Z) - c(E[Z]) + d(E[V \setminus Z])$$

for all $Z \subseteq V$ (take $Z = (S \cap X) \cup (T \setminus X)$). That is, to τ being at least (21.42). ∎

A special case is the following theorem of Folkman and Fulkerson [1969]:

Corollary 21.29a. *Let $G = (V, E)$ be a bipartite graph, let $a, b \in \mathbb{Z}_+^V$, and let $\tau \in \mathbb{Z}_+$. Then E has a subset F with $a_v \le \deg_F(v) \le b_v$ for each $v \in V$ and with $|F| = \tau$ if and only if*

(21.51) $$|E[Z]| \ge \max\{a(Z) - \tau, \tau - b(V \setminus Z), a(S \cap Z) - b(T \setminus Z), a(T \cap Z) - b(S \setminus Z)\}$$

for each $Z \subseteq V$, where S and T are the colour classes of G.

Proof. Directly from Theorems 21.28 and 21.29. ∎

21.13. Further results and notes

21.13a. Complexity survey on weighted bipartite *b*-matching and transportation

Complexity survey for weighted b-matching in bipartite graphs ($*$ indicates an asymptotically best bound in the table):

$O(n^4 B)$	Munkres [1957]
$O(\beta \cdot \mathrm{MF}(n, m, B))$	Ford and Fulkerson [1955,1957b]

≫

continued

	$O(n^2 mB)$	Iri [1960]
*	$O(\beta \cdot \mathrm{SP}_+(n,m,W))$	Edmonds and Karp [1970]
	$O(nW \cdot \mathrm{MF}(n,m,B))$	Edmonds and Karp [1972]
*	$O(m \log B \cdot \mathrm{SP}_+(n,m,W))$	Edmonds and Karp [1972]
	$O(nm \log(nB))$	Dinits [1973a]
	$O(n \log \beta \cdot \mathrm{SP}_+(n,m,W))$	Lawler [1976b]
	$O(n \log W \cdot \mathrm{MF}(n,m,B))$	Röck [1980]
	$O(m^2 \log n \cdot \mathrm{MF}(n,m,B))$	Tardos [1985a]
*	$O(\beta^{3/4} m \log W)$	Gabow [1985b]
*	$O(\beta^{1/2} n^{1/3} m \log W)$	Gabow [1985b] *for simple graphs*
	$O(n^2 \log n \cdot \mathrm{SP}_+(n,m,W))$	Galil and Tardos [1986,1988]
	$O(nm \log(n^2/m) \log(nW))$	Goldberg and Tarjan [1987,1990]
	$O(n \log n(m + n \log n))$	Orlin [1988,1993]
*	$O((\beta^{1/2} m + \beta \log \beta) \log(nW))$	Gabow and Tarjan [1989]
*	$O(n_1 m + n_1^3 \log(n_1 W))$	Ahuja, Orlin, Stein, and Tarjan [1994]
*	$O(n_1 m \log(2 + \frac{n_1^2}{m} \log(n_1 W)))$	Ahuja, Orlin, Stein, and Tarjan [1994]
*	$O(n \log n(m + n_1 \log n_1))$	Kleinschmidt and Schannath [1995]

Here $B := \|b\|_\infty$, $\beta := \|b\|_1$, $W := \|w\|_\infty$ (assumed to be integer), and $n_1 := \min\{|S|,|T|\}$, where S and T are the colour classes of the bipartite graph. By $\mathrm{SP}_+(n,m,W)$ we denote the time required for solving a shortest path problem in a digraph with n vertices, m arcs, and nonnegative integer length function l with $\|l\|_\infty \leq W$. $\mathrm{MF}(n,m,B)$ denotes the time required to solve a maximum flow problem in a digraph with n vertices, m arcs, and integer capacity function c with $\|c\|_\infty \leq B$.

Complexity survey for the uncapacitated transportation problem:

	$O(n^4 B)$	Munkres [1957]
	$O(\beta \cdot \mathrm{MF}(n,n^2,B))$	Ford and Fulkerson [1955,1957b]
*	$O(n^3 \log(nB))$	Edmonds and Karp [1972], Dinits [1973a]
	$O(n^4 W)$	Edmonds and Karp [1972]
*	$O(\beta^{3/4} n^2 \log W)$	Gabow [1985b]
*	$O(\beta^{1/2} n^{7/3} \log W)$	Gabow [1985b]
	$O(n^4 \log n \cdot \mathrm{MF}(n,n^2,W))$	Tardos [1985a]
	$O(n^4 \log n)$	Galil and Tardos [1986,1988]
	$O(n^3 \log(nW))$	Goldberg and Tarjan [1987,1990]

$\gg$

continued

	$O(n^3 \log n)$	Orlin [1988,1993]
$*$	$O((\beta^{1/2}n^2 + \beta \log \beta) \log(nW))$	Gabow and Tarjan [1989]
$*$	$O(n_1 n^2 + n_1^3 \log(n_1 W))$	Ahuja, Orlin, Stein, and Tarjan [1994]
$*$	$O(n_1 n^2 \log(2 + \frac{n_1^2}{n^2} \log(n_1 W)))$	Ahuja, Orlin, Stein, and Tarjan [1994]
$*$	$O(n_1^2 n \log^2 n)$	Tokuyama and Nakano [1992,1995]
$*$	$O(n_1 n^2 \log n)$	Kleinschmidt and Schannath [1995]

Complexity survey for weighted capacitated b-matching in bipartite graphs:

$*$	$O(n \max\{B, C\} \cdot \mathrm{SP}_+(n, m, W))$	Edmonds and Karp [1970]
	$O(nW \cdot \mathrm{MF}(n, m, \max\{B, C\}))$	Edmonds and Karp [1972]
$*$	$O(n \log \beta \cdot \mathrm{SP}_+(n, m, W))$	Lawler [1976b]
	$O(n \log W \cdot \mathrm{MF}(n, m, \max\{B, C\}))$	Röck [1980]
	$O(m^2 \log n \cdot \mathrm{MF}(n, m, \max\{B, C\}))$	Tardos [1985a]
	$O(\beta^{3/4} mC \log W)$	Gabow [1985b]
	$O(\beta^{1/2} n^{1/3} mC \log W)$	Gabow [1985b]
	$O(n^2 \log n \cdot \mathrm{SP}_+(n, m, W))$	Galil and Tardos [1986,1988]
$*$	$O(nm \log(n^2/m) \log(nW))$	Goldberg and Tarjan [1987,1990]
$*$	$O(m \log n \cdot \mathrm{SP}_+(n, m, W))$	Orlin [1988,1993]
$*$	$O(\beta^{1/2} mC \log(nW))$	Gabow and Tarjan [1988b,1989]
$*$	$O(n^{2/3} mC^{4/3} \log(nW))$	Gabow and Tarjan [1989]
$*$	$O((\beta^{1/2} m + \beta \log \beta) \log(nW))$	Gabow and Tarjan [1989]
$*$	$O((nm + \beta \log \beta) \log(nW))$	Gabow and Tarjan [1989]
$*$	$O(n_1 m + n_1^3 \log(n_1 W))$	Ahuja, Orlin, Stein, and Tarjan [1994]
$*$	$O(n_1 m \log(2 + \frac{n_1^2}{m} \log(n_1 W)))$	Ahuja, Orlin, Stein, and Tarjan [1994]

Here $C := \|c\|_\infty$.

Complexity survey for the capacitated transportation problem:

	$O(n^4 W)$	Edmonds and Karp [1972]
	$O(n^3 \log \max\{B, C\})$	Edmonds and Karp [1972]
	$O(n^4 \log W)$	Röck [1980]
	$O(n^4 \log n \cdot \mathrm{MF}(n, n^2, \max\{B, C\}))$	Tardos [1985a]
$*$	$O(n^2 B)$	Gabow [1985b]

≫

continued

	$O(\beta^{3/4}n^2C\log W)$	Gabow [1985b]
	$O(\beta^{1/2}n^{7/3}C\log W)$	Gabow [1985b]
*	$O(n^3\log(nW))$	Goldberg and Tarjan [1987,1990]
*	$O(n^4\log n)$	Galil and Tardos [1986,1988], Orlin [1988,1993]
*	$O(n^2\beta^{1/2}C\log(nW))$	Gabow and Tarjan [1988b,1989]
*	$O(n^{8/3}C^{4/3}\log(nW))$	Gabow and Tarjan [1989]
*	$O((\beta^{1/2}n^2+\beta\log\beta)\log(nW))$	Gabow and Tarjan [1989]
*	$O(n_1n^2+n_1^3\log(n_1W))$	Ahuja, Orlin, Stein, and Tarjan [1994]
*	$O(n_1n^2\log(2+\frac{n_1^2}{n^2}\log(n_1W)))$	Ahuja, Orlin, Stein, and Tarjan [1994]

Let $G=(V,E)$ be a bipartite graph, with colour classes S and T say. The existence of a perfect (capacitated) b-matching can be reduced quite directly to the problem of finding a maximum $s-t$ flow in the digraph obtained from G by adding two new vertices s and t, orienting each edge from S to T, and adding an arc (s,s') for each $s'\in S$, and adding an arc (t',t) for each $t'\in T$. Similarly, a maximum (capacitated) b-matching can be found.

It implies that if $\mathrm{MF}(n,m,C)$ is the running time of a maximum flow algorithm for inputs with n vertices, m arcs, and integer capacity function c with $\|c\|_\infty\le C$, then a maximum-size (capacitated) b-matching can be found in time $O(\mathrm{MF}(n,m,C))$, for bipartite graphs with n vertices, m edges and $b\in\mathbb{Z}^V$ satisfying $\|b\|_\infty\le C$ (and capacity function $c\in\mathbb{Z}^E$ satisfying $\|c\|_\infty\le C$).

In some cases, one can obtain better bounds, in particular if one of the colour classes is considerably smaller than the other. To this end, let $n_1:=\min\{|S|,|T|\}$. Implementing the shortest augmenting path rule described in Section 10.5, then gives an $O(n_1m^2)$ running time, since a shortest $s-t$ path has length at most $2n_1+1=O(n_1)$, implying that the number of iterations is bounded by n_1m.

Similarly, the blocking flow method of Dinits [1970] described in Section 10.6 can be performed in $O(n_1^2m)$ time, since the bound in Theorem 10.6 becomes $O(n_1m)$, while there are $O(n_1)$ blocking flow iterations. The method of Karzanov [1974] can be sharpened to $O(n_1^2n)$, as was shown by Gusfield, Martel, and Fernández-Baca [1987]. Ahuja, Orlin, Stein, and Tarjan [1994] gave a method taking the minimum of $O(n_1m+n_1^3)$, $O(n_1m+n_1^2\sqrt{m})$, $O(n_1m+n_1^2\sqrt{\log C})$, and $O(n_1m\log(2+\frac{n_1^2}{m}))$ time.

For the special case where $b_u=1$ for each u in the smaller colour class, Adel'son-Vel'skiĭ, Dinits, and Karzanov [1975] gave an $O(n_1^{5/3}n)$ algorithm for finding a b-factor.

21.13b. The matchable set polytope

Let $G = (V, E)$ be a graph. A subset X of V is called *matchable*, if G has a matching M with $\bigcup M = X$; that is, if the subgraph $G[X]$ of G induced by X has a perfect matching.

The *matchable set polytope* of G is the convex hull of the incidence vectors of matchable sets. Theorem 21.11 implies a characterization of the matchable set polytope in case G is bipartite.

For any graph, each vector in the matchable set polytope trivially satisfies:

$$\text{(21.52)}\qquad \begin{array}{lll} \text{(i)} & 0 \leq x_v \leq 1 & \text{for each } v \in V, \\ \text{(ii)} & x(C) \leq \frac{1}{2}x(V) & \text{for each stable set } C. \end{array}$$

If G is bipartite, this set of inequalities determines the matchable set polytope, a result of Balas and Pulleyblank [1983]:

Theorem 21.30. *If G is bipartite, the matchable set polytope is determined by (21.52).*

Proof. Let x satisfy (21.52). By Theorem 21.11, there exists an x-transportation $y \in \mathbb{R}^E_+$. That is, $x = Ay$, where A is the $V \times E$ incidence matrix of G.

As x satisfies (21.52)(i), y satisfies $y \geq \mathbf{0}$, $Ay \leq \mathbf{1}$. So, by Corollary 18.1b, y belongs to the matching polytope of G. So y is a convex combination of vectors χ^M, where M ranges over the matchings in G. Then x is a convex combination of the vectors χ^S, where S is matchable (that is, the set of vertices covered by some matching M). This follows from the fact that $A\chi^M = \chi^S$ if M is a matching and S is the set of vertices covered by M.

So x belongs to the matchable set polytope. ∎

It is easy to check that only for bipartite graphs the matchable set polytope is determined by (21.52).

Note that for bipartite graphs $G = (V, E)$, by Theorem 21.11, condition (21.52)(ii) is equivalent to x belonging to the convex cone generated by the incidence vectors (in $\mathbb{R}^V$) of edges, considered as subsets of V.

Qi [1987] gave an algorithm for the separation problem for the matchable set polytope of a bipartite graph. For more on the matchable set polytope, see Balas and Pulleyblank [1983] and Section 25.5d.

21.13c. Existence of matrices

If the bipartite graph is a complete bipartite graph, theorems on the existence of b-matchings and b-edge covers amount to theorems on the existence of matrices obeying prescribed bounds on the row and column sums. This gives the following theorem of Gale [1956,1957] and Ryser [1957]:

Theorem 21.31 (Gale-Ryser theorem)**.** *Let $a, b \in \mathbb{Z}^m_+$ and $a', b' \in \mathbb{Z}^n_+$ with $a \leq b$ and $a' \leq b'$ and satisfying $a_1 \geq a_2 \geq \cdots \geq a_m$ and $a'_1 \geq a'_2 \geq \cdots \geq a'_n$. Then there exists a $\{0, 1\}$-valued $m \times n$ matrix with ith row sum between a_i and b_i $(i = 1, \ldots, m)$ and jth column sum between a'_j and b'_j $(j = 1, \ldots, n)$ if and only if*

(21.53) (i) $\sum_{i=1}^{k} a_i \leq \sum_{j=1}^{n} \min\{k, b'_j\}$ *for all* $k = 1, \ldots, m$,

(ii) $\sum_{j=1}^{k} a'_j \leq \sum_{i=1}^{m} \min\{k, b_i\}$ *for all* $k = 1, \ldots, n$.

Proof. *Necessity.* Consider any inequality in (21.53)(i). The number of 1's in rows $1, \ldots, k$ is at least the left-hand side and at most the right-hand side. This proves necessity of the inequality. Necessity of the inequalities (ii) is shown similarly.

Sufficiency. This follows from Theorem 21.28 applied to the complete bipartite graph $G = K_{m,n}$. Then we must show that for each $I \subseteq \{1, \ldots, m\}$ and $J \subseteq \{1, \ldots, n\}$ one has:

(21.54) $|I| \cdot |J| \geq \max\{a(I) - b'(\overline{J}), a'(J) - b(\overline{I})\}$,

where $\overline{I} := \{1, \ldots, m\} \setminus I$ and $\overline{J} := \{1, \ldots, n\} \setminus J$. By symmetry, it suffices to show

(21.55) $|I| \cdot |J| \geq a(I) - b'(\overline{J})$.

This follows from (21.53)(i), since

(21.56) $a(I) \leq \sum_{i=1}^{|I|} \leq \sum_{j=1}^{n} \min\{|I|, b'_j\} \leq |J| \cdot |I| + b'(\overline{J})$

for any $J \subseteq \{1, \ldots, n\}$. ∎

(Gale [1956,1957] proved this theorem for $a = \mathbf{0}$ and $b' = \infty$, and Ryser [1957] for $a = b$ and $a' = b'$.)

Corollary 21.28a due to Fulkerson [1959a], is equivalent to the following result extending the Gale-Ryser theorem:

Theorem 21.32. *Let $(c_{i,j})$ be a nonnegative $m \times n$ matrix and let $a, b \in \mathbb{Z}_+^m$ and $a', b' \in \mathbb{Z}_+^n$ with $a \leq b$ and $a' \leq b'$. Then there exists an integer $m \times n$ matrix $(x_{i,j})$ satisfying*

(21.57) (i) $0 \leq x_{i,j} \leq c_{i,j}$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$,

(ii) $a_i \leq \sum_{j=1}^{n} x_{i,j} \leq b_i$ for all $i = 1, \ldots, m$,

(iii) $a'_j \leq \sum_{i=1}^{m} x_{i,j} \leq b'_j$ for all $j = 1, \ldots, n$,

if and only if there exist an $m \times n$ matrix $(x'_{i,j})$ satisfying

(21.58) (i) $0 \leq x'_{i,j} \leq c_{i,j}$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$,

(ii) $\sum_{j=1}^{n} x'_{i,j} \leq b_i$ for all $i = 1, \ldots, m$,

(iii) $a'_j \leq \sum_{i=1}^{m} x'_{i,j}$ for all $j = 1, \ldots, n$,

and an $m \times n$ matrix $(x''_{i,j})$ satisfying

(21.59)

$$\begin{array}{lll} \text{(i)} & 0 \leq x''_{i,j} \leq c_{i,j} & \text{for all } i = 1, \ldots, m \text{ and } j = 1, \ldots, n, \\ \text{(ii)} & a_i \leq \sum_{j=1}^{n} x''_{i,j} & \text{for all } i = 1, \ldots, m, \\ \text{(iii)} & \sum_{i=1}^{m} x''_{i,j} \leq b'_j & \text{for all } j = 1, \ldots, n. \end{array}$$

Proof. This is equivalent to Corollary 21.28a. ∎

21.13d. Further notes

Corollary 11.2c implies the following result of Hoffman [1956a]. Let $G = (V, E)$ be a bipartite graph and let $0 < \alpha < 1$. Then E has a subset F such that

(21.60) $$\lfloor \tfrac{\deg_E(v)}{\alpha} \rfloor \leq \deg_F(v) \leq \lceil \tfrac{\deg_E(v)}{\alpha} \rceil$$

for each vertex v.

Ikura and Nemhauser [1982] gave a strongly polynomial-time primal simplex algorithm for the maximum-weight stable set problem in bipartite graphs (the number of pivot steps is at most n^2; the method corresponds to a strongly polynomial-time dual simplex algorithm for the minimum-size b-edge cover problem, which is a special case of a minimum-flow problem). (An improvement was given by Armstrong and Jin [1996].) An interior-point method was described by Mizuno and Masuzawa [1989]. For more on capacitated b-matchings (in terms of matrices), see Anstee [1983].

We refer for further notes on algorithmic aspects of the transportation problem to Section 12.5d on the equivalent transshipment problem.

Heller [1963,1964] gave necessary and sufficient conditions for a linear program to be equivalent to a transportation problem. Katerinis [1987] and Enomoto, Ota, and Kano [1988] gave sufficient conditions for bipartite graphs to have a k-factor.

Goodman, Hedetniemi, and Tarjan [1976] gave a linear-time algorithm finding a maximum-weight simple b-matching in a tree.

Faster algorithms for transportation problems where the cost satisfies a quadrangle inequality where given by Karp and Li [1975] and Aggarwal, Bar-Noy, Khuller, Kravets, and Schieber [1995].

Variants of the transportation problem (minimax, bottleneck) were investigated by Szwarc [1966,1971], Hammer [1969,1971], Garfinkel and Rao [1971], Srinivasan and Thompson [1972a,1972b,1976], Derigs and Zimmermann [1979], Derigs [1982], Russell, Klingman, and Partow-Navid [1983], and Ahuja [1986]. Prager [1957b] and Kellerer [1961] gave a generalization.

Prager [1955] gave an extension to quadratic cost functions, i.e. given $b \in \mathbb{R}^m$, $d \in \mathbb{R}^n$, and $c_{i,j} \geq 0$, $q_{i,j} \geq 0$ ($i = 1, \ldots, m$; $j = 1, \ldots, n$):

(21.61)

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{i,j} x_{i,j} + q_{i,j} x_{i,j}^2), & \\ \text{subject to} & \sum_{j=1}^{n} x_{i,j} = b_i & \text{for } i = 1, \ldots, m, \\ & \sum_{i=1}^{m} x_{i,j} = d_j & \text{for } j = 1, \ldots, n, \\ & x_{i,j} \geq 0 & \text{for } i = 1, \ldots, m; j = 1, \ldots, n. \end{array}$$

Among the books surveying transportation are Ford and Fulkerson [1962], Dantzig [1963], Murty [1976,1983], Bazaraa and Jarvis [1977], Papadimitriou and Steiglitz [1982], Gondran and Minoux [1984], Derigs [1988a], Nemhauser and Wolsey [1988], and Bazaraa, Jarvis, and Sherali [1990].

21.13e. Historical notes on the transportation and transshipment problems

Transportation can be considered as the special case of transshipment where all arcs are oriented from a source to a sink. By the techniques described in Section 21.6a, transshipment problems can be reduced conversely to transportation problems. This makes the history of the two problems intertwined. We should notice also that the transshipment problems studied by Kantorovich and Koopmans were in fact transportation problems, due to the fact that their cost functions are metrics.

Tolstoĭ

The first to study the transportation problem mathematically seems to be A.N. Tolstoĭ. In the collection *Transportation Planning, Volume I* of the National Commissariat of Transportation of the Soviet Union, Tolstoĭ [1930] published an article called *Methods of finding the minimal total kilometrage in cargo-transportation planning in space*. In it, Tolstoĭ described a number of approaches to solve the transportation problem, illuminated by applications to the transportation of salt, cement, and other cargo between sources and destination points along the railway network of the Soviet Union. He seems to be the first to give a negative cycle criterion for optimality. Moreover, a for that time large-scale instance of the transportation problem was solved to optimality.

First, Tolstoĭ considered the problem for the case where there are two sources. He observed that in that case one can order the destination points by the difference between the distances to the two sources. In that case, one source can provide the destinations starting from the beginning of the list, until the supply of that source has been used up. The other source supplies the remaining demands. Tolstoĭ observed that the list is independent of the supplies and demands, and hence

> such table is applicable for the whole life-time of factories, or sources of production.
> Using this table, one can immediately compose an optimal transportation plan every year, given quantities of output produced by these two factories and demands of the destination points.

Next, Tolstoĭ studied the transportation problem for the case where all sources and destinations are along one circular railway line. In this case, considering the negative cycle criterion yields directly the optimum solution. He calls this phenomenon 'circle dependency'.

Finally, Tolstoĭ combined the two methods into a heuristic to solve a concrete transportation problem coming from cargo transportation along the Soviet railway network. The problem has 10 sources and 68 sinks, and 155 links between sources and sinks (all other distances are taken infinite):

	Arkhangelsk	Yaroslavl’	Murom	Balakhonikha	Dzerzhinsk	Kishert’	Sverdlovsk	Artemovsk	Iledzhk	Dekonskaya	*demand:*
Agryz				709	1064	693					2
Aleksandrov					397			1180			4
Almaznaya								81		65	1.5
Alchevskaya								106		114	4
Baku								1554		1563	10
Barybino								985		968	2
Berendeevo		135			430						10
Bilimbai						200	59				1
Bobrinskaya								655		663	10
Bologoe		389						1398			1
Verkhov’e								678		661	1
Volovo								757		740	3
Vologda	634					1236					2
Voskresensk				427				1022		1005	1
V.Volochek		434						1353		1343	5
Galich	815	224				1056					0.5
Goroblagodatskaya						434	196				0.5
Zhlobin								882		890	8
Zverevo								227		235	5
Ivanovo					259						6
Inza				380	735					1272	2
Kagan								2445	2379		0.5
Kasimov			0								1
Kinel’				752		1208			454	1447	2
Kovylkino				355						1213	2
Kyshtym						421	159				3
Leningrad	1237	709						1667		1675	55
Likino			223		328						15
Liski								443		426	1
Lyuberdzhy			268		411					1074	1
Magnitogorskaya						932	678		818		1
Mauk						398	136				5
Moskva			288	378	405			1030		1022	141
Navashino			12	78							2
Nizhegol’								333		316	1
Nerekhta		50			349						5
Nechaevskaya			92								0.5
N.-Novgorod					32						25
Omsk						1159	904		1746		5
Orenburg									76		1.5
Penza				411				1040	883	1023	7
Perm’	1749					121					1
Petrozavodsk	1394										1
Poltoradzhk								1739	3085	1748	4
Pskov								1497		1505	10
Rostov/Don								287		296	20
Rostov/Yarosl		56			454						2
Rtishchevo								880		863	1
Savelovo		325						1206		1196	5
Samara				711					495	1406	7
San-Donato						416	157				1
Saratov								1072		1055	15
Sasovo				504				1096		1079	1
Slavyanoserbsk								119		115	1.1
Sonkovo		193						1337			0.5
Stalingrad								624		607	15.4
St.Russa		558						1507		1515	5
Tambov								783		766	4
Tashkent								3051	1775		3
Tula								840		848	8
Tyumen’						584	329				6
Khar’kov								251		259	60
Chelyabinsk						511	257		949		2
Chishmy				1123		773			889		0.5
Shchigry								566		549	4
Yudino				403	757	999					0.5
Yama								44		52	5
Yasinovataya								85		93	6
supply:	5	11.5	8.5	12	100	12	15	314	10	55	543

Table of distances (in kilometers) between sources and destinations, and of supplies and demands (in kilotons).
Tolstoĭ gave no distance for Kasimov. We have inserted a distance 0 to Murom, since from Tolstoĭ’s solution it appears that Kasimov is connected only to Murom (by a waterway). Hence the distance is irrelevant.

Tolstoĭ's heuristic also makes use of insight into the geography of the Soviet Union. He goes along all sources (starting with the most remote source), where, for each source X, he lists those sinks for which X is the closest source or the second closest source. Based on the difference of the distances to the closest and second closest sources, he assigns cargo from X to the sinks, until the supply of X has been used up. In case Tolstoĭ foresees circle dependency, he deviates from this rule to avoid that a negative-length circuit would arise. No backtracking occurs.

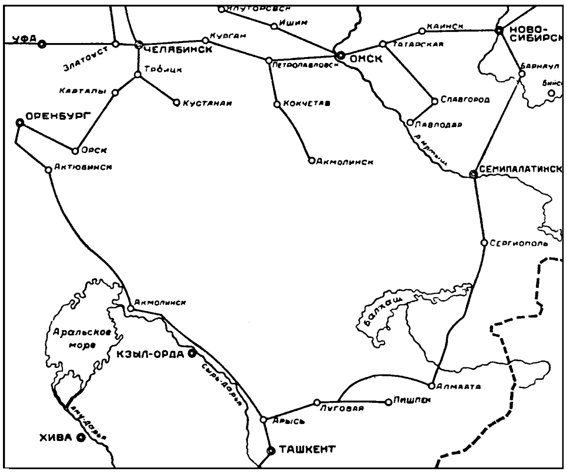


Figure 21.1
Figure from Tolstoĭ [1930] to illustrate a negative cycle.

In the following quotation, Tolstoĭ considers the cycles Dzerzhinsk-Rostov-Yaroslavl'-Leningrad-Artemovsk-Moscow-Dzerzhinsk and Dzerzhinsk-Nerekhta-Yaroslavl'-Leningrad-Artemovsk-Moscow-Dzerzhinsk. It is the sixth step in his method, after the transports from the factories in Iletsk, Sverdlovsk, Kishert', Balakhonikha, and Murom have been set:

> 6. The Dzerzhinsk factory produces 100,000 tons. It can forward its production only in the Northeastern direction, where it sets its boundaries in interdependency with the Yaroslavl' and Artemovsk (or Dekonskaya) factories.

	From Dzerzhinsk	From Yaroslavl'	Difference to Dzerzhinsk
Berendeevo	430 km	135 km	−295 km
Nerekhta	349 ,,	50 ,,	−299 ,,
Rostov	454 ,,	56 ,,	−398 ,,

	From Dzerzhinsk	From Artemovsk	Difference to Dzerzhinsk
Aleksandrov	397 km	1,180 km	+783 km
Moscow	405 ,,	1,030 ,,	+625 ,,

> The method of differences does not help to determine the boundary between the Dzerzhinsk and Yaroslavl' factories. Only the circle dependency, specified to be

an interdependency between the Dzerzhinsk, Yaroslavl' and Artemovsk factories, enables us to exactly determine how far the production of the Dzerzhinsk factory should be advanced in the Yaroslavl' direction.
Suppose we attach point Rostov to the Dzerzhinsk factory; then, by the circle dependency, we get:

Dzerzhinsk-Rostov	454 km	−398 km	Nerekhta	349 km	−299 km
Yaroslavl'- ,,	56 ,,		,,	50 ,,	
Yaroslavl'-Leningrad	709 ,,	+958 ,,	These points remain		
Artemovsk- ,,	1,667 ,,		unchanged because only the		
Artemovsk-Moscow	1,030 ,,	−625 ,,	quantity of production sent		
Dzerzhinsk- ,,	405 ,,		by each factory changes		
Total		−65 km			+34 km

Therefore, the attachment of Rostov to the Dzerzhinsk factory causes over-run in 65 km, and only Nerekhta gives a positive sum of differences and hence it is the last point supplied by the Dzerzhinsk factory in this direction.
As a result, the following points are attached to the Dzerzhinsk factory:

N. Novgorod	25,000 tons	
Ivanova	6,000 ,,	
Nerekhta	5,000 ,,	
Aleksandrov	4,000 ,,	
Berendeevo	10,000 ,,	
Likino	15,000 ,,	
Moscow	35,000 ,,	(remainder of factory's production)
Total	100,000 tons	

After 10 steps, when the transports from all 10 factories have been set, Tolstoĭ 'verifies' the solution by considering a number of cycles in the network, and he concludes that his solution is optimum:

> Thus, by use of successive applications of the method of differences, followed by a verification of the results by the circle dependency, we managed to compose the transportation plan which results in the minimum total kilometrage.

The objective value of Tolstoĭ's solution is 395,052 kiloton-kilometers. Solving the problem with modern linear programming tools (CPLEX) shows that Tolstoĭ's solution indeed is optimum. But it is unclear how sure Tolstoĭ could have been about his claim that his solution is optimum. Geographical insight probably has helped him in growing convinced of the optimality of his solution. On the other hand, it can be checked that there exist feasible solutions that have none of the negative-cost cycles considered by Tolstoĭ in their residual graph, but that are yet not optimum[40].

In the September 1939 issue of *Sotsialisticheskiĭ Transport*, Tolstoĭ [1939] published an article *Methods of removing irrational transportations in planning*, in which he again described his method of 'circle dependency', and applied it to the planning of driving empty cars and transporting heavy cargoes on the U.S.S.R. railway network. In this paper, Tolstoĭ restricted himself to sources and sinks arranged along a circular railway line, for which he gave his 'circle dependency' method:

[40] The maximum objective value of a feasible solution, whose residual graph contains no nonnegative-cost cycle of length 4, and none of the seven longer nonnegative-length cycles considered by Tolstoĭ (of lengths 6 and 8), is equal to 397,226.

> Before counting distances from cargo-senders to points of destination which form a circle dependency, it is necessary to attach points of destination to cargo-senders with complete distribution of waggons. In case of circle dependency determined by geographical location it can be done without special calculations. Then, by calculation of km in circle dependency, the initial attachment can be verified and if not correct, then it can be improved.

Tolstoĭ illustrated the method by the circuit Smolensk - Vitebsk - Velikiye-Luki - Zemtsy - Rzhev - Vyazma - Smolensk of the U.S.S.R. network. A negative-length directed circuit in the auxiliary directed graph gives an improvement, as in the following Table given by Tolstoĭ [1939]:

Source of cargoes	Amount km	Difference of distance	Amount of carriages
Vyazma-Smolensk	176	-37	$4-3=1$
Vitebsk ,,	139		$0+3=4$
Vitebsk-V. Luki	156	-37	$3-3=0$
Zemtsy ,,	119		$2+3=5$
Zemtsy-Rzhev	123	$+7$	$5-3=2$
Vyazma ,,	130		$1+3=4$
	Altogether . . .	-67	

Tolstoĭ then remarked:

> The negative total difference shows that the distribution was wrong and that there is an over-run of 67 km for every waggon which goes from upper cargo-senders.

According to Kantorovich [1987], there were some attempts to introduce Tolstoĭ's work by the appropriate department of the People's Commissariat of Transport. Tolstoĭ's method was also explained in the book *Planning Goods Transportation* by Pariĭskaya, Tolstoĭ, and Mots [1947].

Kantorovich

Apparently unaware (by that time) of the work of Tolstoĭ, L.V. Kantorovich studied a general class of problems, that includes the transportation problem. It formed a major impulse to the study of linear programming. In his memoirs, Kantorovich [1987] writes:

> Once some engineers from the veneer trust laboratory came to me for consultation with a quite skilful presentation of their problems. Different productivity is obtained for veneer-cutting machines for different types of materials; linked to this the output of production of this group of machines depended, it would seem, on the chance factor of which group of raw materials to which machine was assigned. How could this fact be used rationally?
> This question interested me, but nevertheless appeared to be quite particular and elementary, so I did not begin to study it by giving up everything else. I put this question for discussion at a meeting of the mathematics department, where there were such great specialists as Gyunter, Smirnov himself, Kuz'min, and Tartakovskii. Everyone listened but no one proposed a solution; they had already turned to someone earlier in individual order, apparently to Kuz'min. However, this question nevertheless kept me in suspense. This was the year of my marriage, so I was also distracted by this. In the summer or after the vacation

> concrete, to some extent similar, economic, engineering, and managerial situations started to come into my head, that also required the solving of a maximization problem in the presence of a series of linear constraints.
> In the simplest case of one or two variables such problems are easily solved—by going through all the possible extreme points and choosing the best. But, let us say in the veneer trust problem for five machines and eight types of materials such a search would already have required solving about a billion systems of linear equations and it was evident that this was not a realistic method. I constructed particular devices and was probably the first to report on this problem in 1938 at the October scientific session of the Herzen Institute, where in the main a number of problems were posed with some ideas for their solution.
> The universality of this class of problems, in conjunction with their difficulty, made me study them seriously and bring in my mathematical knowledge, in particular, some ideas from functional analysis.

In a footnote, Kantorovich's son V.L. Kantorovich adds:

> In L.V. Kantorovich's archives a manuscript from 1938 is preserved on "Some mathematical problems of the economics of industry, agriculture, and transport" that in content, apparently, corresponds to this report and where, in essence, the simplex method for the machine problem is described.

L.V. Kantorovich recalled that he created in January 1939 'a method of Lagrange (resolving) multipliers'.

> What became clear was both the solubility of these problems and the fact that they were widespread, so representatives of industry were invited to a discussion of my report at the university.

This meeting took place on 13 May 1939 at the Mathematical Section of the Institute of Mathematics and Mechanics of the Leningrad State University. A second meeting, which was devoted specifically to problems connected with construction, was held on 26 May 1939 at the Leningrad Institute for Engineers of Industrial Construction. These meetings provided the basis of the monograph *Mathematical Methods in the Organization and Planning of Production* (Kantorovich [1939]).

According to the Foreword by A.R. Marchenko to this monograph, Kantorovich's work was highly praised by mathematicians, and, in addition, at the special meeting industrial workers unanimously evinced great interest in the work.

The relevance was described by Kantorovich as follows:

> I want to emphasize again that the greater part of the problems of which I shall speak, relating to the organization and planning of production, are connected specifically with the Soviet system of economy and in the majority of cases do not arise in the economy of a capitalist society. There the choice of output is determined not by the plan but by the interests and profits of individual capitalists. The owner of the enterprise chooses for production those goods which at a given moment have the highest price, can most easily be sold, and therefore give the largest profit. The raw material used is not that of which there are huge supplies in the country, but that which the entrepreneur can buy most cheaply. The question of the maximum utilization of equipment is not raised; in any case, the majority of enterprises work at half capacity.
> In the USSR the situation is different. Everything is subordinated not to the interests and advantage of the individual enterprise, but to the task of fulfilling the state plan. The basic task of an enterprise is the fulfillment and overfulfillment of its plan, which is a part of the general state plan. Moreover, this not only means fulfillment of the plan in aggregate terms (i.e. total value of output, total tonnage, and so on), but the certain fulfillment of the plan for all kinds of output; that is, the fulfillment of the assortment plan (the fulfillment of the plan for each kind of output, the completeness of individual items of output, and so on).

In the monograph, Kantorovich outlined a new method to maximize a linear function under given linear constraints. One of the problems studied was a rudimentary form of a transportation problem:

(21.62) given: an $m \times n$ matrix $(a_{i,j})$;
find: an $m \times n$ matrix $(x_{i,j})$ such that:
(i) $x_{i,j} \geq 0$ for all i, j;
(ii) $\sum_{i=1}^{m} x_{i,j} = 1$ for each $j = 1, \ldots, n$;
(iii) $\sum_{j=1}^{n} a_{i,j} x_{i,j}$ is independent of i and is maximized.

Another problem studied by Kantorovich was 'Problem C' which can be stated as follows:

(21.63)
$$\begin{array}{lll} \text{maximize} & \lambda & \\ \text{subject to} & \sum_{i=1}^{m} x_{i,j} = 1 & (j = 1, \ldots, n) \\ & \sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j,k} x_{i,j} = \lambda & (k = 1, \ldots, t) \\ & x_{i,j} \geq 0 & (i = 1, \ldots, m; j = 1, \ldots, n). \end{array}$$

The interpretation is: let there be n machines, which can do m jobs. Let there be one final product consisting of t parts. When machine i does job j, $a_{i,j,k}$ units of part k are produced ($k = 1, \ldots, t$). Now $x_{i,j}$ is the fraction of time machine i does job j. The number λ is the amount of the final product produced. 'Problem C' was later seen (by H.E. Scarf, upon a suggestion by Kantorovich — see Koopmans [1959]) to be equivalent to the general linear programming problem.

Kantorovich's method consists of determining dual variables ('resolving multipliers') and finding the corresponding primal solution. If the primal solution is not feasible, the dual solution is modified following prescribed rules. Kantorovich also indicated the role of the dual variables in sensitivity analysis, and he showed that a feasible primal solution for Problem C can be shown to be optimal by specifying optimal dual variables.

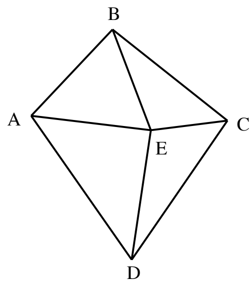
Kantorovich gave a wealth of practical applications of his methods, which he based mainly in the Soviet plan economy:

> Here are included, for instance, such questions as the distribution of work among individual machines of the enterprise or among mechanisms, the correct distribution of orders among enterprises, the correct distribution of different kinds of raw materials, fuel, and other factors. Both are clearly mentioned in the resolutions of the 18th Party Congress.

He described the applications to transportation:

> Let us first examine the following question. A number of freights (oil, grain, machines and so on) can be transported from one point to another by various methods; by railroads, by steamship; there can be mixed methods, in part by railroad, in part by automobile transportation, and so on. Moreover, depending on the kind of freight, the method of loading, the suitability of the transportation, and the efficiency of the different kinds of transportation is different. For example, it is particularly advantageous to carry oil by water transportation if oil tankers

are available, and so on. The solution of the problem of the distribution of a given freight flow over kinds of transportation, in order to complete the haulage plan in the shortest time, or within a given period with the least expenditure of fuel, is possible by our methods and leads to Problems A or C.
Let us mention still another problem of different character which, although it does not lead directly to questions A, B, and C, can still be solved by our methods. That is the choice of transportation routes.



Let there be several points A, B, C, D, E (Fig. 1) which are connected to one another by a railroad network. It is possible to make the shipments from B to D by the shortest route BED, but it is also possible to use other routes as well: namely, BCD, BAD. Let there also be given a schedule of freight shipments; that is, it is necessary to ship from A to B a certain number of carloads, from D to C a certain number, and so on. The problem consists of the following. There is given a maximum capacity for each route under the given conditions (it can of course change under new methods of operation in transportation). It is necessary to distribute the freight flows among the different routes in such a way as to complete the necessary shipments with a minimum expenditure of fuel, under the condition of minimizing the empty runs of freight cars and taking account of the maximum capacity of the routes. As was already shown, this problem can also be solved by our methods.

Kantorovich [1987] wrote in his memoirs:

The university immediately published my pamphlet, and it was sent to fifty People's Commissariats. It was distributed only in the Soviet Union, since in the days just before the start of the World War it came out in an edition of one thousand copies in all.
The number of responses was not very large. There was quite an interesting reference from the People's Commissariat of Transportation in which some optimization problems directed at decreasing the mileage of wagons was considered, and a good review of the pamphlet appeared in the journal *The Timber Industry.*
At the beginning of 1940 I published a purely mathematical version of this work in Doklady Akad. Nauk [76], expressed in terms of functional analysis and algebra. However, I did not even put in it a reference to my published pamphlet—taking into account the circumstances I did not want my practical work to be used outside the country.

In the spring of 1939 I gave some more reports—at the Polytechnic Institute and the House of Scientists, but several times met with the objection that the work used mathematical methods, and in the West the mathematical school in economics was an anti-Marxist school and mathematics in economics was a means for apologists of capitalism. This forced me when writing a pamphlet to avoid the term "economic" as much as possible and talk about the organization and planning of production; the role and meaning of the Lagrange multipliers had to be given somewhere in the outskirts of the second appendix and in the semi Aesopian language.

(Here reference [76] is Kantorovich [1940].) Kantorovich mentioned that the new area opened by his work played a definite role in forming the Leningrad Branch of the Mathematical Institute (LOMI), where he worked with M.K. Gavurin on this area. The problem that they studied occurred to them by itself, but they soon found out that railway workers were already studying the problem of planning haulage on railways, applied to questions of driving empty cars and transport of heavy cargoes.

Kantorovich and Gavurin wrote their method (the method of 'potentials') in a paper *Application of mathematical methods in questions of analysis of freight traffic* (Kantorovich and Gavurin [1949]), which was presented in January 1941 to the mathematics section of the Leningrad House of Scientists, but according to Kantorovich [1987]:

> The publication of this paper met with many difficulties. It had already been submitted to the journal *Railway Transport* in 1940, but because of the dread of mathematics already mentioned it was not printed then either in this or in any other journal, despite the support of Academicians A.N. Kolmogorov and V.N. Obraztsov, a well-known transport specialist and first-rank railway General.

Kantorovich [1987] said that he fortunately made an abstract version of the problem, Kantorovich [1942], in which he considered the following generalization of the transportation problem.

Let R be a compact metric space, with two measures μ and μ'. Let $\mathcal{B}$ be the collection of measurable sets in R. A *translocation (of masses)* is a function Ψ : $\mathcal{B} \times \mathcal{B} \to \mathbb{R}_+$ such that for each $X \in \mathcal{B}$ the functions $\Psi(X, .)$ and $\Psi(., X)$ are measures and such that

(21.64) $$\Psi(X, R) = \mu(X) \text{ and } \Psi(R, X) = \mu'(X)$$

for each $X \in \mathcal{B}$.

Let a continuous function $r : R \times R \to \mathbb{R}_+$ be given. (The value $r(x, y)$ represents the work needed to transfer a unit mass from x to y.) Then the *work* of a *translocation* Ψ is by definition:

(21.65) $$\int_R \int_R r(x, y)\Psi(d\mu, d\mu').$$

Kantorovich argued that, if there exists a translocation, then there exists a *minimal* translocation, that is, a translocation Ψ minimizing (21.65).

He calls a translocation Ψ *potential* if there exists a function $p : R \to \mathbb{R}$ such that for all $x, y \in R$:

(21.66)
(i) $|p(x) - p(y)| \leq r(x, y)$;
(ii) $p(y) - p(x) = r(x, y)$ if $\Psi(U_x, U_y) > 0$ for any neighbourhoods U_x of x and U_y of y.

Kantorovich showed:

Theorem 21.33. *A translocation Ψ is minimal if and only if it is potential.*

This framework applies to the transportation problem (when $m = n$), by taking for R the space $\{1, \ldots, n\}$, with the discrete topology.

Kantorovich's proof of Theorem 21.33 is by a construction of a potential, that however only is correct if r satisfies the triangle inequality. Kantorovich remarked that his method is algorithmic:

> The theorem just demonstrated makes it easy for one to prove that a given mass translocation is or is not minimal. He has only to try and construct the potential in the way outlined above. If this construction turns out to be impossible, i.e. the given translocation is not minimal, he at least will find himself in the possession of the method how to lower the translocation work and eventually come to the minimal translocation.

Beside to a problem of leveling a land area, Kantorovich gave as application:

> Problem 1. *Location of consumption stations with respect to production stations.* Stations $A_1, A_2, \cdots, A_m$, attached to a network of railways deliver goods to an extent of $a_1, a_2, \cdots, a_m$ carriages per day respectively. These goods are consumed at stations $B_1, B_2, \cdots, B_n$ of the same network at a rate of $b_1, b_2, \cdots, b_n$ carriages per day respectively ($\sum a_i = \sum b_k$). Given the costs $r_{i,k}$ involved in moving one carriage from station A_i to station B_k, assign the consumption stations such places with respect to the production stations as would reduce the total transport expenses to a minimum.

As mentioned, Kantorovich's results remained unnoticed for some time by Western researchers. In a note introducing a reprint of the article of Kantorovich [1942], in *Management Science* in 1958, the following reassurance is given:

> It is to be noted, however, that the problem of determining an effective method of actually acquiring the solution to a specific problem is *not* solved in this paper. In the category of development of such methods we seem to be, currently, ahead of the Russians.

Kantorovich's method was elaborated by Kantorovich and Gavurin [1949], where moreover single- and multicommodity transportation models are studied, with applications to the railway network of the U.S.S.R.

Hitchcock

Independently, Hitchcock [1941] studied the transportation problem:

(21.67) given: an $m \times n$ matrix $C = (c_{i,j})$ and vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$;
find: an $m \times n$ matrix $X = (x_{i,j})$ such that:
(i) $x_{i,j} \geq 0$ for all i, j;
(ii) $\sum_{j=1}^{n} x_{i,j} = a_i$ for each $i = 1, \ldots, m$;
(iii) $\sum_{i=1}^{m} x_{i,j} = b_j$ for each $j = 1, \ldots, n$;
(iv) $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j} x_{i,j}$ is as small as possible.

The interpretation of the problem is, in Hitchcock's words:

> When several factories supply a product to a number of cities we desire the least costly manner of distribution. Due to freight rates and other matters the cost of a ton of product to a particular city will vary according to which factory supplies it, and will also vary from city to city.

Hitchcock showed that the minimum is attained at a vertex of the feasible region, and he outlined a scheme for solving the transportation problem which has much in common with the simplex method for linear programming. It includes pivoting (eliminating and introducing basic variables) and the fact that nonnegativity of certain dual variables implies optimality. He showed that the *complementary slackness* conditions characterize optimality: $(x^*_{i,j})$ is an optimum vertex if and only if there exists a combination $\sum_{i,j} \lambda_{i,j} x_{i,j}$ of the left-hand sides of the constraints (ii) and (iii) such that $\lambda_{i,j} \geq c_{i,j}$ for all i, j and such that $\lambda_{i,j} = c_{i,j}$ if $x^*_{i,j} > 0$.

Hitchcock however seemed to have overlooked the possibility of cycling of his method, although he pointed at an example in which some dual variables are negative while yet the primal solution is optimum.

Hitchcock also gave a method to find an initial basic solution, now known as the *north-west rule*: set $x_{1,1} := \min\{a_1, b_1\}$; if the minimum is attained by a_1, reset $b_1 := b_1 - a_1$ and recursively find a basic solution $x_{i,j}$ satisfying $\sum_{j=1}^{n} x_{i,j} = a_i$ for each $i = 2, \ldots, m$ and $\sum_{i=2}^{m} x_{i,j} = b_j$ for each $j = 1, \ldots, n$; if the minimum is attained by b_1, proceed symmetrically. (The north-west rule was also described by Salvemini [1939] and Fréchet [1951] in a statistical context, namely in order to complete correlation tables given the marginal distributions.)

Koopmans

Also independently, Koopmans investigated transportation problems. In March 1942, Koopmans was appointed as a statistician on the staff of the British Merchant Shipping Mission, and later the Combined Shipping Adjustment Board (CSAB), a British-American agency dealing with merchant shipping problems during the Second World War (as they should go in convoys, under military protection). Influenced by his teacher J. Tinbergen (cf. Tinbergen [1934]) he was interested in tanker freights and capacities (cf. Koopmans [1939]). According to Koopmans' personal diary, in August 1942 while the Board was being organized, there was not much work for the statisticians,

> and I had a fairly good time working out exchange ratio's between cargoes for various routes, figuring how much could be carried monthly from one route if monthly shipments on another route were reduced by one unit.

At the Board he studied the assignment of ships to convoys so as to accomplish prescribed deliveries, while minimizing empty voyages (cf. Dorfman [1984]). According to the memoirs of his wife (Wanningen Koopmans [1995]), when Koopmans was with the Board,

> he had been appalled by the way the ships were routed. There was a lot of redundancy, no intensive planning. Often a ship returned home in ballast, when with a little effort it could have been rerouted to pick up a load elsewhere.

In his autobiography (published posthumously), Koopmans [1992] described how he came to the problem:

> My direct assignment was to help fit information about losses, deliveries from new construction, and employment of British-controlled and U.S-controlled ships into a unified statement. Even in this humble role I learned a great deal about the difficulties of organizing a large-scale effort under dual control—or rather in this case four-way control, military and civilian cutting across U.S. and U.K.

> controls. I did my study of optimal routing and the associated shadow costs of transportation on the various routes, expressed in ship days, in August 1942 when an impending redrawing of the lines of administrative control left me temporarily without urgent duties. My memorandum, cited below, was well received in a meeting of the Combined Shipping Adjustment Board (that I did not attend) as an explanation of the "paradoxes of shipping" which were always difficult to explain to higher authority. However, I have no knowledge of any systematic use of my ideas in the combined U.K.-U.S. shipping problems thereafter.

In the memorandum to the Board, Koopmans [1942] analyzed the sensitivity of the optimum shipments for small changes in the demands. In this memorandum, Koopmans did not give a method to find an optimum shipment. Further study led him to a 'local search' method for the transportation problem, stating that it leads to an optimum solution. According to Dorfman [1984], Koopmans found these results in 1943, but, due to wartime restrictions, published them only after the war (Koopmans [1948], Koopmans and Reiter [1949a,1949b,1951]). Koopmans [1948] wrote:

> Let us now for the purpose of argument (since no figures of war experience are available) assume that one particular organization is charged with carrying out a world dry-cargo transportation program corresponding to the actual cargo flows of 1925. How would that organization solve the problem of moving the empty ships economically from where they become available to where they are needed? It seems appropriate to apply a procedure of trial and error whereby one draws tentative lines on the map that link up the surplus areas with the deficit areas, trying to lay out flows of empty ships along these lines in such a way that a minimum of shipping is at any time tied up in empty movements.

The 'trial and error' method mentioned is one of local improvements, corresponding to finding a negative-cost directed circuit in the residual digraph. Koopmans' *first theorem* is that it leads to an optimum solution:

> If, under the assumptions that have been stated, no improvement in the use of shipping is possible by small variations such as have been illustrated, then there is no—however thoroughgoing—rearrangement in the routing of empty ships that can achieve a greater economy of tonnage.

He illustrated the method by giving an optimum solution for a 3×12 transportation problem, with the following supplies and demands:

Net receipt of dry cargo in overseas trade, 1925

Unit: Millions of metric tons per annum

Harbour	Received	Dispatched	Net receipts
New York	23.5	32.7	−9.2
San Francisco	7.2	9.7	−2.5
St. Thomas	10.3	11.5	−1.2
Buenos Aires	7.0	9.6	−2.6
Antofagasta	1.4	4.6	−3.2
Rotterdam	126.4	130.5	− 4.1
Lisbon	37.5	17.0	20.5
Athens	28.3	14.4	13.9
Odessa	0.5	4.7	−4.2
Lagos	2.0	2.4	−0.4
Durban	2.1	4.3	−2.2
Bombay	5.0	8.9	−3.9
Singapore	3.6	6.8	−3.2
Yokohama	9.2	3.0	6.2
Sydney	2.8	6.7	−3.9
Total	266.8	266.8	0.0

Koopmans [1948] moreover claimed that there exist *potentials* $p_1, \ldots, p_n$ and $q_1, \ldots, q_m$ such that $c_{i,j} \geq p_i - q_j$ for all i, j and such that $c_{i,j} = p_i - q_j$ for each i, j for which $x_{i,j} > 0$.

The potentials give the *marginal costs* when modifying the input data. That is, if both a_i and b_j increase by 1, then the minimum cost increases by at least $p_i - q_j$. This is Koopmans' *second theorem*.

In the proof, Koopmans assumed that the cost function is symmetric and satisfies the triangle inequality. Moreover, he assumed that the graph of arcs having a positive transshipment value is weakly connected. The latter restriction was removed in a later paper by Koopmans and Reiter [1951]. In this paper, they investigated the economic implications of the model and the method:

> For the sake of definiteness we shall speak in terms of the transportation of cargoes on ocean-going ships. In considering only shipping we do not lose generality of application since ships may be "translated" into trucks, aircraft, or, in first approximation, trains, and ports into the various sorts of terminals. Such translation is possible because all the above examples involve particular types of movable transportation equipment.

They use the graph model, and in a footnote they remark:

> The cultural lag of economic thought in the application of mathematical methods is strikingly illustrated by the fact that linear graphs are making their entrance into transportation theory just about a century after they were first studied in relation to electrical networks, although organized transportation systems are much older than the study of electricity.

(For a review of Koopmans' research, see Scarf [1992].)

Robinson, 1950

Robinson [1950] might be the earliest reference stating clearly and generally that the absence of a negative-cost directed circuit in the residual digraph is necessary and sufficient for optimality. She mentioned that it can be 'verified directly', and observed that it gives an algorithm to find an optimum transportation. She concluded with:

> The number of steps in the iterative procedure depends on the "goodness" of the initial choice of X_0. The method does not seem to lend itself to machine calculation but may be efficient for hand computation with matrices of small order.

Linear programming and the simplex method

The breakthrough of general linear programming came at the end of the 1940s. In 1947, Dantzig formulated the linear programming problem and designed the *simplex method* for the linear programming problem, published in Dantzig [1951b]. The success of the method was enlarged by a simple tableau-form and a simple pivoting rule, and by the efficiency in practice. In another paper, Dantzig [1951a] described a direct implementation of the simplex method to the transportation problem (including an anti-cycling rule based on perturbation; variants were given by Charnes and Cooper [1954] and Eisemann [1956]).

The simplex method for transportation was described in terms of graphs by Koopmans and Reiter [1951], and Flood [1952,1953] aimed at giving a purely mathematical description of it. A continuous model of transportation was studied by Beckmann [1952].

Votaw and Orden [1952] reported on early computational results (on the SEAC), and claimed (without proof) that the simplex method is polynomial-time for the transportation problem (a statement refuted by Zadeh [1973a]):

> As to computation time, it should be noted that for moderate size problems, say $m \times n$ up to 500, the time of computation is of the same order of magnitude as the time required to type the initial data. The computation time on a sample computation in which m and n were both 10 was 3 minutes. The time of computation can be shown by study of the computing method and the code to be proportional to $(m+n)^3$.

Application to practice

The new ideas of applying linear programming to the transportation problem were quickly disseminated. Applications to routing empty boxcars over the U.S. railroads were given by Fox [1952] and Nerlove [1953]. Dantzig and Fulkerson [1954b,1954a] studied a rudimentary form of a minimum-cost circulation problem in order to determine the minimum number of tankers to meet a fixed schedule. Similarly, Bartlett [1957] and Bartlett and Charnes [1957] studied methods to determine the minimum railway stock to run a given schedule.

Applicability of linear programming to transportation to practice was also met with scepticism. At a Conference on Linear Programming in May 1954 in London, Land [1954] presented a study of applying linear programming to the problem of transporting coal for the British Coke Industry:

> The real crux of this piece of research is whether the saving in transport cost exceeds the cost of using linear programming.

In the discussion which followed, T. Whitwell of Powers Samas Accounting Machines Ltd remarked

> that in practice one could have one's ideas of a solution confirmed or, much more frequently, completely upset by taking a couple of managers out to lunch.

Gleyzal's primal-dual method for the transportation problem

Gleyzal [1955] published the following primal-dual method for the transportation problem (with integer data). Let $x_{i,j}$ be a feasible solution of the transportation problem. Transform $x_{i,j}$ such that the set $\{u_i v_j \mid x_{i,j} > 0\}$ contains no circuit, and transform $c_{i,j}$ such that $c_{i,j} = 0$ if $x_{i,j} > 0$. (These are easy by first cancelling circuits, and next redefining $c_{i,j}$.)

If $c_{i,j} \geq 0$ for all i, j we are done. Suppose that $c_{i_0,j_0} < 0$ for some i_0, j_0. Let $A := \{(u_i, v_j) \mid c_{i,j} \leq 0\} \cup \{(v_j, u_i) \mid x_{i,j} > 0\}$. If u_{i_0} is reachable in A from v_{j_0}, A contains a directed circuit C containing (u_{i_0}, v_{j_0}). Then we can reset $x_{i,j} := x_{i,j} - 1$ if (v_j, u_i) is in C and $x_{i,j} := x_{i,j} + 1$ if (u_i, v_j) is in C. This decreases $c^{\mathsf{T}} x$.

If u_{i_0} is not reachable in A from v_{j_0}, then for any vertex v let $r(v) := 1$ if v is reachable in A from v_{j_0} and $r(v) := 0$ otherwise. Reset $c_{i,j} := c_{i,j} - r(u_i) + r(v_j)$. This increases $\sum(c_{i,j} \mid c_{i,j} < 0)$, and hence the method terminates.

Munkres on the transportation problem

Munkres [1957] extended his variant of the Hungarian method for the assignment problem to the transportation problem. In graph terms, it amounts to the following.

Let $G = (V, E)$ be a complete bipartite graph, with colour classes U and W of size n, and let be given a weight function $w : E \to \mathbb{Z}_+$ and a function $b : V \to \mathbb{Z}_+$ with $b(U) = b(W)$. We must find a function $x : E \to \mathbb{Q}_+$ such that $\sum_{e\in\delta(v)} x_e = b_v$ for each vertex v and such that $\sum_e w_e x_e$ is minimized.

Let F be the set of edges e with $w_e = 0$ and let $H = (V, F)$. Suppose that we have found an $x : E \to \mathbb{Q}_+$ such that $x_e = 0$ if $e \notin F$ and such that $\sum_{e\in\delta(v)} x_e \leq b_v$ for each $v \in V$. Let U' and W' be the sets of vertices v in U and W for which strict inequality holds. If U', and hence W', are empty, x is an optimum solution. Otherwise, perform the following iteratively.

Orient each edge of H from U to W, and orient each edge e of H with $x_e > 0$ also from W to U (so they are two-way). Now determine the set R_M of vertices reachable by a directed path from U'.

Case 1: $R_M \cap W' \neq \emptyset$. Then D has a $U'-W'$ path, on which we can alternatingly increase and decrease the value of x_e, so as to make $\sum_e x_e$ larger.

Case 2: $R_M \cap W' = \emptyset$. So $w(uv) > 0$ for each $u \in U \cap R_M$ and $v \in W \setminus R_M$. Let h be the minimum of these $w(uv)$. Decrease $w(uv)$ by h if $u \in U \cap R_M, v \in W \setminus R_M$, and increase $w(uv)$ by h if $u \in U \setminus R_M, v \in W \cap R_M$.

This describes the iteration. Note that between any two occurrences of Case 1, only n times Case 2 can occur, since at each such iteration the set $R_M \cap W$ increases. Moreover, after Case 2 we can continue the previous search for R_M. So between any two Case 1-iterations, the Case 2-iterations take $O(n^2)$ time altogether.

Now Case 1 can occur at most $\sum_{v\in U} b_v$ times. So the algorithm is finite, and has running time $O(n^4 B)$ where $B := \max\{b_v \mid v \in V\}$. This specializes to the Hungarian method if $b_v = 1$ for all $v \in V$.

Further early methods

Also Ford and Fulkerson [1955,1957b] (cf. Ford and Fulkerson [1956c,1956d]) extended the Hungarian method to general transportation problems. Their method is essentially the same as that of Munkres [1957], except that successive occurrences of Case 1 iterations are combined to a maximum flow computation. A similar primal-dual method for the transportation problem was described by Egerváry [1958].

Ford and Fulkerson [1956a,1957a] extended the method of Ford and Fulkerson [1955,1957b] for the uncapacitated transportation problem to the *capacitated* transportation problem.

Orden [1955] showed the equivalence of the transshipment problem and the transportation problem. He also noted that the class of transportation problems covers the majority of the applications of linear programming which are in practical use or under active development. Also Prager [1957a] studied the transshipment problem by reduction to a transportation problem and by methods of elastostatics (cf. Kuhn [1957]).

Gallai [1957,1958a,1958b] studied the minimum-cost and the maximum-profit circulation problem, for which he gave min-max relations (see Section 12.5b). He also considered vertex capacities and demands. Beside combinatorial proofs based on potentials, Gallai gave proofs based on linear programming duality and total unimodularity.

A minimum-cost flow algorithm (in disguised form) was given by Ford and Fulkerson [1958b], to solve the 'dynamic flow' problem described in Section 12.5c. They described a method which essentially consists of repeatedly finding a zero-length $r-s$ path in the residual graph, making lengths nonnegative by translating the cost with the help of the current potential p. If no zero-length path exists, the potential is updated. (This is Routine I of Ford and Fulkerson [1958b].) The complexity of this was studied by Fulkerson [1958].

Yakovleva [1959] gave some implementations of the method of Kantorovich and Gavurin [1949]. The paper considers three cases of the problem in a digraph with demands (positive, negative, and zero) of vertices and costs of arcs: (i) noncapacitated case, (ii) capacitated case, and (iii) bipartite case (without zero demands). Two methods are developed for finding feasible potentials or improving the current flow. Time bounds are not indicated.

Among the other early algorithms for minimum-cost flow are successive shortest paths methods (Busacker and Gowen [1960], Iri [1960]), *out-of-kilter* methods (Minty [1960], Fulkerson [1961]), cycle-cancelling (Klein [1967]), and successive shortest paths maintaining potentials (Tomizawa [1971], Edmonds and Karp [1972]). An alternative method, which transforms the transportation problem to a nonlinear programming problem, with computational results, was given by Gerstenhaber [1958,1960].

Polynomial-time algorithms

Edmonds and Karp [1972] gave the first polynomial-time algorithm for the minimum-cost flow problem, based on capacity-scaling. They realized that in fact the method is only *weakly* polynomial; that is, the number of steps depends also on the size of the numbers in the input:

> Although it is comforting to know that the minimum-cost flow algorithm terminates, the bounds on the number of augmentations are most unfavorable. The scaling method of the next two sections is a variant of this algorithm in which the bound depends logarithmically, rather than linearly, on the capacities. A challenging open problem is to emulate the results of Section 1.2 for the maximum-value flow problem by giving a method for the minimum-cost flow problem having a bound on computation which is a polynomial in the number of nodes, and is independent of both costs and capacities.

Tarjan [1983] wrote: 'There is still much to be learned about the minimum cost flow problem'. Soon after, Edmonds and Karp's question was resolved by Tardos [1985a], by giving a strongly polynomial-time minimum-cost circulation algorithm. Her work has inspired a stream of further developments, part of which was discussed in Chapter 12.

Chapter 22

Transversals

The study of transversals of a family of sets is close to that of matchings in a bipartite graph, but with a shift in focus. While matchings are subsets of the edge set, transversals are subsets of one of the colour classes. This gives rise to a number of optimization and polyhedral problems and results that deserve special attention.
In this chapter we study transversals of one family of sets, while in the next chapter we go over to *common* transversals of two families of sets.

22.1. Transversals

Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of sets. A set T is called a *transversal* of $\mathcal{A}$ if there exist distinct elements $a_1 \in A_1, \ldots, a_n \in A_n$ such that $T = \{a_1, \ldots, a_n\}$. So T is an *unordered* set with $|T| = n$. (Instead of 'transversal' one uses also the term *system of distinct representatives* or *SDR*.)

Transversals are closely related to matchings in bipartite graphs. In particular, the basic result on the existence of a transversal (Hall [1935]), is a consequence of Kőnig's matching theorem. This can be seen with the following basic construction of a bipartite graph $G = (V, E)$ associated with a family $\mathcal{A} = (A_1, \ldots, A_n)$ of subsets of a set S:

(22.1) $$V := \{1, \ldots, n\} \cup S, \qquad E := \{\{i, s\} \mid i = 1, \ldots, n; s \in A_i\},$$

assuming that S is disjoint from $\{1, \ldots, n\}$ (which for our purposes can be done without loss of generality). So G has colour classes $\{1, \ldots, n\}$ and S. (This construction was given by Skolem [1917].)

Then trivially

(22.2) a set T is a transversal of $\mathcal{A}$ if and only if G has a matching M of size n such that T is the set of vertices in S covered by M.

So the existence of a transversal of $\mathcal{A}$ can be reduced to the existence of a matching in G of size n. Hence Kőnig's matching theorem applies to the existence of transversals.

It is convenient to introduce the following notation, for any family $(A_1, \ldots, A_n)$ of sets and any $I \subseteq \{1, \ldots, n\}$:

(22.3) $$A_I := \bigcup_{i \in I} A_i.$$

Theorem 22.1 (Hall's marriage theorem)**.** *A family* $\mathcal{A} = (A_1, \ldots, A_n)$ *of sets has a transversal if and only if*

(22.4) $$|A_I| \geq |I|$$

for each subset I *of* $\{1, \ldots, n\}$.

Proof. Necessity of the condition being easy, we prove sufficiency. Let G be the graph associated to $\mathcal{A}$ (as in (22.1)). Now the theorem is equivalent to Theorem 16.7 (taking $U := \{1, \ldots, n\}$). ∎

Condition (22.4) is called *Hall's condition.* The name 'marriage theorem' is due to Weyl [1949].

The polynomial-time algorithm given in Section 16.3 for finding a maximum matching in a bipartite graph directly yields a polynomial-time algorithm for finding a transversal of a family $(A_1, \ldots, A_n)$ of sets. In fact, Theorem 16.5 implies an $O(\sqrt{n}\, m)$ algorithm, where $m := \sum_i |A_i|$.

22.1a. Alternative proofs of Hall's marriage theorem

We give two alternative, direct proofs of the sufficiency of Hall's condition (22.4) for the existence of a transversal. Call a subset I of $\{1, \ldots, n\}$ *tight* if equality holds in (22.4).

If there is a $y \in A_n$ such that $A_1 \setminus \{y\}, \ldots, A_{n-1} \setminus \{y\}$ has a transversal, then we are done. Hence, we may assume that for each $y \in A_n$ there is a tight $I \subseteq \{1, \ldots, n-1\}$ with $y \in A_I$ (using induction).

The proof given by Easterfield [1946] (also by M. Hall [1948], Halmos and Vaughan [1950], and Mann and Ryser [1953]) continues as follows. Choose any such tight subset I. Without loss of generality, $I = \{1, \ldots, k\}$. By induction, $(A_1, \ldots, A_k)$ has a transversal, which must be $T := A_I$. Moreover, $(A_{k+1} \setminus T, \ldots, A_n \setminus T)$ has a transversal, Z say. This follows inductively, since for each $J \subseteq \{k+1, \ldots, n\}$,

(22.5) $$\Big|\bigcup_{i \in J}(A_i \setminus T)\Big| = \Big|\bigcup_{i \in I \cup J} A_i\Big| - |T| \geq |I| + |J| - |T| = |J|.$$

Then $T \cup Z$ is a transversal of $(A_1, \ldots, A_k, A_{k+1}, \ldots, A_n)$.

The proof due to Everett and Whaples [1949] continues slightly different. They noted that the collection of tight subsets of $\{1, \ldots, n\}$ is closed under taking intersections and unions. That is, if I and J are tight, then also $I \cap J$ and $I \cup J$ are tight, since

(22.6) $$|I| + |J| = |A_I| + |A_J| \geq |A_{I \cap J}| + |A_{I \cup J}| \geq |I \cap J| + |I \cup J| = |I| + |J|,$$

giving equality throughout. (In (22.6), the first inequality holds as $A_{I \cap J} \subseteq A_I \cap A_J$ and $A_{I \cup J} = A_I \cup A_J$.)

Since for each $y \in A_n$ there is a tight subset I of $\{1, \ldots, n-1\}$ with $y \in A_I$, it follows, by taking the union of them, that there is a tight subset I of $\{1, \ldots, n-1\}$ with $A_n \subseteq A_I$. For $J := I \cup \{n\}$ this gives the contradiction $|A_J| = |A_I| = |I| < |J|$.

The the closedness of tight subsets under intersections and unions was also noticed by Maak [1936] and Weyl [1949], who gave alternative proofs of a theorem of Rado and Hall's marriage theorem, respectively.

Edmonds [1967b] gave a linear-algebraic proof of Hall's marriage theorem (cf. Section 16.2b). Ford and Fulkerson [1958c] derived Hall's marriage theorem from the max-flow min-cut theorem.

22.2. Partial transversals

Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of sets. A set T is called a *partial transversal* if it is a transversal of some subfamily $(A_{i_1}, \ldots, A_{i_k})$ of $(A_1, \ldots, A_n)$. (Instead of 'partial transversal' one uses also the term *partial system of distinct representatives* or *partial SDR*.)

Again, by the construction (22.1), we can study partial transversals with the help of bipartite matching theory. In particular, if G is the graph associated to a family $\mathcal{A}$ of subsets of a set S,

(22.7) a set T is a partial transversal of $\mathcal{A}$ if and only if G has a matching M such that T is the set of vertices in S covered by M.

This yields the following so-called defect form of Hall's marriage theorem, which is equivalent to Kőnig's matching theorem (cf. Ore [1955]):

Theorem 22.2 (defect form of Hall's marriage theorem)**.** *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of subsets of a set S. Then the maximum size of a partial transversal of $\mathcal{A}$ is equal to the minimum value of*

(22.8) $$|S \setminus X| + |\{i \mid A_i \cap X \neq \emptyset\}|,$$

where X ranges over all subsets of S.

Proof. Let G be the graph constructed in (22.1). The maximum size of a partial transversal of $\mathcal{A}$ is equal to the maximum size of a matching in G. By Kőnig's matching theorem, this is equal to the minimum size of a vertex cover of G. This minimum is attained by a vertex cover of form $(S \setminus X) \cup \{i \mid A_i \cap X \neq \emptyset\}$, which shows the theorem. ∎

An equivalent way of characterizing the maximum size of a partial transversal is:

Corollary 22.2a. *The maximum size of a partial transversal of $\mathcal{A}$ is equal to the minimum value of*

(22.9) $$\Big|\bigcup_{i \in I} A_i\Big| + n - |I|,$$

taken over $I \subseteq \{1, \ldots, n\}$.

Proof. Directly from Theorem 22.2, since we can assume that $S \setminus X = A_I$ where $I := \{i \mid A_i \cap X = \emptyset\}$. ∎

Note that it needs an argument to state that each partial transversal is a subset of a transversal, if a transversal exists. This was shown by Hoffman and Kuhn [1956b] (solving a problem of Mann and Ryser [1953]):

Theorem 22.3. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a system of sets having a transversal. Then each partial transversal is contained in a transversal.*

Proof. Directly from Theorem 16.8, using construction (22.1) (taking $R := \{1, \ldots, n\} \cup T$, where T is a partial transversal). ∎

One can generalize this to the case where the family need not have a transversal:

Theorem 22.4. *Let $\mathcal{A}$ be a family of sets. Then each partial transversal is contained in a maximum-size partial transversal.*

Proof. Again directly from Theorem 16.8, using construction (22.1). ∎

In other words, each inclusionwise maximal partial transversal is a maximum-size partial transversal. This is the basis of the fact that partial transversals form the independent sets of a matroid — see Chapter 39. It is equivalent to:

Corollary 22.4a (exchange property of transversals)**.** *Let $\mathcal{A}$ be a family of sets and let T and T' be partial transversals of $\mathcal{A}$, with $|T| < |T'|$. Then there exists an $s \in T' \setminus T$ such that $T \cup \{s\}$ is a partial transversal.*

Proof. To prove this, we can assume that each set in $\mathcal{A}$ is contained in $T \cup T'$. This implies that, if no s as required exists, T is an inclusionwise maximal partial transversal. However, as $|T'| > |T|$, this contradicts Theorem 22.4. ∎

Brualdi and Scrimger [1968] (extending a result of Mirsky and Perfect [1967]) observed:

Theorem 22.5. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of sets, let k be the maximum size of a partial transversal, and let $\mathcal{A}' = (A_1, \ldots, A_k)$ have a transversal. Then each maximum-size partial transversal of $\mathcal{A}$ is a transversal of $\mathcal{A}'$.*

Proof. Via construction (22.1) this follows from Corollary 16.8b. ∎

So when studying the collection of partial transversals of a certain collection $\mathcal{A}$ of sets, we can assume that $\mathcal{A}$ has a transversal.

22.3. Weighted transversals

Consider the problem of finding a minimum-weight transversal: given a family $\mathcal{A} = (A_1, \ldots, A_n)$ of subsets of a set S and a weight function $w : S \to \mathbb{Q}$, find a transversal T of $\mathcal{A}$ minimizing $w(T)$. This problem can be easily reduced to a minimum-weight perfect matching problem, implying that a minimum-weight transversal can be found in strongly polynomial time. In fact:

Theorem 22.6. *A minimum-weight transversal can be found in time $O(nm)$ where n is the number of sets and $m := \sum_i |A_i|$.*

Proof. Make the graph G as in (22.1) and define $w(\{i, s\}) := w(s)$ for each edge $\{i, s\}$ of G. Denote $R := \{1, \ldots, n\}$. Starting with $M = \emptyset$, we can apply the Hungarian method, to obtain an extreme matching of size n. The elements of S covered by M form a maximum-weight transversal. As each iteration of the Hungarian method takes $O(m)$ time, this gives the theorem. ∎

Note that in this algorithm, we grow a partial transversal until it is a (complete) transversal. In this respect it is a ‘greedy method’: we never backtrack. Again, this is a preview of the fact that transversals form a ‘matroid’ — see Chapter 39.

The method similarly solves the problem of finding a maximum-weight partial transversal:

Theorem 22.7. *A maximum-weight partial transversal can be found in time $O(rm)$, where r is the maximum size of a partial transversal and where $m := \sum_i |A_i|$.*

Proof. As above. ∎

22.4. Min-max relations for weighted transversals

We can also obtain a min-max relation for the minimum weight of a transversal:

Theorem 22.8. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of subsets of a set S having a transversal and let $w : S \to \mathbb{Z}$ be a weight function. Then the minimum weight of a transversal of $\mathcal{A}$ is equal to the maximum value of*

$$y(S) + \sum_{i=1}^{n} \min_{s \in A_i} (w(s) - y(s)) \tag{22.10}$$

taken over $y : S \to \mathbb{Z}_+$.

Proof. Let $t := |S|$. For $i = n+1, \ldots, t$, let $A_i := S$. Consider the bipartite graph $G = (V, E)$ defined by (22.1), for the family $(A_1, \ldots, A_t)$. Define a length function l on the edges of G as follows. For any edge $e = is$ of G, with $s \in A_i$, define $l_e := w(s)$ if $i \leq n$ and $l_e := 0$ otherwise. Then the minimum weight of a transversal of $(A_1, \ldots, A_n)$ is equal to the minimum length of a perfect matching in G. By Theorem 17.5 (a variant of Egerváry's theorem), the latter value is equal to the maximum value of $y(V)$ where $y \in \mathbb{Q}^V$ with $y(s) + y(i) \leq l(is)$ for each $i = 1, \ldots, t$ and $s \in A_i$. We can assume that the minimum of $y(s)$ over $s \in S$ is equal to 0 (since subtracting a constant to $y(s)$ for any $s \in S$ and adding it to $y(i)$ for any $i \in \{1, \ldots, t\}$ maintains the properties required for y). Then $y(i) = \min_{s \in A_i}(w(s) - y(s))$ if $i \leq n$ and $y(i) = 0$ if $i > n$. So $y(V)$ is equal to the value of (22.10). ∎

A min-max relation for the maximum weight of a partial transversal follows similarly:

Theorem 22.9. *Let $\mathcal{A} = (A_1, \ldots, A_k)$ be a family of subsets of a set S and let $w : S \to \mathbb{Z}_+$ be a weight function. Then the maximum weight of a partial transversal of $\mathcal{A}$ is equal to the minimum value of*

$$(22.11) \qquad y(S) + \sum_{i=1}^{k} \max\{0, \max_{s \in A_i}(w(s) - y(s))\}$$

over functions $y : S \to \mathbb{Z}_+$.

Proof. Directly from Egerváry's theorem (Theorem 17.1), using construction (22.1). ∎

22.5. The transversal polytope

Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of subsets of a set S. The *partial transversal polytope* $P_{\text{partial transversal}}(\mathcal{A})$ of $\mathcal{A}$ is the convex hull of the incidence vectors (in $\mathbb{R}^S$) of the partial transversals of $\mathcal{A}$. That is,

$$(22.12) \qquad P_{\text{partial transversal}}(\mathcal{A}) = \text{conv.hull}\{\chi^T \mid T \text{ is a partial transversal of } \mathcal{A}\}.$$

It is easy to see that each vector x in the partial transversal polytope satisfies:

$$(22.13) \qquad \begin{array}{lll} \text{(i)} & 0 \leq x_s \leq 1 & \text{for each } s \in S, \\ \text{(ii)} & x(S \setminus A_I) \leq n - |I| & \text{for each } I \subseteq \{1, \ldots, n\}. \end{array}$$

Corollary 22.9a. *System* (22.13) *determines the partial transversal polytope and is TDI.*

Proof. Consider a weight function $w : S \to \mathbb{Z}_+$. Let ω be the maximum weight of a partial transversal. By Theorem 22.9, there exists a function $y : S \to \mathbb{Z}_+$ such that

$$\omega = y(S) + \sum_{i=1}^{n} \max\{0, \max_{s\in A_i}(w(s) - y(s))\}. \tag{22.14}$$

For each $j \in \mathbb{Z}_+$, let I_j be the set of $i \in \{1,\ldots,n\}$ with

$$\max_{s\in A_i}(w(s) - y(s)) \leq j. \tag{22.15}$$

So $I_j = \{1,\ldots,n\}$ for j large enough.

Then

$$w - y \leq \sum_{j=0}^{\infty} \chi^{S\setminus A_{I_j}}, \tag{22.16}$$

since for $k := w(s) - y(s)$, we have for each $j < k$ there is no $i \in I_j$ with $s \in A_i$. Hence $s \in S \setminus A_{I_j}$ for all $j < k$. So y and the I_j give an integer feasible dual solution.

The fact that they are optimum follows from:

$$\begin{aligned} y(S) + \sum_{j=0}^{\infty}(n - |I_j|) &= y(S) + \sum_{j=0}^{\infty} \sum_{\substack{i=1 \\ \max_{s\in A_i}(w(s)-y(s))>j}}^{n} 1 \\ &= y(S) + \sum_{i=1}^{n} \sum_{\substack{j=0 \\ \max_{s\in A_i}(w(s)-y(s))>j}}^{\infty} 1 \\ &= y(S) + \sum_{i=1}^{n} \max\{0, \max_{s\in A_i}(w(s) - y(s))\} = \omega, \end{aligned} \tag{22.17}$$

by (22.14). ∎

Let $\mathcal{A} = (A_1,\ldots,A_n)$ be a family of subsets of a set S. The *transversal polytope* $P_{\text{transversal}}(\mathcal{A})$ of $\mathcal{A}$ is the convex hull of the incidence vectors (in $\mathbb{R}^S$) of the transversals of $\mathcal{A}$. That is,

$$P_{\text{transversal}}(\mathcal{A}) = \text{conv.hull}\{\chi^T \mid T \text{ is a transversal of } \mathcal{A}\}. \tag{22.18}$$

It is easy to see that each vector x in the transversal polytope satisfies:

$$\begin{array}{lll} \text{(i)} & 0 \leq x_s \leq 1 & \text{for each } s \in S, \\ \text{(ii)} & x(A_I) \geq |I| & \text{for each } I \subseteq \{1,\ldots,n\}, \\ \text{(iii)} & x(S) = n. & \end{array} \tag{22.19}$$

Corollary 22.9b. *System* (22.19) *determines the transversal polytope and is TDI.*

Proof. The transversal polytope is the facet of the partial transversal polytope determined by the equality $x(S) = n$. This is constraint (22.13)(ii) for $I = \emptyset$, set to equality. Now each inequality in (22.19) is a nonnegative integer combination of the inequalities in (22.13) and of $-x(S) \leq -n$ (since $-x(A_I) = x(S \setminus A_I) - x(S) \leq (n - |I|) - n = -|I|$). So using Theorem 5.25, the corollary follows. ∎

One may note that the number of facets of the matching polytope of a bipartite graph $G = (V, E)$ is at most $|V| + |E|$, while the number of facets of the closely related partial transversal polytope can be exponential in the size of the input (the family $\mathcal{A}$). In fact, the partial transversal polytope is a projection of the matching polytope of the corresponding graph. Thus we have an illustration of the phenomenon that projection can increase the number of facets dramatically, while this has no negative effect on the complexity of the corresponding optimization problem.

22.6. Packing and covering of transversals

The following min-max relation for the maximum number of disjoint transversals is an easy consequence of Hall's marriage theorem:

Theorem 22.10. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of sets and let k be a natural number. Then $\mathcal{A}$ has k disjoint transversals if and only if*

$$|A_I| \geq k|I| \tag{22.20}$$

for each subset I of $\{1, \ldots, n\}$.

Proof. Replace each set A_i by k copies, yielding the family $\mathcal{A}'$. Then by Hall's marriage theorem and (22.20), $\mathcal{A}'$ has a transversal. This can be split into k transversals of $\mathcal{A}$. ∎

A generalization to disjoint partial transversals of prescribed sizes was given by Higgins [1959] (cf. Mirsky [1966], Mirsky and Perfect [1966]):

Theorem 22.11. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of sets and let $d_1, \ldots, d_k \in \{1, \ldots, n\}$. Then $\mathcal{A}$ has k disjoint partial transversals of sizes $d_1, \ldots, d_k$ respectively if and only if*

$$|A_I| \geq \sum_{j=1}^{k} \max\{0, |I| - n + d_j\} \tag{22.21}$$

for each $I \subseteq \{1, \ldots, n\}$.

Proof. Necessity follows from the fact that if $T_1, \ldots, T_k$ are partial transversals as required, then

$$|A_I| \geq \sum_{j=1}^{k} |A_I \cap T_j| \geq \sum_{j=1}^{k} \max\{0, |I| - n + d_j\} \tag{22.22}$$

for each $I \subseteq \{1, \ldots, n\}$, since $|A_I \cap T_j| + (n - d_j) \geq |I|$.

To see sufficiency, let $B_1, \ldots, B_k$ be disjoint sets, disjoint also from all A_i, with $|B_j| = n - d_j$ for $j = 1, \ldots, k$. Define $A_{i,j} := A_i \cup B_j$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. Then $\mathcal{A}$ has k disjoint partial transversals as required, if $(A_{i,j} \mid i = 1, \ldots, n; j = 1, \ldots, k)$ has a transversal. So it suffices to check Hall's condition (22.4) for the latter family. Take $K \subseteq \{1, \ldots, n\} \times \{1, \ldots, k\}$. Let $I := \{i \mid \exists j : (i,j) \in K\}$ and $J := \{j \mid \exists i : (i,j) \in K\}$. Then

$$\begin{aligned} |\bigcup_{(i,j)\in K} A_{i,j}| &= |\bigcup_{i\in I} A_i| + |\bigcup_{j\in J} B_j| \\ &\geq \sum_{j=1}^{k} \max\{0, |I| - n + d_j\} + \sum_{j\in J}(n - d_j) \geq \sum_{j\in J} |I| = |I| \cdot |J| \\ &\geq |K|. \end{aligned} \tag{22.23}$$

∎

(A proof based on total unimodularity was given by Hoffman [1976b].)

As to covering by partial transversals, Mirsky [1971b] (p. 51) mentioned that R. Rado proved in 1965:

Theorem 22.12. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of subsets of a set S and let k be a natural number. Then S can be covered by k partial transversals if and only if*

$$k \cdot |\{i \mid A_i \cap X \neq \emptyset\}| \geq |X| \tag{22.24}$$

for each subset X of S.

Proof. Let $\mathcal{A}'$ be the family obtained from $\mathcal{A}$ by taking each set k times. Then S can be covered by k partial transversals if and only if S is a partial transversal of $\mathcal{A}'$. By the defect form of Hall's marriage theorem (Theorem 22.2), this last is equivalent to the condition that

$$|S \setminus X| + k \cdot |\{i \mid A_i \cap X \neq \emptyset\}| \geq |S| \tag{22.25}$$

for each $X \subseteq S$. This is equivalent to (22.24). ∎

For covering by partial transversals of prescribed size, there is the following easy consequence of the exchange property of transversals (Corollary 22.4a):

Theorem 22.13. *Let $\mathcal{A}$ be a family of subsets of a set S and let $k \in \mathbb{Z}_+$. If S can be covered by k partial transversals, it can be covered by k partial transversals each of size $\lfloor |S|/k \rfloor$ or $\lceil |S|/k \rceil$.*

Proof. Let $T_1, \ldots, T_k$ be partial transversals partitioning S. If $|T_i| \geq |T_j| + 2$ for some i, j, we can replace T_i and T_j by $T_i \setminus \{s\}$ and $T_j \cup \{s\}$ for some

$s \in T_i$. Repeating this, we finally achieve that $\big||T_i| - |T_j|\big| \leq 1$ for all i, j. Hence $\lfloor |S|/k \rfloor \leq |T_i| \leq \lceil |S|/k \rceil$ for all i. ∎

22.7. Further results and notes

22.7a. The capacitated case

Capacitated versions of the theorems on transversals can be derived straightforwardly from the previous results. First, Halmos and Vaughan [1950] showed the following generalized (but straightforwardly equivalent) version of Hall's marriage theorem:

Theorem 22.14. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of sets and let $b \in \mathbb{Z}_+^n$. Then there exist disjoint subsets $B_1, \ldots, B_n$ of $A_1, \ldots, A_n$ respectively with $|B_i| = b_i$ for $i = 1, \ldots, n$ if and only if*

(22.26) $$|A_I| \geq b(I)$$

for each $I \subseteq \{1, \ldots, n\}$.

Proof. Let $\mathcal{A}'$ be the family of sets obtained from $\mathcal{A}$ by repeating any A_i b_i times. Then the existence of the B_i is equivalent to the existence of a transversal of $\mathcal{A}'$. Moreover, (22.26) is equivalent to Hall's condition for $\mathcal{A}'$. ∎

This theorem concerns taking multiplicities on the sets in $\mathcal{A}$. If we put multiplicities on the elements of S, there is the following observation of R. Rado (as reported by Mirsky and Perfect [1966]):

Theorem 22.15. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of sets and let $r \in \mathbb{Z}_+$. Then there exist $x_i \in A_i$ ($i = 1, \ldots, n$) such that no element occurs more than r times among the s_i if and only if*

(22.27) $$|A_I| \geq |I|/r$$

for each $I \subseteq \{1, \ldots, n\}$.

Proof. Let $\mathcal{A}'$ be the family of sets obtained from $\mathcal{A}$ by replacing any A_i by $A_i \times \{1, \ldots, r\}$. Then the existence of the required s_i is equivalent to the existence of a transversal of $\mathcal{A}'$. Moreover, (22.27) is equivalent to Hall's condition for $\mathcal{A}'$. ∎

These theorems are in fact direct consequences of the general Theorem 21.28. This theorem moreover gives the following result of Vogel [1961], which puts multiplicities both on the sets in $\mathcal{A}$ and on the elements of the underlying set S:

Theorem 22.16. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of subsets of a set S. Let $a \in \mathbb{Z}_+^n$ and $b \in \mathbb{Z}_+^S$. Then there exist subsets $B_1, \ldots, B_n$ of $A_1, \ldots, A_n$ respectively such that $|B_i| = a_i$ for $i = 1, \ldots, n$ and such that each $s \in S$ occurs in at most $b(s)$ of the B_i if and only if*

(22.28) $$b(X) + \sum_{i \in I} |A_i \setminus X| \geq a(I)$$

for each $X \subseteq S$ *and each* $I \subseteq \{1, \ldots, n\}$.

Proof. Consider the system

$$(22.29) \qquad \begin{array}{ll} 0 \leq x(i,s) \leq 1 & \text{for } i \in \{1,\ldots,n\} \text{ and } s \in A_i, \\ a_i \leq x(\delta(i)) \leq a_i & \text{for } i \in \{1,\ldots,n\}, \\ 0 \leq x(\delta(s)) \leq b_s & \text{for } s \in S, \end{array}$$

and apply Theorem 21.28. ∎

This has as special case (Vogel [1961]):

Corollary 22.16a. *Let* $\mathcal{A} = (A_1, \ldots, A_n)$ *be a family of subsets of a set* S *and let* $r, s \in \mathbb{Z}_+$. *Then there exist subsets* $B_1, \ldots, B_n$ *of* $A_1, \ldots, A_n$ *respectively such that* $|B_i| = s$ *for each* i *and such that each element belongs to at most* r *of the* B_i *if and only if*

$$(22.30) \qquad r|X| + \sum_{i \in I} |A_i \setminus X| \geq s|I|$$

for each $I \subseteq \{1, \ldots, n\}$ *and each* $X \subseteq S$.

Proof. This is a special case of Theorem 22.16. ∎

Similar methods apply to systems of *restricted* representatives, considered by Ford and Fulkerson [1958c]. Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a collection of subsets of a set S and let $a, b \in \mathbb{Z}_+^S$ with $a \leq b$. A *system of restricted representatives* (or *SRR*) of $\mathcal{A}$ (with respect to a and b) is a sequence $(s_1, \ldots, s_n)$ such that

$$(22.31) \qquad \begin{array}{l} \text{(i) } s_i \in A_i \text{ for } i = 1, \ldots, n; \\ \text{(ii) } a(s) \leq |\{i \mid s_i = s\}| \leq b(s) \text{ for } s \in S. \end{array}$$

Ford and Fulkerson [1958c] showed:

Theorem 22.17. $\mathcal{A}$ *has a system of restricted representatives if and only if*

$$(22.32) \qquad a(S - \bigcup_{i \notin I} A_i) \leq |I| \leq b(\bigcup_{i \in I} A_i)$$

for each $I \subseteq \{1, \ldots, n\}$.

Proof. Consider the system

$$(22.33) \qquad \begin{array}{ll} 0 \leq x(i,s) \leq \infty & \text{for } i \in \{1,\ldots,n\}, s \in A_i, \\ x(\delta(i)) = 1 & \text{for } i \in \{1,\ldots,n\}, \\ a_s \leq x(\delta(s)) \leq b_s & \text{for } s \in S, \end{array}$$

and apply Theorem 21.28. ∎

(For an alternative proof, see Mirsky [1968a].)

Considering both upper and lower bounds, the following theorem of Hoffman and Kuhn [1956a] follows from Hoffman's circulation theorem:

Theorem 22.18. *Let* $\mathcal{A} = (A_1, \ldots, A_n)$ *be a collection of subsets of a set* S, *let* $\mathcal{P} = (P_1, \ldots, P_m)$ *be a partition of* S, *and let* $a, b \in \mathbb{Z}_+^m$ *with* $a \leq b$. *Then* $\mathcal{A}$ *has a transversal* T *satisfying* $a_i \leq |T \cap P_i| \leq b_i$ *for each* $i = 1, \ldots, m$ *if and only if*

$$|P_I \cap A_J| \geq \max\{|J| - b(\overline{I}), |J| - n + a(I)\} \tag{22.34}$$

for all $I \subseteq \{1, \ldots, m\}$ *and* $J \subseteq \{1, \ldots, n\}$*, where* $\overline{I} := \{1, \ldots, n\} \setminus I$.

Proof. Make a directed graph as follows. Its vertex set is $\{r\} \cup \{u_1, \ldots, u_n\} \cup S \cup \{p_1, \ldots, p_m\} \cup \{t\}$, and there are arcs

(22.35) (r, u_i) for $i = 1, \ldots, n$,
(u_i, s) for $i = 1, \ldots, n$ and $s \in A_i$,
(s, p_j) for $j = 1, \ldots, m$ and $s \in P_j$,
(p_j, t) for $j = 1, \ldots, m$.

Put lower bound a_j and capacity b_j on each arc (p_j, t). On any other arc, put lower bound 0 and capacity 1. Then a transversal as required exists if and only if there is an integer $r - t$ flow of value n satisfying the lower bounds and capacities. Applying Corollary 11.2e gives the present theorem. ∎

(The proof of Hoffman and Kuhn [1956a] is based on the duality theorem of linear programming. Gale [1956,1957] and Fulkerson [1959a] derived the theorem from network flow theory. For further extensions, see Mirsky [1968b].)

22.7b. A theorem of Rado

Rado [1938] proved the following generalization (but also consequence) of Hall's marriage theorem:

Theorem 22.19. *Let* $A_1, \ldots, A_n$*,* $B_1, \ldots, B_n$ *be sets. Then there exists an injection* $f : A_1 \cup \cdots \cup A_n \to B_1 \cup \cdots \cup B_n$ *such that* $f[A_i] \subseteq B_i$ *for* $i = 1, \ldots, n$ *if and only if each set obtained by intersections and unions of sets from* $A_1, \ldots, A_n$ *has size at most the size of the result of the same operations applied to* $B_1, \ldots, B_n$.

Proof. Let $A := A_1 \cup \cdots \cup A_n$. For each $s \in A$, define

$$C_s := \bigcap_{\substack{i \\ s \in A_i}} B_i. \tag{22.36}$$

Then for each subset S of A one has

$$\Big|\bigcup_{s \in S} C_s\Big| = \Big|\bigcup_{s \in S} \bigcap_{\substack{i \\ s \in A_i}} B_i\Big| \geq \Big|\bigcup_{s \in S} \bigcap_{\substack{i \\ s \in A_i}} A_i\Big| \geq |S|. \tag{22.37}$$

Hence, by Hall's marriage theorem, $(C_s \mid s \in A)$ has a transversal. This gives an injection $f : A \to B_1 \cup \cdots \cup B_n$ with $f(s) \in C_s$ for $s \in A$. This is as required. ∎

22.7c. Further notes

Shmushkovich [1939], de Bruijn [1943], Hall [1948], Henkin [1953], Tutte [1953], Mirsky [1967], Rado [1967a] (with H.A. Jung), Brualdi and Scrimger [1968], Folkman [1970], McCarthy [1973], Damerell and Milner [1974], Steffens [1974], Podewski and Steffens [1976], Nash-Williams [1978], Aharoni [1983c], and Aharoni, Nash-Williams, and Shelah [1983] considered extensions of Hall's marriage theorem to the infinite case. Perfect [1968] gave proofs of theorems on transversals with Menger's theorem.

For a 'very general theorem' see Brualdi [1969a]. For counting transversals, see Hall [1948], Rado [1967b], and Ostrand [1970].

Gale [1968] showed that for any family $\mathcal{A}$ of subsets of a finite set S and any total order $<$ on S, there is a transversal T of $\mathcal{A}$ such that for each transversal T' of $\mathcal{A}$ there exists a one-to-one function $\phi : T' \to T$ with $\phi(s) \geq s$ for each $s \in T'$. (Gale showed that this in fact characterizes matroids.)

The standard work on transversal theory is Mirsky [1971b]. Also Brualdi [1975] and Welsh [1976] provide surveys. Surveys on the relations between the theorems of Hall, Kőnig, Menger, and Dilworth were given by Jacobs [1969] and Reichmeider [1984].

22.7d. Historical notes on transversals

Results on transversals go back to the papers by Miller [1910] and Chapman [1912], who showed that if H is a subgroup of a finite group G, then the partitions of G into left cosets and into right cosets have a common transversal. This is an easy result, due to the fact that each component of the intersection graph of left and right cosets is a complete bipartite graph. This implies that any common partial transversal can be extended to a common (full) transversal (Chapman [1912]).

This result was extended by Scorza [1927] to: if H and K are subgroups of a finite group G, with $|H| = |K|$, then there exist $x_1, \ldots, x_m \in G$ with $x_1 H \cup \cdots \cup x_m H = G = K x_1 \cup \cdots \cup K x_m$ and $m = |G|/|H|$. (Again this can be derived easily from the fact that each component of the intersection graph of left cosets of H and right cosets of K is a complete bipartite graph.)

As an extension of these results, in October 1926, van der Waerden [1927] presented the following theorem at the *Mathematisches Seminar* in Hamburg:

> *Es seien zwei Klasseneinteilungen einer endlichen Menge $\mathcal{M}$ gegeben. Die eine soll die Menge in μ zueinander fremde Klassen $\mathcal{A}_1, \ldots, \mathcal{A}_\mu$ zu je n Elementen zerlegen, die andere ebenfalls in μ fremde Klassen $\mathcal{B}_1, \ldots, \mathcal{B}_\mu$ zu je n Elementen. Dann gibt es ein System von Elementen $x_1, \ldots, x_\mu$, derart, daß jede A-Klasse und ebenso jede B-Klasse under den x_i durch ein Element vertreten wird.*[41]

The proof of van der Waerden is based on an augmenting path argument. Moreover, van der Waerden remarked that E. Artin had communicated orally to him that the result can be sharpened to the existence of n disjoint such common transversals.

In a note added in proof, van der Waerden observed that his theorem follows from Kőnig's theorem on the existence of a perfect matching in a regular bipartite graph:

> **Zusatz bei der Korrektur.** Ich bemerke jetzt, daß der hier bewiesene Satz mit einem Satz von DÉNES KŐNIG über reguläre Graphen äquivalent ist.[42]

[41] Let be given two partitions of a finite set $\mathcal{M}$. One of them should decompose the set into μ mutually disjoint classes $\mathcal{A}_1, \ldots, \mathcal{A}_\mu$ each of n elements, the other likewise in μ disjoint classes $\mathcal{B}_1, \ldots, \mathcal{B}_\mu$ each of n elements. Then there exists a system of elements $x_1, \ldots, x_\mu$ such that each $\mathcal{A}$-class and likewise each $\mathcal{B}$-class is represented by one element among the x_i.

[42] **Note added in proof.** I now notice that the theorem proved here is equivalent to a theorem of DÉNES KŐNIG on regular graphs.

Van der Waerden's article is followed by an article of Sperner [1927] (presented at the *Mathematisches Seminar* in Januari 1927) that gives a 'simple proof' of van der Waerden's result. We quote the full article (containing page references to van der Waerden's paper):

> Der auf S. 185 ff. bewiesene Satz gestattet auch folgenden einfachen Beweis.
> Der Satz lautete:
> Zwei beliebige Klasseneinteilungen von $m \cdot n$ Elementen in m Klassen zu je n Elementen haben immer ein gemeinsames Repräsentantensystem (vgl. S. 185).
> Der Satz ist evident für die Klassenzahl 1. Wir nehmen an, er sei bewiesen für die Klassenzahl m (und beliebiges n). Dan folgt für dieses m:
> 1. Die beiden Klasseneinteilungen haben sogar n verschiedene und zueinander fremde Repräsentantensysteme.
> Beweis wie auf S. 187 oben.
> 2. Streicht man daher in beiden Einteilungen dieselben k Elemente, wo $0 \leq k \leq n-1$, dan werden höchstens $n-1$ Repräsentantensysteme verletzt und wenigstens eins bleibt erhalten. Da man auch umgekehrt $m \cdot n - k$ Elemente durch k neue ergänzen kann, um diese nachher wieder zu streichen, so gilt:
> Zwei beliebige Klasseneinteilungen von $m \cdot n - k$ Elementen in m Klassen zu je höchstens n Elementen, wo $0 \leq k \leq n - 1$, haben immer ein gemeinsames Repräsentantensystem.
> Nunmehr wenden wir vollständige Induktion an. Es seien zwei Klasseneinteilungen von $(m+1) \cdot n$ Elementen in $m+1$ Klassen zu je n Elementen gegeben. Dann greifen wir aus beiden Einteilungen je eine Klasse heraus, etwa die Klassen $\mathcal{A}$ und $\mathcal{B}$, die aber wenigstens 1 Element gemeinsam haben sollen, etwa A. Streichen wir dann in beiden Einteilungen die in $\mathcal{A}$ und $\mathcal{B}$ vorkommenden Elemente (also höchstens $2n-1$, aber wenigstens n Elemente), so bleiben zwei Klasseneinteilungen von $m \cdot n - k$ Elementen in je m Klassen zu je höchstens n Elementen übrig, wo $0 \leq k \leq n-1$. Zwei solche Einteilungen haben aber nach 2. ein gemeinsames Repräsentantensystem, das man sofort durch Hinzufügen von A zu einem gemeinsamen Repräsentantensysteme der beiden Einteilungen von $(m+1)n$ Elementen erweitert.[43]

[43] The theorem proved on p. 185 and following pages allows also the following simple proof.
The theorem reads:

Two arbitrary partitions of $m \cdot n$ elements into m classes of n elements each, always have a common system of representatives (cf. p. 185).

The theorem is evident for class number 1. We assume that it be proved for class number m (and arbitrary n). Then the following follows for this m:

1. Both partitions even have n different and disjoint systems of representatives.

Proof like on p. 187 above.

2. Therefore, if one cancels in both partitions the same k elements, where $0 \leq k \leq n-1$, then at most $n-1$ systems of representatives are injured and at least one is preserved. As one can also, reversely, complete $m \cdot n - k$ elements by k new ones, to cancel them after it again, the following therefore holds:

Two arbitrary partitions of $m \cdot n - k$ elements into m classes of at most n elements each, where $0 \leq k \leq n-1$, always have a common system of representatives.

Now we apply complete induction. Let be given two partitions of $(m+1) \cdot n$ elements into $m+1$ classes of n elements each. Then we select from each of the two partitions one class, say the classes $\mathcal{A}$ and $\mathcal{B}$, that however should have at least 1 element in common, say A. If we then cancel in both partitions the elements occurring in $\mathcal{A}$ and $\mathcal{B}$ (so at most $2n-1$, but at least n elements), two partitions of $m \cdot n - k$ elements into m classes of at most n elements each thus remain, where $0 \leq k \leq n-1$. Two such partitions have however, according to 2., a common system of representatives, that one extends, by adding A, to a common system of representatives of both partitions of $(m+1)n$ elements.

Hall

After having mentioned Kőnig's result on the existence of a common transversal for two partitions of a set where all classes have the same size, Hall [1935] said that he is 'concerned with a slightly different problem': to find a transversal

> for a finite collection of (arbitrarily overlapping) subsets of any given set of things. The solution, Theorem 1, is very simple.

Calling a transversal a 'C.D.R. (= complete system of distinct representatives)' and denoting a finite system $T_1, \dots, T_m$ of subsets of a set S by '(1)', Hall formulated his theorem as follows:

> *In order that a C.D.R. of* (1) *shall exist, it is sufficient that for each* $k = 1, 2, \dots, m$ *any selection of* k *of the sets* (1) *shall contain between them at least* k *elements of* S.

This result now is known as 'Hall's marriage theorem'.

In order to prove this theorem, Hall first showed the following lemma. Let $(A_1, \dots, A_n)$ be a system of sets with at least one transversal and let R be the intersection of all transversals. Then there is an $I \subseteq \{1, \dots, n\}$ with $A_I = R$ and $|I| = |R|$.

Hall proved this with the help of an alternating path argument. Having the lemma, the theorem is easy, by induction on n: we may assume that $(A_1, \dots, A_{n-1})$ has a transversal; let R' be the intersection of all these transversals. So by the lemma, $R' = A_{I'}$ for some $I' \subseteq \{1, \dots, n-1\}$ with $|I'| = |R'|$. Hence $A_n \not\subseteq R'$, since otherwise for $I := I' \cup \{n\}$ one has $\left|\bigcup_{i \in I} A_i\right| = |R'| < |I|$. Therefore, $(A_1, \dots, A_{n-1})$ has a transversal not containing A_n as a subset, implying that $(A_1, \dots, A_n)$ has a transversal.

Hall derived as a consequence that if $A_1, \dots, A_n$ and $B_1, \dots, B_n$ are two partitions of a finite set S, then the two partitions have a common transversal if and only if for each subset I of $\{1, \dots, n\}$, the set $\bigcup_{i \in I} A_i$ intersects at least $|I|$ sets among $B_1, \dots, B_n$. Hall remarked that the theorem of Kőnig [1916] on the existence of a perfect matching in a regular bipartite graph follows as an immediate corollary, and that also a theorem of Rado [1933] can be derived (the Kőnig-Rado edge cover theorem — Theorem 19.4), but he did not observe that Hall's marriage theorem is equivalent to a theorem of Kőnig [1931] (Kőnig's matching theorem — Theorem 16.2).

As for *common* transversals, Maak [1936] showed that if $\mathcal{A} = (A_1, \dots, A_n)$ and $\mathcal{B} = (B_1, \dots, B_n)$ are partitions of a finite set S, then $\mathcal{A}$ and $\mathcal{B}$ have a common transversal if and only if for each $I \subseteq \{1, \dots, n\}$, the set $\bigcup_{i \in I} A_i$ contains at most $|I|$ of the sets B_i as a subset. This can be derived from Frobenius' theorem (Frobenius [1917]).

The basic characterization of common transversals of two arbitrary families of sets was given by Ford and Fulkerson [1958c] — see Section 23.1.

Shmushkovich [1939] and de Bruijn [1943] extended the results to the infinite case. Weyl [1949] introduced the name 'marriage theorem' for Hall's marriage theorem. Maak [1952] gave some historical notes.

Chapter 23

Common transversals

We consider sets that are transversals of two families of sets simultaneously. Again we denote, for any family $(A_1, \ldots, A_n)$ of sets and any $I \subseteq \{1, \ldots, n\}$,

$$A_I := \bigcup_{i \in I} A_i.$$

23.1. Common transversals

Let $\mathcal{A}$ and $\mathcal{B}$ be families of sets. A set T is called a *common transversal* of $\mathcal{A}$ and $\mathcal{B}$ if T is a transversal of both $\mathcal{A}$ and $\mathcal{B}$. Similarly, T is called a *common partial transversal* of $\mathcal{A}$ and $\mathcal{B}$ if T is a partial transversal of both $\mathcal{A}$ and $\mathcal{B}$.

When considering two families $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_m)$ of subsets of a set S, it is helpful to construct the following directed graph $D = (V, A)$:

(23.1) $V := \{a_1, \ldots, a_n\} \cup S \cup \{b_1, \ldots, b_m\}$,
$A := \{(a_i, s) \mid i = 1, \ldots, n; s \in A_i\} \cup \{(s, b_i) \mid i = 1, \ldots, m; s \in B_i\}$,

where $a_1, \ldots, a_n, b_1, \ldots, b_m$ are distinct new elements, not in S.

Then one has, if $m = n$:

(23.2) a subset T of S is a common transversal of $\mathcal{A}$ and $\mathcal{B}$ if and only if D has n vertex-disjoint paths from $\{a_1, \ldots, a_n\}$ to $\{b_1, \ldots, b_n\}$ such that T is the set of vertices in S traversed by these paths.

A similar statement can be formulated with respect to common partial transversals.

With Menger's theorem, it yields the following characterization of the existence of a common transversal, due to Ford and Fulkerson [1958c]:

Theorem 23.1. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of sets. Then $\mathcal{A}$ and $\mathcal{B}$ have a common transversal if and only if*

(23.3) $|A_I \cap B_J| \geq |I| + |J| - n$

for all $I, J \subseteq \{1, \ldots, n\}$.

Proof. To see necessity, let T be a common transversal. To prove (23.3), we can assume that $A_i \subseteq T$ and $B_j \subseteq T$ for all i, j. Then

(23.4) $$|A_I \cap B_J| = |A_I| + |B_J| - |A_I \cup B_J| \geq |I| + |J| - |T| \geq |I| + |J| - n$$

for all $I, J \subseteq \{1, \ldots, n\}$.

To see sufficiency, make the digraph D associated to $\mathcal{A}, \mathcal{B}$ as in (23.1). Let $U := \{a_1, \ldots, a_n\}$ and $W := \{b_1, \ldots, b_n\}$. Then by (23.2), $\mathcal{A}$ and $\mathcal{B}$ have a common transversal if D has n disjoint $U - W$ paths. By Menger's theorem, these paths exist if $|C| \geq n$ for each $C \subseteq U \cup S \cup W$ intersecting each $U - W$ path. To check this condition, let $I := \{i \mid a_i \notin C\}$ and $J := \{j \mid b_j \notin C\}$. Then

(23.5) $$C \cap S \supseteq A_I \cap B_J,$$

since $A_I \cap B_J$ is equal to the set of vertices in S that are on a $U - W$ path not intersected by $C \cap (U \cup W)$. So (23.3) implies

(23.6) $$|C \cap S| \geq |A_I \cap B_J| \geq |I| + |J| - n = (n - |C \cap U|) + (n - |C \cap W|) - n,$$

giving $|C| \geq n$. ∎

(For a direct derivation of this theorem from Hall's marriage theorem, see Perfect [1969c]. For a derivation from the Kőnig-Rado edge cover theorem, see Perfect [1980].)

This construction also implies, with Theorem 9.8, that a common transversal of two collections of n subsets of S can be found in time $O(n^{3/2}|S|)$ (cf. Adel'son-Vel'skiĭ, Dinits, and Karzanov [1975]).

Perfect [1968] (cf. McDiarmid [1973]) strengthened Theorem 23.1 to a min-max relation for the maximum size of a common partial transversal:

Corollary 23.1a. *Let* $\mathcal{A} = (A_1, \ldots, A_n)$ *and* $\mathcal{B} = (B_1, \ldots, B_m)$ *be families of sets and let* $k \in \mathbb{Z}_+$. *Then* $\mathcal{A}$ *and* $\mathcal{B}$ *have a common partial transversal of size* k *if and only if*

(23.7) $$|A_I \cap B_J| \geq |I| + |J| - n - m + k$$

for all $I \subseteq \{1, \ldots, n\}$ *and* $J \subseteq \{1, \ldots, m\}$.

Proof. We may assume that $m = n$ (if, say, $n < m$, add $m - n$ copies of $\emptyset$ to $\mathcal{A}$). Let X be a set disjoint from all A_i and B_i with $|X| = n - k$. Replace each A_i by $A'_i := A_i \cup X$ and each B_i by $B'_i := B_i \cup X$. Then $\mathcal{A}$ and $\mathcal{B}$ have a common partial transversal of size k if and only if $\mathcal{A}' = (A'_1, \ldots, A'_n)$ and $\mathcal{B}' = (B'_1, \ldots, B'_n)$ have a common transversal. Applying Theorem 23.1 to $\mathcal{A}'$ and $\mathcal{B}'$ gives this corollary. ∎

Generally, a common partial transversal of families $\mathcal{A}$ and $\mathcal{B}$ need not be contained in a common transversal, even not if a common transversal exists:

let $\mathcal{A} := (\{a\}, \{b, c\})$ and $\mathcal{B} := (\{b\}, \{a, c\})$. Then $\{c\}$ is a common partial transversal, while $\{a, b\}$ is the only common transversal.

The following result of Perfect [1968] and Welsh [1968] characterizes subsets contained in common transversals. It is a special case of a theorem of Ford and Fulkerson [1958c] (cf. Theorem 23.14), and will be derived from Theorem 23.1 with a method of Mirsky and Perfect [1968].

Corollary 23.1b. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S and let $X \subseteq S$. Then $\mathcal{A}$ and $\mathcal{B}$ have a common transversal containing X if and only if*

(23.8) $$|A_I \cap B_J| \geq |I| + |J| - n + |X \setminus (A_I \cup B_J)|$$

for all $I, J \subseteq \{1, \ldots, n\}$.

Proof. To see necessity, we can assume that there is a common transversal T containing each A_i, each B_j, and X. Then for all $I, J \subseteq \{1, \ldots, n\}$:

(23.9) $$|A_I \cap B_J| = |A_I| + |B_J| - |A_I \cup B_J| \geq |I| + |J| + |X \setminus (A_I \cup B_J)| - n$$

since $|A_I \cup B_J| + |X \setminus (A_I \cup B_J)| \leq |T| = n$.

To see sufficiency, let $X = \{x_1, \ldots, x_k\}$ and let $x'_1, \ldots, x'_k$ be new elements. For each $i = 1, \ldots, n$, let A'_i be the set obtained from A_i by replacing any occurrence of x_j by x'_j. Then $\mathcal{A}$ and $\mathcal{B}$ have a common transversal containing X if the families

(23.10) $$\begin{aligned} &\mathcal{A}' := (A'_1, \ldots, A'_n, \{x_1\}, \ldots, \{x_k\}) \text{ and} \\ &\mathcal{B}' := (B_1, \ldots, B_n, \{x'_1\}, \ldots, \{x'_k\}) \end{aligned}$$

have a common transversal. So by Theorem 23.1 we must check condition (23.3) for $\mathcal{A}'$ and $\mathcal{B}'$. Let $I, J \subseteq \{1, \ldots, n\}$ and $I', J' \subseteq \{1, \ldots, k\}$. Define $Y := \{x_i \mid i \in I'\}$ and $Z := \{x_i \mid i \in J'\}$. Then

(23.11) $$\begin{aligned} &\Big|\Big(\bigcup_{i \in I} A'_i \cup \bigcup_{i \in I'} \{x_i\}\Big) \cap \Big(\bigcup_{j \in J} B_j \cup \bigcup_{j \in J'} \{x'_j\}\Big)\Big| \\ &= |(A_I \cap B_J) \setminus X| + |A_I \cap Z| + |B_J \cap Y| \\ &= |(A_I \cap B_J) \setminus X| + |Z| - |Z \setminus A_I| + |Y| - |Y \setminus B_J| \\ &\geq |(A_I \cap B_J) \setminus X| + |Z| - |X \setminus A_I| + |Y| - |X \setminus B_J| \\ &= |A_I \cap B_J| - |X \setminus (A_I \cup B_J)| + |Y| + |Z| - |X| \\ &\geq |I| + |J| + |Y| + |Z| - |X| - n = |I| + |I'| + |J| + |J'| - n - k \end{aligned}$$

(the last inequality follows from (23.8)). ∎

23.2. Weighted common transversals

Consider the problem of finding a minimum-weight common transversal: given families $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ of subsets of a set S and a weight function $w : S \to \mathbb{Q}$, find a common transversal T of $\mathcal{A}$ and $\mathcal{B}$

minimizing $w(T)$. This problem can easily be solved by solving an associated minimum-cost flow problem.

Alternatively, it can be solved with the Hungarian method, as follows. For $s \in S$, introduce a copy s' of s. Let $S' := \{s' \mid s \in S\}$. Let $a_1, \ldots, a_n, b_1, \ldots, b_n$ be vertices. Make a bipartite graph G with colour classes $\{a_1, \ldots, a_n\} \cup S$ and $\{b_1, \ldots, b_n\} \cup S'$. Vertex a_i is connected with vertex $s' \in S'$ if $s \in A_i$. Vertex b_i is connected with vertex $s \in S$ if $s \in B_i$. Moreover, each $s \in S$ is connected with its copy $s' \in S'$. This describes all edges of G.

For any perfect matching M in G, the set of $s \in S$ with $\{s, s'\} \notin M$ is a common transversal of $\mathcal{A}$ and $\mathcal{B}$. Conversely, each common transversal can be obtained in this way from a perfect matching in G.

Therefore, a minimum-weight common transversal of $\mathcal{A}$ and $\mathcal{B}$ can be found by determining a maximum-weight perfect matching in G, taking weight $w(s)$ on any edge $\{s, s'\}$ and weight 0 on any other edge of G. So by Theorem 17.3 we can find a minimum-weight common transversal in time $O(k(m + k \log k))$, where

$$(23.12) \qquad k := n + |S| \text{ and } m := \sum_{i=1}^{n} (|A_i| + |B_i|).$$

Due to the special structure of G and its weight function one can sharpen this to:

Theorem 23.2. *A minimum-weight common transversal can be found in time $O(n(m + k \log k))$, with m and k as in* (23.12).

Proof. We may assume that $w(s) \geq 0$ for each $s \in S$ (we can add a constant to all weights). Then we can start the Hungarian method with the matching M consisting of all edges $\{s, s'\}$ with $s \in S$. This matching is extreme (that is, has maximum weight among all matchings of size $|M|$), and the Hungarian method requires only n iterations to obtain a maximum-weight perfect matching. ∎

Note that, unlike what happened in finding a minimum-weight transversal for *one* family of sets, in the algorithm above we do not grow a common partial transversal — we do backtrack.

We can also obtain a min-max relation for the minimum weight of a common transversal:

Theorem 23.3. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S and let $w : S \to \mathbb{Z}$ be a weight function. Then the minimum weight of a common transversal of $\mathcal{A}$ and $\mathcal{B}$ is equal to the maximum value of*

$$(23.13) \qquad \sum_{i=1}^{n} (\min_{s \in A_i} w_1(s) + \min_{s \in B_i} w_2(s)) + (w(S) - w_1(S) - w_2(S))$$

taken over $w_1, w_2 \in \mathbb{Z}^S$ *satisfying* $w_1 + w_2 \geq w$.

Proof. Consider the graph G above. By Theorem 17.5 (or by total unimodularity), the maximum weight of a perfect matching in G is equal to the minimum value of

$$(23.14) \qquad \sum_{i=1}^{n}(\lambda_i + \mu_i) + \sum_{s \in S}(w_1(s) + w_2(s))$$

taken over $\lambda, \mu \in \mathbb{Z}^n$ and $w_1, w_2 \in \mathbb{Z}^S$ satisfying

$$(23.15) \qquad \begin{array}{ll} \lambda_i + w_1(s) \geq 0 & \text{for } i = 1, \ldots, n \text{ and } s \in A_i, \\ w_1(s) + w_2(s) \geq w(s) & \text{for } s \in S, \\ \mu_i + w_2(s) \geq 0 & \text{for } i = 1, \ldots, n \text{ and } s \in B_i. \end{array}$$

We can assume that $\lambda_i = \max\{-w_1(s) \mid s \in A_i\}$ and $\mu_i = \max\{-w_2(s) \mid s \in B_i\}$ for each $i = 1, \ldots, n$.

Now the minimum weight of a common transversal is equal to $w(S)$ minus the maximum weight of a perfect matching in G. So it is equal to the maximum value of

$$(23.16) \qquad w(S) - \sum_{s \in S}(w_1(s) + w_2(s)) + \sum_{i=1}^{n}(\min_{s \in A_i} w_1(s) + \min_{s \in B_i} w_2(s)),$$

where $w_1, w_2 \in \mathbb{Z}^S$ satisfy $w_1 + w_2 \geq w$. This is equal to (23.13). ∎

23.3. Weighted common partial transversals

A maximum-weight common partial transversal can be found with the Hungarian method, like described at the beginning of Section 23.2. At any stage of the Hungarian method the current matching M is extreme (that is, it has optimum weight among all matchings of size $|M|$). So we can also apply it (like in Theorem 23.2) to find a maximum-weight common partial transversal of two families $\mathcal{A} = (A_1, \ldots, A_k)$ and $\mathcal{B} = (B_1, \ldots, B_l)$ of subsets of a set S. Taking

$$(23.17) \qquad n := k + l + |S| \text{ and } m := \sum_{i=1}^{k} |A_i| + \sum_{i=1}^{l} |B_i|,$$

we have:

Theorem 23.4. *A maximum-weight common partial transversal can be found in time* $O(\min\{k, l\}(m + n \log n))$.

Proof. As above. ∎

Note that, even if all weights are positive, a maximum-weight common partial transversal need not be a common transversal (a statement that is true if we delete 'common'). To see this, let $\mathcal{A} = (\{a\}, \{b, c\})$, $\mathcal{B} = (\{b\}, \{a, c\})$, and $w(a) = w(b) = 1$, $w(c) = 3$. Then $\{c\}$ is the only maximum-weight common partial transversal, while $\{a, b\}$ is the only common transversal.

A min-max relation for the maximum weight of a common partial transversal can be derived from a min-max relation for the maximum weight of a matching in a bipartite graph, or from linear programming duality using total unimodularity, as we do in the proof below:

Theorem 23.5. *Let $\mathcal{A} = (A_1, \ldots, A_k)$ and $\mathcal{B} = (B_1, \ldots, B_l)$ be families of subsets of a set S and let $w : S \to \mathbb{Z}_+$ be a weight function. Then the maximum weight of a common partial transversal of $\mathcal{A}$ and $\mathcal{B}$ is equal to the minimum value of*

$$\sum_{i=1}^{k} \max_{s \in A_i} w_1(s) + \sum_{i=1}^{l} \max_{s \in B_i} w_2(s) + (w - w_1 - w_2)(S) \tag{23.18}$$

where $w_1, w_2 \in \mathbb{Z}_+^S$ with $w_1 + w_2 \leq w$.

Proof. The maximum weight of a common partial transversal is equal to the maximum of $w^{\mathsf{T}} x$ where $x \in \mathbb{Z}^S$ such that there exist $y_1(i, s) \in \mathbb{Z}_+$ ($i = 1, \ldots, k$; $s \in A_i$) and $y_2(i, s) \in \mathbb{Z}_+$ ($i = 1, \ldots, l$; $s \in B_i$) satisfying

$$\begin{array}{lll}
(23.19) & \displaystyle\sum_{s \in A_i} y_1(i, s) \leq 1 & \text{for } i = 1, \ldots, k, \\
& \displaystyle\sum_{s \in B_i} y_2(i, s) \leq 1 & \text{for } i = 1, \ldots, l, \\
& x_s = \displaystyle\sum_{i, s \in A_i} y_1(i, s) & \text{for } s \in S, \\
& x_s = \displaystyle\sum_{i, s \in B_i} y_2(i, s) & \text{for } s \in S, \\
& 0 \leq x_s \leq 1 & \text{for } s \in S.
\end{array}$$

By linear programming duality and the total unimodularity of the constraint matrix in (23.19), the maximum value is equal to the minimum value of

$$\sum_{i=1}^{k} z_1(i) + \sum_{i=1}^{l} z_2(i) + \sum_{s \in S} u(s), \tag{23.20}$$

where $z_1, z_2 \in \mathbb{Z}_+^k$ and $u \in \mathbb{Z}_+^S$ satisfy

$$\begin{array}{lll}
(23.21) & z_1(i) \geq w_1(s) & \text{for } i = 1, \ldots, k \text{ and } s \in A_i, \\
& z_2(i) \geq w_2(s) & \text{for } i = 1, \ldots, l \text{ and } s \in B_i, \\
& w_1(s) + w_2(s) + u(s) \geq w(s) & \text{for } s \in S,
\end{array}$$

for some $w_1, w_2 \in \mathbb{Z}^E$. We may assume that $w_1, w_2 \geq \mathbf{0}$, since replacing any negative $w_j(s)$ by 0 does not violate (23.21). We may assume that $w_1 + w_2 + u = w$, since $w \geq \mathbf{0}$, and hence we can decrease $w_1(s)$, $w_2(s)$ or $u(s)$ if $w_1(s) + w_2(s) + u(s) > w(s)$. This gives the theorem. ∎

By specializing w to the all-one function, Theorem 23.5 reduces to Corollary 23.1a on the maximum size of a common partial transversal. We can also derive an alternative min-max relation for the maximum weight of a common partial transversal, expressed in

(23.22) $\quad m(\mathcal{C}, w) :=$ maximum weight of a partial transversal of $\mathcal{C}$

for any family $\mathcal{C}$ and weight function w (so we can plug in a min-max relation for $m(\mathcal{C}, w)$ to obtain a genuine min-max relation):

Corollary 23.5a. *Let $\mathcal{A} = (A_1, \ldots, A_k)$ and $\mathcal{B} = (B_1, \ldots, B_l)$ be families of subsets of a set S and let $w : S \to \mathbb{Z}_+$ be a weight function. Then the maximum weight of a common partial transversal of $\mathcal{A}$ and $\mathcal{B}$ is equal to the minimum value of $m(\mathcal{A}, w_1) + m(\mathcal{B}, w_2)$, taken over $w_1, w_2 \in \mathbb{Z}_+^S$ with $w_1 + w_2 = w$.*

Proof. Clearly, the maximum value here cannot be larger than the minimum value, since $w(T) = w_1(T)+w_2(T) \leq m(\mathcal{A}, w_1)+m(\mathcal{B}, w_2)$ for any maximum-weight common partial transversal T.

To see equality, consider w_1 and w_2 of Theorem 23.5, and let $w_2' := w-w_1$. Then for any partial transversal T_1 of $\mathcal{A}$ one has

$$w_1(T_1) \leq \sum_{i=1}^{k} \max_{s \in A_i} w_1(s). \tag{23.23}$$

Moreover, for any partial transversal T_2 of $\mathcal{B}$ one has

$$\begin{aligned} w_2'(T_2) &= w_2(T_2) + (w - w_1 - w_2)(T_2) \\ &\leq \sum_{i=1}^{k} \max_{s \in B_i} w_2(s) + (w - w_1 - w_2)(S). \end{aligned} \tag{23.24}$$

So by Theorem 23.5 we have that $m(\mathcal{A}, w_1) + m(\mathcal{B}, w_2')$ is not more than the maximum w-weight of a common partial transversal. ∎

The obvious generalization to common partial transversals of *three* families is not true: take

(23.25) $\quad \mathcal{A} = (\{a\}, \{b, c\})$, $\mathcal{B} = (\{b\}, \{a, c\})$, and $\mathcal{C} = (\{c\}, \{a, b\})$,

and $w(a) = w(b) = w(c) = 1$. Then the maximum weight of a common partial transversal is 1, but one cannot decompose w as $w = w_1 + w_2 + w_3$ with $m(\mathcal{A}, w_1) + m(\mathcal{B}, w_2) + m(\mathcal{C}, w_3) = 1$.

23.4. The common partial transversal polytope

Let $\mathcal{A} = (A_1, \ldots, A_k)$ and $\mathcal{B} = (B_1, \ldots, B_l)$ be families of subsets of a set S. The *common partial transversal polytope* $P_{\text{common partial transversal}}(\mathcal{A}, \mathcal{B})$ of

$\mathcal{A}$ and $\mathcal{B}$ is the convex hull of the incidence vectors (in $\mathbb{R}^S$) of the common partial transversals of $\mathcal{A}$ and $\mathcal{B}$. That is,

(23.26) $P_{\text{common partial transversal}}(\mathcal{A},\mathcal{B}) = \text{conv.hull}\{\chi^T \mid T$ is a common partial transversal of $\mathcal{A}$ and $\mathcal{B}\}$.

It is easy to see that each vector x in the common partial transversal polytope satisfies:

(23.27)
$$\begin{array}{rll} \text{(i)} & 0 \le x_s \le 1 & \text{for } s \in S, \\ \text{(ii)} & x(S \setminus A_I) \le k - |I| & \text{for } I \subseteq \{1,\ldots,k\}, \\ \text{(iii)} & x(S \setminus B_I) \le l - |I| & \text{for } I \subseteq \{1,\ldots,l\}. \end{array}$$

In fact, this fully determines the common partial transversal polytope:

Theorem 23.6. *The common partial transversal polytope is determined by* (23.27).

Proof. We must show that for any weight function $w \in \mathbb{Z}_+^S$, the maximum value of $w^\mathsf{T} x$ over (23.27) is equal to the maximum weight μ of any common partial transversal. By Corollary 23.5a, there exist weight functions $w_1, w_2 \in \mathbb{Z}^S$ with $w = w_1 + w_2$ and $\mu = m(\mathcal{A}, w_1) + m(\mathcal{B}, w_2)$. Now any x satisfying (23.27) belongs to the partial transversal polytopes of $\mathcal{A}$ and $\mathcal{B}$. So $w_1^\mathsf{T} x \le m(\mathcal{A}, w_1)$ and $w_2^\mathsf{T} x \le m(\mathcal{B}, w_2)$. Hence $w^\mathsf{T} x \le \mu$. ∎

Since (23.27) is the union of the systems that determine the partial transversal polytope of $\mathcal{A}$ and of $\mathcal{B}$, we have:

Corollary 23.6a. *Let $\mathcal{A}$ and $\mathcal{B}$ be families of subsets of a set S. Then*

(23.28)
$$\begin{aligned} &P_{\text{common partial transversal}}(\mathcal{A},\mathcal{B}) \\ &= P_{\text{partial transversal}}(\mathcal{A}) \cap P_{\text{partial transversal}}(\mathcal{B}). \end{aligned}$$

Proof. Directly from Theorem 23.6 and Corollary 22.9a. ∎

Also:

Theorem 23.7. *System* (23.27) *is TDI.*

Proof. Directly from Corollaries 23.5a and 22.9a. ∎

Again one cannot make the obvious extension to three families of sets, by considering the families (23.25). In that case, the vector $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ belongs to the intersection of the three partial transversal polytopes, but does not belong to the common partial transversal polytope.

23.5. The common transversal polytope

Similar results hold for the common transversal polytope. Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S. The *common transversal polytope* $P_{\text{common transversal}}(\mathcal{A})$ of $\mathcal{A}$ and $\mathcal{B}$ is the convex hull of the incidence vectors (in $\mathbb{R}^S$) of the common transversals of $\mathcal{A}$ and $\mathcal{B}$. That is,

(23.29) $P_{\text{common transversal}}(\mathcal{A}, \mathcal{B}) = \text{conv.hull}\{\chi^T \mid T$ is a common transversal of $\mathcal{A}$ and $\mathcal{B}\}$.

It is easy to see that each vector x in the common transversal polytope satisfies:

(23.30)
(i) $0 \leq x_s \leq 1$ for $s \in S$,
(ii) $x(A_I) \geq |I|$ for $I \subseteq \{1, \ldots, n\}$,
(iii) $x(B_I) \geq |I|$ for $I \subseteq \{1, \ldots, n\}$,
(iv) $x(S) = n$.

Corollary 23.7a. *The common transversal polytope is determined by* (23.30).

Proof. The common transversal polytope is the facet of the common partial transversal polytope determined by the equality $x(S) = n$. So we must show that (23.30) implies (23.27), which is trivial, since if x satisfies (23.30), then $x(S \setminus A_I) = x(S) - x(A_I) \leq n - |I|$ and $x(S \setminus B_I) = x(S) - x(B_I) \leq n - |I|$ for any $I \subseteq \{1, \ldots, n\}$. ∎

Again this implies:

Corollary 23.7b. *Let $\mathcal{A}$ and $\mathcal{B}$ be families of subsets of a set S. Then*

(23.31) $P_{\text{common transversal}}(\mathcal{A}, \mathcal{B}) = P_{\text{transversal}}(\mathcal{A}) \cap P_{\text{transversal}}(\mathcal{B})$.

Proof. Directly from Corollaries 23.7a and 22.9b. ∎

In fact:

Theorem 23.8. *System* (23.30) *is TDI.*

Proof. This follows from Theorem 23.7, using Theorem 5.25. ∎

Weinberger [1976] proved the following conjecture of Fulkerson [1971a], which generalizes Theorem 18.8. Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S. Then the up hull $P^{\uparrow}_{\text{common transversal}}(\mathcal{A}, \mathcal{B})$ of the common transversal polytope is determined by:

(23.32) $x(U) \geq n-$ maximum size of a common partial transversal contained in $S \setminus U$, for $U \subseteq S$. This will follow from Theorem 46.3 on polymatroids.

23.6. Packing and covering of common transversals

Fulkerson [1971b] and de Sousa [1971] detected that results on bipartite edge-colouring (or related results) imply characterizations of packings of common transversals. It was noticed by Brualdi [1971b] that the methods in fact yield more general results.

Basic is the following exchange property given by de Sousa [1971]:

Theorem 23.9. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_m)$ be families of subsets of a set S and let $k \in \mathbb{Z}_+$. Suppose that S can be covered by k partial transversals of $\mathcal{A}$ and that S can also be covered by k partial transversals of $\mathcal{B}$. Then S can be covered by k common partial transversals of $\mathcal{A}$ and $\mathcal{B}$.*

Proof. Let $T_1, \ldots, T_k$ be a partition of S into k partial transversals of $\mathcal{A}$. Since each T_i is a partial transversal of $\mathcal{A}$, it follows that each A_i has a subset A'_i such that $|A'_i| \leq k$ and such that $A'_1, \ldots, A'_n$ partition S. We can assume that $A'_i = A_i$ for each i, and hence that $\mathcal{A}$ is a partition of S into classes of size at most k.

Similarly, we can assume that $\mathcal{B}$ is a partition of S into classes of size at most k.

Now make a bipartite graph G, with colour classes $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_m\}$, connecting a_i and b_j by $|A_i \cap B_j|$ parallel edges. So G has maximum degree k, and hence, by Kőnig's edge-colouring theorem, the edges of G can be coloured with k colours. It implies that S can be partitioned as required. ∎

A consequence is a min-max formula for the minimum number of common partial transversals needed to cover S, stated by Brualdi [1971b]:

Corollary 23.9a. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S, each with union S. Then the minimum number of common partial transversals of $\mathcal{A}$ and $\mathcal{B}$ needed to cover S is equal to*

$$\left\lceil \max_{\substack{X \subseteq S \\ X \neq \emptyset}} \max\left\{\frac{|X|}{|\{i \mid A_i \cap X \neq \emptyset\}|}, \frac{|X|}{|\{i \mid B_i \cap X \neq \emptyset\}|}\right\}\right\rceil. \tag{23.33}$$

Proof. From Theorem 23.9, using Theorem 22.12. ∎

Theorem 23.9 also gives a variant of the exchange property (de Sousa [1971]):

Corollary 23.9b. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S and let $k \in \mathbb{Z}_+$. Suppose that S can be partitioned into k*

transversals of $\mathcal{A}$, and also can be partitioned into k transversals of $\mathcal{B}$. Then S can be partitioned into k common transversals of $\mathcal{A}$ and $\mathcal{B}$.

Proof. Directly from Theorem 23.9, since $|S| = nk$. ∎

This implies another variant (de Sousa [1971]):

Corollary 23.9c. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S and let $k \in \mathbb{Z}_+$. Suppose that S has a partition $(S_1, \ldots, S_n)$ with $|S_i| = k$ and $S_i \subseteq A_i$ for $i = 1, \ldots, n$. Suppose moreover that S has a partition $(Z_1, \ldots, Z_n)$ with $|Z_i| = k$ and $Z_i \subseteq B_i$ for $i = 1, \ldots, n$. Then S can be partitioned into common transversals of $\mathcal{A}$ and $\mathcal{B}$.*

Proof. Note that if S has a partition $(S_1, \ldots, S_n)$ with $|S_i| = k$ and $S_i \subseteq A_i$ for $i = 1, \ldots, n$, then S can be partitioned into k transversals of $\mathcal{A}$. Similarly for $\mathcal{B}$. So the present corollary follows from Corollary 23.9b. ∎

This gives the following basic min-max relation for the maximum number of disjoint common transversals, given by Fulkerson [1971b] and de Sousa [1971]:

Corollary 23.9d. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of sets and let k be a natural number. Then $\mathcal{A}$ and $\mathcal{B}$ have k disjoint common transversals if and only if*

(23.34) $$|A_I \cap B_J| \geq k(|I| + |J| - n).$$

for all $I, J \subseteq \{1, \ldots, n\}$.

Proof. Necessity of (23.34) being easy, we show sufficiency.

Let $\mathcal{A}'$ arise by taking k copies of $\mathcal{A}$ and let $\mathcal{B}'$ arise from taking k copies of $\mathcal{B}$. Condition (23.34) implies that $\mathcal{A}'$ and $\mathcal{B}'$ have a common transversal, S say (by Theorem 23.1). Then we can partition S into subsets $A'_1, \ldots, A'_n$, with $A'_i \subseteq A_i$ and $|A'_i| = k$. Similarly, we can partition S into subsets $B'_1, \ldots, B'_n$, with $B'_i \subseteq B_i$ and $|B'_i| = k$. Then by Corollary 23.9c, S has a partition into k common transversals of $\mathcal{A}$ and $\mathcal{B}$. ∎

(Note that if $\mathcal{A}$ and $\mathcal{B}$ are partitions of a set, this corollary reduces to Corollary 20.9a.)

The following open problem, dealing with packing common transversals, was mentioned by Fulkerson [1971b]: Let $\mathcal{A}$ and $\mathcal{B}$ be families of subsets of a set S and let $c \in \mathbb{Z}_+^S$. What is the maximum number k of common transversals $T_1, \ldots, T_k$ such that

(23.35) $$\chi^{T_1} + \cdots + \chi^{T_k} \leq c?$$

More generally than Corollary 23.9d, one has for disjoint common *partial* transversals of prescribed size (Fulkerson [1971b]):

Theorem 23.10. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_m)$ be families of sets and let $k, p \in \mathbb{Z}_+$. Then there exist k disjoint common partial transversals of size p if and only if*

$$(23.36) \qquad |A_I \cap B_J| \geq k(|I| + |J| + p - n - m)$$

for all $I \subseteq \{1, \ldots, n\}$ and $J \subseteq \{1, \ldots, m\}$.

Proof. Construct $\mathcal{A}'$ and $\mathcal{B}'$ as in Corollary 23.9d. By Corollary 23.1a, (23.36) implies that $\mathcal{A}'$ and $\mathcal{B}'$ have a common partial transversal, T say, of size pk. Then each A_i has a subset A'_i such that $|A'_i| \leq k$ and such that $A'_1, \ldots, A'_n$ partition T. We can assume that $A'_i = A_i$ for each i, and hence that $\mathcal{A}$ is a partition of T into classes of size at most k.

Similarly, we can assume that $\mathcal{B}$ is a partition of T into classes of size at most k.

Now make a bipartite graph G, with colour classes $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_m\}$, connecting a_i and b_j by $|A_i \cap B_j|$ parallel edges. So G has kp edges and maximum degree k, and hence, by Theorem 20.8, the edges of G can be coloured with k colours, each of size p. It implies that T can be partitioned into common partial transversals of $\mathcal{A}$ and $\mathcal{B}$ of size p. ∎

Similarly to Theorem 23.9 one can prove the following exchange property:

Theorem 23.11. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S and let $k \in \mathbb{Z}_+$. Suppose that $\mathcal{A}$ has k disjoint transversals and that also $\mathcal{B}$ has k disjoint transversals. Then S has k disjoint subsets $S_1, \ldots, S_k$ such that each S_i contains a transversal of $\mathcal{A}$ and contains a transversal of $\mathcal{B}$.*

Proof. As $\mathcal{A}$ has k disjoint transversals, there exist disjoint sets $A'_1, \ldots, A'_n$ with $A'_i \subseteq A_i$ and $|A'_i| = k$ for $i = 1, \ldots, k$. For our purposes, we can assume that $A'_i = A_i$. Let Y be the union of the A_i. Similarly, we can assume that $B_1, \ldots, B_n$ have size k each and partition some set Z.

Again, make a bipartite graph G, with colour classes $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$, connecting a_i and b_j by $|A_i \cap B_j|$ parallel edges. Then G has maximum degree at most k, and hence, by Kőnig's edge-colouring theorem (Theorem 20.1), G is k-edge-colourable. It gives a partition of $Y \cap Z$ into k classes each intersecting any A_i and B_i in at most one element. We can extend this partition to a partition of $Y \cup Z$ into classes each intersecting any A_i and any B_i in exactly one element. This is a partition as required. ∎

This implies another min-max relation:

Corollary 23.11a. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S. Then the maximum number k for which there exist disjoint subsets $S_1, \ldots, S_k$ each containing a transversal of $\mathcal{A}$ and a transversal of $\mathcal{B}$ is equal to*

$$\lfloor \min_{\emptyset \neq I \subseteq \{1,\ldots,n\}} \min\{\tfrac{|A_I|}{|I|}, \tfrac{|B_I|}{|I|}\}\rfloor. \tag{23.37}$$

Proof. Directly from Theorems 22.10 and 23.11. ∎

An analogue of Corollary 23.9d for covering by common transversals is:

Theorem 23.12. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of S and let $X \subseteq S$. Then X can be covered by k common transversals if and only if*

$$k|A_I \cap B_J| \geq k(|I| + |J| - n) + |X \setminus (A_I \cup B_J)| \tag{23.38}$$

for all $I, J \subseteq \{1, \ldots, n\}$.

Proof. To see necessity, let $T_1, \ldots, T_k$ be common transversals covering X and let $I, J \subseteq \{1, \ldots, n\}$. Then

$$\begin{aligned} k|A_I \cap B_J| &\geq \sum_{j=1}^{k} |A_I \cap B_J \cap T_j| \\ &= \sum_{j=1}^{k} (|A_I \cap T_j| + |B_J \cap T_j| - |T_j \cap (A_I \cup B_J)|) \\ &\geq \sum_{j=1}^{k} (|I| + |J| - |T_j \cap (A_I \cup B_J)|) \\ &= k(|I| + |J| - n) + \sum_{j=1}^{k} |T_j \setminus (A_I \cup B_J)| \\ &\geq k(|I| + |J| - n) + |X \setminus (A_I \cup B_J)|. \end{aligned} \tag{23.39}$$

To see sufficiency, make a directed graph D, with vertex set

$$\{r\} \cup \{a_1, \ldots, a_n\} \cup S \cup S' \cup \{b_1, \ldots, b_n\}, \tag{23.40}$$

where S' is a set consisting of, for each $s \in S$, a (new) copy s' of s, and with arcs, with demands and capacities, as follows:

(23.41) (r, a_i) with demand k and capacity k, for $i = 1, \ldots, n$,
(a_i, s) with demand 0 and capacity ∞ for $i = 1, \ldots, n$ and $s \in A_i$,
(s, s') with demand 1 (if $s \in X$) or 0 (if $s \notin X$) and capacity k, for $s \in S$,
(s', b_i) with demand 0 and capacity ∞, for $i = 1, \ldots, n$ and $s \in B_i$,
(b_i, r) with demand k and capacity k, for $i = 1, \ldots, n$.

Then by Hoffman's circulation theorem (Theorem 11.2), (23.38) implies the existence of a circulation f obeying the demands and capacities. Indeed, consider any set U of vertices of D. Let $I := \{i \mid a_i \in U\}$, $J := \{j \mid b_j \notin U\}$, $Y := U \cap S$ and $Z := \{s \in S \mid s' \notin U\}$. We can assume that the capacity of the arcs leaving U is finite, and hence, if $i \in I$, then $A_i \subseteq Y$ and if $j \in J$, then $B_j \subseteq Z$. That is, $A_I \subseteq Y$ and $B_J \subseteq Z$.

If $r \notin U$, then the total demand of the arcs entering U is equal to

(23.42) $\quad k|I| + |X \setminus (Y \cup Z)|$

and the total capacity of the arcs leaving U is equal to

(23.43) $\quad k|Y \cap Z| + k(n - |J|).$

Since $A_I \subseteq Y$ and $B_J \subseteq Z$, (23.38) implies that (23.42) is at most (23.43).

If $r \in U$, then the total demand of the arcs entering U is equal to

(23.44) $\quad k|J| + |X \setminus (Y \cup Z)|$

and the total capacity of the arcs leaving U is equal to

(23.45) $\quad k|Y \cap Z| + k(n - |I|).$

Since $A_I \subseteq Y$ and $B_J \subseteq Z$, (23.38) implies that (23.44) is at most (23.45).

So Hoffman's condition is satisfied, and hence there exists a circulation f.

Now f is at most k on any arc. Hence, by Corollary 11.2b, f is the sum of k $\{0,1\}$-valued circulations $f_1, \ldots, f_k$. For each circulation f_i, the set T_i of $s \in S$ with $f_i(s, s') = 1$ is a common transversal of $\mathcal{A}$ and $\mathcal{B}$. Moreover, since $f(s, s') \geq 1$ for each $s \in X$, these common transversals cover X. ∎

A covering theorem different from Theorem 23.12 is due to Brualdi [1971b]:

Theorem 23.13. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_m)$ be families of subsets of a set S. Suppose that S can be covered by k common partial transversals of $\mathcal{A}$ and $\mathcal{B}$. Then S can be covered by k common partial transversals each of size $\lfloor |S|/k \rfloor$ or $\lceil |S|/k \rceil$.*

Proof. The assumption implies that each A_i contains a subset A'_i with $|A'_i| \leq k$, such that the A'_i partition S. For our purposes, we can assume that $A'_i = A_i$ for each i. Similarly, we can assume that $\mathcal{B}$ is a partition of S into classes of size at most k.

Again, make a bipartite graph G, with colour classes $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_m\}$, connecting a_i and b_j by $|A_i \cap B_j|$ parallel edges. Then G has maximum degree at most k, and hence, by Theorem 20.8, G is k-edge-colourable, where each colour has size $\lfloor |S|/k \rfloor$ or $\lceil |S|/k \rceil$. This yields a partition of S into k common partial transversals as required. ∎

23.7. Further results and notes

23.7a. Capacitated common transversals

Recall the definition of system of restricted representatives: Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a collection of subsets of a set S and let $a, b \in \mathbb{Z}_+^S$ with $a \leq b$. A *system of restricted representatives* (or *SRR*) of $\mathcal{A}$ (with respect to a and b) is a sequence $(s_1, \ldots, s_n)$ such that

(23.46) (i) $s_i \in A_i$ for $i = 1, \ldots, n$;
(ii) $a(s) \leq |\{i \mid s_i = s\}| \leq b(s)$ for $s \in S$.

Ford and Fulkerson [1958c] derived the following characterization of the existence of a common system of restricted representatives from the max-flow min-cut theorem (we give the derivation from Corollary 23.1b based on splitting elements, due to Mirsky and Perfect [1968]):

Theorem 23.14. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be families of subsets of a set S and let $a, b \in \mathbb{Z}_+^S$ with $a \leq b$. Then $\mathcal{A}$ and $\mathcal{B}$ have a common system of restricted representatives if and only if*

(23.47) $$b(A_I \cap B_J) \geq |I| + |J| - n + a(S \setminus (A_I \cup B_J))$$

for all $I, J \subseteq \{1, \ldots, n\}$.

Proof. Let for any $s \in S$, Z_s be a set of $b(s)$ (new) elements. Replace in each A_i and B_j, any occurrence of any $s \in S$ by the elements of Z_s. Choose from each Z_s, $a(s)$ elements, forming the set X. Then $\mathcal{A}$ and $\mathcal{B}$ have a common system of restricted representatives if and only if the new families have a common transversal containing X. Trivially, condition (23.47) is equivalent to condition (23.8) for the new families, and hence the theorem follows from Corollary 23.1b. ∎

(More can be found in Mirsky [1968b].)

23.7b. Exchange properties

Mirsky [1968a] showed the following exchange property of common transversals:

Theorem 23.15. *Let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_m)$ be families of sets. Let $I', I'' \subseteq \{1, \ldots, n\}$ and $J', J'' \subseteq \{1, \ldots, m\}$. Suppose that $(A_i \mid i \in I')$ and $(B_j \mid j \in J')$ have a common transversal, and also that $(A_i \mid i \in I'')$ and $(B_j \mid j \in J'')$ have a common transversal. Then there exist I and J with $I' \subseteq I \subseteq I' \cup I''$ and $J'' \subseteq J \subseteq J' \cup J''$ such that $(A_i \mid i \in I)$ and $(B_j \mid j \in J)$ have a common transversal.*

Proof. Directly from Corollary 9.12a applied to the digraph defined in (23.1). ∎

This implies (Mirsky [1968a]):

Corollary 23.15a. *Let $\mathcal{A}$ and $\mathcal{B}$ be families of sets and let $\mathcal{A}'$ and $\mathcal{B}'$ be subfamilies of $\mathcal{A}$ and $\mathcal{B}$ respectively. Then there exist subfamilies $\mathcal{A}_0$ and $\mathcal{B}_0$ of $\mathcal{A}$ and $\mathcal{B}$*

respectively satisfying $\mathcal{A}' \subseteq \mathcal{A}_0$ *and* $\mathcal{B}' \subseteq \mathcal{B}_0$ *and having a common transversal if and only if* (i) $\mathcal{A}'$ *and some subfamily of* $\mathcal{B}$ *have a common transversal and* (ii) $\mathcal{B}'$ *and some subfamily of* $\mathcal{A}$ *have a common transversal.*

Proof. Directly from Theorem 23.15. ∎

23.7c. Common transversals of three families

It is NP-complete to test if *three* families of sets have a common transversal, even if each of the three families is a partition of S (E.L. Lawler — cf. Karp [1972b]).

Theorem 23.16. *Testing if three partitions have a common transversal is* NP-*complete.*

Proof. I. It suffices to show the NP-completeness of the following problem:

(23.48) given disjoint sets X, Y, Z with $|X| = |Y| \geq |Z|$ and a collection $\mathcal{C}$ of subsets U of $W := X \cup Y \cup Z$ with $|U \cap X| = |U \cap Y| = 1$ and $|U \cap Z| \leq 1$, decide if $\mathcal{C}$ contains a partition of W as subcollection.

To see this, first observe that we can assume that $|X| = |Y| = |Z|$. Indeed, we can extend Z by a set R of size $|X| - |Z|$ and replace each doubleton $\{x, y\}$ in $\mathcal{C}$ by all sets $\{x, y, w\}$ with $w \in R$. Then the new collection contains a partition if and only if the original collection contains one.

So we can assume that $|X| = |Y| = |Z|$. For $w \in W$, define $\mathcal{C}_w := \{C \in \mathcal{C} \mid w \in C\}$. Then the collection $\{\mathcal{C}_w \mid w \in W\}$ is the union of three partitions of $\mathcal{C}$. Moreover, these three partitions have a common transversal if and only if $\mathcal{C}$ contains a partition of W. So this reduces problem (23.48) to the problem of finding a common transversal of three partitions of a set.

II. So it suffices to show the NP-completeness of (23.48). We derive this from the NP-completeness of the (more general) *partition problem*: decide if a given collection $\mathcal{B}$ of subsets of a set Z contains a partition of Z as a subcollection (Corollary 4.1b).

Let $V := \{(B, z) \mid z \in B \in \mathcal{B}\}$. Make, for each $B \in \mathcal{B}$, an (arbitrary) directed circuit on $\{(B, z) \mid z \in B\}$. This makes the directed graph D on V (consisting of vertex-disjoint directed circuits). Define $X := V \times \{1\}$ and $Y := V \times \{2\}$. Let $\mathcal{C}$ be the collection of

(23.49) all triples $\{(B, z, 1), (B, z, 2), z\}$ for all $B \in \mathcal{B}$ and $z \in B$, and all pairs $\{(B, z, 1), (B, z', 2)\}$, for all $B \in \mathcal{B}$ and $z, z' \in B$ such that D contains an arc from z to z'.

So each element of $X \cup Y$ is in precisely two sets in $\mathcal{C}$: a triple and a pair. Any partition $\mathcal{P} \subseteq \mathcal{C}$ of $X \cup Y \cup Z$ contains, for any $B \in \mathcal{B}$, either all triples containing B or all pairs containing B. (Here containing B means: containing (B, z, i) for some z, i.)

This implies that $\mathcal{C}$ contains a partition of $X \cup Y \cup Z$ if and only if $\mathcal{B}$ contains a partition of Z. ∎

As indicated in this proof, the problem of finding a common transversal of three partitions is equivalent to the 3-*dimensional matching problem*: given a partition

U, V, W of a finite set S and a collection $\mathcal{C}$ of subsets X of S satisfying $|X \cap U| = |X \cap V| = |X \cap W| = 1$, does $\mathcal{C}$ have a subcollection that partitions S?

The following *necessary* condition for the existence of a common transversal of three families $\mathcal{A} = (A_1, \ldots, A_n)$, $\mathcal{B} = (B_1, \ldots, B_n)$, and $\mathcal{C} = (C_1, \ldots, C_n)$ of sets is not sufficient: for all $I, J, K \subseteq \{1, \ldots, n\}$

$$|A_I \cap B_J \cap C_K| \geq |I| + |J| + |K| - 2n. \tag{23.50}$$

(This would generalize condition (23.3).) To see this, consider $\mathcal{A} = (\{a\}, \{b, c\})$, $\mathcal{B} = (\{b\}, \{a, c\})$, $\mathcal{C} = (\{c\}, \{a, b\})$.

More on common transversals of more than two families is given by Brown [1976,1984], Dacić [1977,1979], Longyear [1977], and Zaverdinos [1981]. Woodall [1982] studied fractional transversals, and described a good characterization for the existence of a common fractional transversal for more than two families, based on linear programming.

23.7d. Further notes

Weinberger [1974b] observed that if the families $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ of subsets of a set S are uniform (that is, all sets have the same size) and regular (that is, each $s \in S$ is in the same number of sets), then $\mathcal{A}$ and $\mathcal{B}$ have a common transversal.

Further work on common transversals (including extensions to the infinite case) is reported by Perfect [1969b], Brualdi [1970b,1971a], and Davies and McDiarmid [1976].

Part III

Nonbipartite Matching and Covering

Part III: Nonbipartite Matching and Covering

Nonbipartite matching is a highlight of combinatorial optimization, thanks to pioneering work of Tutte and Edmonds. In particular the 1965 papers of Edmonds on nonbipartite matching opened up areas that were not accessible with the 'classical' methods based on flows, linear programming, and total unimodularity found in the 1950s. The papers are pioneering in polyhedral combinatorics, giving the first nontrivial characterizations of combinatorially defined polytopes.
The techniques are highly self-refining, and extend to b-matchings, b-factors, T-joins, shortest paths in undirected graphs, and the Chinese postman problem. Nonbipartite matching also applies to practical problems where an optimal pairing has to be found, like in seat or room assignment, crew planning, and two-processor scheduling.

Chapters:

Chapter 24

Cardinality nonbipartite matching

In this chapter we consider maximum-cardinality matching, with as key results Tutte's characterization of the existence of a perfect matching (implying the Tutte-Berge formula for the maximum-size of a matching) and Edmonds' polynomial-time algorithm to find a maximum-size matching.
As in Section 16.1, we call a path P an *M-augmenting path* if P has odd length and connects two vertices not covered by M, and its edges are alternatingly out of and in M. By Theorem 16.1, a matching M has maximum size if and only if there is no M-augmenting path.
We say that a matching M *covers* a vertex v if v is incident with an edge in M. If M does not cover v, we say that M *misses* v.
In this chapter, graphs can be assumed to be simple.

24.1. Tutte's 1-factor theorem and the Tutte-Berge formula

A basic result of Tutte [1947b] characterizes graphs that have a perfect matching. Berge [1958a] observed that it implies a min-max formula for the maximum size of a matching in a graph, the Tutte-Berge formula.

Call a component of a graph *odd* if it has an odd number of vertices. For any graph G, let

(24.1) $\quad o(G) :=$ number of odd components of G.

Let $\nu(G)$ denotes the maximum size of a matching. Then:

Theorem 24.1 (Tutte-Berge formula)**.** *For each graph $G = (V, E)$,*

(24.2) $$\nu(G) = \min_{U \subseteq V} \tfrac{1}{2}(|V| + |U| - o(G - U)).$$

Proof. To see $\leq$, we have for each $U \subseteq V$:

(24.3) $$\nu(G) \leq |U| + \nu(G - U) \leq |U| + \tfrac{1}{2}(|V \setminus U| - o(G - U)) = \tfrac{1}{2}(|V| + |U| - o(G - U)).$$

We prove the reverse inequality by induction on $|V|$, the case $V = \emptyset$ being trivial. We can assume that G is connected, since otherwise we can apply induction to the components of G.

First assume that there exists a vertex v covered by all maximum-size matchings. Then $\nu(G-v) = \nu(G)-1$, and by induction there exists a subset U' of $V \setminus \{v\}$ with

$$(24.4) \qquad \nu(G-v) = \tfrac{1}{2}(|V \setminus \{v\}| + |U'| - o(G-v-U')).$$

Then $U := U' \cup \{v\}$ gives equality in (24.2), since

$$(24.5) \qquad \begin{aligned}\nu(G) &= \nu(G-v)+1 = \tfrac{1}{2}(|V \setminus \{v\}| + |U'| - o(G-v-U')) + 1 \\ &= \tfrac{1}{2}(|V| + |U| - o(G-U)).\end{aligned}$$

So we can assume that there is no such v. In particular, $\nu(G) < \frac{1}{2}|V|$. We show that there exists a matching of size $\frac{1}{2}(|V|-1)$, which implies the theorem (taking $U := \emptyset$).

Indeed, suppose to the contrary that each maximum-size matching M misses at least two distinct vertices u and v. Among all such M, u, v, choose them such that the distance $\mathrm{dist}(u,v)$ of u and v in G is as small as possible.

If $\mathrm{dist}(u,v) = 1$, then u and v are adjacent, and hence we can augment M by the edge uv, contradicting the maximality of $|M|$. So $\mathrm{dist}(u,v) \geq 2$, and hence we can choose an intermediate vertex t on a shortest $u-v$ path. By assumption, there exists a maximum-size matching N missing t. Choose such an N with $|M \cap N|$ maximal.

By the minimality of $\mathrm{dist}(u,v)$, N covers both u and v. Hence, as M and N cover the same number of vertices, there exists a vertex $x \neq t$ covered by M but not by N. Let $x \in e = xy \in M$. Then y is covered by some edge $f \in N$, since otherwise $N \cup \{e\}$ would be a matching larger than N. Replacing N by $(N \setminus \{f\}) \cup \{e\}$ would increase its intersection with M, contradicting the choice of N. ∎

(This proof is based on the proof of Lovász [1979b] of Edmonds' matching polytope theorem.)

The Tutte-Berge formula immediately implies Tutte's 1-factor theorem. A *perfect matching* (or 1-*factor*) is a matching covering all vertices of the graph.

Corollary 24.1a (Tutte's 1-factor theorem)**.** *A graph $G = (V,E)$ has a perfect matching if and only if $G-U$ has at most $|U|$ odd components, for each $U \subseteq V$.*

Proof. Directly from the Tutte-Berge formula (Theorem 24.1), since G has a perfect matching if and only if $\nu(G) \geq \frac{1}{2}|V|$. ∎

24.1a. Tutte's proof of his 1-factor theorem

The original proof of Tutte [1947b] of his 1-factor theorem (Corollary 24.1a), with a simplification of Maunsell [1952], and smoothed by Halton [1966] and Lovász [1975d], is as follows.

Suppose that there exist graphs $G = (V, E)$ satisfying the condition, but not having a perfect matching. Fixing V, take such a graph G with G simple and $|E|$ as large as possible. Let $U := \{v \in V \mid v \text{ is adjacent to every other vertex of } G\}$. We show that each component of $G - U$ is a complete graph.

Suppose to the contrary that there are distinct $a, b, c \notin U$ with $ab, bc \in E$ and $ac \notin E$. By the maximality of $|E|$, adding ac to E makes that G has a perfect matching (since the condition is maintained under adding edges). So G has a matching M missing precisely a and c. As $b \notin U$, there exists a vertex d with $bd \notin E$. Again by the maximality of $|E|$, G has a matching N missing precisely b and d. Now each component of $M \triangle N$ contains the same number of edges in M as in N — otherwise there would exist an M- or N-augmenting path, and hence a perfect matching in G, a contradiction. So the component P of $M \triangle N$ containing d is a path starting at d, with first edge in M and last edge in N, and hence ending at a or c; by symmetry we may assume that it ends at a. Moreover, P does not traverse b. Then extending P by the edge ab gives an N-augmenting path, and hence a perfect matching in G — a contradiction.

So each component of $G - U$ is a complete graph. Moreover, by the condition, $G - U$ has at most $|U|$ odd components. This implies that G has a perfect matching, contradicting our assumption.

More proofs were given by Gallai [1950,1963b], Edmonds [1965d], Balinski [1970], Anderson [1971], Brualdi [1971d], Hetyei [1972,1999], Mader [1973], and Lovász [1975a,1979b].

24.1b. Petersen's theorem

The following theorem of Petersen [1891] is a consequence of Tutte's 1-factor theorem (a graph is *cubic* if it is 3-regular):

Corollary 24.1b (Petersen's theorem). *A bridgeless cubic graph has a perfect matching.*

Proof. Let $G = (V, E)$ be a bridgeless cubic graph. By Tutte's 1-factor theorem, we should show that $G - U$ has at most $|U|$ odd components, for each $U \subseteq V$.

Each odd component of $G - U$ is left by an odd number of edges (as G is cubic), and hence by at least three edges (as G is bridgeless). On the other hand, U is left by at most $3|U|$ edges, since G is cubic. Hence $G - U$ has at most $|U|$ odd components. ∎

24.2. Cardinality matching algorithm

The idea of finding an M-augmenting path to increase a matching M is fundamental in finding a maximum-size matching. However, the simple trick for bipartite graphs, of orienting the edges based on the colour classes of the graph, does not extend to the nonbipartite case. Yet one could try to find an M-augmenting path by finding an 'M-alternating walk', but such a walk can run into a loop that cannot simply be deleted. It was Edmonds [1965d] who found the trick to resolve this problem, namely by 'shrinking' the loop (for which he introduced the term 'blossom'). Then applying recursion to a smaller graph solves the problem[1].

Let $G = (V, E)$ be a graph, let M be a matching in G, and let X be the set of vertices missed by M. A walk $P = (v_0, v_1, \ldots, v_t)$ is called *M-alternating* if for each $i = 1, \ldots, t-1$ exactly one of the edges $v_{i-1}v_i$ and v_iv_{i+1} belongs to M. Note that one can find a shortest M-alternating $X - X$ walk of positive length, by considering the auxiliary directed graph $D = (V, A)$ with

(24.6) $$A := \{(u, v) \mid \exists x \in V : ux \in E, xv \in M\}.$$

Then each M-alternating $X - X$ walk of positive length yields a directed $X - N(X)$ path in D, and vice versa (where $N(X)$ denotes the set of neighbours of X).

An M-alternating walk $P = (v_0, v_1, \ldots, v_t)$ is called an *M-flower* if t is odd, $v_0, \ldots, v_{t-1}$ are distinct, $v_0 \in X$, and $v_t = v_i$ for some even $i < t$. Then the circuit $(v_i, v_{i+1}, \ldots, v_t)$ is called an *M-blossom* (associated with the M-flower).

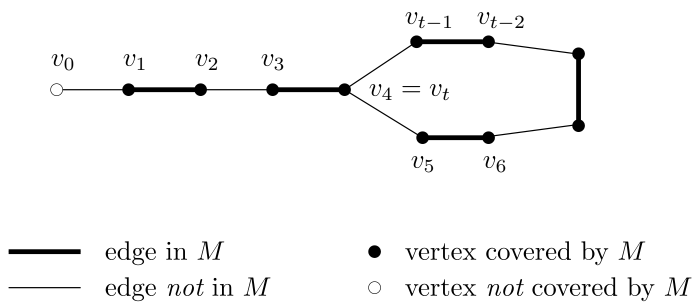


Figure 24.1
An M-flower

The core of the algorithm is the following observation. Let $G = (V, E)$ be a graph and let B be a subset of V. Denote by G/B the graph obtained by *contracting* (or *shrinking*) B to one new vertex, called B. That is, G/B has vertex set $(V \setminus B) \cup \{B\}$, and for each edge e of G an edge obtained from e by replacing any end vertex in B by the new vertex B. (We ignore loops that may arise.) We denote the new edge again by e. (So its ends are modified,

[1] The idea of applying shrinking recursively to matching problems was introduced by Petersen [1891], and was applied in an algorithmic way by Brahana [1917].

but not its name.) We say that the new edge is the *image* (or *projection*) of the original edge.

For any matching M, let M/B denote the set of edges in G/B that are images of edges in M not spanned by B. Obviously, if M intersects $\delta(B)$ in at most one edge, then M/B is a matching in G/B. In the following, we identify a blossom with its set of vertices.

Theorem 24.2. *Let B be an M-blossom in G. Then M is a maximum-size matching in G if and only if M/B is a maximum-size matching in G/B.*

Proof. Let $B = (v_i, v_{i+1}, \ldots, v_t)$.

First assume that M/B is not a maximum-size matching in G/B. Let P be an M/B-augmenting path in G/B. If P does not traverse vertex B of G/B, then P is also an M-augmenting path in G. If P traverses vertex B, we may assume that it enters B with some edge uB that is not in M/B. Then $uv_j \in E$ for some $j \in \{i, i+1, \ldots, t\}$.

(24.7) If j is odd, replace vertex B in P by $v_j, v_{j+1}, \ldots, v_t$.
If j is even, replace vertex B in P by $v_j, v_{j-1}, \ldots, v_i$.

In both cases we obtain an M-augmenting path in G. So M is not maximum-size.

Conversely, assume that M is not maximum-size. We may assume that $i = 0$, that is, $v_i \in X$, since replacing M by $M \triangle EQ$, where Q is the path $(v_0, v_1, \ldots, v_i)$, does not modify the theorem. Let $P = (u_0, u_1, \ldots, u_s)$ be an M-augmenting path in G. If P does not intersect B, then P is also an M/B-augmenting path in G/B. If P intersects B, we may assume that $u_0 \notin B$. (Otherwise replace P by its reverse.) Let u_j be the first vertex of P in B. Then $(u_0, u_1, \ldots, u_{j-1}, B)$ is an M/B-augmenting path in G/B. So M/B is not maximum-size. ∎

Another useful observation is:

Theorem 24.3. *Let $P = (v_0, v_1, \ldots, v_t)$ be a shortest M-alternating $X - X$ walk. Then either P is an M-augmenting path or $(v_0, v_1, \ldots, v_j)$ is an M-flower for some $j \leq t$.*

Proof. Assume that P is not a path. Choose $i < j$ with $v_j = v_i$ and with j as small as possible. So $v_0, \ldots, v_{j-1}$ are all distinct.

If $j - i$ would be even, we can delete $v_{i+1}, \ldots, v_j$ from P so as to obtain a shorter M-alternating $X - X$ walk. So $j - i$ is odd. If j is even and i is odd, then $v_{i+1} = v_{j-1}$ (as it is the vertex matched to $v_i = v_j$), contradicting the minimality of j.

Hence j is odd and i is even, and therefore $(v_0, v_1, \ldots, v_j)$ is an M-flower. ∎

We now describe an algorithm (the *matching-augmenting algorithm*) for the following problem:

(24.8) given: a matching M;
find: an M-augmenting path, if any.

Denote the set of vertices missed by M by X.

(24.9) If there is no M-alternating $X-X$ walk of positive length, there is no M-augmenting path.
If there exists an M-alternating $X-X$ walk of positive length, choose a shortest one, $P = (v_0, v_1, \ldots, v_t)$ say.
Case 1: $\boldsymbol{P}$ is a path. Then output P.
Case 2: $\boldsymbol{P}$ is not a path. Choose j such that $(v_0, \ldots, v_j)$ is an M-flower, with M-blossom B. Apply the algorithm (recursively) to G/B and M/B, giving an M/B-augmenting path P in G/B. Expand P to an M-augmenting path in G (cf. (24.7)).

The correctness of this algorithm follows from Theorems 24.2 and 24.3. It gives a polynomial-time algorithm to find a maximum-size matching, which is a basic result of Edmonds [1965d].

Theorem 24.4. *Given a graph, a maximum-size matching can be found in time $O(n^2m)$.*

Proof. The algorithm directly follows from algorithm (24.9), since, starting with $M = \emptyset$, one can iteratively apply it to find an M-augmenting path P and replace M by $M \triangle EP$. It terminates if there is no M-augmenting path, whence M is a maximum-size matching.

By using (24.6), path P in (24.9) can be found in time $O(m)$. Moreover, the graph G/B can be constructed in time $O(m)$. Since the recursion has depth at most n, an M-augmenting path can be found in time $O(nm)$. Since the number of augmentations is at most $\frac{1}{2}n$, the time bound follows. ∎

This implies for perfect matchings:

Corollary 24.4a. *A perfect matching in a graph (if any) can be found in time $O(n^2m)$.*

Proof. Directly from Theorem 24.4, as a perfect matching is a maximum-size matching. ∎

24.2a. An $O(n^3)$ algorithm

The matching algorithm described above consists of a series of matching augmentations. Each matching augmentation itself consists of a series of two steps performed alternatingly:

(24.10) finding an M-alternating walk, and
shrinking an M-blossom,

until the M-alternating walk is simple, that is, is an M-augmenting path.

Each of these two steps can be done in time $O(m)$. Since there are at most n shrinkings and at most n matching augmentations, we obtain the $O(n^2m)$ time bound.

If we want to save time we must consider speeding up both the walk-finding step and the shrinking step. In a sense, our description above gives a brute-force polynomial-time method. The $O(m)$ time bound for shrinking gives us time to construct the shrunk graph completely, by copying all vertices that are not in the blossom, by introducing a new vertex for the shrunk blossom, and by introducing for each original edge its 'image' in the shrunk graph. The $O(m)$ time bound for finding an M-alternating walk gives us time to find, after any shrinking, a walk starting just from scratch.

In fact, we cannot do much better if we explicitly construct the shrunk graph. But if we modify the graph only locally, by shrinking the M-blossom B and removing loops and parallel edges, this can be done in time $O(|B|n)$. Since the sum of $|B|$ over all M-blossoms B is $O(n)$, this yields a time bound of $O(n^2)$ for shrinking.

To reduce the $O(m)$ time for walk-finding, we keep data from the previous walk-search for the next walk-search, with the help of an M-alternating forest, defined as follows.

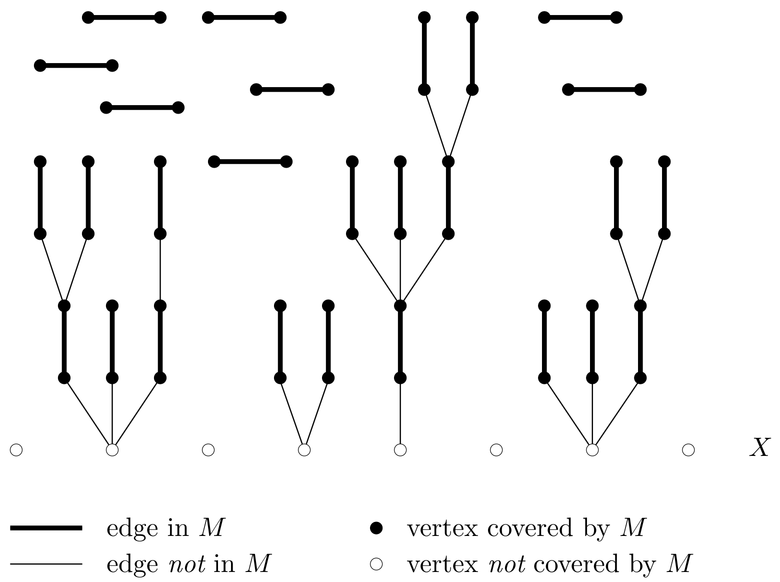


Figure 24.2
An M-alternating forest

Let $G = (V, E)$ be a simple graph and let M be a matching in G. Define X to be the set of vertices missed by M. An *M-alternating forest* is a subset F of E satisfying:

(24.11) F is a forest with $M \subseteq F$, each component of (V, F) contains either exactly one vertex in X or consists of one edge in M, and each path in F starting in X is M-alternating

(cf. Figure 24.2). For any M-alternating forest F, define

(24.12) $\text{even}(F) := \{v \in V \mid F \text{ contains an even-length } X - v \text{ path}\}$,
$\text{odd}(F) := \{v \in V \mid F \text{ contains an odd-length } X - v \text{ path}\}$,
$\text{free}(F) := \{v \in V \mid F \text{ contains no } X - v \text{ path}\}$.

Then each $u \in \text{odd}(F)$ is incident with a unique edge in $F \setminus M$ and a unique edge in M. Moreover:

(24.13) if there is no edge connecting $\text{even}(F)$ and $\text{even}(F) \cup \text{free}(F)$, then M is a maximum-size matching.

Indeed, if there is no such edge, $\text{even}(F)$ is a stable set in $G - \text{odd}(F)$. Hence, setting $U := \text{odd}(F)$:

(24.14) $o(G - U) \geq |\text{even}(F)| = |X| + |\text{odd}(F)| = (|V| - 2|M|) + |U|$,

and hence M has maximum size by (24.2).

Now algorithmically, we keep, next to E and M, an M-alternating forest F. We keep the set of vertices by a doubly linked list. We keep for each vertex v, the edges in E, M, and F, incident with v as doubly linked lists. We also keep the incidence functions $\chi^{\text{even}(F)}$ and $\chi^{\text{odd}(F)}$. Moreover, we keep for each vertex v of G one edge $e_v = vu$ with $u \in \text{even}(F)$, if such an edge exists.

Initially, $F := M$ and for each $v \in V$ we select an edge $e_v = vu$ with $u \in X$ (if any). The iteration is:

(24.15) Find a vertex $v \in \text{even}(F) \cup \text{free}(F)$ for which $e_v = vu$ exists.
Case 1: $v \in \text{free}(F)$. Add uv to F. Let vw be the edge in M incident with v. For each edge wx incident with w, set $e_x := wx$.
Case 2: $v \in \text{even}(F)$. Find the $X - u$ and $X - v$ paths P and Q in F.
Case 2a: P and Q are disjoint. Then P and Q form with uv an M-augmenting path.
Case 2b: P and Q are not disjoint. Then P and Q contain an M-blossom B. For each edge bx with $b \in B$ and $x \notin B$, set $e_x := Bx$. Replace G by G/B and remove all loops and parallel edges from E, M, and F.

The number of iterations is at most $|V|$, since, in each iteration, $|V| + |\text{free}(F)|$ decreases by at least 2 (one of these terms decreases by at least 2 and the other does not change). We end up either with a matching augmentation or with the situation that there is no edge connecting $\text{even}(F)$ and $\text{even}(F) \cup \text{free}(F)$, in which case M has maximum size by (24.13).

It is easy to update the data structure in Case 1 in time $O(n)$. In Case 2, the paths P and Q can be found in time $O(n)$, and hence in Case 2a, the M-augmenting path is found in time $O(n)$.

Finally, the data structure in Case 2b can be updated in $O(|B|n)$ time[2]. Also a matching augmentation in G/B can be transformed to a matching augmentation in G in time $O(|B|n)$. Since $|B|$ is bounded by twice the decrease in the number of vertices of the graph, this takes time $O(n^2)$ overall.

Hence a matching augmentation can be found in time $O(n^2)$, and therefore:

Theorem 24.5. *A maximum-size matching can be found in time $O(n^3)$.*

Proof. From the above. ∎

The first $O(n^3)$-time cardinality matching algorithm was published by Balinski [1969], and consists of a depth-first strategy to find an M-alternating forest, replacing shrinking by a clever labeling technique.

Bottleneck in a further speedup is storing the shrinking. With the disjoint set union data structure of Tarjan [1975] one can obtain an $O(nm\alpha(m,n))$-time algorithm (Gabow [1976a]). A special set union data structure of Gabow and Tarjan [1983,1985] gives an $O(nm)$-time algorithm. An $O(\sqrt{n}\,m)$-time algorithm was announced (with partial proof) by Micali and Vazirani [1980]. A proof was given by Blum [1990], Vazirani [1990,1994], and Gabow and Tarjan [1991] (cf. Peterson and Loui [1988]).

24.3. Matchings covering given vertices

Brualdi [1971d] derived from Tutte's 1-factor theorem the following extension of the Tutte-Berge formula:

Theorem 24.6. *Let $G = (V,E)$ be a graph and let $T \subseteq V$. Then the maximum size of a subset S of T for which there is a matching covering S is equal to the minimum value of*

$$|T| + |U| - o_T(G-U) \tag{24.16}$$

over $U \subseteq V$. Here $o_T(G-U)$ denotes the number of odd components of $G-U$ contained in T.

Proof. For any matching M in G and any $U \subseteq V$, at most $|U|$ odd components of $G-U$ can be covered completely by M. So M misses at least $o_T(G-U) - |U|$ vertices in T. This shows that the minimum is not less than the maximum.

To see equality, let μ be equal to the minimum. Let C be a set disjoint from V with $|C| = |V|$ and let $C' \subseteq C$ with $|C'| = |T| - \mu$. Make a new graph H by extending G by C, in such a way that C is a clique, each vertex in C'

[2] For each $Z \in \{E, M, F\}$, we scan the vertices b in B, and for $b \in B$ we scan the Z-neighbours w of b. If w does not belong to B and was not met as a Z-neighbour of an earlier scanned vertex in B, we replace bw by Bw in Z. Otherwise, we delete bw from Z.

is adjacent to each vertex in V, and each vertex in $C \setminus C'$ is adjacent to each vertex in $V \setminus T$.

If H has a perfect matching M, then M contains at most $|C'| = |T| - \mu$ edges connecting T and C (since T is not connected to $C \setminus C'$). Hence at least μ vertices in T are covered by edges in M spanned by V, as required.

So we may assume that H has no perfect matching. Then by Tutte's 1-factor theorem, there is a set W of vertices of H such that $H - W$ has at least $|W| + 2$ odd components (since $|V| + |C|$ is even).

If $C' \not\subseteq W$, then $H - W$ has only one component (since each vertex in C' is adjacent to every other vertex), a contradiction. If $C \subseteq W$, then $H - W$ has at most $|V|$ components, while $|W| + 2 \geq |C| + 2 = |V| + 2$, a contradiction.

So $C' \subseteq W$ and $C \setminus C' \not\subseteq W$. Then at most one component of $H - W$ is not contained in T (since $C \setminus C'$ is a clique and each vertex in $C \setminus C'$ is adjacent to each vertex in $V \setminus T$). Let $U := W \cap V$. Then

$$\begin{aligned} (24.17) \qquad o_T(G - U) &= o_T(H - W) \geq o(H - W) - 1 > |W| \geq |C'| + |U| \\ &= |T| - \mu + |U|, \end{aligned}$$

contradicting the definition of μ. ∎

(This theorem was also given by Las Vergnas [1975b].)

A consequence is a result of Lovász [1970c] on sets of vertices covered by matchings:

Corollary 24.6a. *Let $G = (V, E)$ be a graph and let T be a subset of V. Then G has a matching covering T if and only if T contains at most $|U|$ odd components of $G - U$, for each $U \subseteq V$.*

Proof. Directly from Theorem 24.6. ∎

(This theorem was also given by McCarthy [1975].)

24.4. Further results and notes

24.4a. Complexity survey for cardinality nonbipartite matching

$O(n^2 m)$	Edmonds [1965d] (cf. Witzgall and Zahn [1965])
$O(n^3)$	Balinski [1969] (also Gabow [1973,1976a], Karzanov [1976], Lawler [1976b])
$O(nm\alpha(m, n))$	Gabow [1976a]
$O(n^{5/2})$	Even and Kariv [1975], Kariv [1976] (also Bartnik [1978])
$O(\sqrt{n}\, m \log n)$	Even and Kariv [1975], Kariv [1976]

≫

continued

	$O(\sqrt{n}\, m \log\log n)$	Kariv [1976]
	$O(\sqrt{n}\, m + n^{1.5+\varepsilon})$	Kariv [1976] for each $\varepsilon > 0$
	$O(\sqrt{n}\, m)$	announced by Micali and Vazirani [1980], full proof in Blum [1990], Vazirani [1990,1994], and Gabow and Tarjan [1991](cf. Gabow and Tarjan [1983,1985])
*	$O(\sqrt{n}\, m \log_n \frac{n^2}{m})$	Goldberg and Karzanov [1995]

Here $*$ indicates an asymptotically best bound in the table. (Kameda and Munro [1974] claim to give an $O(nm)$-time cardinality matching algorithm, but the proof contains some errors which I could not resolve.)

Gabow and Tarjan [1988a] observed that the method of Micali and Vazirani [1980] also implies that one can find, for given k, a matching of size at least $\nu(G) - \frac{n}{k}$ in time $O(km)$. They derived that a maximum-size matching M minimizing $\max_{e\in M} w(e)$ can be found in time $O(\sqrt{n \log n}\, m)$. (the 'bottleneck matching problem').

Mulmuley, Vazirani, and Vazirani [1987a,1987b] showed that 'matching is as easy as matrix inversion', which is especially of interest for the parallel complexity.

24.4b. The Edmonds-Gallai decomposition of a graph

There is a canonical set U that attains the minimum in (24.2). It has the property that the odd components of $G-U$ cover an inclusionwise minimal set of vertices, and is given by the *Edmonds-Gallai decomposition*, independently found by Edmonds [1965d] and Gallai [1963a,1964].

Let $G = (V, E)$ be a graph. The Edmonds-Gallai decomposition of G is the partition of V into $D(G)$, $A(G)$, and $C(G)$ defined as follows (recall that $N(U) := \{v \in V \setminus U \mid \exists u \in U\colon uv \in E\}$):

(24.18)
$$D(G) := \{v \in V \mid \text{there exists a maximum-size matching missing } v\},$$
$$A(G) := N(D(G)),$$
$$C(G) := V \setminus (D(G) \cup A(G)).$$

It yields a 'canonical' certificate of maximality of a matching:

Theorem 24.7. *$U := A(G)$ attains the minimum in* (24.2)*, $D(G)$ is the union of the odd components of $G-U$, and (hence) $C(G)$ is the union of the even components of $G - U$.*

Proof. *Case 1: $D(G)$ is a stable set.* Let M be a maximum-size matching and let X be the set of vertices missed by M. Then each vertex v in $A(G)$ is contained in an edge $uv \in M$ (as $v \notin D(G)$). We show that $u \in D(G)$. Assume that $u \notin D(G)$.

Since $v \in A(G) = N(D(G))$, there is an edge vw with $w \in D(G)$. Let N be a matching missing w. Then $M \triangle N$ contains a path component starting at a vertex in X and ending at w. Let $(v_0, v_1, \ldots, v_t)$ be this path, with $v_0 \in X$ and $v_t = w$. Then t is even and $v_i \in D(G)$ for each even i (because $M \triangle \{v_0v_1, v_2v_3, \ldots, v_{i-1}v_i\}$ is a maximum-size matching missing v_i). Hence, assuming $u \notin D(G)$, the edge vu is not on P. So extending P by wv and vu gives a path Q. Then $M \triangle Q$ is a maximum-size matching missing u. So $u \in D(G)$.

As this is true for any $v \in A(G)$, we see that part of M matches $A(G)$ and $D(G) \setminus X$. Hence

(24.19) $$o(G - U) \geq |D(G)| = |X| + |A(G)| = |V| - 2|M| + |U|.$$

So U attains the minimum in (24.2), and moreover $o(G - U) = |D(G)|$, that is, $D(G)$ is the union of the odd components of $G - U$.

Case 2: $D(G)$ spans some edge $e = uv$. Let M and N be maximum-size matchings missing u and v, respectively. Then $M \cup N$ contains a path component P starting at u. If it does not end at v, then $P \cup \{e\}$ forms an N-augmenting path, contradicting the maximality of N. So P ends at v, and hence $P \cup \{e\}$ gives an M-blossom B.

Let $G' := G/B$ and $M' := M/B$ and let X' be the set of vertices of G' missed by M'. By Theorem 24.2, $|M'| = \nu(G')$. Then

(24.20) $$D(G') = (D(G) \setminus B) \cup \{B\},$$

since $B \in D(G')$ and since for each $v \in V \setminus B$:

(24.21) $v \in D(G') \iff G'$ has an even-length M'-alternating $X' - v$ path $\iff G$ has an even-length M-alternating $X - v$ path $\iff v \in D(G)$.

This proves (24.20), which implies that $A(G') = A(G)$ and $C(G') = C(G)$. By induction, $D(G')$ is the union of the odd components of $G' - U$. Hence $D(G)$ is the union of the odd components of $G - U$ (since $B \subseteq D(G)$ by (24.20)). Also by induction, $|M'| = \frac{1}{2}(|V'| + |U| - o(G' - U))$. Hence $|M| = \frac{1}{2}(|V| + |U| - o(G - U))$, since $|V| - 2|M| = |V'| - 2|M'|$. ∎

So $U = A(G)$ is the unique set attaining the minimum in (24.2) for which the union of the odd components of $G - U$ is inclusionwise minimal.

Note that:

(24.22) for any U attaining the minimum in (24.2), each maximum-size matching M has exactly $\lfloor \frac{1}{2}|K| \rfloor$ edges contained in any component K of $G - U$, and each edge of M intersecting U also intersects some odd component of $G - U$.

This implies the following. Call a graph $G = (V, E)$ *factor-critical* if $G - v$ has a perfect matching for each vertex v.

Corollary 24.7a. *Let $G = (V, E)$ be a graph. Then each component K of $G[D(G)]$ is factor-critical.*

Proof. Directly from Theorem 24.7 and (24.22): if $v \in K$, then $v \in D(G)$, and hence $G - v$ has a maximum-size matching M missing v. By (24.22), M has $\lfloor \frac{1}{2}|K| \rfloor$ edges contained in K. So $K - v$ has a perfect matching. ∎

The Edmonds-Gallai decomposition can be found in polynomial time, since the set $D(G)$ of vertices missed by at least one maximum-size matching can be determined in polynomial time (with the cardinality matching algorithm). In fact, with the alternating forest approach of Section 24.2a one can find the Edmonds-Gallai decomposition in time $O(n^3)$. If we have a maximum-size matching, it takes $O(n^2)$ time.

24.4c. Strengthening of Tutte's 1-factor theorem

Tutte's 1-factor theorem can be (self-)refined as follows (this theorem also can be derived from Theorem 24.7 and Corollary 24.7a; we give a direct derivation from Tutte's 1-factor theorem):

Theorem 24.8. *A graph $G = (V, E)$ has a perfect matching if and only if for each $U \subseteq V$, the graph $G - U$ has at most $|U|$ factor-critical components.*

Proof. Necessity is easy, since each factor-critical component is odd. To see sufficiency, let the condition be satisfied, and suppose that G has no perfect matching. By Tutte's 1-factor theorem, there is a subset U of V such that $G - U$ has more than $|U|$ odd components. Choose an inclusionwise maximal such set U.

By the condition, at least one component K of $G-U$ is not factor-critical. That is, K contains a vertex v such that $K - v$ has no perfect matching. Then by Tutte's 1-factor theorem, there exists a subset U' of $K - v$ such that $K - v - U'$ has more than $|U'|$ odd components, and hence at least $|U'|+2$ odd components (since $K-v$ has an even number of vertices). Now define $U'' := U \cup U' \cup \{v\}$. Then $G - U''$ has more than $|U''|$ odd components. As $U'' \supset U$, this contradicts the maximality of U. ∎

24.4d. Ear-decomposition of factor-critical graphs

As mentioned, a graph $G = (V, E)$ is *factor-critical* if, for each $v \in V$, the graph $G - v$ has a perfect matching. Lovász [1972b] showed that all factor-critical graphs can be constructed by 'odd ear-decompositions' in the following sense. We say that graph H arises by *adding an odd ear* from G, if H arises from G by adding an odd-length path at two (not necessarily distinct) vertices of G. That is, if there is a path or circuit $(v_0, v_1, \ldots, v_t)$ in H with t odd, $v_1, \ldots, v_{t-1}$ each having degree 2, and $G = H - \{v_1, \ldots, v_{t-1}\}$.

It is easy to see that if H arises by adding an odd ear to a factor-critical graph G, then H is again factor-critical. Now each factor-critical graph arises in this way from the one-vertex graph:

Theorem 24.9. *A graph G is factor-critical if and only if there exists a series of graphs $G_0, \ldots, G_k$ with G_0 being a one-vertex graph, $G_k = G$, and G_i arising by adding an odd ear to G_{i-1} ($i = 1, \ldots, k$).*

Proof. For sufficiency, see above. To see necessity, fix, for each vertex v of G, a perfect matching M_v of $G-v$. Choose a vertex u of G. Let H be a maximal subgraph of G such that

(24.23)
- (i) H arises by a series of odd ear addings from the one-vertex graph on u;
- (ii) for each edge $e \in M_u$, if e intersects VH, then $e \in EH$.

Such a graph trivially exists, as the one-vertex graph on u satisfies (24.23).

If $EH = EG$ we are done, so assume $EH \neq EG$. As G is factor-critical, G is connected, and hence there is an edge $e = vw \in EG \setminus EH$ with $v \in VH$. Consider $M_w \cup M_u$. One of its components is an even-length $w-u$ path $P = (v_1, \ldots, v_t)$ with $v_1 = w$ and $v_t = u$. So $v_t \in VH$. Let j be the smallest index with $v_j \in VH$. Then j is odd, since otherwise $v_{j-1}v_j \in M_u$ with $v_{j-1} \notin VH$ and $v_j \in VH$, contradicting (24.23)(ii).

Let Q be the path $(v, v_1, \ldots, v_j)$. Then $H \cup Q$ arises by adding an odd ear to H, and moreover, it satisfies (24.23)(ii) again, contradicting the maximality of H. ∎

(This is the original proof of Lovász [1972b].)

As a consequence we have a recursive characterization of factor-critical graphs:

Corollary 24.9a. *Let $G = (V, E)$ be a graph with $|V| \geq 2$. Then G is factor-critical if and only if G has an odd circuit C with G/C factor-critical.*

Proof. To see sufficiency, let C be an odd circuit with G/C factor-critical. We show that G is factor-critical. Choose $v \in V$. If $v \in C$, let M' be a perfect matching of $G[C \setminus \{v\}]$. Since G/C is factor-critical, $G - C$ has a perfect matching M''. Then $M \cup M''$ is a perfect matching of $G - v$.

If $v \notin C$, let M'' be a perfect matching of $(G/C) - v$. In G this gives a matching covering all vertices in $V \setminus (C \cup \{v\})$ and exactly one vertex, u say, in C. Let M' be a perfect matching in $G[C \setminus \{u\}]$. Then $M' \cup M''$ is a perfect matching of $G - v$. This shows sufficiency.

Necessity is shown with Theorem 24.9. Let G be factor-critical. Consider an odd ear-decomposition of G, and let C be the first odd ear. Then the remaining ears form an odd ear-decomposition of G/C, and hence G/C is factor-critical. ∎

(Related results were given by Cornuéjols and Pulleyblank [1983].)

24.4e. Ear-decomposition of matching-covered graphs

A graph $G = (V, E)$ is called *matching-covered* if each edge of G belongs to a perfect matching of G. Matching-covered graphs can be constructed similarly to factor-critical graphs, but now starting from an even circuit (however, the decomposition does not characterize matching-covered graphs). This will be used in proving Theorem 29.11 on the maximum size of a join.

Theorem 24.10. *For each connected matching-covered graph G with at least four vertices there exists a series of graphs $G_0, \ldots, G_k$ with G_0 being an even circuit, $G_k = G$, and G_i arising by adding an odd ear to G_{i-1} ($i = 1, \ldots, k$).*

Proof. For each edge e of G, fix a perfect matching M_e of G containing e. Fix a perfect matching M of G. One easily checks that G contains an M-alternating even circuit C. Let H be a maximal subgraph of G such that

(24.24) (i) H arises by a series of odd ear addings from C;
(ii) for each edge $e \in M$, if e intersects VH, then $e \in EH$.

Such a graph trivially exists, as C satisfies (24.24).

If $EH = EG$ we are done, so assume $EH \neq EG$. As G is connected, there is an edge $e \in EG \setminus EH$ intersecting VH. Consider $M_e \cup M$. Then the component of $M_e \cup M$ containing e gives an odd ear that can be added to H, contradicting the maximality of H. ∎

A direct algorithmic proof was given by Little and Rendl [1989]. Little [1974] showed that in a matching-covered graph, any two edges belong to a circuit that is in the symmetric difference of two perfect matchings. Carvalho, Lucchesi, and Murty [1999] gave more results on ear-decompositions of matching-covered graphs.

24.4f. Barriers in matching-covered graphs

A *barrier* in a graph $G = (V, E)$ is a subset B of V such that $G - B$ has $|B|$ odd components. Note that if B is a barrier in a connected matching-covered graph G, then B is a stable set and each component of $G - B$ is odd.

Lovász and Plummer [1975,1986] showed:

Theorem 24.11. *Let B and C be barriers in a connected matching-covered graph $G = (V, E)$ with $B \cap C \neq \emptyset$. Then $B \cap C$ and $B \cup C$ are barriers again.*

Proof. We first show:

(24.25) if B and C are distinct barriers with $B \cap C \neq \emptyset$, then there exists a nonempty set D with $D \subseteq B \setminus C$ or $D \subseteq C \setminus B$ such that $B \triangle D$ and $C \triangle D$ are barriers again.

As B and C are stable sets, there is a path from $B \cap C$ to $B \triangle C$. Consider a shortest such path, say it runs from $B \cap C$ to $C \setminus B$. It implies that $G - B$ has a component K with a neighbour in $B \cap C$ and intersecting $C \setminus B$. Define $D := K \cap C$. We show that $B \cup D$ and $C \setminus D$ are barriers again.

Fix an edge e connecting $B \cap C$ and K. Let L be the component of $G - C$ incident with e. Let M be a perfect matching containing e. As e connects $K \cap L$ and $B \cap C$, all other edges in M incident with K are contained in K. So if some edge $f \in M$ leaves $K \cap L'$ for some component L' of $G - C$, and $f \neq e$, then f does not leave K. Hence f leaves L', implying $L' \neq L$ (otherwise, L is left by two edges in M). It also implies that f connects $K \cap L'$ and $K \cap C$ and that f is the only edge in M leaving $K \cap L'$. Moreover, each vertex in D is covered by an edge in M, and hence it is such an edge f. Hence the number of components L' of $G - C$ with $K \cap L'$ odd is equal to $|D| + 1$.

Now $B \cup D$ is a barrier, since $G[K \setminus D]$ has $|D|+1$ odd components. So $G-(B \cup D)$ has at least $|B| + |D|$ odd components, and hence $B \cup D$ is a barrier.

Hence, as G is matching-covered, each component of $G - B - D$ is odd. So each component of $G[K \setminus D]$ is odd, and therefore $G[K \setminus D]$ has exactly $|D| + 1$ components. So all but at most $|D| + 1$ components of $G - C$ are also components of $G - (C \setminus D)$. Hence the number of odd components of $G - (C \setminus D)$ is at least $|C| - |D| - 1$, and hence, by parity, at least $|C \setminus D|$. So $C \setminus D$ is a barrier. This proves (24.25).

Now to prove that $B \cup C$ is a barrier, we can assume that we have chosen B and C inclusionwise maximal barriers contained in $B \cup C$. Then $B = C$ by (24.25).

Similarly, to prove that $B \cap C$ is a barrier, we can assume that we have chosen B and C inclusionwise minimal barriers containing $B \cap C$. Again we have $B = C$ by (24.25). ∎

This has the following consequence due to Lovász [1972e] (cf. Kotzig [1960] (Theorem 31)):

Corollary 24.11a. *Any two distinct maximal barriers in a connected matching-covered graph are disjoint.*

Proof. Directly from Theorem 24.11. ∎

Since each singleton is a barrier, Corollary 24.11a implies that the maximal barriers in a connected matching-covered graph partition the vertex set of G. This gives the result of Kotzig [1959b] (Theorem 11):

Corollary 24.11b. *Let $G = (V, E)$ be a connected matching-covered graph. For $u, v \in V$ define $u \sim v$ by:*

(24.26) $\quad$ *$u \sim v$ if and only if $G - u - v$ has no perfect matching.*

Then $\sim$ is an equivalence relation.

Proof. Note that $u \sim v$ if and only if $\{u, v\}$ is contained in some barrier. So the corollary follows directly from Corollary 24.11a. ∎

For much more on barriers in matching-covered graphs, see Lovász and Plummer [1986].

24.4g. Two-processor scheduling

The following problem was considered by Fujii, Kasami, and Ninomiya [1969]. Suppose that we have to carry out certain jobs, where some of the jobs have to be done before other. We can represent this by a partially ordered set $(V, \leq)$ where V is the set of jobs and $x < y$ indicates that job x has to be done before job y. Each job takes one time-unit, say one hour.

Suppose now that there are two workers, each of which can do one job at a time. Alternatively, suppose that you have one machine, that can do at each moment two jobs simultaneously (a *two-processor*).

We wish to do all jobs within a minimum total time span. This problem can be solved with the matching algorithm as follows. Make a graph $G = (V, E)$, with vertex set V (the set of jobs) and with edge set

(24.27) $\quad E := \{\{u, v\} \mid u \not\leq v \text{ and } v \not\leq u\}.$

(So (V, E) is the complementary graph of the 'comparability graph' associated with $(V, \leq)$.)

Consider now a possible schedule of the jobs. That is, we have a sequence $p_1, \ldots, p_t$, where each p_i is either a singleton vertex or an edge of G such that $p_1, \ldots, p_t$ partition V and such that if $u < v$ and $u \in p_i$ and $v \in p_j$, then $i < j$.

Now the pairs in this list should form a matching M in G. Hence $t = |V| - |M|$. In particular, t cannot be smaller than $|V| - \nu(G)$, where $\nu(G)$ is the matching number of G.

Fujii, Kasami, and Ninomiya [1969] showed that in fact one can always make a schedule with $t = |V| - \nu(G)$. For that it is sufficient to show:

Theorem 24.12. *G contains a maximum-size matching $M = \{e_1, \ldots, e_t\}$ such that if $u \in e_i$ and $v \in e_j$ with $u < v$, then $i < j$.*

Proof. The proof is by induction on $|V|$. Let M be a maximum-size matching in G. We may assume that M is a perfect matching, since otherwise we can delete all vertices missed by M, and apply induction.

Let $V^{\min}$ be the set of minimal elements of $(V \leq)$. If $V^{\min}$ contains an edge $uv \in M$ as a subset, we can delete u and v from V, and apply induction. So we may assume that each $s \in V^{\min}$ is contained in an edge $st \in M$ with $t \notin V^{\min}$. Choose an edge $st \in M$ with $s \in V^{\min}$ and with the height of t as small as possible. (The *height* of an element t is the maximum size of a chain in $(V, \leq)$ with maximum element t.) As $t \notin V^{\min}$ there exists an $s't' \in M$ with $s' \in V^{\min}$ and $s' < t$.

Now clearly ss' is an edge of G, as s and s' are minimal elements. Moreover, tt' is an edge of G. For if $t < t'$, then $s' < t < t'$, contradicting the fact that $s't' \in E$; and if $t' < t$, then t' would have smaller height than t.

So replacing st and $s't'$ in M by ss' and tt', we have $ss' \subseteq V^{\min}$, and so by deleting s and s' from V we can apply induction as before. ∎

The theorem implies that there is a linear extension $\preceq$ of $\leq$ and a maximum-size matching M in G such that if $uv \in M$, then u and v are neighbouring in $\preceq$.

Coffman and Graham [1972] gave a direct, $O(n^2)$-time algorithm. (Muntz and Coffman [1969] gave an algorithm for the two-processor scheduling problem if jobs may be interrupted and continued later.) This was improved to $O(m + n\alpha(m,n))$ by Gabow [1982] and to $O(m + n)$ by Gabow and Tarjan [1983,1985].

24.4h. The Tutte matrix and an algebraic matching algorithm

Tutte [1947b] observed the following. Let $G = (V, E)$ be a graph. Choose for each edge e an indeterminate x_e. Let M be a skew-symmetric[3] $V \times V$ matrix with $M_{u,v} = \pm x_e$ if $e = \{u, v\} \in E$, and $M_{u,v} = 0$ otherwise (including $u = v$) (the *Tutte matrix*). Then the rank of M is equal to twice the matching number of G.

Lovász [1979c] showed that substituting random integers for the x_e, gives an efficient randomized algorithm for finding the matching number of G. This idea was extended by Geelen [2000], who proved the following:

(24.28) Let M' arise from M by substituting the x_e by integers from $\{1, \ldots, n\}$, where $n := |V|$. If $\operatorname{rank}(M') < \operatorname{rank}(M)$, then there is an edge e of G and a number $b \in \{1, \ldots, n\}$ such that for the matrix M'' arising from

[3] A matrix M is *skew-symmetric* if $M^{\mathsf{T}} = -M$.

M' by resetting the $\pm x_e$ entries to $\pm b$, we have $\text{rank}(M'') > \text{rank}(M')$, or $\text{rank}(M'') = \text{rank}(M')$ and $D(M'') \supset D(M')$.

Here $D(A)$ denotes the set of $v \in V$ such that the $V \setminus \{v\} \times V \setminus \{v\}$ submatrix of A has the same rank as A.

(24.28) implies a polynomial-time algorithm to compute the matching number of G (and hence to find a maximum-size matching in G): start with an arbitrary matrix M' obtained by substituting the x_e by numbers in $\{1, \ldots, n\}$, and iteratively try to reset an entry to another number from $\{1, \ldots, n\}$, as long as it either increases the rank of M', or maintains the rank and increases $D(M')$. The final matrix has rank equal to the matching number of G.

L. Lovász (cf. Geelen [1995]) extended Tutte's result to the rank of any (not necessarily principal) submatrix of M. Geelen [1995] described the corresponding system of linear inequalities and proved its total dual integrality, generalizing Edmonds' matching polytope theorem.

24.4i. Further notes

Biedl, Bose, Demaine, and Lubiw [1999,2001] gave an $O(n \log^4 n)$ time algorithm to find a perfect matching in cubic bridgeless graphs (linear-time if the graph is moreover planar). Biedl [2001] gave a linear-time reduction of the general matching problem to the matching problem for cubic graphs.

Lower bounds on the maximum size of a matching were given by Nishizeki and Baybars [1979] for planar graphs and by Biedl, Demaine, Duncan, Fleischer, and Kobourov [2001] for several other classes of graphs.

Fulkerson, Hoffman, and McAndrew [1965] showed that any regular graph with an even number of vertices and with the property that each two vertex-disjoint odd circuits are connected by an edge, has a perfect matching (cf. Mahmoodian [1977], Berge [1978b,1981]). Other sufficient conditions were given by Anderson [1972], Sumner [1974a], Las Vergnas [1975a], and Chartrand, Goldsmith, and Schuster [1979].

Plesník [1972] showed that in a k-regular $(k-1)$-edge-connected graph with an even number of vertices, there is a perfect matching not containing $k-1$ prescribed edges (cf. Chartrand and Nebeský [1979]). For $k = 3$ this was proved by Schönberger [1934]. For general k, it can also be derived from Edmonds' perfect matching polytope theorem (Theorem 25.1 below). See also Plesník [1979].

Further studies on the structure of *matching-covered graphs* (graphs in which each edge belongs to a perfect matching) were made by Kotzig [1959a,1959b,1960], Hetyei [1964], Lovász [1970d,1972f,1972d,1972e,1983a], Little, Grant, and Holton [1975], Lovász and Plummer [1975], Gabow [1979], Edmonds, Lovász, and Pulleyblank [1982], Naddef [1982], and Szigeti [1998b].

Gabow, Kaplan, and Tarjan [1999,2001] gave fast algorithms to test if a given perfect matching is unique, to find it, and if it not unique to find another perfect matching.

Sumner [1974b,1976] studied sets U with $o(G-U) > |U|$. Weinstein [1963,1974] and Bollobás and Eldridge [1976] related the matching number to the minimum and maximum degree and the connectivity. Chvátal and Hanson [1976] evaluated the maximum number $f(n, b, d)$ of edges of a graph with n vertices having no vertex of degree $> d$ and no matching of size $> b$.

Implementing cardinality matching algorithms were studied by Burkard and Derigs [1980], Crocker [1993], and Mattingly and Ritchey [1993]. A simulated annealing approach was described by Sasaki and Hajek [1988].

Books covering nonbipartite matching algorithms include Christofides [1975], Lawler [1976b], Minieka [1978], Papadimitriou and Steiglitz [1982], Sysło, Deo, and Kowalik [1983], Tarjan [1983], Gondran and Minoux [1984], Derigs [1988a], Nemhauser and Wolsey [1988], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], Mehlhorn and Näher [1999], and Korte and Vygen [2000]. Surveys on matching algorithms were given by Galil [1983,1986a,1986b].

Motwani [1989,1994] investigated the expected running time of matching algorithms.

Gallai [1950], Tutte [1950], Kaluza [1953], Steffens [1977], and Aharoni [1984a, 1984c,1984d,1988] gave extensions to infinite graphs. The Edmonds-Gallai decomposition was extended to locally finite graphs by Bry and Las Vergnas [1982] (cf. Steffens [1985]).

The behaviour of a greedy heuristic for finding a large matching was investigated by Dyer and Frieze [1991], Dyer, Frieze, and Pittel [1993], and Aronson, Dyer, Frieze, and Suen [1994].

The standard work on matching theory is Lovász and Plummer [1986]. Other books discussing nonbipartite matching include Berge [1973b], Bondy and Murty [1976], Bollobás [1978,1979], Tutte [1984], and Diestel [1997]. Survey articles on matchings were given by Akiyama and Kano [1985b] and Lovász and Plummer [1986], Gerards [1995a], Pulleyblank [1995], and Cunningham [2002].

24.4j. Historical notes on nonbipartite matching

Petersen and Sylvester

Petersen [1891] was among the first to study perfect matchings (1-factors) in graphs, introducing several basic concepts and methods, like factors and alternating paths. He was motivated by finite basis theorems in invariant theory, especially by the question which polynomials form a finite basis. Petersen cooperated with J.J. Sylvester, who did similar studies, leading to an intensive correspondence on the topic in the years 1889-1890 — see Sabidussi [1992] (unfortunately, the letters of Petersen to Sylvester were not found).

In particular, they considered homogeneous polynomials of the form

(24.29) $$\prod_{i<j}(x_i - x_j)^{r_{i,j}},$$

and were interested in conditions under which such a polynomial can be factorized into other homogeneous polynomials of the same form. This is equivalent to characterizing the existence of k-factors in regular graphs. (Graph terms like 'factor' and 'degree' introduced by Petersen are motivated by this interpretation.)

In a letter of 18 October 1889, Sylvester expressed to Petersen the conjecture that each graph of minimum degree at least two has a 2-factor. He had checked it for graphs with up to 7 vertices, and said that he had 'not much doubt of being able to establish the proof for all values of n by the same process which has been successful for the earlier numbers'. Sylvester considered this as the most important theorem discovered hitherto in the science of chemical graphology, a field initiated by Sylvester [1878].

Two days later, Sylvester wrote a letter in which he restricted his conjecture to the case of regular graphs, and he was more doubtful on whether it is true. After a reply of Petersen, Sylvester gave in a letter of 27 October 1889 an example of a graph with 7 vertices, with degrees 2 and 3, not having a 2-factor. In this letter, Sylvester also remarked that as a consequence of his conjecture, each regular graph of odd order has a 2-factorization.

Then, in a letter of 8 November 1889, Sylvester observed that there is a cubic graph on 10 vertices that has no factorization (Figure 24.3). (A graph is *cubic* if it

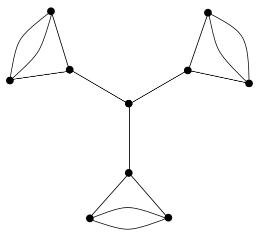


Figure 24.3
Sylvester's graph

is 3-regular.)

Subsequently, on 16 November 1889, Sylvester wrote to Petersen:

> Thanks for your interesting note—I also have a proof of the 'theorem of Ablation' for even equifrequencies.

Apparently, Petersen had written about his theorem that each regular graph of even degree has a 2-factorization, for which Sylvester also said to have a proof.

Next follows correspondence on the proofs the two have, with a lot of mutual misunderstanding. However, after hearing Petersen's proof at a visit of Petersen to Sylvester, at the end of December 1889, Sylvester became convinced of the correctness of Petersen's proof, and found it 'a very beautiful method'. On the other hand, Petersen remained very sceptical about Sylvester's proof, which Sylvester said was by induction on the number of vertices. They decided to publish their proofs separately. However, Sylvester did not publish on the topic; Petersen's proof appeared in the paper Petersen [1891].

Petersen's 1891 paper

In this paper, Petersen first observed that Gordan's finite basis theorem implies that for each n there exists a finite set $\mathcal{G}$ of regular graphs on n vertices (of nonzero degree) with the property that each regular graph on n vertices contains at least one graph in $\mathcal{G}$ as spanning subgraph (factor). (This result can also be proved by elementary means.) Petersen next puts as his goal to characterize all *primitive* graphs, that is, all regular graphs that have no other factors than itself and the 0-regular subgraph.

First, Petersen observed that a 2-regular graph is primitive if and only if at least one of its components is odd. Next, he showed that each 4-regular graph has a 2-factor. To this end, he made an Eulerian tour along all edges, colouring them alternatingly blue and red. The blue edges then form a 2-factor. He observed that similarly one can show more generally that each $2k$-regular graph with an even number of edges has a k-factor.

Next, Petersen showed that each $2k$-regular graph has a 2-factorization. His proof is by observing that the existence of a 2-factorization is invariant under replacing any two disjoint edges ab and cd by ac and bd (by using the result that each 4-regular graph has a 2-factorization).

This solves the factorization problem for k-regular graphs with k even. Petersen next considered the case of odd k. He gave an example of primitive k-regular graphs for arbitrary odd k. He showed that each k-regular graph on n vertices with $k > \frac{1}{2}n + 1$ has a perfect matching. To this end, he considered a matching M and observed that

(24.30) M has maximum size if and only if there is no M-augmenting path.

To formulate this, Petersen coloured the edges in M red, and all other edges blue. A *Wechselweg* (alternating path) is a path coloured alternatingly red and blue. Let $2n$ be the number of vertices of the graph and let α be the size of the matching (thus it misses $2n - 2\alpha$ vertices). Then:

> Wir sahen oben, dass α grösser gemacht werden konnte, wenn wir zwischen zwei von den $2n - 2\alpha$ Punkten einen Wechselweg *cabd* finden konnten; dasselbe gilt wenn wir zwischen zwei von den $2n - 2\alpha$ Punkten überhaupt einen Wechselweg finden können, denn verändert man die Farben der Seiten eines solches Weges, so wird die Anzahl der rothen Linien um eins vergrössert. Man beweist leicht, dass diese Bedingung auch notwendig is.[4]

This brought Petersen to propose an algorithm to find a 1-factor:

> Indem wir die α Linien aufs Geradewohl ausnehmen und dann mittelst Wechselwege α zu vergrössern suchen, können wir untersuchen, ob ein gegebener *graph* primitiv ist oder nicht;[5]

Petersen however preferred a direct characterization:

> es entsteht aber die Frage, ob die primitiven *graphs* sich nicht durch einfache Kennzeichen von den zerlegbaren scheiden.[6]

He conjectured:

---

[4] We saw above that α can be increased, if we could find an alternating path *cabd* between two of the $2n-2\alpha$ points; the same holds if we can find an alternating path at all between two of the $2n-2\alpha$ points, because if one changes the colours of the edges in such a path, then the number of red edges increases by one. One easily proves that this condition is also necessary.

[5] While we select the α edges arbitrarily and then try to increase α by alternating paths, we can investigate if a given *graph* is primitive or not;

[6] the question however arises if the primitive *graphs* are not distinguished from the factorizable by simple characteristics.

> Es spricht etwas dafür, dass ein primitiver *graph Blätter* haben muss, indem ein Blatt ein solcher Theil des *graphs* ist, der nur durch eine einzelne Linie mit dem übrigen Theil in Verbindung steht. Ich habe daher versucht dieses zu beweisen, habe aber die Schwierigkeiten so gross gefunden, dass ich die Untersuchung auf den *graph* dritten Grades beschränkt habe.[7]

Petersen [1891] described the cubic graph on 10 vertices found by Sylvester that has no 1-factor (Figure 24.3), which he called *Sylvester's graph*.

Petersen showed that each primitive cubic graph has at least three leaves. As mentioned, a *leaf* is a subset U of the vertices with $|\delta(U)| = 1$. (A graph is *cubic* if it is 3-regular.)

Again, Petersen showed his theorem with the help of studying alternating paths. Those edges that can be traversed in both directions by alternating paths starting at a 'free' vertex are called 'zweipfeilig' (two-arrow as adjective). He then reduced the problem by shrinking and stated:

> Wir ziehen jetzt jedes zweipfeiliges System in einen Punkt zusammen;[8]

Proofs and extensions of Petersen's theorem

Brahana [1917] gave a shorter proof of Petersen's theorem. He restricted the concept of leaf to a *minimal* set of vertices connected by only one edge to the remainder of the graph. (In fact, also Petersen's proof is valid for this restricted interpretation of leaf.)

Brahana's method is again based on augmenting paths and shrinking. Moreover, he used a reduction to smaller graphs by deleting two adjacent vertices u and v and connecting the two further vertices adjacent to u and v by new edges. This can be done in such a way that the number of leaves remains at most 2.

In fact, part of Brahana's method is algorithmic, and can be considered as a specialization of Edmonds' cardinality matching algorithm. Brahana needs to find a 1-factor, given a matching M of size $\frac{1}{2}n - 1$ (where n is the number of vertices). He described a depth-first method to find an M-augmenting path starting from a vertex missed by M. If it runs into a loop (a 'bicursal circuit'), it can be removed by shrinking:

> We continue this shrinking process as long as there are such bicursal circuits.

Also Errera [1921,1922], Frink [1925], Schönberger [1934], Kőnig [1936], and Baebler [1954] gave alternative proofs of Petersen's theorem (see also Sainte-Laguë [1926b]). The proof of Frink is 'by induction, no shrinking or counting processes being used.' He overlooked however some complications (in relation to the construction of a new 2-connected graph in the proof of his 'Theorem II') — they were resolved by Kőnig [1936]. The proof yields a polynomial-time algorithm to find a perfect matching in a 2-connected cubic graph.

Schönberger [1934] showed that in any 2-connected cubic graph each edge is in a perfect matching, and (more generally) for any two prescribed edges there is a perfect matching not containing these edges.

[7] Something speaks for it that a primitive *graph* must have *leaves*, while a leaf is such a part of the *graph* that is in connection with the remaining part only by one single edge. I therefore have tried to prove this, but have found the difficulties that big, that I have restricted the investigation to the *graph* of third degree.

[8] We now contract each two-arrow system to one point;

Baebler [1937] showed that each k-regular l-edge-connected graph, with k odd and l even, has an l-factor. His proof is based on shrinking.

Tutte

Tutte [1947b] characterized the graphs that have a perfect matching. His proof is essentially that given in Section 24.1a, defining a graph to be 'hyperprime' if it has no perfect matching, but adding any edge creates a perfect matching. He used 'pfaffians' in order to show that, in a hyperprime graph, each component of the subgraph induced by the set of vertices that are not adjacent to all other vertices, is complete. A combinatorial proof of this fact was given by Maunsell [1952].

Tutte's theorem was extended to arbitrary l-factors ($l \in \mathbb{Z}_+$) by Belck [1950] (see Chapter 33); the proof is by extension of Tutte's method. This in turn was generalized by Tutte [1952] to b-factors where $b \in \mathbb{Z}_+^V$. As an 'allied problem', Tutte [1952] considered perfect b-matchings, that is, functions $f \in \mathbb{Z}_+^E$ with $f(\delta(v)) = b(v)$ for each vertex v. The proof is by reduction to the b-factor case, by replacing each edge by several parallel edges.

Then in Tutte [1954b] it is realized that the b-factor and b-matching theorems can be reduced to the case $b = \mathbf{1}$ by splitting vertices and by the construction given in the proof of Theorem 32.1.

Gallai [1950] gave a short proof of Tutte's 1-factor theorem. He showed the following. Let G be a graph without a perfect matching, let M be a maximum-size matching in G, and let v be a vertex missed by M. Let U be the set of vertices u for which there is an M-alternating $v-u$ path of odd length. Then $G-U$ has more than $|U|$ odd components. Gallai [1950] also gave several characterizations for the existence of l-factors in regular graphs, and he considered the infinite case.

Also Tutte [1950] and Kaluza [1953] gave extensions to the infinite case. The main theorem of Ore [1957] is an alternative characterization of the existence of a b-factor. Berge [1958a] extended Tutte's 1-factor theorem to a min-max relation for the maximum size of a matching, the Tutte-Berge formula.

Kotzig [1959a,1959b,1960] studied the structure of matching-covered graphs, leading to a decomposition of any graph (cf. Ore [1959]).

Augmenting paths

Like Petersen, Berge [1957] observed that a matching M is maximum if and only if there is no M-augmenting path, and he suggested the following procedure for solving the cardinality matching problem:

> Construct a maximal matching V, and determine whether there exists an alternating chain W connecting two neutral points. (The procedure is known.) If such a chain exists, change V into $(V \setminus W) \cup (W \setminus V)$, and look again for a new alternating chain; if such a chain does not exist, V is maximum.

In Berge [1958b], a depth-first search approach to finding an augmenting path is sketched, however without shrinking, and not leading to a polynomial-time algorithm.

Also Norman and Rabin [1958,1959] found the augmenting path criterion for maximality of a matching (and similarly, for minimality of an edge cover):

> These results immediately lead to algorithms for a minimum cover and a maximum matching respectively.

Edmonds [1962] and Ray-Chaudhuri [1963] extended the augmenting path criterion to arbitrary hypergraphs.

Edmonds

Edmonds observed that Berge's proposal for finding an augmenting path (quoted above) does not lead to a polynomial-time algorithm. In his personal recollections, Edmonds [1991] stated:

> It is really hard for anyone to see that it isn't easy that when you've got a matching in a graph and you are starting at a deficient node, that you cannot just grow a tree looking for a Berge augmenting path.

Edmonds [1965d] argued:

> Berge proposed searching for augmenting paths as an algorithm for maximum matching. In fact, he proposed to trace out an alternating path from an exposed vertex until it must stop and, then, if it is not augmenting, to back up a little and try again, thereby exhausting possibilities.
> His idea is an important improvement over the completely naive algorithm. However, depending on what further directions are given, the task can still be one of exponential order, requiring an equally large memory to know when it is done.

In the summer of 1963, at a Workshop at the RAND Corporation, Edmonds discovered that shrinking leads to a polynomial-time algorithm to find a maximum-size matching in any graph. The result was described in the paper Edmonds [1965d] (received 22 November 1963), in which paper he also described his views on algorithms and complexity:

> For practical purposes computational details are vital. However, my purpose is only to show as attractively as I can that there is an efficient algorithm. According to the dictionary, "efficient" means "adequate in operation or performance". This is roughly the meaning I want — in the sense that it is conceivable for maximum matching to have no efficient algorithm. Perhaps a better word is "good".
> I am claiming, as a mathematical result, the existence of a *good* algorithm for finding a maximum cardinality matching in a graph.
> There is an obvious finite algorithm, but that algorithm increases in difficulty exponentially with the size of the graph. It is by no means obvious whether *or not* there exists an algorithm whose difficulty increases only algebraically with the size of the graph.

Moreover:

> For practical purposes the difference between algebraic and exponential order is often more crucial than the difference between finite and non-finite.

Edmonds described his algorithm, in terms of paths, trees, flowers, and blossoms, and concluded that the 'order of difficulty' is n^4 (more precisely, it is $O(n^2m)$).

In this paper, Edmonds also introduced the decomposition of any graph which is now called the Edmonds-Gallai decomposition. Also in 1963, Gallai submitted a paper (Gallai [1963a]), in which this decomposition is described implicitly, which was made more explicit in Gallai [1964].

In the Proceedings of the IBM Scientific Computing Symposium on Combinatorial Problems in March 1964 in Yorktown Heights, New York, at the end of Gomory [1966], the following discussion is reported:

J. EDMONDS: I have a comment on the polyhedral approach to complete analysis, supplementing Professor Kuhn's remarks. I do not believe there is any reason for taking as a measure of the algorithmic difficulty of a class of combinatorial extremum problems the number of faces in the associated polyhedra. For example, consider the generalization of the assignment problem from bipartite graphs to arbitrary graphs. Unlike the case of bipartite graphs, the number of faces in the associated polyhedron increases exponentially with the size of the graph. On the other hand, there is an algorithm for this generalized assignment problem which has an upper bound on the work involved just as good as the upper bound for the bipartite assignment problem.
H.W. KUHN: I could not agree with you more. That is shown by the unreasonable effectiveness of the Norman-Rabin scheme for solving this problem. Their result is unreasonable only in the sense that the number of faces of the polyhedron suggests that it ought to be a harder problem than it actually turned out to be. It is not impossible that some day we will have a practical combinatorial algorithm for this problem.
J. EDMONDS: Actually, the amount of work in carrying out the Norman-Rabin scheme generally increases exponentially with the size of the graph.
The algorithm I had in mind is one I introduced in a paper submitted to the Canadian Journal of Mathematics (see Edmonds, 1965). This algorithm depends crucially on what amounts to knowing all the bounding inequalities of the associated convex polyhedron—and, as I said, there are many of them. The point is that the inequalities are known by an easily verifiable characterization rather than by exhaustive listing—so their number is not important.

Chapter 25

The matching polytope

As a by-product of his weighted matching algorithm (to be discussed in Chapter 26), Edmonds obtained a characterization of the matching polytope in terms of defining inequalities. It forms the first class of polytopes whose characterization does not simply follow just from total unimodularity, and its description was a breakthrough in polyhedral combinatorics.

25.1. The perfect matching polytope

The *perfect matching polytope* of a graph $G = (V, E)$ is the convex hull of the incidence vectors of the perfect matchings in G. It is denoted by $P_{\text{perfect matching}}(G)$:

(25.1) $$P_{\text{perfect matching}}(G) = \text{conv.hull}\{\chi^M \mid M \text{ perfect matching in } G\}.$$

So $P_{\text{perfect matching}}(G)$ is a polytope in $\mathbb{R}^E$.

Consider the following set of linear inequalities for $x \in \mathbb{R}^E$:

(25.2)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(\delta(v)) = 1$ for each $v \in V$,
(iii) $x(\delta(U)) \geq 1$ for each $U \subseteq V$ with $|U|$ odd.

In Section 18.1 we saw that if G is bipartite, the perfect matching polytope is fully determined by the inequalities (25.2)(i) and (ii). These inequalities are not enough for, say, K_3: taking $x_e := \frac{1}{2}$ for each edge e of K_3 gives a vector x satisfying (25.2)(i) and (ii) but not belonging to the perfect matching polytope of K_3 (as it is empty).

Edmonds [1965b] showed that for general graphs, adding (25.2)(iii) is enough. It is clear that for any perfect matching M in G, the incidence vector χ^M satisfies (25.2). So $P_{\text{perfect matching}}(G)$ is contained in the polytope determined by (25.2). The essence of Edmonds' theorem is that one needs no more inequalities.

Theorem 25.1 (Edmonds' perfect matching polytope theorem)**.** *The perfect matching polytope of any graph $G = (V, E)$ is determined by* (25.2).

Proof. Clearly, the perfect matching polytope is contained in the polytope Q determined by (25.2). Suppose that the converse inclusion does not hold. So we can choose a vertex x of Q that is not in the perfect matching polytope.

We may assume that we have chosen this counterexample such that $|V|+|E|$ is as small as possible. Hence $0 < x_e < 1$ for all $e \in E$ (otherwise, if $x_e = 0$, we can delete e, and if $x_e = 1$, we can delete e and its ends). So each degree of G is at least 2, and hence $|E| \geq |V|$. If $|E| = |V|$, each degree is 2, in which case the theorem is trivially true. So $|E| > |V|$. Note also that $|V|$ is even, since otherwise $Q = \emptyset$ (consider $U := V$ in (25.2)(iii)).

As x is a vertex, there are $|E|$ linearly independent constraints among (25.2) satisfied with equality. Since $|E| > |V|$, there is an odd subset U of V with $3 \leq |U| \leq |V| - 3$ and $x(\delta(U)) = 1$.

Consider the projections x' and x'' of x to the edge sets of the graphs $G/\overline{U}$ and G/U, respectively (where $\overline{U} := V \setminus U$). Here we keep parallel edges.

Then x' and x'' satisfy (25.2) for $G/\overline{U}$ and G/U, respectively, and hence belong to the perfect matching polytopes of $G/\overline{U}$ and G/U, by the minimality of $|V| + |E|$.

So $G/\overline{U}$ has perfect matchings $M'_1, \ldots, M'_k$ and G/U has perfect matchings $M''_1, \ldots, M''_k$ with

$$(25.3) \qquad x' = \frac{1}{k}\sum_{i=1}^{k} \chi^{M'_i} \text{ and } x'' = \frac{1}{k}\sum_{i=1}^{k} \chi^{M''_i}.$$

(Note that x is rational as it is a vertex of Q.)

Now for each $e \in \delta(U)$, the number of i with $e \in M'_i$ is equal to $kx'(e) = kx(e) = kx''(e)$, which is equal to the number of i with $e \in M''_i$. Hence we can assume that, for each $i = 1, \ldots, k$, M'_i and M''_i have an edge in $\delta(U)$ in common. So $M_i := M'_i \cup M''_i$ is a perfect matching of G. Then

$$(25.4) \qquad x = \frac{1}{k}\sum_{i=1}^{k} \chi^{M_i}.$$

Hence x belongs to the perfect matching polytope of G. ∎

Notes. This proof was given by Aráoz, Cunningham, Edmonds, and Green-Krótki [1983] and Schrijver [1983c], with ideas of Seymour [1979a]. For other proofs, see Balinski [1972], Hoffman and Oppenheim [1978], and Lovász [1979b]. A proof can also be derived from Edmonds' weighted matching algorithm (Chapter 26).

25.2. The matching polytope

The characterization of the perfect matching polytope implies Edmonds' matching polytope theorem. It characterizes the *matching polytope* of a graph $G = (V, E)$, denoted by $P_{\text{matching}}(G)$, which is the convex hull of the incidence vectors of the matchings in G:

(25.5) $$P_{\text{matching}}(G) = \text{conv.hull}\{\chi^M \mid M \text{ matching in } G\}.$$

Again, $P_{\text{matching}}(G)$ is a polytope in $\mathbb{R}^E$.

Corollary 25.1a (Edmonds' matching polytope theorem)**.** *For any graph $G = (V, E)$, the matching polytope is determined by:*

(25.6)
$$\begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for each } e \in E, \\ \text{(ii)} & x(\delta(v)) \leq 1 & \text{for each } v \in V, \\ \text{(iii)} & x(E[U]) \leq \lfloor \frac{1}{2}|U| \rfloor & \text{for each } U \subseteq V \text{ with } |U| \text{ odd.} \end{array}$$

Proof. Clearly, each vector x in the matching polytope satisfies (25.6). To see that the inequalities (25.6) are enough, let x satisfy (25.6). Make a copy $G' = (V', E')$ of G, and add edges vv' for each vertex $v \in V$, where v' is the copy of v in V'. This makes the graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$.

Define $\tilde{x}_e := \tilde{x}_{e'} := x_e$ for each $e \in E$, where e' is the copy of e in E', and $\tilde{x}(vv') := 1 - x(\delta(v))$ for each $v \in V$. Then by Theorem 25.1, $\tilde{x}$ belongs to the perfect matching polytope of $\widetilde{G}$, since $\tilde{x}$ satisfies (25.2) with respect to $\widetilde{G}$.

Indeed, for each $v \in V$ one has $\tilde{x}(\tilde{\delta}(v)) = \tilde{x}(\tilde{\delta}(v')) = 1$ (where $\tilde{\delta} := \delta_{\widetilde{G}}$). Moreover, consider any odd subset U of $\widetilde{V} = V \cup V'$, say $U = W \cup X'$ with $W, X \subseteq V$. Then $\tilde{x}(\tilde{\delta}(U)) \geq \tilde{x}(\tilde{\delta}(W \setminus X)) + \tilde{x}(\tilde{\delta}(X' \setminus W'))$. So we may assume that $W \cap X = \emptyset$, and by symmetry we may assume that W is odd, and hence that $X = \emptyset$. So it suffices to show that for any odd $U \subseteq V$ one has $\tilde{x}(\tilde{\delta}(U)) \geq 1$. Now

(25.7) $$\tilde{x}(\tilde{\delta}(U)) + 2\tilde{x}(\widetilde{E}[U]) = \sum_{v \in U} \tilde{x}(\tilde{\delta}(v)) = |U|,$$

and hence

(25.8) $$\tilde{x}(\tilde{\delta}(U)) = |U| - 2\tilde{x}(\widetilde{E}[U]) \geq |U| - 2\lfloor \tfrac{1}{2}|U| \rfloor = 1.$$

So by Theorem 25.1, $\tilde{x}$ belongs to the perfect matching polytope of $\widetilde{G}$, and hence x belongs to the matching polytope of G. ∎

25.3. Total dual integrality: the Cunningham-Marsh formula

With linear programming duality one can derive from Corollary 25.1a a min-max relation for the maximum weight of a matching:

Corollary 25.1b. *Let $G = (V, E)$ be a graph and let $w \in \mathbb{R}^E_+$ be a weight function. Then the maximum weight of a matching is equal to the minimum value of*

(25.9) $$\sum_{v \in V} y_v + \sum_{U \in \mathcal{P}_{\mathrm{odd}}(V)} z_U \lfloor \tfrac{1}{2}|U| \rfloor,$$

where $y \in \mathbb{R}_+^V$ *and* $z \in \mathbb{R}_+^{\mathcal{P}_{\mathrm{odd}}(V)}$ *satisfy*

(25.10) $$\sum_{v \in V} y_v \chi^{\delta(v)} + \sum_{U \in \mathcal{P}_{\mathrm{odd}}(V)} z_U \chi^{E[U]} \geq w.$$

Proof. Directly with LP-duality from Corollary 25.1a. ∎

The constraints (25.6) determining the matching polytope in fact are totally dual integral, as was shown by Cunningham and Marsh [1978]. This implies that a stronger min-max relation holds than obtained by linear programming duality from the matching polytope inequalities: if w is integer-valued, then in Corollary 25.1b we can restrict y and z to integer vectors:

Theorem 25.2 (Cunningham-Marsh formula)**.** *In Corollary* 25.1b*, if w is integer, we can take y and z integer. We can take z moreover such that the collection $\{U \in \mathcal{P}_{\mathrm{odd}}(V) \mid z_U > 0\}$ is laminar.*[9]

Proof. We prove the theorem by induction on $|E| + w(E)$. If $w(e) = 0$ for some $e \in E$, we can delete e and apply induction. So we may assume that $w(e) \geq 1$ for each $e \in E$.

First assume that there exists a vertex u of G covered by every maximum-weight matching. Let $w' := w - \chi^{\delta(u)}$. By induction, there exist integer y'_v, z'_U that are optimum with respect to w'. Now increasing y'_u by 1, gives y_v, z_U as required for w, since the maximum of $w'(M)$ over all matchings M is strictly less than the maximum of $w(M)$ over all matchings M, as each maximum-weight matching M contains an edge e incident with u.

So we may assume that for each vertex v there exists a maximum-weight matching missing v. Hence if $y \in \mathbb{R}_+^V$ and $z \in \mathbb{R}_+^{\mathcal{P}_{\mathrm{odd}}(V)}$ satisfying (25.10) attain the minimum of (25.9), then $y = \mathbf{0}$. (If $y_u > 0$, then each maximum-weight matching covers u, by complementary slackness.)

Now choose z attaining the minimum (with $y = \mathbf{0}$) such that

(25.11) $$\sum_{U \in \mathcal{P}_{\mathrm{odd}}(V)} z_U \lfloor \tfrac{1}{2}|U| \rfloor^2$$

is as large as possible. Let $\mathcal{F} := \{U \in \mathcal{P}_{\mathrm{odd}}(V) \mid z_U > 0\}$. Then $\mathcal{F}$ is laminar. For suppose not. Let $U, W \in \mathcal{F}$ with $U \cap W \neq \emptyset$ and $U \not\subseteq W \not\subseteq U$. Then $|U \cap W|$ is odd. To see this, choose $v \in U \cap W$. Then there is a maximum-weight matching M missing v. Since $z_U > 0$, $E[U]$ contains $\lfloor \frac{1}{2}|U| \rfloor$ edges in M, and hence each vertex in $U \setminus \{v\}$ is covered by an edge in M contained in U. Similarly, each vertex in $W \setminus \{v\}$ is covered by an edge in M contained in

[9] A collection $\mathcal{F}$ of sets is called *laminar* if $U \cap W = \emptyset$ or $U \subseteq W$ or $W \subseteq U$ for all $U, W \in \mathcal{F}$.

W. Hence each vertex in $(U \cap W) \setminus \{v\}$ is covered by an edge in M contained in $U \cap W$. So $|(U \cap W) \setminus \{v\}|$ is even, and hence $|U \cap W|$ is odd.

Now let $\alpha := \min\{z_U, z_W\}$, and decrease z_U and z_W by α and increase $z_{U \cap W}$ and $z_{U \cup W}$ by α. This resetting maintains (25.10), does not change (25.9), but increases (25.11), contradicting our assumption.

This shows that $\mathcal{F}$ is laminar. Now suppose that z is not integer-valued, and let U be an inclusionwise maximal set in $\mathcal{F}$ with $z_U \notin \mathbb{Z}$. Let $U_1, \ldots, U_k$ be the inclusionwise maximal sets in $\mathcal{F}$ properly contained in U (possibly $k = 0$). As $\mathcal{F}$ is laminar, the U_i are disjoint. Let $\alpha := z_U - \lfloor z_U \rfloor$. Then decreasing z_U by α and increasing each z_{U_i} by α would maintain (25.10) (by the integrality of w), but would strictly decrease (25.9) (since $\sum_{i=1}^{k} \lfloor \frac{1}{2}|U_i| \rfloor < \lfloor \frac{1}{2}|U| \rfloor$). This contradicts the minimality of (25.9). ∎

(This proof follows the method given by Schrijver and Seymour [1977]. Other proofs were given by Hoffman and Oppenheim [1978], Schrijver [1983a,1983c], and Cook [1985].)

Note that the Cunningham-Marsh formula has the Tutte-Berge formula (Corollary 24.1) as special case. The previous theorem is equivalent to:

Corollary 25.2a. *System* (25.6) *is totally dual integral.*

Proof. This follows from Theorem 25.2. ∎

25.3a. Direct proof of the Cunningham-Marsh formula

We give a direct proof of the Cunningham-Marsh formula, as given in Schrijver [1983a] (generalizing the proof of Lovász [1979b] of Edmonds' matching polytope theorem). It does not use Edmonds' matching polytope theorem, which rather follows as a consequence.

Let $G = (V, E)$ be a graph. For each weight function $w \in \mathbb{Z}_+^E$, let ν_w denote the maximum weight of a matching. We must show that for each $w \in \mathbb{Z}_+^E$ there exist $y \in \mathbb{Z}_+^V$ and $z \in \mathbb{Z}_+^{\mathcal{P}_{\mathrm{odd}}(V)}$ such that

$$\sum_{v \in V} y_v + \sum_{U \in \mathcal{P}_{\mathrm{odd}}(V)} z_U \lfloor \tfrac{1}{2}|U| \rfloor \leq \nu_w \tag{25.12}$$

and

$$\sum_{v \in V} y_v \chi^{\delta(v)} + \sum_{U \in \mathcal{P}_{\mathrm{odd}}(V)} z_U \chi^{E[U]} \geq w. \tag{25.13}$$

Suppose that G and w contradict this, with $|V|+|E|+w(E)$ as small as possible. Then G is connected (otherwise one of the components of G will form a smaller counterexample) and $w(e) \geq 1$ for each edge e (otherwise we can delete e). Now there are two cases.

Case 1: There is a vertex u covered by every maximum-weight matching. In this case, let $w' := w - \chi^{\delta(u)}$. Then $\nu_{w'} = \nu_w - 1$. Since $w'(E) < w(E)$, there are y' and z' satisfying (25.12) and (25.13) with respect to w'. Increasing y'_u by 1 gives y and z satisfying (25.12) and (25.13) with respect to w.

Case 2: No vertex is covered by every maximum-weight matching. Now let w' arise from w by decreasing all weights by 1. Let M be a matching with $w'(M) = \nu_{w'}$ and with $|M|$ as large as possible.

Then M does not cover all vertices, as, otherwise, for any matching N of maximum w-weight not covering all vertices:

$$(25.14) \qquad w'(N) = w(N) - |N| > w(N) - |M| \geq w(M) - |M| = w'(M) = \nu_{w'},$$

contradicting the definition of $\nu_{w'}$.

Suppose that M covers all but one vertex (in particular, $|V|$ is odd). Then

$$(25.15) \qquad \nu_w \geq w(M) = w'(M) + |M| = \nu_{w'} + \lfloor \tfrac{1}{2}|V| \rfloor.$$

Since $w'(E) < w(E)$, there are y' and z' satisfying (25.12) and (25.13) with respect to w'. Increasing z'_V by 1 gives y and z satisfying (25.12) and (25.13) with respect to w (by (25.15)), a contradiction.

So we know that M leaves at least two vertices in V uncovered. Let u and v be not covered by M. We can assume that we have chosen M, u, v under the additional condition that the distance $d(u, v)$ of u and v in G is as small as possible. Then $d(u, v) > 1$, since otherwise we could augment M by edge $\{u, v\}$, thereby increasing $|M|$ while not decreasing $w'(M)$. Let t be an internal vertex of a shortest $u - v$ path. Let N be a matching not covering t, with $w(N) = \nu_w$.

Let P be the component of $M \cup N$ containing t. Then P forms a path starting at t and not covering both u and v (as t is not covered by N and u and v are not covered by M). We can assume that P does not cover u. Now the symmetric differences $M' := M \triangle P$ and $N' := N \triangle P$ are matchings again, and $|M'| \leq |M|$ (as M covers t), implying

$$(25.16) \qquad \begin{aligned} w'(M') - w'(M) &= w(M') - |M'| - w(M) + |M| \geq w(M') - w(M) \\ &= w(N) - w(N') = \nu_w - w(N') \geq 0. \end{aligned}$$

So $w'(M') \geq w'(M) = \nu_{w'}$ and hence we have equality throughout. So $w(M') = w(M)$, $w'(M') = w'(M)$, and $|M'| = |M|$. However, M' does not cover t and u while $d(u, t) < d(u, v)$, contradicting our choice of M, u, v.

25.4. On the total dual integrality of the perfect matching constraints

System (25.2) determining the perfect matching polytope is generally not totally dual integral. Indeed, consider the complete graph $G = K_4$ on four vertices, with $w(e) := 1$ for each edge e; then the maximum weight of a perfect matching is 2, while the dual of optimizing $w^{\mathsf{T}}x$ subject to (25.2) is attained only by taking $y(\{v\}) = \frac{1}{2}$ for each vertex v.

However, consider the following system, again determining the perfect matching polytope (by Corollary 25.1a):

$$(25.17) \qquad \begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for each } e \in E; \\ \text{(ii)} & x(\delta(v)) = 1 & \text{for each } v \in V; \\ \text{(iii)} & x(E[U]) \leq \lfloor \tfrac{1}{2}|U| \rfloor & \text{for each } U \subseteq V \text{ with } |U| \text{ odd.} \end{array}$$

Corollary 25.2b. *System* (25.17) *is totally dual integral.*

Proof. Directly from Corollary 25.2a, with Theorem 5.25. ∎

This implies a result stated by Edmonds and Johnson [1970]:

Corollary 25.2c. *The perfect matching inequalities* (25.2) *form a totally dual half-integral system.*

Proof. Let $w \in \mathbb{Z}^E$, and minimize $w^\mathsf{T} x$ subject to (25.2). As it is the same as minimizing $w^\mathsf{T} x$ subject to (25.17), by Corollary 25.2b there is an optimum dual solution $y \in \mathbb{Z}^V$, $z \in \mathbb{Z}_+^{\mathcal{P}_{\text{odd}}(V)}$. Since $x(E[U]) \leq \lfloor \frac{1}{2}|U| \rfloor$ is half of the sum of the inequalities $x(\delta(v)) = 1$ ($v \in U$) and $-x(\delta(U)) \leq -1$, we obtain the total dual half-integrality of (25.2). ∎

This can be strengthened to (Barahona and Cunningham [1989]):

Corollary 25.2d. *If $w \in \mathbb{Z}^E$ and $w(C)$ is even for each circuit C, then the problem of minimizing $w^\mathsf{T} x$ subject to* (25.2) *has an integer optimum dual solution.*

Proof. If $w(C)$ is even for each circuit, there is a subset T of V with $\{e \in E \mid w(e) \text{ is odd}\} = \delta(T)$. Now replace w by $\tilde{w} := w + \sum_{v \in T} \chi^{\delta(v)}$. Then $\tilde{w}(e)$ is an even integer for each edge e. Hence by Corollary 25.2c there is an optimum dual solution $\tilde{y} \in \mathbb{Z}^V$, $z \in \mathbb{Z}_+^{\mathcal{P}_{\text{odd}}(V)}$ for the problem of minimizing $\tilde{w}^\mathsf{T} x$ subject to (25.2). Now setting $y_v := \tilde{y}_v - 1$ if $v \in T$ and $y_v := \tilde{y}_v$ if $v \notin T$ gives an integer optimum dual solution for w. ∎

25.5. Further results and notes

25.5a. Adjacency and diameter of the matching polytope

Balinski and Russakoff [1974] and Chvátal [1975a] characterized adjacency on the matching polytope:

Theorem 25.3. *Let M and N be distinct matchings in a graph $G = (V, E)$. Then χ^M and χ^N are adjacent vertices of the matching polytope if and only if $M \triangle N$ is a path or circuit.*

Proof. To see necessity, let χ^M and χ^N be adjacent. Let P be any nontrivial component of $M \triangle N$ and let $M' := M \triangle P$ and $N' := N \triangle P$. So M' and N' are matchings again. Then

$$(25.18) \qquad \tfrac{1}{2}(\chi^M + \chi^N) = \tfrac{1}{2}(\chi^{M'} + \chi^{N'}).$$

As χ^M and χ^N are adjacent, it follows that $\{M', N'\} = \{M, N\}$. So $M' = N$ and $N' = M$, and therefore $M \triangle N = P$.

To see sufficiency, let $P := M \triangle N$ be a path or circuit. Suppose that χ^M and χ^N are not adjacent. Then there exists a matching $L \neq M, N$ that belongs to the smallest face of the matching polytope containing $x := \frac{1}{2}(\chi^M + \chi^N)$. As $x_e = 0$ for each edge $e \notin M \cup N$ and $x_e = 1$ for each edge $e \in M \cap N$, we know that $M \cap N \subseteq L \subseteq M \cup N$. Moreover, $x(\delta(v)) = 1$ for each vertex v covered both by M and by N. Hence each vertex v covered both by M and by N is covered by L. As P is a path or a circuit, it follows that $L = M$ or $L = N$, a contradiction. ∎

This has as consequence for the diameter:

Corollary 25.3a. *The diameter of the matching polytope of any graph $G = (V, E)$ is equal to the maximum size $\nu(G)$ of the matchings.*

Proof. First, by Theorem 25.3, for any two matchings M and N, the distance of χ^M and χ^N is at most the number of nontrivial components of $M \triangle N$. Since each such component contains at least one edge and since these edges are pairwise disjoint, this number is at most $\nu(G)$. So the diameter is at most $\nu(G)$.

Equality follows from the fact that $\emptyset$ and any matching M have distance $|M|$. This follows from the fact that if M and N are adjacent, then $\big||M| - |N|\big| \leq 1$ by Theorem 25.3. ∎

Another direct consequence concerns adjacency on the *perfect* matching polytope:

Corollary 25.3b. *Let M and N be perfect matchings in a graph $G = (V, E)$. Then χ^M and χ^N are adjacent vertices of the perfect matching polytope if and only if $M \triangle N$ is a circuit.*

Proof. Directly from Theorem 25.3. ∎

This in turn implies for the diameter of the perfect matching polytope:

Corollary 25.3c. *The perfect matching polytope of a graph $G = (V, E)$ has diameter at most $\frac{1}{2}|V|$ ($\frac{1}{4}|V|$ if G is simple).*

Proof. For any two perfect matching M, N, the symmetric difference has at most $\frac{1}{2}|V|$ components (each being a circuit). Hence Corollary 25.3b implies that χ^M and χ^N have distance at most $\frac{1}{2}|V|$.

If G is simple the bounds can be sharpened to $\frac{1}{4}|V|$, as each even circuit has at least four vertices. ∎

Padberg and Rao [1974] showed that if G is a complete graph with an even number $2n$ of vertices, then $P_{\text{perfect matching}}(G)$ has diameter at most 2. (This can be derived from Theorem 18.5, since any two perfect matchings belong to some $K_{n,n}$-subgraph of G, which subgraph gives a face of $P_{\text{perfect matching}}(G)$.)

25.5b. Facets of the matching polytope

Pulleyblank and Edmonds [1974] (cf. Pulleyblank [1973]) characterized which of the inequalities (25.6) give a facet of the matching polytope:

Let $G = (V, E)$ be a graph. Define

(25.19) $I := \{v \in V \mid \deg_G(v) \geq 3$, or $\deg_G(v) = 2$ and v is contained in no triangle, or $\deg_G(v) = 1$ and the neighbour of v also has degree $1\}$,
$\mathcal{T} := \{U \subseteq V \mid |U| \geq 3$, $G[U]$ is factor-critical and 2-vertex-connected$\}$.

(Recall that graph G is *factor-critical* if, for each vertex v of G, $G - v$ has a perfect matching.)

Consider the system

(25.20)
(i) $x_e \geq 0$ for $e \in E$,
(ii) $x(\delta(v)) \leq 1$ for $v \in I$,
(iii) $x(E[U]) \leq \lfloor \frac{1}{2}|U| \rfloor$ for $U \in \mathcal{T}$.

We first show:

Theorem 25.4. *Each inequality in* (25.6) *is a nonnegative integer combination of inequalities* (25.20).

Proof. First consider a vertex $v \notin I$. If $\deg_G(v) = 1$, let u be the neighbour of v. Then $u \in I$ and

(25.21) $$x(\delta(v)) = x(\delta(u)) - \sum_{e \in \delta(u) - \delta(v)} x_e.$$

If $\deg_G(v) = 2$ and v is contained in a triangle $G[U]$, then $x(\delta(v)) = x(E[U]) - x_e$, where e is the edge in $E[U]$ not incident with v.

Next consider a subset U of V with $|U|$ odd and $|U| \geq 3$. We show that $x(E[U]) \leq \lfloor \frac{1}{2}|U| \rfloor$ is a sum of constraints (25.20), by induction on $|U|$. If $U \in \mathcal{T}$ we are done. So assume that $U \notin \mathcal{T}$. Let $H := G[U]$. If H is not factor-critical, there is a vertex v such that $H - v$ has no perfect matching. Let $U' = U \setminus \{v\}$. Then $x(E[U']) \leq \lfloor \frac{1}{2}|U| \rfloor - 1$ for the incidence vector x of any matching, and hence also for each vector x in the matching polytope. By the total dual integrality of the matching constraints (Corollary 25.2a), this constraint is a sum of constraints (25.6), and hence, by induction, of constraints (25.20). So $x(E[U]) \leq \lfloor \frac{1}{2}|U| \rfloor$ is a sum of constraints (25.20), as $E[U] \subseteq E[U'] \cup \delta(v)$.

If H is factor-critical, it has a cut vertex v. Let $K_1, \ldots, K_t$ be the components of $H - v$ and let $U_i := K_i \cup \{v\}$ for each i. As H is factor-critical, each $|U_i|$ is odd. Hence $x(E[U]) \leq \lfloor \frac{1}{2}|U| \rfloor$ is a sum of the constraints $x(E[U_i]) \leq \lfloor \frac{1}{2}|U_i| \rfloor$. ∎

This implies that (25.20) is sufficient:

Corollary 25.4a. (25.20) *determines the matching polytope.*

Proof. Directly from Corollary 25.1a and Theorem 25.4. ∎

Another consequence is the result of Cunningham and Marsh [1978] that the irredundant system still is totally dual integral:

Corollary 25.4b. (25.20) *is TDI.*

Proof. Directly from Theorem 25.4, using the total dual integrality of system (25.6). ∎

(For a short proof of this result, see Cook [1985].)

Next we show that each inequality in (25.20) determines a facet. To this end, we first show:

Lemma 25.5α. *Let $G = (V, E)$ be a 2-vertex-connected factor-critical graph and let W be a proper subset of V with $|W|$ odd and ≥ 3. Then G has a matching of size $\lfloor \frac{1}{2}|V| \rfloor$ containing less than $\lfloor \frac{1}{2}|W| \rfloor$ edges in $E[W]$.*

Proof. Choose a vertex $v \in W$ that is adjacent to at least one vertex in $V \setminus W$. If v has no neighbour in W, choose $u \in W \setminus \{v\}$ and let M be a perfect matching in $G - u$. This matching has the required properties.

So we may assume that v has a neighbour in W. Make from G a graph G', by splitting v into two vertices v' and v'', where v' is adjacent to all vertices in W adjacent to v and where v'' is adjacent to all vertices in $V \setminus W$ adjacent to v.

If G' has a perfect matching M', then deleting the edge in M' covering v', and identifying v' and v'', gives a matching M in G with $|M| = \lfloor \frac{1}{2}|V| \rfloor$, but with $|M \cap E[W]| < \lfloor \frac{1}{2}|W| \rfloor$.

So we can assume that G' has no perfect matching. Then by Tutte's 1-factor theorem, there is a subset U of VG' such that $G' - U$ has more than $|U|$ odd components. Since the graph $G' \cup \{v'v''\}$ has a perfect matching[10] (as G is factor-critical), we know that $v', v'' \notin U$.

If $U = \emptyset$, G' has an odd component, contradicting the fact that G' is connected (since G is 2-vertex-connected) and has an even number of vertices. So $U \neq \emptyset$. Choose $u \in U$, and let M be a perfect matching in $G - u$. Then M yields a matching M' in G' missing u and exactly one of v', v''. So $G' \cup \{uv'\}$ or $G' \cup \{uv''\}$ has a perfect matching, contradicting the fact that $u \in U$ and $G' - U$ has more than $|U|$ odd components. ∎

This lemma is used in proving:

Theorem 25.5. *Each inequality in* (25.20) *determines a facet.*

Proof. We clearly cannot delete any inequality $x_e \geq 0$, since otherwise the vector x defined by $x_e := -1$ and $x_{e'} := 0$ for each $e' \neq e$ would be a solution. So it determines a facet.

Consider next an inequality

(25.22) $\qquad x(\delta(v)) \leq 1$

for some $v \in I$. Let F be the set of vectors x in the matching polytope satisfying $x(\delta(v)) = 1$. Suppose that F is not a facet. Then there is a facet F' with $F' \supset F$. So F' is determined by one of the inequalities (25.20).

[10] By $G' \cup \{uv\}$ we denote the graph obtained from G' by adding edge uv.

If F' is determined by $x_e = 0$ for some $e \in E$, choose a matching M with $e \in M$ and covering v (the existence of such a matching follows from the definition of I). Then $\chi^M \in F \setminus F'$, a contradiction.

If F' is determined by $x(\delta(u)) = 1$ for some $u \in I$, then $u \neq v$ (since $F' \neq F$) and there is an edge e incident with u but not with v (since $u, v \in I$). Hence for matching $M := \{e\}$ we have $\chi^M \in F \setminus F'$, a contradiction.

If F' is determined by $x(E[U]) = \lfloor \frac{1}{2}|U| \rfloor$ for some $U \in \mathcal{T}$, then $\delta(u) \subseteq E[U]$ and $\lfloor \frac{1}{2}|U| \rfloor = 1$ (since $\chi^M \in F \subseteq F'$ for $M = \{e\}$, for each $e \in \delta(v)$). So $|U| = 3$. Since $F' \neq F$, U determines a triangle, contradicting the fact that $v \in I$.

Finally consider an inequality

$$x(E[U]) \leq \lfloor \tfrac{1}{2}|U| \rfloor \tag{25.23}$$

for some $U \in \mathcal{T}$. Let F be the set of vectors x in the matching polytope satisfying $x(E[U]) = \lfloor \frac{1}{2}|U| \rfloor$.

Suppose that F is not a facet, and let F' be a facet with $F' \supset F$.

First assume that F' is determined by $x_e = 0$ for some $e \in E$. If e is not spanned by U, there is a $v \in U$ such that $U \setminus \{v\}$ is not intersected by e. Let M be a perfect matching of $G[U] - v$. Then $\chi^{M \cup \{e\}} \in F \setminus F'$, a contradiction. If e is spanned by U, choose $v \in e$ and let M be a perfect matching of $G[U] - v$. Let $f \in M$ intersect e, and define $M' := (M \setminus \{f\}) \cup \{e\}$. Then $\chi^{M'} \in F \setminus F'$, a contradiction.

Next assume that F' is determined by $x(\delta(v)) = 1$ for some $v \in I$. Then, as $G[U]$ is factor-critical, there is a matching M with $|M \cap E[U]| = \lfloor \frac{1}{2}|U| \rfloor$ and $M \cap \delta(v) = \emptyset$. So $\chi^M \in F \setminus F'$, a contradiction.

Finally assume that F' is determined by $x(E[U']) = \lfloor \frac{1}{2}|U'| \rfloor$ for some $U' \in \mathcal{T}$. If $U' \not\subseteq U$, there is a matching M with $|M \cap E[U]| = \lfloor \frac{1}{2}|U| \rfloor$ missing at least two vertices in U' and hence $|M \cap E[U']| < \lfloor \frac{1}{2}|U'| \rfloor$. Then $\chi^M \in F \setminus F'$, a contradiction.

So $U' \subset U$. By Lemma 25.5α, $G[U]$ has a matching M of size $\lfloor \frac{1}{2}|U| \rfloor$ such that less than $\lfloor \frac{1}{2}|U'| \rfloor$ edges in M are spanned by U'. Then $\chi^M \in F \setminus F'$, a contradiction. ∎

(This proof is due to L. Lovász (cf. Cornuéjols and Pulleyblank [1982]). For another proof, see Cook [1985]. See also Giles [1978b].)

Edmonds, Lovász, and Pulleyblank [1982] gave an irredundant system of linear inequalities describing the *perfect* matching polytope. More on the combinatorial structure of the (perfect) matching polytope is given by Naddef and Pulleyblank [1981a].

25.5c. Polynomial-time solvability with the ellipsoid method

In Chapter 26 we shall describe Edmonds' strongly polynomial-time algorithm for the weighted matching problem. This algorithm gives as a by-product the inequalities describing the perfect matching polytope, as we shall see in Section 26.3b.

It turns out that conversely one can derive the strong polynomial-time solvability of the weighted matching problem from the description of the perfect matching polytope (albeit that the method is impractical).

Indeed, the weighted perfect matching problem is equivalent to the optimization problem over the perfect matching polytope. So, by the ellipsoid method, there exists a polynomial-time weighted perfect matching algorithm if and only if there exists a polynomial-time separation algorithm for the perfect matching polytope.

Such a polynomial-time algorithm indeed exists (and would follow conversely also with the ellipsoid method from the polynomial-time solvability of the weighted matching problem). A direct proof was given by Padberg and Rao [1982], and is as follows.

The separation problem for the perfect matching polytope is: given a graph $G = (V, E)$ and a vector $x \in \mathbb{R}_+^E$, decide if x belongs to the perfect matching polytope, and if not, find a separating hyperplane. To answer this question we can first check the constraints (25.2)(i)(ii) in polynomial time. If one of them is violated, it gives a separating hyperplane. If each of them is satisfied, we should check if $x(\delta(U)) < 1$ for some odd subset U of V. Considering x as a capacity function, we should find an odd cut of capacity less than 1. Here an *odd cut* is a cut $\delta(U)$ with $|U|$ odd.

Such a cut can be found in strongly polynomial time. For a graph $G = (V, E)$ and a tree $T = (V, F)$, a *fundamental cut determined by* T is a cut $\delta_E(W_f)$, where $f \in F$ and W_f is one of the components of $T - f$. Then:

Theorem 25.6. *Let $G = (V, E)$ be a graph with $|V|$ even, let $c \in \mathbb{R}_+^E$ be a capacity function, and let $T = (V, F)$ be a Gomory-Hu tree for G and c. Then one of the fundamental cuts determined by T is a minimum-capacity odd cut in G.*

Proof. For each $f \in F$, choose W_f as one of the two components of $T - f$. Let $\delta_G(U)$ be a minimum-capacity odd cut of G. Then U or $V \setminus U$ is equal to the symmetric difference of the W_f over $f \in \delta_F(U)$. Hence $|W_f|$ is odd for at least one $f \in \delta_F(U)$. So $\delta_G(W_f)$ is an odd cut. Let $f = uv$. As $\delta_G(W_f)$ is a minimum-capacity $u - v$ cut and as $\delta_G(U)$ is a $u - v$ cut, we have $c(\delta_G(W_f)) \leq c(\delta_G(U))$. So $\delta_G(W_f)$ is a minimum-capacity odd cut. ∎

This gives algorithmically:

Corollary 25.6a. *A minimum-capacity odd cut can be found in strongly polynomial time.*

Proof. This follows from Theorem 25.6, since a Gomory-Hu tree can be found in strongly polynomial time, by Corollary 15.15a. ∎

As the separation problem for the perfect matching polytope can be reduced to finding a minimum-capacity odd cut, this implies:

Corollary 25.6b. *The separation problem for the perfect matching polytope can be solved in strongly polynomial time.*

Proof. See above. ∎

Corollary 25.6c. *A minimum-weight perfect matching can be found in strongly polynomial time.*

Proof. This follows from Corollary 25.6b, with Theorem 5.11. ∎

25.5d. The matchable set polytope

Let $G = (V, E)$ be a graph. A subset U of V is called *matchable* if the graph $G[U]$ has a perfect matching. The *matchable set polytope* of G is the convex hull (in $\mathbb{R}^V$) of the incidence vectors of matchable sets.

Balas and Pulleyblank [1989] characterized the matchable set polytope as follows (where $N(U)$ is the set of neighbours of U and $o(G[U])$ is the number of odd components of $G[U]$):

Theorem 25.7. *The matchable set polytope of a graph $G = (V, E)$ is determined by:*

(25.24)
$$\begin{array}{lll} \text{(i)} & 0 \leq x_v \leq 1 & \text{for } v \in V, \\ \text{(ii)} & x(U) - x(N(U)) \leq |U| - o(G[U]) & \text{for } U \subseteq V. \end{array}$$

Proof. Each vector in the matchable set polytope of G satisfies (25.24), since the incidence vector of any matchable set satisfies (25.24), since if any odd component K of $G[U]$ is covered by a matching M, then M has an edge connecting K and $N(U)$.

To see the reverse, choose a counterexample with $|V| + |E|$ minimal, and let x be a vertex of the polytope determined by (25.24) that is not in the matchable set polytope.

Then $x_v > 0$ for each vertex v, since otherwise we can obtain a smaller counterexample by deleting v. Moreover, there exists at least one vertex v with $x_v < 1$, since otherwise $x = \chi^V$, while V is matchable (as follows from Tutte's theorem, using (25.24)(ii)).

Hence, since x is a vertex of the polytope determined by (25.24), at least one constraint in (25.24)(ii) is attained with equality for some U with $o(G[U]) \geq 1$ (for any other U, (ii) follows from (i)).

Choose such a U with U inclusionwise minimal. Let $\mathcal{K}$ be the collection of components of $G[U]$. Then

(25.25) $G[K]$ is factor-critical for each $K \in \mathcal{K}$.

Otherwise, if K is even, then

(25.26)
$$\begin{aligned} & x(U \setminus K) - x(N(U \setminus K)) \geq x(U) - x(K) - x(N(U)) \\ & \geq x(U) - |K| - x(N(U)) = |U| - o(G[U]) - |K| \\ & = |U \setminus K| - o(G[U \setminus K]), \end{aligned}$$

contradicting the minimality of U.

So K is odd. If $G[K]$ is not factor-critical, then by Tutte's 1-factor theorem, K has a nonempty subset C with $o(G[K] - C) \geq |C| + 1$. Then

(25.27)
$$\begin{aligned} & x(U \setminus C) - x(N(U \setminus C)) \geq x(U) - 2x(C) - x(N(U)) \\ & = |U| - o(G[U]) - 2x(C) \geq |U| - o(G[U]) - 2|C| \\ & = |U \setminus C| - o(G[U]) - |C| \geq |U \setminus C| - o(G[U]) - o(G[K \setminus C]) + 1 \\ & = |U \setminus C| - o(G[U \setminus C]). \end{aligned}$$

So we have equality by (25.24)(ii), contradicting the minimality of U. This shows (25.25).

Let $S := U \cup N(U)$. Let $G' := G - S$ and let x' be the restriction of x to $V \setminus S$. Then x' satisfies (25.24) with respect to G'. Indeed, (i) is trivial. To see (ii), choose a subset $U' \subseteq V \setminus S$. Then (since no edge connects U and U'):

$$
\begin{aligned}
(25.28)\qquad & x'(U') - x'(N_{G'}(U')) = x(U') - x(N(U') \setminus S) \\
& = x(U \cup U') - x(N(U \cup U')) - (x(U) - x(N(U))) \\
& \leq |U| + |U'| - o(G[U \cup U']) - (|U| - o(G[U])) = |U'| - o(G'[U']),
\end{aligned}
$$

as required.

Hence, by the minimality of G, x' belongs to the matchable set polytope of G'. Hence we are done if we have shown that the restriction of x to $G[S]$ belongs to the matchable set polytope of G.

Let H be the bipartite graph obtained from $G[S]$ by deleting all edges spanned by $N(U)$ and by contracting each $K \in \mathcal{K}$ to one vertex, u_K say. Define y on the vertices of H by: $y(v) := x(v)$ if $v \in N(U)$ and $y(u_K) := x(K) - |K| + 1$ for $K \in \mathcal{K}$.

Then y belongs to the matchable set polytope of H. To see this, we apply Theorem 21.30. Trivially $0 \leq y(v) \leq 1$ for each $v \in N(U)$. Moreover, $y(u_K) \geq 0$ for each $K \in \mathcal{K}$, since otherwise $x(K) < |K| - 1$ implying

$$
\begin{aligned}
(25.29)\qquad & x(U \setminus K) - x(N(U \setminus K)) \geq x(U) - x(K) - x(N(U)) \\
& = |U| - o(G[U]) - x(K) > |U| - o(G[U]) - |K| + 1 \\
& = |U \setminus K| - o(G[U \setminus K]),
\end{aligned}
$$

contradicting (25.24)(ii). The inequality $y(u_K) \leq 1$ follows from the fact that $x(K) \leq |K|$.

Now

$$
(25.30)\qquad \sum_{K \in \mathcal{K}} y(u_K) = x(U) - |U| + |\mathcal{K}| = x(N(U)) = \sum_{v \in N(U)} y(v).
$$

This implies, by Theorem 21.30, that if y is not in the matchable set polytope of H, then there exists a subcollection $\mathcal{L}$ of $\mathcal{K}$ with

$$
(25.31)\qquad y(N(U')) < \sum_{K \in \mathcal{L}} y(u_K),
$$

where $U' := \bigcup \mathcal{L}$. However, by (25.24) we have

$$
\begin{aligned}
(25.32)\qquad & \sum_{K \in \mathcal{L}} y(u_K) = \sum_{K \in \mathcal{L}} (x(K) - |K| + 1) = x(U') - |U'| + |\mathcal{L}| \\
& = x(U') - |U'| + o(G[U']) \leq x(N(U')) = y(N(U')).
\end{aligned}
$$

So y belongs to the matchable set polytope of H. Assuming that the restriction of x to S does not belong to the matchable set polytope of $G[S]$, there exists a vector $w \in \mathbb{R}^V$ with $w^{\mathsf{T}}x > w(Y)$ for each matchable set Y of $G[S]$ and with $w(v) = 0$ if $v \notin S$. For each $K \in \mathcal{K}$, let $v_K \in K$ minimize $w(v)$ over K. Define w' on the vertices of H by: $w'(v) := w(v)$ for $v \in N(U)$ and $w'(u_K) := w(v_K)$ for $K \in \mathcal{K}$. Since y belongs to the matchable set polytope of H, H has a matchable set Y' satisfying $w'(Y') \geq {w'}^{\mathsf{T}}y$. Let Y be the union of Y', of all K with $u_K \in Y'$, and of all $K \setminus \{v_K\}$. Since each $G[K]$ is factor-critical, Y is matchable. Moreover,

$$
\begin{aligned}
(25.33)\qquad & w(Y) = w'(Y') + \sum_{K \in \mathcal{K}} w(K \setminus \{v_K\}) \geq {w'}^{\mathsf{T}}y + \sum_{K \in \mathcal{K}} w(K \setminus \{v_K\}) \\
& = \sum_{v \in N(U)} w(v)x(v) + \sum_{K \in \mathcal{K}} w(v_K)(x(K) - |K| + 1) + \sum_{K \in \mathcal{K}} w(K \setminus \{v_K\}) \\
& \geq \sum_{v \in N(U)} w(v)x(v) + \sum_{K \in \mathcal{K}} \sum_{v \in K} (w(v) - w(v_K) + w(v_K)x(v)) \\
& \geq \sum_{v \in N(U)} w(v)x(v) + \sum_{K \in \mathcal{K}} \sum_{v \in K} w(v)x(v) = w^{\mathsf{T}}x
\end{aligned}
$$

(the last inequality follows from $(w(v) - w(v_K))(1 - x(v)) \geq 0$), contradicting our assumption. ∎

Cunningham and Green-Krótki [1994] gave a combinatorial, polynomial-time separation algorithm for the matchable set polytope that implies a proof of Theorem 25.7. A combinatorial, strongly polynomial-time algorithm was given by Cunningham and Geelen [1996,1997]. Qi [1987] characterized adjacency of vertices on the matchable set polytope. Related work can be found in Barahona and Mahjoub [1994a].

25.5e. Further notes

We postpone a discussion of the dimension of the perfect matching polytope to Chapter 37.

Note that Edmonds' matching polytope theorem gives the linear inequalities determining the convex hull of all *symmetric* permutation matrices.

Hoffman and Oppenheim [1978] showed that for each graph $G = (V, E)$ and for each vertex x of the matching polytope of G, there exist $|E|$ linearly independent constraints among (25.6) satisfied by x with equality and yielding a matrix of determinant ± 1. This also implies the total dual integrality of the constraints (25.6).

Unlike in the bipartite case, the convex hull of incidence vectors of edge sets containing a perfect matching is not determined by linear inequalities with $0, 1$ coefficients (in the left-hand side), as was shown by Cunningham and Green-Krótki [1986]. They showed that for each integer $n > 0$ there exists a graph $G = (V, E)$ with $|V| = 2n+4$ such that the convex hull of the incidence vectors of supersets of perfect matchings has facet-inducing inequalities with coefficient set $\{0, 1, \ldots, n\}$. They also showed that for odd n a similar result holds for subsets of perfect matchings. So the polyhedra $P^{\uparrow}_{\text{perfect matching}}(G)$ and $P^{\downarrow}_{\text{perfect matching}}(G)$ are not determined by $0, 1$ inequalities.

Naddef and Pulleyblank [1981b] observed that Edmonds' perfect matching polytope theorem implies that any $(k-1)$-edge connected k-regular graph $G = (V, E)$ with an even number of vertices, is matching-covered. (This can be seen by showing that the all-$\frac{1}{k}$ vector in $\mathbb{R}^E$ belongs to the perfect matching polytope.)

Rispoli [1992] noticed that the 'monotonic diameter' of the perfect matching polytope of K_n is equal to $\lfloor \frac{n}{4} \rfloor$. So for any weight function w there is a polytopal path with monotonically increasing $w^{\mathsf{T}}x$ and leading from any vertex to a vertex maximizing $w^{\mathsf{T}}x$, of length at most $\lfloor \frac{n}{4} \rfloor$.

Chapter 26

Weighted nonbipartite matching algorithmically

In the previous chapter we gave good characterizations for the maximum-weight matching problem. In the present chapter we go over to the algorithmic side, and describe Edmonds' strongly polynomial-time algorithm for finding a minimum-weight perfect matching in any graph. It implies a strongly polynomial-time algorithm for finding a maximum-weight matching.

In this chapter, graphs can be assumed to be simple.

26.1. Introduction and preliminaries

As an extension of the cardinality matching algorithm, Edmonds [1965b] proved that also a maximum-*weight* matching can be found in strongly polynomial time. Equivalently, a minimum-weight perfect matching can be found in strongly polynomial time.

Like the cardinality matching algorithm, the weighted matching algorithm is based on shrinking sets of vertices. Unlike the cardinality matching algorithm however, for weighted matchings one has, at times, to 'deshrink' sets of vertices (the reverse operation of shrinking). For this purpose we have to keep track of the shrinking history throughout the iterations.

Let $G = (V, E)$ be a graph and let $w \in \mathbb{Q}^E$ be a weight function. We describe a strongly polynomial-time algorithm to find a minimum-weight perfect matching in G. We can assume that G has at least one perfect matching and that $w \geq \mathbf{0}$.

The algorithm is 'primal-dual'. The 'vehicle' carrying us to a minimum-weight perfect matching is a pair of a laminar[11] collection Ω of odd-size subsets of V and a function $\pi : \Omega \to \mathbb{Q}$ satisfying:

(26.1)
$$\begin{array}{lll} \text{(i)} & \pi(U) \geq 0 & \text{if } U \in \Omega \text{ and } |U| \geq 3, \\ \text{(ii)} & \displaystyle\sum_{\substack{U \in \Omega \\ e \in \delta(U)}} \pi(U) \leq w(e) & \text{for each } e \in E. \end{array}$$

[11] A collection Ω of sets is called *laminar* if $U \cap W = \emptyset$ or $U \subseteq W$ or $W \subseteq U$ for any $U, W \in \Omega$.

Condition (26.1) implies

$$w(M) \geq \sum_{U\in\Omega} \pi(U) \tag{26.2}$$

for each perfect matching M in G, since

$$\begin{aligned} w(M) &= \sum_{e\in M} w(e) \geq \sum_{e\in M} \sum_{\substack{U \in \Omega \\ e \in \delta(U)}} \pi(U) = \sum_{U\in\Omega} \pi(U)|M \cap \delta(U)| \\ &\geq \sum_{U\in\Omega} \pi(U). \end{aligned} \tag{26.3}$$

Hence M is a minimum-weight perfect matching if equality holds throughout in (26.3).

Notation. Let be given Ω and $\pi : \Omega \to \mathbb{Q}$. Define for any edge e:

$$w_\pi(e) := w(e) - \sum_{\substack{U \in \Omega \\ e \in \delta(U)}} \pi(U). \tag{26.4}$$

So (26.1)(ii) says that $w_\pi(e) \geq 0$ for each $e \in E$. Let E_π denote the set of edges e with $w_\pi(e) = 0$, and let $G_\pi = (V, E_\pi)$.

Throughout the algorithm we will have that $\{v\} \in \Omega$ for each $v \in V$. Hence, as Ω is laminar, the collection $\Omega^{\max}$ of inclusionwise maximal sets in Ω is a partition of V.

By G' we denote the graph obtained from G_π by shrinking all sets in $\Omega^{\max}$:

$$G' := G_\pi/\Omega^{\max}. \tag{26.5}$$

(So G' depends on Ω and π.) The vertex set of G' is $\Omega^{\max}$, with two distinct elements $U, U' \in \Omega^{\max}$ adjacent if and only if G_π has an edge connecting U and U'. We denote any edge of G' by the original edge in G.

Finally, for $U \in \Omega$ with $|U| \geq 3$, we denote by H_U the graph obtained from $G_\pi[U]$ by contracting each inclusionwise maximal proper subset of U that belongs to Ω.

26.2. Weighted matching algorithm

We keep a laminar collection Ω of odd-size subsets of V, a function $\pi : \Omega \to \mathbb{Q}$ satisfying (26.1), a matching M in G', and for each $U \in \Omega$ with $|U| \geq 3$, a Hamiltonian circuit C_U in H_U. We assume that G is simple and has at least one perfect matching.

Initially, we set $\Omega := \{\{v\} \mid v \in V\}$, $\pi(\{v\}) := 0$ for each $v \in V$, and $M := \emptyset$. The iteration is as follows. Let X be the set of vertices of G' missed by M. (In the algorithm, 'positive length' means: having at least one edge.)

(26.6) **Case 1: G' has an M-alternating $X-X$ walk of positive length.** Choose a shortest such walk P. If P is a path, it is an M-augmenting path in G'. Reset $M := M \triangle EP$ (*matching augmentation*) and iterate.

If P is not a path, it contains an M-flower (Theorem 24.3). Let C be the circuit in it. Add $U := \bigcup VC$ to Ω (*shrinking*), set $\pi(U) := 0$, $M := M \setminus EC$, and $C_U := C$, and iterate.

Case 2: G' has no M-alternating $X-X$ walk of positive length. Let $\mathcal{S}$ be the set of vertices U of G' for which G' has an odd-length M-alternating $X-U$ walk and let $\mathcal{T}$ be the set of vertices U of G' for which G' has an even-length M-alternating $X-U$ walk. Reset $\pi(U) := \pi(U) + \alpha$ if $U \in \mathcal{T}$ and $\pi(U) := \pi(U) - \alpha$ if $U \in \mathcal{S}$, where α is the largest value maintaining (26.1). If after this resetting $\pi(U) = 0$ for some $U \in \mathcal{S}$ with $|U| \geq 3$, delete U from Ω (*deshrinking*), extend M by the perfect matching of $C_U - v$, where v is the vertex of C_U covered by M, and iterate.

In Case 2, α is bounded, since $|\mathcal{T}| > |\mathcal{S}|$ if M is not perfect and since by (26.3), $\sum_{U \in \Omega} \pi(U)$ is bounded (as there exists at least one perfect matching by assumption).

The iterations stop if M is a perfect matching in G', and then we are done: using the C_U we can expand M to a perfect matching N in G with $w_\pi(N) = 0$ and $|N \cap \delta(U)| = 1$ for each $U \in \Omega$. Then N has equality throughout in (26.3), and hence it is a minimum-weight perfect matching.

As for estimating the number of iterations, it is good to observe that the laminarity of Ω implies (cf. Theorem 3.5)

(26.7) $|\Omega| \leq 2|V|$,

assuming $V \neq \emptyset$.

Theorem 26.1. *There are at most* $2|V|^2$ *iterations.*

Proof. There are at most $\frac{1}{2}|V|$ matching augmentations, since at each matching augmentation the size of X decreases by 2, and remains unchanged in any other iteration.

The further proof is based on the following observation:

(26.8) Any set U added to Ω ('shrinking') will not be removed from Ω ('deshrinking') before the next matching augmentation.

Indeed, after shrinking U, there exists an even-length M-alternating $X-U$ path. Until the next matching augmentation, this remains the case, or U is swallowed by a larger set that is shrunk. So U is not in $\mathcal{S}$ before the next matching augmentation, proving (26.8).

Consider any sequence of iterations between two consecutive matching augmentations. By (26.8), the number of deshrinkings is not more than the size of Ω at the start of the sequence. Similarly by (26.8), the number of shrinkings is not more than the size of Ω at the end of the sequence. So, by (26.7), both the number of shrinkings and the number of deshrinkings are at most $2|V|$.

If in Case 2 we do not deshrink, then there is an edge e connecting a vertex $U \in \mathcal{T}$ with a vertex $W \notin \mathcal{S}$ of G' for which $w_\pi(e)$ has decreased to 0. If $W \notin \mathcal{T}$, then after resetting π, $W \in \mathcal{S}$, and hence the number of vertices of G' not in $\mathcal{S} \cup \mathcal{T}$ decreases. If $W \in \mathcal{T}$, then, in the next iteration, Case 1 applies. So the number of Case 2 iterations in which we do not deshrink is at most $|V|$. This proves the theorem. ∎

This gives the theorem of Edmonds [1965b]:

Corollary 26.1a. *A minimum-weight perfect matching can be found in time $O(n^2m)$.*

Proof. By Theorem 26.1, since each iteration can be performed in time $O(m)$. ∎

This implies that also a maximum-weight matching can be found in time $O(n^2m)$:

Corollary 26.1b. *A maximum-weight matching can be found in time $O(n^2m)$.*

Proof. Let $G = (V, E)$ be a graph with weight function $w \in \mathbb{Q}^E$. Extend G as follows. Make copies G' and w' of G and w. Connect each $v \in V$ to its copy in V', by an edge of weight 0. Let M be a maximum-weight perfect matching in the extended graph. The restriction of M to the original edges is a maximum-weight matching in G. ∎

Notes. In fact, a bound of $\frac{3}{2}|V|$ can be shown in (26.7) (as the size of any set in Ω is odd), implying a bound of $|V|^2$ on the number of iterations in Theorem 26.1.

26.2a. An $O(n^3)$ algorithm

In the above description, we estimated the time required for any iteration by $O(m)$. This leaves time to find the walk in each iteration just from scratch, and to construct the graph $G' = G_\pi/\Omega$ from scratch, after any shrinking or deshrinking step.

Like in the cardinality case, we can speed this up (i) by using the result of the previous walk-search in the next walk-search, and (ii) by constructing the graph G' only in an implicit way. In this way we can reduce the time per iteration from $O(m)$ to $O(n)$ on average, leading to an overall time bound of $O(n^3)$.

Again we use M-alternating forests to reach this goal. Thus, next to Ω, π, M, and the C_U, we keep an M-alternating forest F in $G' := G_\pi/\Omega^{\max}$.

We do not keep the graph G'. Instead, we keep for each pair Y, Z of disjoint sets in Ω an edge e_{YZ} of G connecting Y and Z and minimizing $w_\pi(e_{YZ})$. We take e_{YZ} void if no such edge exists. We keep the e_{YZ} as lists: for each $Y \in \Omega$ we have a list containing the e_{YZ}.

Moreover, for each $Y \in \Omega$ we keep an edge e_Y with $e_Y = e_{YZ}$ for some $Z \in$ even(F) and with $w_\pi(e_{YZ})$ minimal. Again, if no such e_{YZ} exists, e_Y is void.

Finally, for each $v \in V$ we keep

$$p(v) := \sum_{\substack{U \in \Omega \\ v \in U}} \pi(U). \tag{26.9}$$

Initially, we set $\Omega := \{\{v\} \mid v \in V\}$, $\pi(\{v\}) := 0$ and $p(v) := 0$ for each $v \in V$, and $M := \emptyset$, $F := \emptyset$. The e_{YZ} and e_Y are easily set.

Next we apply the following iteratively:

(26.10) Reset $\pi(U) := \pi(U) - \alpha$ for $U \in \text{odd}(F)$ and $\pi(U) := \pi(U) + \alpha$ for $U \in \text{even}(F)$, where α is the largest value maintaining (26.1). Update p accordingly. After that, at least one of the following three cases applies.

Case 1: $w_\pi(e_U) = 0$ for some $U \in$ free(F). Extend F by e_U and update the e_Y (*forest augmentation*).

Case 2: $w_\pi(e_U) = 0$ for some $U \in$ even(F). Let e_U connect vertices U and W in even(F). Let P and Q be the $X - U$ and the $X - W$ path in $(\Omega^{\max}, F)$, respectively.

Case 2a: Paths P and Q are disjoint. Then P and Q form with e_U an M-augmenting path, yielding a matching M' in G' with $|M'| = |M| + 1$. Reset $M := M'$, $F := M'$, and update the e_Y (*matching augmentation*).

Case 2b: Paths P and Q intersect. Then they contain (with e_U) an M-blossom B. Let T be the union of the sets (in $\Omega^{\max}$) forming the vertices of B. Add T to Ω, setting $C_T := B$ and $\pi(T) := 0$. Reset $F := F \setminus EB$ and $M \setminus EB$, and update the e_{YZ} and e_Y (*shrinking*).

Case 3: $\pi(U) = 0$ for some $U \in$ odd(F) with $|U| \geq 3$. Let v be the vertex in C_U covered by an edge in M and let u be the vertex in C_U covered by an edge in $F \setminus M$. Let P be the even-length $u - v$ path in C_U and let N be the matching in $C_U - v$. Delete U from Ω, reset $F := F \cup EP \cup N$ and $M := M \cup N$, and update the e_{YZ} and e_Y (*deshrinking*).

(In updating F and M, we update them as graphs on $\Omega^{\max}$.)

The number of iterations between any two matching augmentations is at most $|V|$, as may be proved similarly to the proof of Theorem 26.1 (replacing $\mathcal{S}$ by odd(F) and $\mathcal{T}$ by even(F)).

In the iteration (26.10), we can find the value α in $O(n)$ time, as it is the minimum of $w_\pi(e_U)$ over $U \in$ free(F), of $\frac{1}{2}w_\pi(e_U)$ over $U \in$ even(F), and of $\pi(U)$ over $U \in$ odd(F) with $|U| \geq 3$. So we can update π and p in $O(n)$ time. Also F and M can be updated in $O(n)$ time (as they have $O(n)$ edges).

Note that each time we need the value of $w_\pi(e)$ for some edge e (when determining α or the e_{YZ} and e_Y), then e connects two disjoint sets in $\Omega^{\max}$, and hence $w_\pi(e) = w(e) - p(u) - p(v)$. Note also that the resetting of π on $\Omega^{\max}$ changes no e_{YZ} and e_Y.

In Case 1, Ω, π, p, and the e_{YZ} are unchanged. The set $U \in \Omega^{\max}$ is moved from free(F) to odd(F), and a set $W \in \Omega^{\max}$ (the mate of U in M) is moved from free(F) to even(F). To update the e_Y, it suffices to scan the list of the e_{WZ}. This can be done in $O(n)$ time.

In Case 2a, Ω, π, p, and the e_{YZ} are unchanged. Since (in the new situation) $F = M$, we delete from even(F) and odd(F) all sets in $\Omega^{\max}$ covered by M. We can find the e_Y by scanning all e_{YZ}. We have $O(n^2)$ time for this, since there are only $\frac{1}{2}|V|$ matching augmentations.

In Case 2b, set T is inserted into $\Omega^{\max}$ and into even(F), and the sets in VB are removed from even(F) and odd(F). We need to find the e_{TZ}, which can be done by scanning the e_{YZ} for each $Y \in VB$. At the same time, the e_Z can be updated. This can be done in $O(|VB|n)$ time.

In Case 3, set U is removed from $\Omega^{\max}$ and from odd(F), and the sets in VC_U become members of $\Omega^{\max}$ and are inserted into even(F) or odd(F). This modifies no e_{YZ} (except that all e_{UZ} disappear). By scanning the e_{YZ} for each $Y \in VC_U$, we can update the e_Z. This can be done in $O(|VC_U|n)$ time.

Now, between any two matching augmentations, the sum of the $|VC_U|$ over the U added or removed is $O(n)$, since any set added will not be removed before the next matching augmentation (cf. (26.8)). So between any two matching augmentations, the iterations can be done in $O(n^2)$ time.

This gives the result of Gabow [1973] and Lawler [1976b]:

Theorem 26.2. *A minimum-weight perfect matching can be found in $O(n^3)$ time.*

Proof. See above. ∎

Several ingredients in this method can be implemented so as to require only $O(m)$ time between any two matching augmentations. However, reducing the time needed to administer Ω requires additional data structure — see the references in Section 26.3a.

26.3. Further results and notes

26.3a. Complexity survey for weighted nonbipartite matching

Complexity survey for weighted nonbipartite matching ($*$ indicates an asymptotically best bound in the table):

$O(n^4)$	Edmonds [1965b]
$O(n^3)$	Gabow [1973], Lawler [1976b]
$O(nm \log n)$	Galil, Micali, and Gabow [1982, 1986] (cf. Ball and Derigs [1983])
$O(n(m \log\log\log_{m/n} n + n \log n))$	Gabow, Galil, and Spencer [1984, 1989]
$O(n^{3/4} m \log W)$	Gabow [1985a,1985b]

≫

continued

*	$O(n(m + n \log n))$	Gabow [1990]
*	$O(m \log(nW) \sqrt{n\alpha(m,n) \log n})$	Gabow and Tarjan [1991]

Here W is the maximum absolute value of the weights, assuming they are integer.

Cunningham and Marsh [1978] gave a *primal* algorithm for weighted nonbipartite matching that takes $O(n^2 m)$ time (where, throughout the algorithm, there is a perfect matching at hand, the weight of which is improved iteratively). They state that it can be improved to $O(n^3)$. Derigs [1981] gave a shortest augmenting path method of running time $O(n^3)$. In Derigs [1988b] an $O(\min\{n^3, nm \log n\})$ algorithm is given based on successive improvement of a perfect matching by choosing an improving alternating circuit.

26.3b. Derivation of the matching polytope characterization from the algorithm

Edmonds' weighted matching algorithm directly yields the description of the perfect matching polytope. Indeed, one can derive from Edmonds' algorithm the following. Let $G = (V, E)$ be a graph and let $w \in \mathbb{Q}^E$ be a weight function. Then:

(26.11) the minimum weight of a perfect matching is equal to the maximum value of $\sum_{U \in \mathcal{P}_{\text{odd}}(V)} \pi(U)$ where π ranges over all functions $\pi : \mathcal{P}_{\text{odd}}(V) \to \mathbb{Q}$ satisfying (26.1),

where $\mathcal{P}_{\text{odd}}(V)$ denotes the collection of odd-size subsets of V.

To see this, we may assume that w is nonnegative: if μ is the minimum value of $w(e)$ over all edges e, decreasing each $w(e)$ by μ decreases both the maximum and the minimum by $\frac{1}{2}|V|\mu$.

That the minimum is not smaller than the maximum follows from (26.3). Equality follows from the fact that in the algorithm the final perfect matching and the final function π have equality throughout in (26.1). This shows (26.11).

It implies Edmonds' perfect matching polytope theorem: the perfect matching polytope of any graph $G = (V, E)$ is determined by (25.2). Indeed, by (weak) LP-duality, for any weight function $w \in \mathbb{Q}^E$, the minimum weight of a perfect matching is equal to the minimum of $w^{\mathsf{T}} x$ taken over the polytope determined by (25.2). Hence the two polytopes coincide.

26.3c. Further notes

Weber [1981] and Derigs [1985a] analyzed the sensitivity of minimum-weight perfect matchings to changing edge weights. White [1974] studied the maximum weight of a matching of size k, as a function of k.

An outstanding open problem is to formulate the weighted matching problem as a linear programming problem of size polynomial in the size of the graph, by extending the set of variables. That is, is the matching polytope of a graph $G = (V, E)$ equal to the projection of some polytope $\{x \mid Ax \le b\}$ with A and b having size polynomial in $|V| + |E|$?

Yannakakis [1988,1991] showed that this is not possible in a symmetric fashion. (That is, for $G = K_n$ there is not a system $Ax \leq b$ which is invariant under each permutation of the vertex set.) For further partial results, see Yannakakis [1988, 1991], Gerards [1991], and Barahona [1993a,1993b].

Gabow, Kaplan, and Tarjan [1999,2001] gave fast algorithms to test uniqueness of a minimum-weight perfect matching.

For heuristics and fast approximation methods for the weighted matching problem if the weight function satisfies the triangle inequality (including matching points in Euclidean space), see Papadimitriou [1977b], Avis [1978,1981,1983], Supowit, Plaisted, and Reingold [1980], Iri, Murota, and Matsui [1981,1983], Reingold and Tarjan [1981], Bartholdi and Platzman [1983], Reingold and Supowit [1983], Supowit and Reingold [1983], Supowit, Reingold, and Plaisted [1983], Plaisted [1984], Grigoriadis and Kalantari [1986,1988], Grigoriadis, Kalantari, and Lai [1986], Imai [1986], Weber and Liebling [1986], Avis, Davis, and Steele [1988], Vaidya [1988, 1989a,1989b], Kalyanasundaram and Pruhs [1991,1993], Marcotte and Suri [1991], Goemans and Williamson [1992,1995a], Osiakwan and Akl [1994], Williamson and Goemans [1994], Jünger and Pulleyblank [1995], Arora [1997,1998], Varadarajan [1998], and Varadarajan and Agarwal [1999].

For studies of implementing weighted matching algorithms, see Cunningham and Marsh [1978], Burkard and Derigs [1980], Derigs [1981,1986a,1986b,1988b], Lessard, Rousseau, and Minoux [1989], Derigs and Metz [1991], Applegate and Cook [1993], and Cook and Rohe [1999].

Grötschel and Holland [1985] report on implementing a cutting plane algorithm for the weighted matching problem based on the simplex method (cf. Derigs and Metz [1991]). For an alternative approach, see Lessard, Rousseau, and Minoux [1989]. Derigs and Metz [1986b] showed how solving the matching problem fractionally can help in finding a shortest augmenting path.

Megiddo and Tamir [1978] gave an $O(n \log n)$ algorithm to find a maximum-weight matching in a graph $G = (V, E)$, if each weight $w(uv)$ is equal to $a(u) + b(v)$ for $u < v$, where the vertices are ordered by $<$ and where $a, b : V \to \mathbb{Q}$.

For weighted matching problems with side constraints, see Ball, Derigs, Hilbrand, and Metz [1990].

For a survey on weighted matching algorithms, see Galil [1983,1986a,1986b]. Books covering weighted nonbipartite matching algorithms include Christofides [1975], Lawler [1976b], Minieka [1978], Papadimitriou and Steiglitz [1982], Gondran and Minoux [1984], Derigs [1988a], Nemhauser and Wolsey [1988], Cook, Cunningham, Pulleyblank, and Schrijver [1998], Jungnickel [1999], and Korte and Vygen [2000].

Chapter 27

Nonbipartite edge cover

Edge cover is closely related to matching, through a construction described by Gallai. In this chapter we derive basic results on edge covers (min-max relation, polyhedral characterization, strongly polynomial-time algorithm) from the results on matchings given in the previous chapters.
In this chapter, graphs can be assumed to be loopless.

27.1. Minimum-size edge cover

With Gallai's theorem, the Tutte-Berge formula implies a formula for the edge cover number $\rho(G)$ (where $o(G[U])$ denotes the number of odd components of $G[U]$):

Theorem 27.1. *Let $G = (V, E)$ be a graph without isolated vertices. Then*

$$\rho(G) = \max_{U \subseteq V} \frac{|U| + o(G[U])}{2}. \tag{27.1}$$

Proof. By Gallai's theorem (Theorem 19.1) and the Tutte-Berge formula (Theorem 24.1),

$$\begin{aligned} \rho(G) &= |V| - \nu(G) = |V| - \min_{U \subseteq V} \frac{|V| + |U| - o(G - U)}{2} \\ &= \max_{U \subseteq V} \frac{|U| + o(G[U])}{2}. \end{aligned} \tag{27.2}$$ ∎

This min-max relation is equivalent to: $\rho(G)$ is equal to the maximum value of

$$\sum_{U \in \mathcal{U}} \lceil \tfrac{1}{2}|U| \rceil, \tag{27.3}$$

where $\mathcal{U}$ is a collection of disjoint odd subsets of V such that no edge of G connects two distinct sets in $\mathcal{U}$.

By the method of Gallai's theorem, one can derive a minimum-size edge cover from a maximum-size matching M, just by adding for each vertex v

missed by M, an arbitrary edge incident with v. Hence a minimum-size edge cover can be found in polynomial time.

One can reduce the problem of finding a minimum-weight edge cover to that of finding a minimum-weight perfect matching, as described in Section 19.3. It gives the following result of Edmonds and Johnson [1970]:

Theorem 27.2. *A minimum-weight edge cover can be found in $O(n^3)$ time.*

Proof. From Corollary 26.1b, with the method of Section 19.3. ∎

27.2. The edge cover polytope and total dual integrality

The *edge cover polytope* of a graph $G = (V, E)$ is the convex hull of the incidence vectors of edge covers. We will show that the edge cover polytope is determined by

(27.4)
$$\begin{array}{lll} \text{(i)} & 0 \leq x_e \leq 1 & \text{for each } e \in E, \\ \text{(ii)} & x(E[U] \cup \delta(U)) \geq \lceil \frac{1}{2}|U| \rceil & \text{for each } U \subseteq V \text{ with } |U| \text{ odd,} \end{array}$$

and moreover, that this system is totally dual integral. The latter statement will be derived from the Cunningham-Marsh formula (Theorem 25.2), and is equivalent to:

Theorem 27.3. *Let $G = (V, E)$ be a graph without isolated vertices and let $w \in \mathbb{Z}_+^E$ be a weight function. Then the minimum weight of an edge cover is equal to the maximum value of*

(27.5)
$$\sum_{U \in \mathcal{P}_{\text{odd}}(V)} z_U \lceil \tfrac{1}{2}|U| \rceil,$$

where $z_U \in \mathbb{Z}_+$ for each $U \in \mathcal{P}_{\text{odd}}(V)$ such that

(27.6)
$$\sum_{U \in \mathcal{P}_{\text{odd}}(V)} z_U \chi^{E[U] \cup \delta(U)} \leq w.$$

Proof. We first show:

(27.7) in the Cunningham-Marsh formula one can assume that for each $v \in V$ there is an edge $e \in \delta(v)$ with $y_v + \sum_{U \ni v} z_U \leq w(e)$.

Indeed, by Theorem 25.2 we can take y, z such that $\mathcal{F} := \{U \mid z_U > 0\}$ is laminar. Now choose $v \in V$. Suppose that $y_v + \sum_{U \ni v} z_U > w(e)$ for each edge $e \in \delta(v)$. If no set in $\mathcal{F}$ covers v, then reducing y_v by 1 would maintain the conditions, contradicting the fact that y, z attain the minimum in the Cunningham-Marsh formula.

So some $T \in \mathcal{F}$ covers v. Choose an inclusionwise minimal set $T \in \mathcal{F}$ covering v. As $\mathcal{F}$ is laminar, $U \supseteq T$ for each $U \in \mathcal{F}$ containing v. Then for each edge $e = uv$ with $v \in e \subseteq T$ one has for each $U \in \mathcal{F}$: if $v \in U$, then $e \subseteq U$. So for each such edge $e = uv$,

$$y_u + y_v + \sum_{U \supseteq e} z_U \geq y_v + \sum_{U \ni v} z_U > w(e). \tag{27.8}$$

Hence, if we choose $s \in T \setminus \{v\}$, then decreasing z_T by 1 and increasing y_s and $z_{T \setminus \{v,s\}}$ by 1, gives again an optimum solution. Iterating this for all v, gives a solution as in (27.7).

We next show the theorem. For each vertex v, let e_v be an edge incident with v of minimum weight and let $\mu(v) := w(e_v)$. For each edge $e = uv$, define $w'(e) := \mu(u) + \mu(v) - w(e)$.

By the Cunningham-Marsh formula, there exists a matching M and $y_v \in \mathbb{Z}_+$ $(v \in V)$ and $z'_U \in \mathbb{Z}_+$ $(U \in \mathcal{P}_{\text{odd}}(V))$ such that

$$\begin{aligned} &\text{(i) } y_u + y_v + \sum_{U \supseteq e} z'_U \geq w'(e) \text{ for each edge } e = uv; \\ &\text{(ii) } w'(M) = \sum_{v \in V} y_v + \sum_{U \in \mathcal{P}_{\text{odd}}(V)} z'_U \lfloor \tfrac{1}{2}|U| \rfloor. \end{aligned} \tag{27.9}$$

We may assume that $z'_U = 0$ if $|U| = 1$. By (27.7) we may assume that for each $v \in V$:

$$y_v + \sum_{T \ni v} z'_T \leq w'(e) \tag{27.10}$$

for some edge e incident with v.

Let F be the edge cover obtained from M by adding the edge e_v for each vertex v missed by M. For each $U \in \mathcal{P}_{\text{odd}}(V)$, define:

$$z_U := \begin{cases} \mu(v) - y_v - \sum_{T \ni v} z'_T & \text{if } U = \{v\}, \\ z'_U & \text{if } |U| \geq 3. \end{cases} \tag{27.11}$$

Clearly $z_U \geq 0$ if $|U| \geq 3$. If $U = \{v\}$, then let $e = uv \in \delta(v)$ satisfy satisfying (27.10). Hence

$$z_{\{v\}} = \mu(v) - y_v - \sum_{T \ni v} z'_T \geq \mu(v) - w'(e) = w(e) - \mu(u) \geq 0. \tag{27.12}$$

So z is nonnegative.

Now for each edge $e = uv$ one has:

$$\begin{aligned} \sum_{U \cap e \neq \emptyset} z_U &= z_{\{u\}} + z_{\{v\}} + \sum_{U \cap e \neq \emptyset} z'_U \\ &= \mu(u) - y_u - \sum_{U \ni u} z'_U + \mu(v) - y_v - \sum_{U \ni v} z'_U + \sum_{U \cap e \neq \emptyset} z'_U \\ &= \mu(u) + \mu(v) - y_u - y_v - \sum_{U \supseteq e} z'_U \leq \mu(u) + \mu(v) - w'(e) \\ &= w(e). \end{aligned} \tag{27.13}$$

Moreover,

$$\begin{aligned}(27.14)\qquad \sum_U z_U\lceil\tfrac{1}{2}|U|\rceil &= \sum_{v\in V}(\mu(v)-y_v-\sum_{U\ni v}z'_U)+\sum_U z'_U\lceil\tfrac{1}{2}|U|\rceil\\ &=\sum_{v\in V}\mu(v)-\sum_{v\in V}y_v-\sum_U z'_U\lfloor\tfrac{1}{2}|U|\rfloor=\sum_{v\in V}\mu(v)-w'(M)=w(F).\end{aligned}$$

∎

(The idea of using w' was given by J.F. Geelen.)

Equivalently, we can state:

Corollary 27.3a. *System* (27.4) *determines the edge cover polytope and is TDI.*

Proof. This is equivalent to Theorem 27.3. ∎

27.3. Further notes on edge covers

27.3a. Further notes

Inspired by Edmonds' algorithm for maximum-weight matching, White [1967] and Murty and Perin [1982] described minimum-weight edge cover algorithms based on blossoms.

White and Gillenson [1975] and Murty and Perin [1982] described a blossom-type algorithm to find a minimum-weight edge cover of given size k. Also White [1971] considered the problem of finding a minimum-weight edge cover of a given size, by parametrizing the weight function.

In fact, the convex hull of incidence vectors of edge covers F with $k \leq |F| \leq l$ is equal to the edge cover polytope intersected with $\{x \in \mathbb{R}^E \mid k \leq x(E) \leq l\}$. This can be proved similarly to the proof of Corollary 18.10a.

Hurkens [1991] characterized adjacency on the edge cover polytope and derived that its diameter is equal to $|E| - \rho(G)$. (This turns out to be harder to prove than the corresponding results for the matching polytope given in Section 25.5a.)

27.3b. Historical notes on edge covers

The nonbipartite edge cover problem was considered by Gallai [1959a] and Norman and Rabin [1959]. The latter were motivated by a problem of Roth [1958] related to minimizing the number of switches in a switching systems, for which they considered the problem of finding a minimum cover for a cubical complex.

Norman and Rabin [1959] showed that an edge cover F in a graph has minimum size if and only if there is no path P such that the end vertices of P are covered more than once by F, while all intermediate vertices are covered exactly once by F, and such that the edges of P are alternatingly in and out F, with the first and last edge in F. (Thus $F\triangle P$ is an edge cover of smaller size than F.)

Chapter 28

Edge-colouring

Edge-colouring means covering the edge set by matchings. The problem goes back to Tait [1878b], who showed that the four-colour conjecture is equivalent to the 3-edge-colourability of any bridgeless cubic planar graph. Nonbipartite edge-colouring is less tractable than in the special case of bipartite graphs. No tight min-max relation is known and finding a minimum edge-colouring is NP-complete. In this chapter we prove Vizing's theorem, which gives an almost tight min-max relation. Moreover, we consider the 'fractional' edge-colouring number, which approximates the edge-colouring number. It can be characterized and computed with the help of matching results. We also consider the related problem of packing edge covers.

28.1. Vizing's theorem for simple graphs

We recall some definitions and notation. Let $G = (V, E)$ be a graph. An *edge-colouring* is a partition of E into matchings. Each matching in an edge-colouring is called a *colour* or an *edge-colour*. A *k-edge-colouring* is an edge-colouring with k colours. G is *k-edge-colourable* if a k-edge-colouring exists. The smallest k for which G is k-edge-colourable is called the *edge-colouring number* of G, denoted by $\chi'(G)$. Since an edge-colouring of G is a vertex-colouring of the line-graph $L(G)$ of G, we have that $\chi'(G) = \chi(L(G))$.

Clearly $\chi'(G) \geq \Delta(G)$, where $\Delta(G)$ denotes the maximum degree of G. We saw that $\chi'(G) = \Delta(G)$ if G is bipartite (Kőnig's edge-colouring theorem (Theorem 20.1)). On the other hand, $\chi'(G) > \Delta(G)$ if $G = K_3$. It was proved by Holyer [1981] that deciding if $\chi(G) \leq 3$ is NP-complete.

Nevertheless, $\Delta(G)$ is a good estimate of the edge-colouring number as Vizing [1964,1965a] showed the following (our proof roots in Ehrenfeucht, Faber, and Kierstead [1984]):

Theorem 28.1 (Vizing's theorem for simple graphs)**.** $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$ *for any simple graph G.*

Proof. The inequality $\Delta(G) \leq \chi'(G)$ being trivial, we show $\chi'(G) \leq \Delta(G) + 1$. To prove this inductively, it suffices to show for any simple graph G:

(28.1) Let v be a vertex such that v and all its neighbours have degree at most k, while at most one neighbour has degree precisely k. Then if $G-v$ is k-edge-colourable, also G is k-edge-colourable.

We prove (28.1) by induction on k, the case $k=0$ being trivial. We can assume that each neighbour u of v has degree $k-1$, except for one neighbour having degree exactly k, since otherwise we can add a new vertex w and an edge uw without violating the condition in (28.1).

Consider any k-edge-colouring of $G-v$. For $i=1,\ldots,k$, let X_i be the set of neighbours of v that are missed by colour i. Choose the colouring such that $\sum_{i=1}^{k}|X_i|^2$ is minimized.

First assume that $|X_i|\neq 1$ for all i. Since all but one neighbour of v is in precisely two of the X_i, and one neighbour is in precisely one X_i, we have

$$\sum_{i=1}^{k}|X_i| = 2\deg(v)-1 < 2k. \tag{28.2}$$

Hence there exist i,j with $|X_i|<2$ and $|X_j|$ odd. So $|X_i|=0$ and $|X_j|\geq 3$. Consider the subgraph H made by all edges of colours i and j, and consider a component of H containing a vertex in X_j. This component is a path P starting in X_j. Exchanging colours i and j on P reduces $|X_i|^2+|X_j|^2$, contradicting our minimality assumption.

So we can assume that $|X_k|=1$, say $X_k=\{u\}$. Let G' be the graph obtained from G by deleting edge vu and deleting all edges of colour k. So $G'-v$ is $(k-1)$-edge-coloured. Moreover, in G', vertex v and all its neighbours have degree at most $k-1$, and at most one neighbour has degree $k-1$. So by the induction hypothesis, G' is $(k-1)$-edge-colourable. Restoring colour k, and giving edge vu colour k, gives a k-edge-colouring of G. ∎

Notes. This theorem was also announced in an abstract of Gupta [1966].

The above proof implies the stronger result of Fournier [1973] that a simple graph G is $\Delta(G)$-edge-colourable if the maximum-degree vertices span no circuit (since this last condition implies that the maximum-degree vertices induce a forest as subgraph, and hence there exists a maximum-degree vertex v with at most one neighbour that has maximum degree).

Petersen [1898] gave the example of the (now-called) Petersen graph (Figure 28.1) which is 2-connected and cubic but not 3-edge-colourable. It was conjectured by Tutte [1966] that each 2-connected cubic graph without Petersen graph minor, is 3-edge-colourable. This conjecture was proved (using the 4-colour theorem) by the combined efforts of Robertson, Seymour, and Thomas [1997], Sanders, Seymour, and Thomas [2000], and Sanders and Thomas [2000].

Complexity. The proof gives a polynomial-time algorithm to find a $(\Delta+1)$-edge-colouring of a simple graph, in fact, $O(\Delta n^2)$-time. As we can assume that $\Delta n = O(m)$ (since we can merge vertices of degree at most $\frac{1}{2}\Delta$), this implies an $O(nm)$-time algorithm.

Gabow, Nishizeki, Kariv, Leven, and Terada [1985] gave algorithms finding a $(\Delta+1)$-edge-colouring of a simple graph G of maximum degree Δ, with running

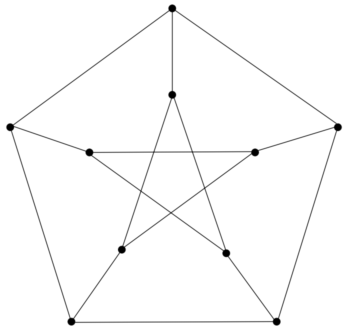


Figure 28.1
The Petersen graph

times $O(m\Delta \log n)$ and $O(m\sqrt{n \log n})$ (improving $O(nm)$ of Terada and Nishizeki [1982]).

28.2. Vizing's theorem for general graphs

In Theorem 28.1 we cannot delete the condition that G be simple: the graph G obtained from K_3 by replacing each edge by two parallel edges, has $\chi'(G) = 6$ and $\Delta(G) = 4$. However, Vizing's theorem can be extended so as to take also the nonsimple case into account. For any graph $G = (V, E)$ and $u, v \in V$, let $\mu(u, v)$ denote the number of edges connecting u and v, called the *multiplicity* of $\{u, v\}$. Let $\mu(G)$ denote the maximum of $\mu(u, v)$ over all distinct $u, v \in V$. Then Vizing [1964,1965a] showed (again, our proof roots in Ehrenfeucht, Faber, and Kierstead [1984]):

Theorem 28.2 (Vizing's theorem)**.** *$\Delta(G) \leq \chi'(G) \leq \Delta(G) + \mu(G)$ for any graph G.*

Proof. The inequality $\Delta(G) \leq \chi'(G)$ being trivial, we show $\chi'(G) \leq \Delta(G) + \mu(G)$. To prove this inductively, it suffices to show for any graph G:

(28.3) Let v be a vertex of degree at most k such that each neighbour u of v satisfies $\deg(u) + \mu(u, v) \leq k + 1$, with equality for at most one neighbour. Then if $G - v$ is k-edge-colourable, also G is k-edge-colourable.

We prove (28.3) by induction on k, the case $k = 0$ being trivial. We can assume that for each vertex u in $N(v)$ (the set of neighbours of v) we have $\deg(u) + \mu(u, v) = k$, except for one satisfying $\deg(u) + \mu(u, v) = k + 1$, since otherwise we can add a new vertex w and an edge uw without violating the condition in (28.3).

Consider any k-edge-colouring of $G - v$. For $i = 1, \ldots, k$, let X_i be the set of neighbours of v that are missed by colour i. Choose the colouring such that $\sum_{i=1}^{k} |X_i|^2$ is minimized.

First assume that $|X_i| \neq 1$ for all i. As each $u \in N(v)$ is in precisely $2\mu(u,v)$ of the X_i, except for one $u \in N(v)$ being in $2\mu(u,v) - 1$ of the X_i, we know

$$(28.4) \qquad \sum_{i=1}^{k} |X_i| = -1 + 2 \sum_{u \in N(v)} \mu(u,v) = 2\deg(v) - 1 < 2k.$$

Hence there exist i, j with $|X_i| < 2$ and $|X_j|$ odd. So $|X_i| = 0$ and $|X_j| \geq 3$. Consider the subgraph H made by all edges of colours i and j, and consider a component of H containing a vertex in X_j. This component is a path P starting in X_j. Exchanging colours i and j on P reduces $|X_i|^2 + |X_j|^2$, contradicting our minimality assumption.

So we can assume that $|X_k| = 1$, say $X_k := \{u\}$. Let G' be the graph obtained from G by deleting one of the edges vu and deleting all edges of colour k. So $G' - v$ is $(k-1)$-edge-coloured. Moreover, in G', vertex v has degree at most $k-1$ and each neighbour w of v satisfies $\deg_{G'}(w) + \mu_{G'}(w,v) \leq k$, with equality for at most one neighbour. So by the induction hypothesis, G' is $(k-1)$-edge-colourable. Restoring colour k, and giving the deleted edge vu colour k, gives a k-edge-colouring of G. ∎

Notes. The proof of Theorem 28.2 in fact implies that the edge-colouring number of a graph G is at most

$$(28.5) \qquad \max_{u \in V} \Big(\deg(u) + \max\{1, \max_{\substack{v \in V \\ \deg(v) \geq \deg(u)}} \mu(u,v)\}\Big),$$

where $\mu(u,v)$ is the number of edges connecting u and v (cf. Ore [1967]).

Other proofs of Vizing's theorem were given by Ore [1967], Fournier [1973], Berge and Fournier [1991], Misra and Gries [1992], Rao and Dijkstra [1992], and Chew [1997b].

28.3. NP-completeness of edge-colouring

Vizing's theorem gives us a close approximation to the edge-colouring number of a simple graph. The error is at most 1. However, it turns out to be NP-complete to determine the edge-colouring number precisely, even for cubic graphs, which was shown by Holyer [1981]:

Theorem 28.3. *It is* NP*-complete to decide if a given cubic graph is* 3*-edge-colourable.*

Proof. We show that the 3-satisfiability problem (3-SAT) can be reduced to the edge-colouring problem of graphs of maximum degree 3. One easily reduces this last problem to the edge-colouring problem for cubic graphs (by deleting iteratively all vertices of degree ≤ 1, next making a copy of the graph left, and adding an edge between each degree-2 vertex and its copy).

Consider the graph fragment, called the *inverting component*, given by the left-hand picture of Figure 28.2, where the right-hand picture gives its symbolic representation if we take it as part of larger graphs. The pairs a, b and c, d are called the *output pairs*.

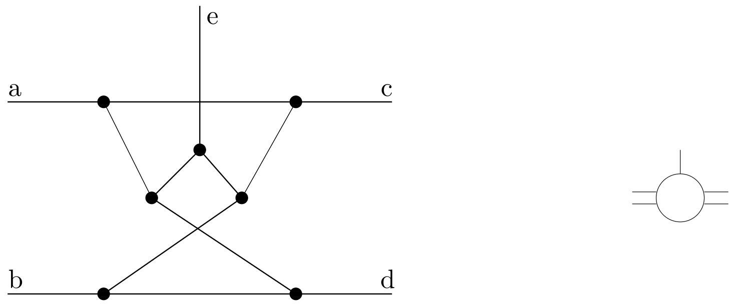


Figure 28.2

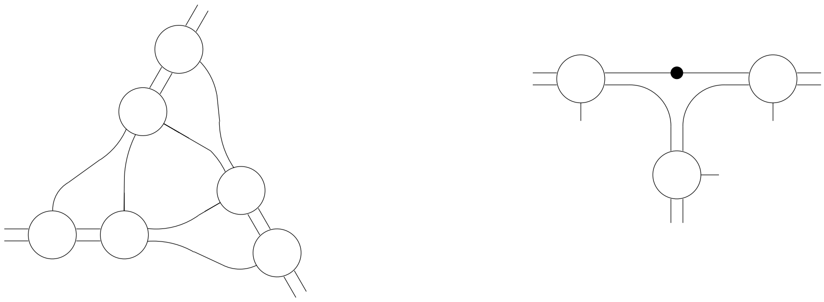
The *inverting component* and its symbolic representation.

This graph fragment has the property that a 3-colouring of the edges a, b, c, d, and e is extendible to a 3-edge-colouring of the fragment if and only if either a and b have the same colour while c, d, and e have three distinct colours, or c and d have the same colour while a, b, and e have three distinct colours.

Consider now an instance of the 3-satisfiability problem. From the inverting component we build larger graph fragments. A *splitting component* is given in Figure 28.3(a). For each variable u, occurring k times, as u or $\neg u$, we introduce a fragment Γ_u by concatenating $k-2$ splitting components. So Γ_u has k output pairs, and it has the property that in any colouring either all output pairs are monochromatic, or they all are nonmonochromatic.

For each clause C we introduce a component Δ_C given by Figure 28.3(b). If a variable u occurs in a clause C as u, we connect one of the output pairs of Γ_u with one of the output pairs of Δ_C. If a variable u occurs in a clause C as $\neg u$, we connect one of the output pairs of Γ_u with one side of an inverting component, and connect the other side of this inverting component with one of the output pairs of Δ_C.

In this way we can match up all output pairs of the Γ_u and those of the Δ_C. Deleting all loose ends, we obtain a graph G of maximum degree 3. Now, given the properties of the fragments, one easily checks that the input of the 3-satisfiability problem is satisfiable if and only if G is 3-edge-colourable. ∎



(a) (b)

Figure 28.3

Fragment (a) (the *splitting component*) has the property that for any 3-edge-colouring either all three output pairs are monochromatic or all are nonmonochromatic.

Fragment (b) has the property that a colouring of the output edges is extendible to a 3-edge-colouring of the fragment if and only if at least one of the output pairs is monochromatic.

Leven and Galil [1983] showed more generally that for each k, finding the edge-colouring number of a k-regular graph is NP-complete. (This does not seem to follow from the case $k = 3$.)

28.4. Nowhere-zero flows and edge-colouring

Let $D = (V, A)$ be a directed graph and let Γ be an additive abelian group. A *flow over* Γ is a function $f : A \to \Gamma$ such that for each $v \in V$:

(28.6) $$f(\delta^{\text{in}}(v)) = f(\delta^{\text{out}}(v)).$$

The flow is called *nowhere-zero* if all values of f are nonzero.

If G is an undirected graph, then a *flow over* Γ is a flow over Γ in some orientation of G. We say that an undirected graph G has a nowhere-zero flow over Γ if G has an orientation having a nowhere-zero flow over Γ.

Colouring the edges of an undirected graph is related to the problem of finding a nowhere-zero flow over a finite abelian group in the graph. This might be illustrated best by the following easy fact:

(28.7) a cubic graph G is 3-edge-colourable $\iff$ G has a nowhere-zero flow over $\mathrm{GF}(4)$.

Since $-x = x$ for each $x \in \mathrm{GF}(4)$, the orientation is irrelevant in this case.

Statement (28.7) implies that the four-colour theorem is equivalent to:

(28.8) each bridgeless cubic planar graph has a nowhere-zero flow over GF(4)

(since the four-colour theorem is equivalent to each bridgeless cubic planar graph being 3-edge-colourable (Tait [1878b])).

In studying nowhere-zero flows, the following theorem shows that for the existence of a nowhere-zero flow, only the size of the group is relevant (the equivalence (i)⇔(ii) was shown by Tutte [1947a], the equivalence (i)⇔(iii) by Tutte [1949], and the equivalence (iii)⇔(iv) by Minty [1967]):

Theorem 28.4. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}$ with $k \geq 1$. Then the following are equivalent:*

(28.9)
(i) *G has a nowhere-zero flow over some abelian group with precisely k elements;*
(ii) *G has a nowhere-zero flow over each abelian group with at least k elements;*
(iii) *G has a flow over $\mathbb{Z}$ taking values in the interval $[1, k-1]$ only;*
(iv) *G has an orientation $D = (V, A)$ with $d_A^{\text{in}}(U) \geq \frac{1}{k} d_E(U)$ for each $U \subseteq V$.*

Proof. The implication (ii)⇒(i) is trivial, while the implication (iii)⇒(i) is easy, by considering the integer values of (iii) as values in the group of integers mod k.

For any graph $G = (V, E)$ and any finite abelian group Γ, let $\phi_\Gamma(G)$ denote the number of nowhere-zero flows over Γ in G. Then for any nonloop edge e of G one has (where G/e is the graph obtained from G by contracting e):

(28.10) $$\phi_\Gamma(G) = \phi_\Gamma(G/e) - \phi_\Gamma(G - e).$$

Moreover, if each edge of G is a loop, then:

(28.11) $$\phi_\Gamma(G) = (|\Gamma| - 1)^{|E|}.$$

This proves that if Γ and Γ' are finite abelian groups with $|\Gamma| = |\Gamma'|$, then $\phi_\Gamma(G) = \phi_{\Gamma'}(G)$. Hence G has a nowhere-zero flow over Γ if and only if G has a nowhere-zero flow over Γ'. Therefore:

(28.12) if G has a nowhere-zero flow over some abelian group of size k, then it has one over each abelian group of size k.

We now consider (i)⇒(iii). By (28.12), (i) implies that G has a nowhere-zero flow over the group of integers mod k. This implies that there is an orientation $D = (V, A)$ of G and a function $f : A \to \{1, \ldots, k-1\}$ such that for each $v \in V$:

(28.13) $$f(\delta^{\text{in}}(v)) \equiv f(\delta^{\text{out}}(v)) \pmod{k}.$$

We choose the orientation D and the function f such that the sum

(28.14) $$\sum_{v \in V} |f(\delta^{\text{in}}(v)) - f(\delta^{\text{out}}(v))|$$

is minimized. If the sum is 0, we are done. So assume that the sum is nonzero. Define

(28.15) $$U_+ := \{v \in V \mid f(\delta^{\text{in}}(v)) > f(\delta^{\text{out}}(v))\} \text{ and}$$
$$U_- := \{v \in V \mid f(\delta^{\text{in}}(v)) < f(\delta^{\text{out}}(v))\}.$$

Necessarily, there is a directed path P in D from U_- to U_+ (Theorem 11.1). Now reverse the orientation of each arc a on P to its reverse a^{-1}, and define $f(a^{-1}) := k - f(a)$. This maintains (28.13) but reduces the sum (28.14), a contradiction.

This proves (i)⇒(iii), and hence (i)⇔(iii). Since (iii) is maintained if we increase k, also (i) is maintained if we increase k. So with (28.12), (i) implies (ii) if (ii) is restricted to finite groups. Since each infinite abelian group has $\mathbb{Z}$ as subgroup or has arbitrarily large finite subgroups, (iii)⇒(ii) also follows for infinite groups.

The equivalence of (iii) and (iv) follows directly from Hoffman's circulation theorem (Theorem 11.2). ∎

This theorem implies that in studying the existence of nowhere-zero flows, we can restrict ourselves to the group $\mathbb{Z}_k$ with elements $0, \ldots, k-1$ and addition mod k. A *nowhere-zero k-flow* is a nowhere-zero flow over $\mathbb{Z}_k$.

It is easy to characterize the graphs having a nowhere-zero 2-flow: they are precisely the Eulerian graphs. As to larger values of k there are the following three famous conjectures of Tutte. The 5-*flow conjecture* (Tutte [1954a]):

(28.16) (?) each bridgeless graph has a nowhere-zero 5-flow, (?)

the 4-*flow conjecture* (Tutte [1966]):

(28.17) (?) each bridgeless graph without Petersen graph minor has a nowhere-zero 4-flow, (?)

and the 3-*flow conjecture* (W.T. Tutte, 1972 (cf. Bondy and Murty [1976], Unsolved problem 48)):

(28.18) (?) each 4-edge-connected graph has a nowhere-zero 3-flow. (?)

For planar graphs this is equivalent to the theorem of Grötzsch [1958] that each loopless triangle-free planar graph is 3-vertex-colourable.

It may be seen that a cubic graph G has a nowhere-zero 3-flow if and only if G is bipartite. This follows from the fact that the existence of such a flow implies that G has an orientation such that in each vertex the indegree and outdegree differ by a multiple of 3. Hence, one of them is 3, the other 0. Hence each arc is oriented from a source to a sink, and so G is bipartite. The reverse implication is easy, by orienting each edge from one colour class to the other.

Jaeger [1979] showed that each 4-edge-connected graph has a nowhere-zero 4-flow: a 4-edge-connected graph $G = (V, E)$ has two edge-disjoint spanning trees T_1 and T_2 (by Corollary 51.1a). For $i = 1, 2$, let C_i be the symmetric difference of all fundamental circuits of T_i. Then C_1 and C_2 are cycles covering E. This gives a nowhere-zero 4-flow.

Jaeger [1988] proposed a weakened version of the 3-flow conjecture, the *weak 3-flow conjecture*:

(28.19) (?) there exists a number k such that each k-edge-connected graph has a nowhere-zero 3-flow. (?)

By (28.8), the 4-flow conjecture implies the four-colour theorem. For cubic graphs, (28.17) was proved by Robertson, Seymour, and Thomas [1997], Sanders, Seymour, and Thomas [2000], and Sanders and Thomas [2000].

One should note that having a nowhere-zero 4-flow is equivalent to the existence of two cycles covering the edge set. In other words, there exist two disjoint T-joins, where T is the set of odd-degree vertices (see Chapter 29).

It was proved by Seymour [1981b] that each bridgeless graph has a nowhere-zero 6-flow. (Inspired by Seymour's method, Younger [1983] gave a polynomial-time algorithmic proof.)

Seymour's theorem improves an earlier result of Jaeger [1976,1979] that each bridgeless graph has a nowhere-zero 8-flow. This is equivalent to: each bridgeless graph contains three cycles covering all edges.

Jaeger [1984] offered a conjecture, the *circular flow conjecture*, that implies both the 3-flow and the 5-flow conjecture:

(28.20) (?) for each $k \geq 1$, each $4k$-connected graph has an orientation such that in each vertex, the indegree and the outdegree differ by an integer multiple of $2k + 1$. (?)

For $k = 1$, this is equivalent to the 3-flow conjecture. For $k = 2$, it implies the 5-flow conjecture: Let $G = (V, E)$ be a 3-edge-connected graph, and replace each edge by 3 parallel edges. The new graph, H say, is 9-edge-connected. If (28.20) is true for $k = 2$, H has an orientation such that in each vertex, the indegree and the outdegree differ by a multiple of 5. This can easily be transformed to a nowhere-zero 5-flow in G.[12]

More on the 3-flow conjecture can be found in Fan [1993] and Kochol [2001]. Jaeger [1979,1988] and Seymour [1995a] gave surveys on nowhere-zero flows, and a book on this topic was written by Zhang [1997b]. We continue discussing nowhere-zero flows in Section 38.8.

[12] Orient any edge e of G in the direction of the majority of the direction of the three parallel edges in H made from e, with flow equal to 3 if all three edges have the same orientation, and 1 otherwise.

28.5. Fractional edge-colouring

Determining the edge-colouring number of a graph is NP-complete, but with matching techniques one can determine a fractional version of it in polynomial time.

Let $G = (V, E)$ be a graph. The *fractional edge-colouring number* $\chi'^*(G)$ of G is defined as

$$(28.21) \qquad \chi'^*(G) := \min\{\sum_{M\in\mathcal{M}} \lambda_M \mid \lambda \in \mathbb{R}_+^{\mathcal{M}}, \sum_{M\in\mathcal{M}} \lambda_M \chi^M = \mathbf{1}\},$$

where $\mathcal{M}$ denotes the collection of all matchings in G.

So if we require the λ_M to be integer, this would define the edge-colouring number of G. Therefore, we have

$$(28.22) \qquad \chi'^*(G) \leq \chi'(G).$$

The Petersen graph is an example of a graph G with $\chi'^*(G) = 3$ and $\chi'(G) = 4$. In Section 28.7 we shall see that $\chi'^*(G)$ can be computed in polynomial time.

$\chi'^*(G)$ can be characterized as follows. For any natural number $k \geq 1$, let G_k be the graph obtained from G by replacing each edge by k parallel edges. Then

$$(28.23) \qquad \chi'^*(G) = \min_{k\geq 1} \frac{\chi'(G_k)}{k}.$$

This follows from the fact that the minimum in (28.21) is attained by rational λ_M. Then the minimum in (28.23) is attained by $k :=$ the l.c.m. of the denominators of the λ_M.

From Edmonds' matching polytope theorem (Corollary 25.1a), a characterization of the fractional edge-colouring number follows:

Theorem 28.5. *The fractional edge-colouring number $\chi'^*(G)$ satisfies:*

$$(28.24) \qquad \chi'^*(G) = \max\{\Delta(G), \max_{U\subseteq V, |U|\geq 3} \frac{|E[U]|}{\lfloor \frac{1}{2}|U| \rfloor}\}.$$

Proof. Let μ be equal to the maximum in (28.24). Then $\chi'^*(G) \geq \mu$, since if λ_M attains minimum (28.21) and if vertex v has maximum degree, then

$$(28.25) \qquad \chi'^*(G) = \sum_M \lambda_M \geq \sum_M \lambda_M |M \cap \delta(v)| = \sum_{e\in\delta(v)} \sum_{M\ni e} \lambda_M$$
$$= \sum_{e\in\delta(v)} 1 = \Delta(G).$$

Moreover, for each $U \subseteq V$ with $|U| \geq 3$, (28.26) $$\chi'^*(G) = \sum_M \lambda_M \geq \sum_M \lambda_M \frac{|M \cap E[U]|}{\lfloor \frac{1}{2}|U| \rfloor} = \frac{1}{\lfloor \frac{1}{2}|U| \rfloor} \sum_{e \in E[U]} \sum_{M \ni e} \lambda_M = \frac{|E[U]|}{\lfloor \frac{1}{2}|U| \rfloor}.$$

To see that $\chi'^*(G) = \mu$, let x be the all-$\frac{1}{\mu}$ vector in $\mathbb{R}^E$. Then $x(\delta(v)) \leq 1$ for each $v \in V$ and $x(E[U]) \leq \lfloor \frac{1}{2}|U| \rfloor$ for each $U \subseteq V$ with $|U| \geq 3$. Hence x belongs to the matching polytope of G. So x is a convex combination of incidence vectors of matchings. Therefore $\mathbf{1} = \mu \cdot x = \sum_M \lambda_M \chi^M$ for some $\lambda_M \geq 0$ with $\sum_M \lambda_M = \mu$, showing that $\chi'^*(G) \leq \mu$. ∎

This implies for regular graphs:

Corollary 28.5a. *Let $G = (V, E)$ be a k-regular graph. Then $\chi'^*(G) = k$ if and only if $|\delta(U)| \geq k$ for each odd subset U of V.*

Proof. By Theorem 28.5, $\chi'^*(G) = k$ if and only if $|E[U]| \leq k\lfloor \frac{1}{2}|U| \rfloor$ for each subset U of V. This last is equivalent to $|\delta(U)| \geq k$ for each odd subset U of V. ∎

Call a graph $G = (V, E)$ a *k-graph* if G is regular of degree k and if $|\delta(U)| \geq k$ for each odd subset U of V. So by Corollary 28.5a, a k-regular graph G is a k-graph if and only if $\chi'^*(G) = k$.

28.6. Conjectures

Seymour [1979a] conjectures that

(28.27) (?) $\lceil \chi'^*(G) \rceil = \lceil \frac{1}{2}\chi'(G_2) \rceil$ (?)

for each graph G, where G_2 arises from G by replacing each edge by two parallel edges. Conjecture (28.27) is equivalent to the conjecture that, for each k,

(28.28) (?) for each k-graph G one has $\chi'(G_2) = 2k$ (?);

equivalently, for each k-graph G, the minimum (28.21) for $\chi'^*(G)$ is attained by half-integer λ_M. In other words, it is conjectured that any k-graph has $2k$ perfect matchings covering each edge exactly twice. (The equivalence of (28.27) and (28.28) can be seen as follows. The implication (28.27)⇒(28.28) is easy. To see the reverse implication, let G be any graph and define $k := \lceil \chi'^*(G) \rceil$. Make a disjoint copy G' of G, and connect each vertex v of G by $k - \deg_G(v)$ parallel edges to its copy v' in G'. This makes a k-regular graph H with $\chi'^*(H) = k$. So H is a k-graph, and hence by (28.28), $\chi'(H_2) = 2k$. Hence $\chi'(G_2) \leq 2k$, implying (28.27).)

Seymour called (28.28) the *generalized Fulkerson conjecture*, as it generalizes the special case $k = 3$ asked (but not conjectured) by Fulkerson [1971a]. This special case is called the 'Fulkerson conjecture'[13]. (By Corollary 28.5a, a cubic graph G has $\chi'^*(G) = 3$ if and only if G is bridgeless.) For a partial result, see Corollary 38.11e.

Berge [1979a] conjectured that the edges of any bridgeless cubic graph can be covered by 5 perfect matchings. This would follow from the Fulkerson conjecture.

A conjecture of Gol'dberg [1973] (and also of Seymour [1979a]) is that for each (possibly nonsimple) graph G one has

(28.29) (?) $\chi'(G) \leq \max\{\Delta(G) + 1, \lceil\chi'^*(G)\rceil\}$. (?)

(An equivalent conjecture was stated by Andersen [1977].)

As $\chi'(G) \geq \max\{\Delta(G), \lceil\chi'^*(G)\rceil\}$, validity of (28.29) would yield a tight (gap 1) bound for $\chi'(G)$ also for nonsimple graphs. In particular, if $\Delta(G) < \chi'^*(G)$, we would have equality in (28.29). Seymour [1979a] mentioned that he has shown that $\chi'(G) \leq \lceil\chi'^*(G)\rceil + 1$ for graphs G with $\chi'^*(G) \leq 6$.

Conjecture (28.29) would generalize Theorem 28.2 due to Vizing. For let $\mu(G)$ again denote the maximum multiplicity of any edge of $G = (V, E)$. Then for any subset U of V,

(28.30)
$$|E[U]| \leq \tfrac{1}{2}\Delta(G[U])|U| \leq (\Delta(G[U]) + \mu(G))\tfrac{1}{2}(|U| - 1) \leq (\Delta(G) + \mu(G))\lfloor\tfrac{1}{2}|U|\rfloor$$

(The second inequality follows from $\Delta(G[U]) \leq \mu(G)(|U| - 1)$.) So with Theorem 28.5 we know that $\chi'^*(G) \leq \Delta(G) + \mu(G)$.

A well-known equivalent form of the four-colour theorem is that each bridgeless cubic planar graph is 3-edge-colourable. This equivalence was discovered by Tait [1878b]. Seymour [1981c] conjectures the following generalization:

(28.31) (?) each planar k-graph is k-edge-colourable. (?)

This was proved for $k = 4$ and $k = 5$ by Guenin [2002b].

A consequence of the 4-flow conjecture of Tutte [1966] is:

(28.32) each bridgeless cubic graph without Petersen graph minor is 3-edge-colourable.

This was proved jointly by Robertson, Seymour, and Thomas [1997], Sanders, Seymour, and Thomas [2000], and Sanders and Thomas [2000].

(28.31) and (28.32) made Lovász [1987] conjecture:

(28.33) (?) each k-graph without Petersen graph minor is k-edge-colourable. (?)

[13] Seymour [1979a] says that it was first conjectured by C. Berge, but that it is usually called Fulkerson's conjecture because the latter put it into print.

(An equivalent conjecture was given by Rizzi [1997,1999].) This is equivalent to stating that the incidence vectors of perfect matchings in a graph without Petersen graph minor, form a Hilbert base (cf. Section 5.18). It relates to Lovász's work on the perfect matching lattice — see Chapter 38.

Goddyn [1993] noted that (28.33) would not yield a full characterization, since also the perfect matchings of the Petersen graph form a Hilbert base. (This is due to the fact that the all-one vector does not belong to the perfect matching lattice of the Petersen graph.)

Notes. Seymour [1979a] conjectured that if $k \geq 4$, any k-graph $G = (V, E)$ has a perfect matching M such that $G - M$ is a $(k-1)$-graph. However, this was disproved by Rizzi [1997,1999], who showed that for any $k \geq 3$, there exists a k-graph in which any two perfect matchings intersect. Hence, for any $k \geq 3$ there exists a k-graph that cannot be decomposed into a k_1- and a k_2-graph for any $k_1, k_2 \geq 1$ with $k_1 + k_2 = k$.

Nishizeki and Kashiwagi [1990] showed that

(28.34) $$\chi'(G) \leq \max\{\tfrac{11}{10}\Delta(G) + \tfrac{4}{5}, \lceil \chi'^*(G) \rceil\},$$

and they gave a polynomial-time algorithm finding an edge-colouring fulfilling this bound. (This improves earlier results of Andersen [1977], Goldberg [1984], and Hochbaum, Nishizeki, and Shmoys [1986].)

Marcotte [1986,1990a,1990b,2001], Seymour [1990a], Lee and Leung [1993], and Caprara and Rizzi [1998] gave other partial results on conjecture (28.29).

28.7. Edge-colouring polyhedrally

Let $G = (V, E)$ be a graph and let Q be the polytope determined by

(28.35) $$\begin{array}{ll} x_e \geq 0 & (e \in E), \\ x(M) \leq 1 & (M \text{ matching}). \end{array}$$

So Q is the antiblocking polyhedron of the matching polytope. By the description of the matching polytope and by the theory of antiblocking polyhedra, Q is equal to the convex hull of the following set of vectors:

(28.36) $$\begin{array}{ll} \chi^S & S \text{ substar}, \\ \frac{1}{\lfloor \frac{1}{2} |\bigcup F| \rfloor} \chi^F & \text{for nonempty } F \subseteq E. \end{array}$$

Here a *substar* is any set S of edges with $S \subseteq \delta(v)$ for some $v \in V$. By $\bigcup F$ we denote the set of vertices covered by F.

Now the fractional edge-colouring number $\chi'^*(G)$ is equal to the maximum value of $\mathbf{1}^{\mathsf{T}} x$ over Q (by LP-duality). The ellipsoid method then gives:

Theorem 28.6. *The fractional edge-colouring number of a graph can be determined in polynomial time.*

Proof. The separation problem over Q is equivalent to the weighted matching problem, and hence is solvable in polynomial time. Therefore, with the ellipsoid method, also the optimization problem over Q is solvable in polynomial time. This gives the fractional edge-colouring number. ∎

For any weight function $w \in \mathbb{R}_+^E$, the maximum of $w^\mathsf{T} x$ where x ranges over the vectors (28.36), is equal to the minimum value of $\sum_M \lambda_M$ where $\lambda_M \geq 0$ for $M \in \mathcal{M}$ such that $\sum_M \lambda_M \chi^M = w$. Thus we have a min-max relation for the 'weighted fractional edge-colouring number'.

We should note that (generally) the matching polytope does not have the integer decomposition property, and (equivalently) that system (28.35) does not have the integer rounding property. Indeed, for the Petersen graph, the maximum of $\mathbf{1}^\mathsf{T} x$ over (28.35) is equal to 3. So it has a fractional optimum dual solution of value 3. However, there is no integer optimum dual solution, since the edges of the Petersen graph cannot be decomposed into three matchings.

28.8. Packing edge covers

The results on edge-colouring (which is essentially covering by matchings), can be dualized to packing edge covers, as observed by Gupta [1974] (where $\delta(G)$ denotes the minimum degree of G):

Theorem 28.7. *A simple graph $G = (V, E)$ has $\delta(G)-1$ disjoint edge covers.*

Proof. Make an auxiliary graph H as follows. For each $v \in V$, do the following. Make $\deg_G(v) - \delta(G)$ new vertices, and reconnect $\deg_G(v) - \delta(G)$ of the edges incident with v with the new vertices, in such a way that v has degree $\delta(G)$, while each new vertex has degree 1.

Then H has maximum degree $\delta(G)$ and there is a one-to-one mapping between the edges of G and those of H. By Vizing's theorem for simple graphs (Theorem 28.1), H has matchings $M_1, \ldots, M_{\delta(G)+1}$ partitioning E. We denote the corresponding edge sets in G also by M_i.

Then each vertex v of G is covered by all but at most one of the matchings $M_1, \ldots, M_{\delta(G)+1}$. Let U be the set of vertices of G missed by one of $M_1, \ldots, M_{\delta(G)-1}$. Then each vertex in U is covered by both $M_{\delta(G)}$ and $M_{\delta(G)+1}$. So $M_{\delta(G)} \cup M_{\delta(G)+1}$ forms a graph on V where each vertex in U has degree at least 2. Hence we can orient the edges in $M_{\delta(G)} \cup M_{\delta(G)+1}$ such that each vertex in U is head of at least one of the oriented edges.

Now for each $i = 1, \ldots, \delta(G) - 1$, add to M_i all edges in $M_{\delta(G)} \cup M_{\delta(G)+1}$ that are oriented towards a vertex missed by M_i. This gives $\delta(G) - 1$ disjoint edge covers. ∎

This can be formulated in terms of the *edge cover packing number* $\xi(G)$ of G, which is the maximum number of disjoint edge covers in G. Then, if G is simple,

(28.37) $\xi(G) \geq \delta(G) - 1.$

Gupta [1974] showed more generally for not necessarily simple graphs (where $\mu(G)$ denotes the maximum multiplicity of the edges of G):

Theorem 28.8. *Any graph G has $\delta(G) - \mu(G)$ disjoint edge covers.*

Proof. Let $\delta := \delta(G)$ and $\mu := \mu(G)$. Make an auxiliary graph H as follows. For each $v \in V$, do the following. Make $\deg_G(v) - \delta$ new vertices, and reconnect $\deg_G(v) - \delta$ of the edges incident with v with the new vertices, in such a way that v has degree δ, while each new vertex has degree 1.

Then H has maximum degree $\delta(G)$, and there is a one-to-one mapping between the edges of G and those of H. By Vizing's theorem (Theorem 28.2), H has matchings $M_1, \ldots, M_{\delta+\mu}$ partitioning E. We denote the corresponding edge sets in G also by M_i. Let

(28.38) $F := M_{\delta-\mu+1} \cup \cdots \cup M_{\delta+\mu}.$

Orient the edges in F such that each vertex v is entered by at least $\lfloor \frac{1}{2} \deg_F(v) \rfloor$ of the edges incident with v.

Consider any vertex v, and let v be missed by α of the $M_1, \ldots, M_{\delta-\mu}$. Let k be the number of $M_{\delta-\mu+1}, \ldots, M_{\delta+\mu}$ covering v. As v is covered by at least δ of the $M_1, \ldots, M_{\delta+\mu}$, we know $k + (\delta - \mu - \alpha) \geq \delta$, that is, $k \geq \mu + \alpha$. Since $k \leq 2\mu$, it follows that $\deg_F(v) \geq k = 2k - k \geq 2(\mu + \alpha) - 2\mu = 2\alpha$. Hence v is entered by at least α edges.

So for each $i = 1, \ldots, \delta - \mu$, if v is missed by M_i, then we can extend M_i by an edge in F oriented towards v. Doing this for each vertex v, we obtain $\delta - \mu$ disjoint edge covers. ∎

Equivalently, for any graph,

(28.39) $\xi(G) \geq \delta(G) - \mu(G).$

Gupta [1974] announced (without proof) and Fournier [1977] showed that for any graph $G = (V, E)$ and any $k \in \mathbb{Z}_+$, E can be partitioned into classes $E_1, \ldots, E_k$ such that each vertex v is covered by at least

(28.40) $\min\{k, \deg(v), \max\{k, \deg(v)\} - \mu(v)\}$

of the E_i, where $\mu(v)$ denotes the maximum multiplicity of the edges incident with v.

28.9. Further results and notes

28.9a. Shannon's theorem

Shannon [1949] gave the following upper bound on the edge-colouring number that can be better than Vizing's bound if G is not simple. As Vizing [1965a] observed, the bound can be derived from Vizing's theorem, as below.

Theorem 28.9. *The edge-colouring number $\chi'(G)$ of a graph $G = (V, E)$ is at most $\lfloor \frac{3}{2}\Delta(G) \rfloor$.*

Proof. Let G be a counterexample with a minimum number of edges. Define $k := \lfloor \frac{3}{2}\Delta(G) \rfloor$. So $\chi'(G) > k$ and by Vizing's theorem (Theorem 28.2), $\chi'(G) \leq \Delta(G) + \mu(G)$, where $\mu(G)$ is the maximum edge-multiplicity of G. Hence $\mu(G) > \frac{1}{2}\Delta(G)$.

Let u and v be vertices connected by $\mu(G)$ parallel edges. Choose one such edge, e say. By the minimality of G, $\chi'(G - e) \leq k$. Consider a k-edge-colouring of $G - e$. Let I_u and I_v be the sets of colours covering u and v respectively. Then $|I_u \cap I_v| \geq \mu(G) - 1$, since $\mu(G) - 1$ edges of $G - e$ connect u and v. Moreover, $|I_u| \leq \Delta(G) - 1$, since u has degree less than $\Delta(G)$ in $G - e$; similarly, $|I_v| \leq \Delta(G) - 1$. So

$$
\begin{aligned}
(28.41) \qquad |I_u \cup I_v| &= |I_u| + |I_v| - |I_u \cap I_v| \leq 2(\Delta(G) - 1) - (\mu(G) - 1) \\
&= 2\Delta(G) - \mu(G) - 1 < \tfrac{3}{2}\Delta(G) - 1 < k
\end{aligned}
$$

(since $\mu(G) > \frac{1}{2}\Delta(G)$), and hence at least one colour does not occur in $I_u \cup I_v$. This colour can be given to edge e to obtain a k-edge-colouring of G. ∎

The bound in this theorem is sharp, as is shown by a graph H on three vertices u, v, and w, with $\lceil \frac{1}{2}\Delta \rceil$ parallel arcs connecting u and v, $\lfloor \frac{1}{2}\Delta \rfloor$ parallel arcs connecting u and w, and $\lfloor \frac{1}{2}\Delta \rfloor$ parallel arcs connecting v and w. Then $\Delta(H) = \Delta$ and $\chi'(H) = \lfloor \frac{3}{2}\Delta \rfloor$.

Vizing [1965a] showed that any graph G with $\Delta(G) \geq 4$ and $\chi'(G) = \lfloor \frac{3}{2}\Delta(G) \rfloor$ contains this graph H as a subgraph.

The case $\Delta(G)$ even in Theorem 28.9 can be proved simpler as follows. We may assume that each degree of G is even (we can pair up the odd-degree vertices by new edges). Let $k := \frac{1}{2}\Delta(G)$. Make an Eulerian orientation of G. Split each vertex v into two vertices v' and v'', and replace any edge oriented from u to v, by an edge connecting u' and v''. In this way we obtain a bipartite graph H, of maximum degree k. Hence, by Kőnig's edge-colouring theorem, H has a k-edge-colouring. This yields a decomposition of the edges of G into classes $E_1, \ldots, E_k$ such that each graph $G_i = (V, E_i)$ has maximum degree 2. Hence each G_i is 3-edge-colourable, and therefore G is $3k$-colourable.

28.9b. Further notes

For simple *planar* graphs, if $\Delta(G) \geq 7$, then $\chi'(G) = \Delta(G)$ (for $\Delta \geq 8$, this was proved by Vizing [1965b], and for $\Delta = 7$ by Sanders and Zhao [2001] and Zhang [2000]). For $2 \leq \Delta \leq 5$ there exist simple planar graphs of maximum degree Δ with $\chi'(G) = \Delta + 1$. This is unknown for $\Delta = 6$ (and constitutes a question of Vizing

[1968]). For $\Delta \geq 8$, polynomial-time algorithms finding a Δ-edge-colouring of a simple planar graph were given by Terada and Nishizeki [1982] ($O(n^2)$), Chrobak and Yung [1989] ($O(n)$ if $\Delta \geq 19$), and Chrobak and Nishizeki [1990] ($O(n \log n)$ if $\Delta \geq 9$).

Kotzig [1957] showed the following theorem:

Theorem 28.10. *Let $G = (V, E)$ be a connected cubic graph with an even number of edges. Then G is 3-edge-colourable if and only if the line graph $L(G)$ of G is 4-edge-colourable.*

Proof. I. First assume that $L(G)$ is 4-edge-colourable, say with colours 0, 1, 2, and 3. We colour the edges of G with colours labeled by the three partitions of $\{0, 1, 2, 3\}$ into pairs. Consider an edge $e = uv$ of G. Let f_1 and f_2 be the two other edges incident with u and let g_1 and g_2 be the two other edges incident with v. Let i_1 and i_2 be the colours of the edges ef_1 and ef_2 of $L(G)$ and let j_1 and j_2 be the colours of the edges eg_1 and eg_2 of $L(G)$. Give e the colour labeled by the partition of $\{0, 1, 2, 3\}$ into the pairs $\{i_1, i_2\}$ and $\{j_1, j_2\}$. This gives a 3-edge-colouring of G.

II. Conversely, assume that G is 3-edge-colourable. We first show that $L(G)$ has a perfect matching. Indeed, there is a subset M of the edge set of $L(G)$ such that each vertex of $L(G)$ is covered an odd number of times. To see this, choose an arbitrary partition Π of the vertices of $L(G)$ into pairs, and for each pair $\{e, f\} \in \Pi$, we choose an $e - f$ path $P_{e,f}$ in $L(G)$. Then the symmetric difference of all these paths is a subset M as required.

Now choose such an M with $|M|$ as small as possible. We claim that each vertex of $L(G)$ is covered exactly once by M; that is, M is a perfect matching in $L(G)$. Suppose that vertex e of $L(G)$ is covered by three edges in M, say ee_1, ee_2, and ee_3. We can assume that e, e_1 and e_2 are pairwise adjacent in $L(G)$. Hence, replacing M by $M \triangle \{ee_1, ee_2, e_1e_2\}$, gives a subset M' covering each vertex an odd number of times, however with $|M'| < |M|$. This contradicts our assumption.

So M is a perfect matching in $L(G)$, forming our first colour 0. Let G be edge-coloured with colours 1, 2, and 3. Consider an edge e_1e_2 of $L(G)$ not having colour 0. Let e_0 be the third edge of G incident with the common vertex of e_1 and e_2. If e_0e_1 has colour 0, give e_1e_2 the colour of edge e_1. If e_0e_2 has colour 0, give e_1e_2 the colour of edge e_2. If neither e_0e_1 nor e_0e_2 has colour 0, give e_1e_2 the colour of edge e_0. It is straightforward to check that this gives a 4-edge-colouring of $L(G)$. ∎

For more on edge-colouring cubic graphs, see Kotzig [1975,1977].

McDiarmid [1972] observed that in any graph $G = (V, E)$, if $p \geq \chi'(G)$, then there is a p-edge-colouring with $\lfloor |E|/p \rfloor \leq |M| \leq \lceil |E|/p \rceil$ for each colour M. This can be proved in the same way as Theorem 20.8.

Meredith [1973] gave k-regular non-Hamiltonian non-k-edge colourable graphs with an even number of vertices, for each $k \geq 3$ (cf. Isaacs [1975]). Johnson [1966a] gave a short proof that any cubic graph is 4-edge-colourable.

Vizing [1965a] asked if the minimum number of colours of the edges of a graph can be obtained from any edge-colouring by iteratively swapping colours on a colour-alternating path or circuit and deleting empty colours.

Marcotte and Seymour [1990] observed that the following is a necessary condition for extending a partial k-edge colouring a graph $G = (V, E)$ to a complete k-edge colouring:

(28.42) $$|F| \leq \sum_{i=1}^{k} \mu_i(F) \text{ for each } F \subseteq E,$$

where $\mu_i(F)$ is the maximum size of a matching $M \subseteq F$ not covering any vertex covered by the ith colour. They studied graphs where this condition is sufficient as well.

Vizing [1965a] showed that if G is nonsimple and $\Delta(G) = 2\mu(G) + 1$, then $\chi'(G) \leq 3\mu(G)$ (where $\mu(G)$ is the maximum edge-multiplicity of G).

The *list-edge-colouring number* $\chi^l(G)$ of a graph $G = (V, E)$ is the minimum number k such that for each choice of sets L_e for $e \in E$ with $|L_e| = k$, one can select $l_e \in L_e$ for $e \in E$ such that for any two incident edges e, f one has $l_e \neq l_f$. Vizing [1976] conjectures that $\chi^l(G)$ is equal to the edge-colouring number of G, for each graph G (see Häggkvist and Chetwynd [1992]).

The *total colouring number* of a graph $G = (V, E)$ is a colouring of $V \cup E$ such that each colour consists of a stable set and a matching, vertex-disjoint. Behzad [1965] and Vizing [1968] conjecture that the total colouring number of a simple graph G is at most $\Delta(G) + 2$. Molloy and Reed [1998] showed that there exists a constant C such that the total colouring number of any simple graph is at most $\Delta(G) + C$. A polynomial-time algorithm finding a total colouring with $\Delta(G)$+poly($\log \Delta$) colours is given by Hind, Molloy, and Reed [1999].

More generally, Vizing [1968] conjectures that the total colouring number of a graph G is at most $\Delta(G) + \mu(G) + 1$, where $\mu(G)$ is the maximum size of a parallel class of edges. Partial results have been found by Kostochka [1977], Hind [1990, 1994], Kilakos and Reed [1992], and McDiarmid and Reed [1993].

For extensions of Vizing's theorem, see Vizing [1965b], Fournier [1973], Jakobsen [1973], Gol'dberg [1974], Fiorini [1975], Hilton [1975], Jakobsen [1975], Andersen [1977], Kierstead and Schmerl [1983], Kostochka [1983], Ehrenfeucht, Faber, and Kierstead [1984], Goldberg [1984], Hilton and Jackson [1987], Berge [1990], and Chew [1997a]. The fractional edge-colouring number $\chi'^*(G)$ was studied by Hilton [1975] and Stahl [1979]. A computational study based on fractional edge-colouring was made by Nemhauser and Park [1991]. Equitable edge-colourings were investigated by de Werra [1981].

Generalizations of edge-colouring were studied by Hakimi and Kariv [1986] and Nakano, Nishizeki, and Saito [1988,1990]. Fiorini and Wilson [1977,1978], Fiorini [1978], Jensen and Toft [1995], Nakano, Zhou, and Nishizeki [1995], and Zhou and Nishizeki [2000] gave surveys on edge-colouring and extensions.

28.9c. Historical notes on edge-colouring

Historically, studying edge-colouring was motivated by the equivalence of the four-colour conjecture with the 3-edge-colourability of planar bridgeless cubic graphs. The four-colour conjecture was raised in 1852 by F. Guthrie. An early attempt to prove the conjecture by Kempe [1879,1880] was shown by Heawood in 1890 to contain an error — see below.

Also Tait [1878a] studied the four-colour problem. He claimed without proof that each triangulated planar graph has two disjoint sets of edges such that each triangular face is incident with exactly one edge in each of these sets. From this he derived (correctly) that each loopless planar graph is 4-vertex-colourable. He also observed that his claim implies that each planar bridgeless cubic graph is 3-edge-colourable.

In a note following a note of Guthrie [1878] (describing the very early history of the four-colour problem, which note itself was a reaction to the article of Tait [1878a]), Tait [1878b] remarked that in his earlier paper

> I gave a series of proofs of the theorem that four colours suffice for a map. All of these were long, and I felt that, while more than sufficient to prove the truth of the theorem, they gave little insight into its real nature and bearings. A somewhat similar remark may, I think, be made about Mr Kempe's proof.

He therefore withdrew the former paper, and replaced it by the present note, in which, without proof, the following 'elementary theorem' is formulated:

> if an even number of points be joined, so that three (and only three) lines meet in each, these lines may be coloured with *three* colours only, so that no two conterminous lines shall have the same colour. (When an odd number of the points forms a group, connected by *one* line only with the rest, the theorem is not true.)

Tait next gave the now well-known construction of deriving 3-edge-colourability of bridgeless planar cubic graphs from the 4-vertex-colourability of planar loopless graphs. At that time, the error in Kempe's proof of the four-colour conjecture was not yet detected.

But Tait also said:

> The proof of the elementary theorem is given easily by induction; and then the proof that four colours suffice for a map follows almost immediately from the theorem, by an inversion of the demonstration just given.

Tait [1880] claimed that in Tait [1878b] the 3-edge-colourability of bridgeless planar cubic graphs had been shown:

> *If* 2n *points be joined by* 3n *lines, so that three lines, and three only, meet at each point, these lines can be divided (usually in many different ways) into three groups of* n *each, such that one of each group ends at each of the points.*

While Tait did not mention it explicitly, he restricted himself to *planar* cubic graphs, since he considered them equivalently as the skeletons of polytopes[14]. Also the figures given in Tait [1880] are planar (and also those in Tait [1884], where similar claims are made).

The validity of Kempe's proof of the four-colour conjecture was accepted until Heawood [1890] discovered a fatal error in Kempe's proof, and showed that it in fact gives a five-colour theorem for planar graphs. The error in his proof was acknowledged by Kempe [1889]. (For an account of the early history of the four-colour problem, see Biggs, Lloyd, and Wilson [1976].)

After that, the problem of the 3-edge-colourability of bridgeless planar cubic graphs was open again. At several occasions, this problem was advanced (cf. Goursat [1894]). It was only resolved in 1977 when Appel and Haken proved the four-colour theorem.

Petersen [1898] asserted that Tait had claimed to have proved the 3-edge-colourability of any (also nonplanar) bridgeless cubic graph. It motivated him to present, as a counterexample, the now-called *Petersen graph*, which is a bridgeless cubic graph that is not 3-edge-colourable:

[14] Tait's polytopes are 3-dimensional, since each vertex has degree 3.

j'ai réussi à construire un graphe où le théorème de Tait ne s'applique pas.[15]

Petersen [1898] drew the Petersen graph as follows:

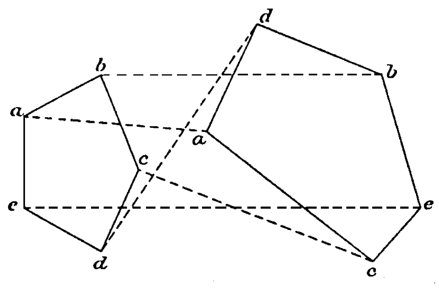


Figure 28.4

For another purpose, the Petersen graph was given earlier by Kempe [1886], who represented it as follows:

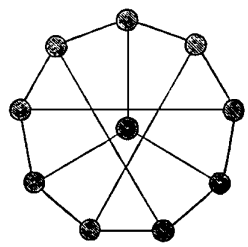


Figure 28.5

Sainte-Laguë [1926a] introduced the term *class* for the edge-colouring number of a graph. He noted (without exact argumentation) that Petersen's theorem on the existence of a perfect matching in a bridgeless cubic graph implies that each cubic graph has edge-colouring number 3 or 4.

[15] I have succeeded in constructing a graph where the theorem of Tait does not apply.

Chapter 29

T-joins, undirected shortest paths, and the Chinese postman

The methods for weighted matching also apply to shortest paths in undirected graphs (provided that each circuit has nonnegative length) and to the Chinese postman problem — more generally, to *T-joins*.

29.1. T-joins

Let $G = (V, E)$ be a graph and let $T \subseteq V$. A subset J of E is called a *T-join* if T is equal to the set of vertices of odd degree in the graph (V, J). So if a T-join exists, then $|T|$ is even. More precisely,

(29.1) G has a T-join if and only if $|K \cap T|$ is even for each component K of G.

T-joins are close to matchings. In fact, from Corollary 26.1a it can be derived that a shortest T-join can be found in strongly polynomial time. To see this, one should observe the following elementary graph-theoretical fact representing T-joins as sets of paths:

(29.2) each T-join is the edge-disjoint union of circuits and $\frac{1}{2}|T|$ paths connecting disjoint pairs of vertices in T;
the symmetric difference of a set of circuits and $\frac{1}{2}|T|$ paths connecting disjoint pairs of vertices in T is a T-join.

This is used in showing that a shortest T-join can be found in strongly polynomial time:

Theorem 29.1. *Given a graph $G = (V, E)$, a length function $l \in \mathbb{Q}^E$, and a subset T of V, a shortest T-join can be found in strongly polynomial time.*

Proof. First we dispose of negative lengths. Let N be the set of edges e with $l(e) < 0$, let U be the set of vertices v with $\deg_N(v)$ odd, let $T' := T \triangle U$, and let $l' \in \mathbb{Q}^E_+$ be defined by $l'(e) := |l(e)|$ for $e \in E$.

Then, if J' is a T'-join minimizing $l'(J')$, the set $J := J' \triangle N$ is a T-join minimizing $l(J)$. To see this, let $\widetilde{J}$ be any T-join. Then $\widetilde{J} \triangle N$ is a T'-join, and hence $l'(J') \leq l'(\widetilde{J} \triangle N)$. Therefore, (29.3) $$l(J) = l'(J') + l(N) \leq l'(\widetilde{J} \triangle N) + l(N) = l(\widetilde{J}).$$

So we can assume $l \geq \mathbf{0}$. Now consider the complete graph K_T with vertex set T. For each edge st of K_T, determine a path P_{st} in G of minimum length, say, $w(st)$. Find a perfect matching M in K_T minimizing $w(M)$. Then the symmetric difference of the paths P_{st} for $st \in M$ is a shortest T-join in G. This follows directly from (29.2). ∎

(This method is due to Edmonds [1965e].)

Note that a T-join J has minimum length if and only if $l(C \cap J) \leq \frac{1}{2} l(C)$ for each circuit C. (This was observed essentially by Guan [1960].)

Theorem 29.1 implies that also a *longest* T-join can be found in strongly polynomial time:

Corollary 29.1a. *Given a graph $G = (V, E)$, a length function $l \in \mathbb{Q}^E$, and a subset T of V, a longest T-join can be found in strongly polynomial time.*

Proof. Apply Theorem 29.1 to $-l$. ∎

An application is finding a maximum-capacity cut in a planar graph $G = (V, E)$ (Orlova and Dorfman [1972][16], Hadlock [1975]): it amounts to finding a maximum-capacity $\emptyset$-join in the planar dual graph. (Barahona [1990] gave an $O(n^{3/2} \log n)$ time bound.)

Another consequence is:

Corollary 29.1b. *Given a graph $G = (V, E)$ and a length function $l : E \to \mathbb{Q}$, one can check if there is a negative-length circuit in strongly polynomial time.*

Proof. There is a negative-length circuit if and only if there exists an $\emptyset$-join J with $l(J) < 0$. So Theorem 29.1 gives the corollary. ∎

Complexity. With Dijkstra's shortest path method (Theorem 7.3) one derives from Theorem 26.2 that a shortest T-join can be found in $O(n^3)$ time. Generally, one has an $O(\mathrm{APSP}_+(n, m, L) + \mathrm{WM}(n, n^2, nL))$-time algorithm, where L is the maximum absolute value of the lengths of the edges in G (assuming they are integer), where $\mathrm{APSP}_+(n, m, L)$ is the time in which the all-pairs shortest paths problem can be solved, in an undirected graph, with n vertices and m edges and with nonnegative integer lengths at most L, and where $\mathrm{WM}(n, m, W)$ is the time in which a minimum-weight perfect matching can be found, in a graph with n vertices and m edges and with integer weights at most W in absolute value.

[16] Orlova and Dorfman observed that finding a maximum-size cut in a planar graph amounts to finding shortest paths connecting the odd-degree vertices in the dual graph, but described a branch-and-bound method for it, and did not state that it can be solved in polynomial time by matching techniques.

29.2. The shortest path problem for undirected graphs

In Chapter 8 we saw that a shortest path in a directed graph without negative-length directed circuits, can be found in strongly polynomial time. It implies a strongly polynomial-time shortest path algorithm for undirected graphs, provided that all lengths are nonnegative. This, because the reduction replaces each undirected edge uv by two directed edges (u, v) and (v, u) — which would create a negative-length directed circuit if uv has negative length.

However, Theorem 29.1 implies that one can find (in strongly polynomial-time) a shortest path in undirected graphs even if there are negative-length edges, provided that all circuits have nonnegative length:

Corollary 29.1c. *Given a graph $G = (V, E)$, $s, t \in V$, and a length function $l \in \mathbb{Q}^E$ such that each circuit has nonnegative length, a shortest $s-t$ path can be found in strongly polynomial time.*

Proof. Define $T := \{s, t\}$ and apply Theorem 29.1. By observation (29.2), a shortest T-join J can be partitioned into an $s-t$ path and a number of circuits. Since by assumption any circuit has nonnegative length, we can delete the circuits from J. ∎

Complexity. Since by Gabow [1990] the weighted matching problem is solvable in $O(n(m + n \log n))$ time, a shortest path in an undirected graph, without negative-length circuits, can be found in $O(n(m + n \log n))$ time. This can be derived as follows: If we want to find a shortest $s-t$ path, add to each vertex v a 'copy' v', for each edge uv add edges uv', $u'v$, and $u'v'$ (each with the same length as uv), and for each vertex v add an edge vv', of length 0. Let G' be the graph obtained. Then a minimum-weight perfect matching in $G' - s' - t'$ gives a shortest $s-t$ path in G.

Gabow [1983a] gave an $O(n \min\{m \log n, n^2\})$-time algorithm for the all-pairs shortest paths problem in undirected graphs.

29.3. The Chinese postman problem

Call a walk $C = (v_0, e_1, v_1, \ldots, e_t, v_t)$ in a graph G a *Chinese postman tour* if $v_t = v_0$ and each edge of G occurs at least once in C. The *Chinese postman problem*, first studied by Guan [1960] (and named by Edmonds [1965e]), is:

(29.4) given: a connected graph $G = (V, E)$ and a length function $l \in \mathbb{Q}_+^E$,
find: a shortest Chinese postman tour C.

By Euler's theorem, if each vertex has even degree, there is an Eulerian tour, that is, a walk traversing each edge *exactly* once. So in that case, any Eulerian tour is a shortest Chinese postman tour.

But if not all degrees are even, certain edges have to be traversed more than once. These edges form in fact a shortest T-join for $T := \{v \mid \deg_G(v)$ odd$\}$. This is the base of the following consequence of Theorem 29.1:

Corollary 29.1d. *The Chinese postman problem can be solved in strongly polynomial time.*

Proof. Let $T := \{v \mid \deg_G(v) \text{ odd}\}$. Find a shortest T-join J. Add to each edge e in J an edge e' parallel to e. This gives the Eulerian graph G'. Then any Eulerian tour in G' gives a shortest Chinese postman tour (by identifying any new edge e' with its parallel e).

To see this, note that obviously the Eulerian tour gives a Chinese postman tour C of length $l(E) + l(J)$. Suppose that there is a shorter tour C'. Let J' be the set of edges traversed an even number of times by C'. Then J' is a T-join, and so $l(J') \geq l(J)$. Hence $l(C') \geq l(E) + l(J') \geq l(E) + l(J) = l(C)$. ∎

Observe that a postman never has to traverse any street more than twice.

Complexity. The above gives an $O(n^3)$-time algorithm for the Chinese postman problem (more precisely, $O(k(m + n \log n) + k^3 + m)$, where k is the number of vertices of odd degree).

29.4. *T*-joins and *T*-cuts

There is an interesting min-max relation for the minimum size of T-joins. To this end, call, for any graph $G = (V, E)$ and any $T \subseteq V$, a subset C of E a *T-cut* if $C = \delta(U)$ for some $U \subseteq V$ with $|U \cap T|$ odd.

Trivially, each T-cut intersects each T-join. Moreover, each edge set C intersecting each T-join contains a T-cut (since otherwise each component of $(V, E \setminus C)$ has an even number of vertices in T, and hence there is a T-join disjoint from C). So the inclusionwise minimal T-cuts are exactly the inclusionwise minimal edge sets intersecting all T-joins. Hence the inclusionwise minimal T-joins are exactly the inclusionwise minimal edge sets intersecting all T-cuts.

Call a family $\mathcal{F}$ of cuts in $G = (V, E)$ *cross-free* if $\mathcal{F} = \{\delta(U) \mid U \in \mathcal{C}\}$ for some cross-free collection $\mathcal{C}$ of subsets of V; that is, a collection $\mathcal{C}$ with

(29.5) $\quad U \subseteq W$ or $W \subseteq U$ or $U \cap W = \emptyset$ or $U \cup W = V$

for all $U, W \in \mathcal{C}$.

The following min-max relation for minimum-size T-joins in bipartite graphs was proved by Seymour [1981d] — we give the simple proof due to Sebő [1987]:

Theorem 29.2. *Let $G = (V, E)$ be a bipartite graph and let $T \subseteq V$. Then the minimum size of a T-join is equal to the maximum number of disjoint T-cuts. The maximum is attained by a cross-free family of cuts.*

Proof. We may assume $T \neq \emptyset$. Let J be a minimum-size T-join in G. Define a length function $l : E \to \{+1, -1\}$ by: $l(e) := -1$ if $e \in J$ and $l(e) := +1$ if $e \notin J$. Then every circuit C has nonnegative length, since $J \triangle C$ is again a T-join, and hence $|J \triangle C| \geq |J|$, implying $l(C) = |C \setminus J| - |C \cap J| \geq 0$.

Let P be a minimum-length walk in G traversing no edge more than once. Choose P such that it traverses a minimum number of edges. So P is a path (as we can delete any circuit occurring in P). Let t be an end vertex of P and let f be the last edge of P.

Then $f \in J$, since otherwise we could make the walk shorter by deleting f from P. Moreover, $\deg_J(t) = 1$, as if J has another edge, e say, incident with t, then extending P by e would make the walk shorter.

We next show:

(29.6) Each circuit C traversing t and not traversing f has positive length.

If C has only vertex t in common with P, let e be the first edge of C. So $l(e) = 1$. Consider the walk $P' := P \cup (C - e)$. Then $l(P') \geq l(P)$ and hence $l(C - e) \geq 0$. So $l(C) > 0$.

If C has another vertex in common with P, let u be the last vertex on P with $u \neq t$ and traversed by C. Let P' be the $u - t$ part of P. Split C into two $u - t$ paths C' and C''. By the minimality of $|P|$, $l(P') < 0$. Hence, as $C' \cup P'$ and $C'' \cup P'$ are circuits, $l(C') > 0$ and $l(C'') > 0$. This implies $l(C) > 0$.

Now shrink $\{t\} \cup N(t)$ to one new vertex v_0, giving the graph G'. If $|T \cap (\{t\} \cup N(t))|$ is odd, let $T' := (T \setminus (\{t\} \cup N(t))) \cup \{v_0\}$, and otherwise let $T' := T \setminus (\{t\} \cup N(t))$. Let $J' := J \setminus \{f\}$.

Then J' is a minimum-size T'-join in G'. For suppose to the contrary that G' contains a circuit C' with $|C' \setminus J'| < |C' \cap J'|$. If C' comes from a circuit C in G not traversing t, we would have $|C \setminus J| < |C \cap J|$, a contradiction. So C' comes from a circuit C in G traversing t.

If C traverses f, then $|C' \setminus J'| - |C' \cap J'| = |C \setminus J| - |C \cap J| \geq 0$, a contradiction. If C does not traverse f, then, by (29.6), $l(C) > 0$, and hence $l(C) \geq 2$. So $|C' \setminus J'| = |C \setminus J| - 2 \geq |C \cap J| = |C' \cap J'|$, again a contradiction.

Now by induction (on $|V| + |T|$), G' has disjoint cross-free T'-cuts $D_1, \ldots, D_{|J'|}$. With the T-cut $\delta(t)$ this gives $|J|$ disjoint cross-free T-cuts in G. ∎

(For another, algorithmic proof, see Barahona [1990].)

This implies for not necessarily bipartite graphs (Lovász [1975a]):

Corollary 29.2a. *Let $G = (V, E)$ and let $T \subseteq V$ with $|T|$ even. Then the minimum size of a T-join is equal to half of the maximum number of T-cuts covering each edge at most twice. The maximum is attained by a cross-free family of cuts.*

Proof. Replace each edge of G by a path of length two, thus obtaining the bipartite graph G'. Applying Theorem 29.2 to G' gives the corollary. ∎

In general it is not true that the minimum size of a T-cut is equal to the maximum number of disjoint T-joins — see Section 29.11c.

Notes. Frank, Tardos, and Sebő [1984] sharpened Theorem 29.2 to the following. Let $G = (V, E)$ be a bipartite graph, with colour classes U and W, and let $T \subseteq V$. Then the minimum size of a T-join is equal to the maximum of

$$(29.7) \qquad \sum_{S \in \Pi} q_T(S),$$

where Π ranges over all partitions of U. Here $q_T(S)$ denotes the number of components K of $G - S$ with $|K \cap T|$ odd. If G is arbitrary one takes the maximum of $\frac{1}{2}\sum_{S \in \Pi} q_T(S)$ over all partitions Π of V. (For extensions, see Kostochka [1994].)

29.5. The up hull of the T-join polytope

The last corollary implies a polyhedral result due to Edmonds and Johnson [1973] (also stated by Seymour [1979b]). Let $G = (V, E)$ be a graph and let $T \subseteq V$. The *T-join polytope*, denoted by $P_{T\text{-join}}(G)$, is the convex hull of the incidence vectors of T-joins. So it is a polytope in $\mathbb{R}^E$.

We first consider the 'up hull' of $P_{T\text{-join}}(G)$, that is,

$$(29.8) \qquad P^{\uparrow}_{T\text{-join}}(G) := P_{T\text{-join}}(G) + \mathbb{R}^E_+,$$

which turns out to be determined by the system:

$$(29.9) \qquad \begin{array}{lll} \text{(i)} & x_e \geq 0 & \text{for each } e \in E, \\ \text{(ii)} & x(C) \geq 1 & \text{for each } T\text{-cut } C. \end{array}$$

Corollary 29.2b. *The polyhedron $P^{\uparrow}_{T\text{-join}}(G)$ is determined by* (29.9).

Proof. It is easy to see that $P^{\uparrow}_{T\text{-join}}(G)$ is contained in the polyhedron determined by (29.9). If the converse inclusion does not hold, there is a weight function $w \in \mathbb{Q}^E$ with $w > \mathbf{0}$ such that the minimum value of $w^{\mathsf{T}}x$ subject to (29.9) is less than the minimum weight α of any T-join. We may assume that each $w(e)$ is an even integer.

We make a new graph $G' = (V', E')$ as follows. Replace each edge $e = uv$ of G by a path from u to v of length $w(e)$. Then α is equal to the minimum size of a T-join in G'. Hence by Theorem 29.2, G' has α disjoint T-cuts. This gives a family of α T-cuts in G such that each edge e of G is in at most $w(e)$ of these T-cuts. Let y_C be the number of times that T-cut C occurs in this list. Then the y_C give a feasible dual solution to the problem of minimizing $w^{\mathsf{T}}x$ over (29.9), with value $\sum_C y_C = \alpha$. This contradicts our assumption that the minimum value of $w^{\mathsf{T}}x$ subject to (29.9) is less than α. ∎

(Gastou and Johnson [1986] gave a proof based on binary groups.)

By adding $x_e \leq 1$ for each $e \in E$ we obtain from (29.9) the system

(29.10) (i) $0 \leq x_e \leq 1$ for each $e \in E$,
(ii) $x(C) \geq 1$ for each T-cut C.

Corollary 29.2c. *The convex hull of the incidence vectors of edge sets containing a T-join as a subset is determined by* (29.10).

Proof. Directly from Corollary 29.2b, with Theorem 5.19. ∎

These systems are totally dual half-integral:

Corollary 29.2d. *Systems* (29.9) *and* (29.10) *are totally dual half-integral.*

Proof. This follows from the proof of Corollary 29.2b, observing that the y_C are integer if each w_e is an even integer. ∎

Generally these systems are not TDI, as is shown by taking $G = K_4$ and $T = V$ — see Section 29.11b.

Barahona [2002] gave an $O(n^6)$-time algorithm to decompose a vector in the up hull of the T-join polytope as a convex combination of incidence vectors of T-joins, added with a nonnegative vector.

29.6. The T-join polytope

In the previous section we considered the up hull of the T-join polytope. We can derive from it an inequality system determining the T-join polytope itself. Consider the following system of linear inequalities for $x \in \mathbb{R}^E$:

(29.11) (i) $0 \leq x_e \leq 1$ $(e \in E)$,
(ii) $x(\delta(U) \setminus F) - x(F) \geq 1 - |F|$ $(U \subseteq V, F \subseteq \delta(U), |U \cap T| + |F| \text{ odd})$.

Corollary 29.2e. *The T-join polytope is determined by* (29.11).

Proof. First, the incidence vector x of any T-join J satisfies (29.11). Indeed, if $U \subseteq V$, then $|\delta(U) \cap J| \equiv |U \cap T| \pmod 2$. Hence if $F \subseteq \delta(U)$ with $|U \cap T| + |F|$ odd, then $|\delta(U) \cap J| + |F|$ is odd, and hence if $x(F) = |F|$ one has $x(\delta(U) \setminus F) \geq 1$. This shows (29.11). So the T-join polytope is contained in the polytope determined by (29.11).

To see the reverse inclusion, choose a weight function $w \in \mathbb{Q}^E$. We show that the minimum value of $w^{\mathsf{T}}x$ subject to (29.11) is equal to $w(J)$ for some T-join J.

Define

(29.12) $$N := \{e \mid w(e) < 0\} \text{ and } T' := T \triangle \{v \mid \deg_N(v) \text{ odd}\}.$$

Let $w'(e) := |w(e)|$ for each $e \in E$. Let J' be a T'-join minimizing $w'(J')$. By Corollary 29.2c, there exist λ_U for $U \subseteq V$ with $|U \cap T'|$ odd, such that

(29.13) (i) $\lambda_U \geq 0$ for each U with $|U \cap T'|$ odd, with equality if $|J' \cap \delta(U)| > 1$,

(ii) $\displaystyle\sum_{\substack{U \\ e \in \delta(U)}} \lambda_U \leq w'(e)$ for each $e \in E$, with equality if $e \in J'$.

Define $\mu, \nu : E \to \mathbb{R}_+$ by the conditions that $\mu(e)\nu(e) = 0$ for each $e \in E$ and that

(29.14) $$\nu - \mu + \sum_U \lambda_U(\chi^{\delta(U)\setminus N} - \chi^{\delta(U)\cap N}) = w.$$

So the $\nu(e)$, $\mu(e)$, and λ_U give a feasible dual solution to the problem of minimizing $w^{\mathsf{T}}x$ subject to (29.11) (taking $F := \delta(U) \cap N$).

Let $J := J' \triangle N$. So J is a T-join. We show that J, $\mu(e)$, $\nu(e)$, λ_U satisfy the complementary slackness conditions, thus finishing our proof.

First we show that if $e \in J$, then $\nu(e) = 0$. Indeed, if $e \in J \setminus N$, then $e \in J'$, and hence

(29.15) $$\sum_{U, e \in \delta(U)\setminus N} \lambda_U - \sum_{U, e \in \delta(U)\cap N} \lambda_U$$

is equal to $w'(e) = w(e)$ by (29.13)(ii), and hence $\nu(e) = 0$. If $e \in J \cap N$, then (29.15) is at least $-w'(e) = w(e)$ by (29.13)(ii), and hence $\nu(e) = 0$.

Second we show that if $e \notin J$, then $\mu(e) = 0$. If $e \notin J \cup N$, then (29.15) is at most $w'(e) = w(e)$ by (29.13)(ii), implying $\mu(e) = 0$. If $e \in N \setminus J$, then $e \in J'$, and hence (29.15) is equal to $-w'(e) = w(e)$ by (29.13)(ii), implying $\mu(e) = 0$.

Finally if $\lambda_U > 0$, then (as $J = J' \triangle N$ and $|J' \cap \delta(U)| = 1$ by (29.13)(i))

(29.16) $$\begin{aligned} &|J \cap (\delta(U) \setminus N)| - |J \cap (\delta(U) \cap N)| \\ &= |(J' \setminus N) \cap \delta(U)| - |(N \setminus J') \cap \delta(U)| \\ &= |J' \cap \delta(U)| - |N \cap \delta(U)| = 1 - |\delta(U) \cap N|. \end{aligned}$$ ∎

In Section 29.11b we show that (29.11) is TDI if and only if G is series-parallel.

29.7. Sums of circuits

Given a graph $G = (V, E)$, the *circuit cone* is the cone in $\mathbb{R}^E$ generated by the incidence vectors of circuits. Seymour [1979b] showed that this cone is determined by:

(29.17) (i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(D) \geq 2x_e$ for each cut D and $e \in D$.

As J. Edmonds (cf. Seymour [1979b]) pointed out, this can be derived from (essentially) matching theory:

Corollary 29.2f. *The circuit cone is determined by* (29.17).

Proof. Since the incidence vector x of any circuit satisfies (29.17), the circuit cone is contained in the cone determined by (29.17).

To see the converse inclusion, let x satisfy (29.17). To show that x belongs to the circuit cone, we may assume (by scaling) that $x(E) \leq 1$. It suffices to show that x belongs to the $\emptyset$-join polytope of G. Hence, by Corollary 29.2e, it suffices to show that $x(\delta(U)) - 2x(F) \geq 1 - |F|$ for each $U \subseteq V$ and $F \subseteq \delta(U)$ with $|F|$ odd. If $|F| = 1$, this follows from (29.17)(ii). If $|F| \geq 3$, then $x(\delta(U)) - 2x(F) \geq -x(E) \geq -1 \geq 1 - |F|$. ∎

(This proof is due to Aráoz, Cunningham, Edmonds, and Green-Krótki [1983]. Hoffman and Lee [1986] gave a 'different (but not shorter) proof'. Coullard and Pulleyblank [1989] gave a short elementary proof, together with decomposition results.)

Seymour [1979b] in fact characterized when a box has a nonempty intersection with the circuit cone:

Corollary 29.2g. *Let $G = (V, E)$ be a graph and let $l, u \in \mathbb{R}^E_+$ satisfying $l \leq u$. Then there exists an x in the circuit cone of G with $l \leq x \leq u$ if and only if*

(29.18) $u(D \setminus \{e\}) \geq l(e)$ *for each cut D and each $e \in D$.*

Proof. Necessity being trivial, we show sufficiency. Choose a counterexample with $\sum_{e \in E}(u_e - l_e)$ minimum. Suppose that $u_e > l_e$ for some edge e. Then there exist a cut D and $e \in D$ with $u(D \setminus \{e\}) = l(e)$ and there exist a cut D' containing e, and $f \in D' \setminus \{e\}$ with $u(D' \setminus \{f\}) = l(f)$. Then $f \notin D$, since otherwise $e, f \in D \cap D'$, implying

(29.19)
$$u(D \triangle D') \leq u(D \setminus \{e, f\}) + u(D' \setminus \{e, f\})$$
$$= l(e) - u(f) + l(f) - u(e) < 0.$$

Hence the cut $D \triangle D'$ satisfies

(29.20)
$$u(D \triangle D' \setminus \{f\}) \leq u(D \setminus \{e\}) + u(D' \setminus \{e, f\}) = l(e) - u(e) + l(f)$$
$$< l(f),$$

contradicting (29.18).

So $u_e = l_e$ for each edge e, and hence the corollary follows from Corollary 29.2f. ∎

Let $G = (V, E)$ be a graph. A function $l : E \to \mathbb{R}$ is called *conservative* if $l(C) \geq 0$ for each circuit C. The conservative functions form a polyhedral convex cone, and Corollary 29.2f gives functions that generate this cone:

Corollary 29.2h. *The cone of conservative functions is generated by the nonnegative functions and by the functions l for which there is a subset U of V and an edge $e \in \delta(U)$ such that*

(29.21) $$l = \chi^{\delta(U)\setminus\{e\}} - \chi^e.$$

Proof. Directly by polarity (cf. Section 5.7) from Corollary 29.2f. ∎

In Section 29.11b we show that system (29.17) is TDI if and only if G is series-parallel.

29.8. Integer sums of circuits

Seymour [1979b] gave the following characterization of integer sums of circuits in planar graphs. It is equivalent to saying that the incidence vectors of circuits in a planar graph form a Hilbert base. (We follow a proof suggested by A.V. Karzanov, which starts like Seymour's proof but does not use the four-colour theorem.)

Theorem 29.3. *Let $G = (V, E)$ be a planar graph and let $x \in \mathbb{R}^E$. Then x is a nonnegative integer combination of incidence vectors of circuits if and only if x is an integer vector in the circuit cone with $x(\delta(v))$ even for each vertex v.*

Proof. Necessity being easy, we show sufficiency. Consider a counterexample with

(29.22) $$|V| + \sum_{e \in E} (x(e) + 1)^2$$

minimal. Then G is connected (otherwise one of the components forms a counterexample with (29.22) smaller), $x(e) \geq 1$ for each $e \in E$ (otherwise we can delete e), and each vertex v has degree at least 3 (the degree is at least 2 by (29.17)(ii); if it is precisely 2, then x has the same value on the two edges incident with v (by (29.17)(ii)), and hence we can replace them by one edge).

Consider any edge e_0 with $x(e_0) \geq 2$ and $x(e_0)$ minimal. Let e_0 connect vertices p and q say. Let G' be the graph obtained from G by adding a new (parallel) edge f between p and q. Define $x'(e_0) := x(e_0) - 1$, $x'(f) := 1$, and

$x'(e) := x(e)$ for all other edges e of G'. Then condition (29.17) is maintained, but sum (29.22) decreases. So x' is a sum of circuits[17] in G'. If none of these circuits consist of e_0 and f, then x is a sum of circuits in G, a contradiction. So $\{e_0, f\}$ is one of the circuits. Therefore, in G, the vector

(29.23) $\quad y := x - 2\chi^{e_0}$

is a sum of circuits, say

(29.24) $\quad y = \sum_{C \in \mathcal{C}} \lambda_C \chi^{EC},$

where $\mathcal{C}$ is a collection of circuits and where the λ_C are positive integers. Let $\mathcal{C}_0$ be the collection of circuits in $\mathcal{C}$ traversing e_0, and let $\mathcal{C}_1 := \mathcal{C} \setminus \mathcal{C}_0$.

We construct a directed graph $D = (V, A)$. We say that a circuit C *generates* a pair (u, v) of distinct vertices if C traverses both u and v, in such a way that if C traverses e_0, then C traverses p, q, u, v cyclically in this order (possibly $u = q$ or $v = p$). The arc set A of D consists of all pairs (u, v) generated by at least one $C \in \mathcal{C}$. Then:

(29.25) $\quad D$ contains a directed path from p to q.

For suppose not. Let U be the set of vertices reachable in D from p. So $q \notin U$, and no arc of D leaves U. Hence no $C \in \mathcal{C}_1$ intersects $\delta_E(U)$, and each $C \in \mathcal{C}_0$ intersects $\delta_E(U)$ precisely twice: once in e_0 and once elsewhere. So

(29.26) $\quad x(\delta_E(U) \setminus \{e_0\}) = y(\delta_E(U) \setminus \{e_0\}) = y(e_0) < x(e_0),$

contradicting (29.17). This shows (29.25).

Now choose a shortest directed $p-q$ path P in D, say $P = (v_0, v_1, \ldots, v_k)$, with $v_0 = p$ and $v_k = q$. Let $\mathcal{C}'$ be an inclusionwise minimal subcollection of $\mathcal{C}$ with the property that each arc of P is generated by some C in $\mathcal{C}'$. Define $\mathcal{C}'_0 := \mathcal{C}' \cap \mathcal{C}_0$, and

(29.27) $\quad z := 2\chi^{e_0} + \sum_{C \in \mathcal{C}'} \chi^{EC}.$

We show:

(29.28) $\quad z = x$, $\mathcal{C}' = \mathcal{C}$, and $\lambda_C = 1$ for each $C \in \mathcal{C}$.

It suffices to show that $z = x$. Suppose $z \neq x$. Then, since $z \leq x$, by the minimality of (29.22), z is a sum of circuits. To see this, it suffices to show that (29.17) is satisfied by z. To this end, let $U \subseteq V$ and $e \in \delta(U)$. If $e \neq e_0$, then (29.17)(ii) follows since $z - 2\chi^{e_0}$ is a sum of circuits. If $e = e_0$, then we can assume that $p \in U$, $q \notin U$. Hence some arc (v_{i-1}, v_i) leaves U. Let $C' \in \mathcal{C}'$ generate (v_{i-1}, v_i). Then C' has at least two edges in $\delta(U)$, and at least four if $C' \in \mathcal{C}'_0$. Moreover, any $C \in \mathcal{C}'_0$ has at least two edges in $\delta(U)$. Hence

[17] By a 'sum of circuits' we mean a sum of incidence vectors of circuits.

(29.29) $$z(\delta(U) \setminus \{e_0\}) = \sum_{C \in \mathcal{C}'} |EC \cap \delta(U) \setminus \{e_0\}| \geq |\mathcal{C}'_0| + 2 = z(e_0).$$

So (29.17) is satisfied by z. Hence z is a sum of circuits. But then also x is a sum of circuits, since

(29.30) $$\begin{aligned} x - z = y - \sum_{C \in \mathcal{C}'} \chi^{EC} &= \sum_{C \in \mathcal{C}} \lambda_C \chi^{EC} - \sum_{C \in \mathcal{C}'} \chi^{EC} \\ &= \sum_{C \in \mathcal{C} \setminus \mathcal{C}'} \lambda_C \chi^{EC} + \sum_{C \in \mathcal{C}'} (\lambda_C - 1) \chi^{EC}. \end{aligned}$$

This contradicts our assumption, proving (29.28).

Then:

(29.31) each vertex v is traversed by at most two circuits in $\mathcal{C}_1$.

Otherwise, there exist three arcs on P generated by circuits in $\mathcal{C}_1$ traversing v. Hence there exist arcs (v_{i-1}, v_i) and (v_{j-1}, v_j) on P generated by circuits C and C' in $\mathcal{C}_1$ traversing v, with $i < j - 1$. This implies that we can make P shorter (by replacing $v_i, v_{i+1}, \ldots, v_{j-1}$ by v), a contradiction. This shows (29.31).

Consider now any vertex $v \neq p, q$ and any $f \in \delta(v)$ with $x(f) \geq 2$. By the choice of e_0 we know $x(f) \geq x(e_0)$. Hence, using (29.31),

(29.32) $$2x(f) \leq x(\delta(v)) \leq 2|\mathcal{C}_0| + 4 = 2(y(e_0) + 2) = 2x(e_0) \leq 2x(f).$$

So we have equality throughout. In particular, v is traversed by precisely two circuits in $\mathcal{C}_1$, and $x(f) = x(e_0)$.

It follows that, for any $i = 1, \ldots, k-1$, the arcs (v_{i-1}, v_i) and (v_i, v_{i+1}) are generated by circuits in $\mathcal{C}_1$ (by taking $v = v_i$). Trivially, if $k = 1$, the arc (v_0, v_1) is not generated by any circuit in $\mathcal{C}_0$, and hence by some circuit in $\mathcal{C}_1$. Therefore, by the minimality of $\mathcal{C}$, $\mathcal{C}_0 = \emptyset$ and $\mathcal{C}_1 = \mathcal{C}$. Hence $y(e_0) = 0$, and so $x(e_0) = 2$. Therefore, $x(e) \in \{1, 2\}$ for each edge e.

Since each vertex $v \neq p, q$ is traversed by precisely two circuits in $\mathcal{C}$, we know that v is incident with at most one edge e with $x(e) = 2$. Since any e with $x(e) = 2$ can play the role of e_0, this also holds for $v \in \{p, q\}$. So

(29.33) the edges e with $x(e) = 2$ form a matching M in G.

Consider the path P above. Let arc (v_{i-1}, v_i) be generated by circuit $C_i \in \mathcal{C}$, for $i = 1, \ldots, k$. By the minimality of k, C_i and C_j are vertex-disjoint if $j > i + 1$. Let D_1 be the union of the EC_i for odd i, and let D_2 be the union of the EC_i for even i. So (for each $i = 1, 2$) D_i consists of vertex-disjoint circuits, and $D_1 \cap D_2 = M \setminus \{e_0\}$.

This is used in proving:

(29.34) each nonempty cut D contained in M is odd.

Indeed, by symmetry we may assume that $e_0 \in D$. Then $D \setminus \{e_0\} = D \cap D_1$ (since $D \setminus \{e_0\} \subseteq M \setminus \{e_0\} \subseteq D_1$ and since $e_0 \notin D_1$). Moreover, $|D \cap D_1|$ is even, since D_1 is a disjoint union of circuits.

This proves (29.34), which implies that

(29.35) $G - M$ has at most two components,

since if K and L are components with $K \cup L \neq V$, then at least one of $\delta_E(K)$, $\delta_E(L)$, and $\delta_E(K \cup L)$ is nonempty and even, contradicting (29.34).

Moreover:

(29.36) M forms a cut in G.

Otherwise, M has an edge spanned by a component of $G-M$. Hence G has a circuit C with $|C \cap M| = 1$ By symmetry, we may assume that $C \cap M = \{e_0\}$. Then $C \triangle D_1$ and $C \triangle D_2$ form cycles whose incidence vectors add up to x. Hence x is a sum of circuits, a contradiction. So we have (29.36).

Now let K_1 and K_2 be the components of $G - M$. They are connected Eulerian graphs. Since M forms a cut, we can assume that the attachments of M at K_1 and at K_2 are at the outer boundaries B_1 of K_1 and B_2 of K_2. By the planarity of G, the attachments of M occur in the same order on B_1 as on B_2. So $\chi^{EB_1} + \chi^{EB_2} + 2\chi^M$ is a sum of circuits. Since $EK_1 \setminus EB_1$ and $EK_2 \setminus EB_2$ are cycles, this gives x as a sum of circuits. ∎

(The proof of Seymour [1979b] of Theorem 29.3 uses the four-colour theorem. Fleischner and Frank [1990] showed that a method of Fleischner [1980] gives a proof not using the four-colour theorem. Also Alspach and Zhang [1993] gave a proof not using the four-colour theorem.)

In Theorem 29.3 we cannot delete the planarity condition, as is shown by the Petersen graph: fix a perfect matching M, and set $x_e := 2$ if $e \in M$ and $x_e := 1$ if $e \notin M$. Alspach, Goddyn, and Zhang [1994] (extending Alspach and Zhang [1993]) proved that the Petersen graph is the critical example:

Theorem 29.4. *For any graph $G = (V, E)$, the following are equivalent:*

(29.37) (i) *each integer vector x in the circuit cone with $x(\delta(v))$ even for each vertex v is a nonnegative integer combination of incidence vectors of circuits;*
(ii) *G has no Petersen graph minor.*

(This was generalized to binary matroids by Fu and Goddyn [1999] — see Section 81.9.)

Seymour [1979b] conjectures that each *even* integer vector x in the circuit cone is a nonnegative integer combination of incidence vectors of circuits. A special case of this is the *circuit double cover conjecture* (it was asked by Szekeres [1973] and conjectured by Seymour [1979b]): each bridgeless graph has circuits such that each edge is covered by precisely two of them. Thus Theorem 29.4 implies that the circuit double cover conjecture is true for graphs without Petersen graph minor.

It has been proved that for any even integer $k \geq 4$, each bridgeless graph has circuits such that each edge is covered by precisely k of them. (For $k = 6$ by Jaeger [1979] and for $k = 4$ by Fan [1992] — hence any even $k \geq 4$ follows.)

This relates to the 4-*flow conjecture* of Tutte [1966], which generalizes the four-colour theorem:

(29.38) (?) The edges of any bridgeless graph without Petersen graph minor can be covered by two Eulerian subgraphs. (?)

(It is called the 4-flow conjecture, since it is equivalent to saying that for each bridgeless graph $G = (V, E)$ without Petersen graph minor, there is an orientation $D = (V, A)$ of G and a function $f : A \to \{1, 2, 3\}$ with $f(\delta^{\text{in}}(v)) = f(\delta^{\text{out}}(v))$ for each $v \in V$ — see Section 28.4.)

Conjecture (29.38) was proved for 4-edge-connected graphs by Jaeger [1979], and for cubic graphs jointly by Robertson, Seymour, and Thomas [1997], Sanders, Seymour, and Thomas [2000], and Sanders and Thomas [2000].

(29.38) is equivalent to:

(29.39) (?) Any bridgeless graph without Petersen graph minor has two disjoint T-joins, where T is the set of vertices of odd degree (?)

(since J is a T-join if and only if $E \setminus T$ yields an Eulerian graph).

It is NP-complete to decide if a graph has two disjoint T-joins, since for cubic graphs it is equivalent to 3-edge-colourability (cf. Theorem 28.3).

Related work can be found in Zhang [1993c]. Surveys on the circuit double cover conjecture were given by Jaeger [1985], Jackson [1993], and Zhang [1993a,1993b,1997b], and on integer decomposition of the circuit cone (and more general decompositions) by Goddyn [1993].

29.9. The T-cut polytope

The *T-cut polytope* $P_{T\text{-cut}}(G)$ — the convex hull of the incidence vectors of T-cuts — is a 'hard' polytope, even if $|T| = 2$, since finding a maximum cut separating two given vertices in a graph is NP-complete. However, the up hull of the T-cut polytope:

(29.40) $$P^{\uparrow}_{T\text{-cut}}(G) := P_{T\text{-cut}}(G) + \mathbb{R}^E_+$$

is tractable, as follows directly with the theory of blocking polyhedra from the results above on the up hull of the T-join polytope, and is determined by:

(29.41)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(J) \geq 1$ for each T-join J.

Theorem 29.5. *The up hull $P^{\uparrow}_{T\text{-cut}}(G)$ of the T-cut polytope of G is determined by* (29.41)*.*

Proof. Directly with the theory of blocking polyhedra from Corollary 29.2b. ∎

This implies that the following system:

(29.42) (i) $0 \leq x_e \leq 1$ for each $e \in E$,
(ii) $x(J) \geq 1$ for each T-join J,

describes a convex hull as follows.

Corollary 29.5a. *The convex hull of the incidence vectors of edge sets containing a T-cut is determined by* (29.42)*.*

Proof. Directly from Theorem 29.5 with Theorem 5.19. ∎

(For a direct derivation from Edmonds' perfect matching polytope theorem, see Seymour [1979a].)

In general, (29.41) is not TDI, not even totally dual half-integral (Seymour [1979a]). Seymour [1977b] characterized pairs of G, T for which (29.41) is TDI — see Section 29.11c.

Rizzi [1997] showed that the minimal TDI-system for the up hull of the T-cut polytope can have arbitrarily large coefficients and right-hand sides.

29.10. Finding a minimum-capacity T-cut

Like in Section 25.5c we can find a minimum-capacity T-cut by constructing a Gomory-Hu tree (for a graph $G = (V, E)$ and a tree $H = (V, F)$, a *fundamental cut* is a cut $\delta_E(W_f)$, where $f \in F$ and W_f is a component of $H - f$):

Theorem 29.6. *Let $G = (V, E)$ be a graph and let $T \subseteq V$ with $|T|$ even. Let $c \in \mathbb{R}^E_+$ be a capacity function and let $H = (V, F)$ be a Gomory-Hu tree. Then one of the fundamental cuts of H is a minimum-capacity T-cut in G.*

Proof. For each $f \in F$, choose W_f to be one of the two components of $H - f$. Let $\delta_G(U)$ be a minimum-capacity T-cut of G. So $|U \cap T|$ is odd.

Then U or $V \setminus U$ is equal to the symmetric difference of the W_f over $f \in \delta_F(U)$. Hence $|W_f \cap T|$ is odd for at least one $f \in \delta_F(U)$. So $\delta_G(W_f)$ is a T-cut.

Let $f = uv$. As $\delta_G(W_f)$ is a minimum-capacity $u-v$ cut and as $\delta_G(U)$ is a $u-v$ cut, we have $c(\delta_G(W_f)) \leq c(\delta_G(U))$. So $\delta_G(W_f)$ is a minimum-capacity T-cut. ∎

This gives algorithmically (Padberg and Rao [1982]):

Corollary 29.6a. *A minimum-capacity T-cut can be found in strongly polynomial time.*

Proof. This follows from Theorem 29.6, since a Gomory-Hu tree can be found in strongly polynomial time, by Corollary 15.15a. ∎

Barahona and Conforti [1987] showed that a cut $\delta(U)$ with $T \cap U$ and $T \setminus U$ even and nonempty, and of minimum capacity, can be found in strongly polynomial time.

Barahona [2002] gave a combinatorial strongly polynomial-time algorithm to solve the dual of maximizing $c^{\mathsf{T}}x$ over (29.41) (yielding a fractional packing of T-joins).

29.11. Further results and notes

29.11a. Minimum-mean length circuit

Let $G = (V, E)$ be an undirected graph and let $l \in \mathbb{Q}^E$ be a length function. The *mean length* of a circuit C is equal to $l(C)/|C|$. Barahona [1993b] showed (using an argument of Cunningham [1985c]) that a minimum-mean length circuit in an undirected graph can be found in strongly polynomial time, by solving at most m Chinese postman problems.

Theorem 29.7. *A minimum-mean length circuit in an undirected graph can be found in strongly polynomial time.*

Proof. Let $G = (V, E)$ be an undirected graph and let $l \in \mathbb{Q}^E$ be a length function. Note that by adding a constant γ to all edge-lengths, the collection of minimum-mean length circuits does not change (as the mean length of any circuit increases by exactly γ). So we can assume that there exists a circuit C with $l(C) < 0$.

The algorithm is as follows:

(29.43) Find a minimum-length $\emptyset$-join J.
If $l(J) = 0$, output a circuit of length 0, and stop.
If $l(J) < 0$, add $\gamma := -l(J)/|J|$ to all edge-lengths, and iterate.

We first show that the algorithm stops; in fact, in at most $|E| + 1$ iterations. To this end, consider two subsequent iterations. Let l and l' be two subsequent length functions and let J and J' be the shortest $\emptyset$-joins found. So $l'(e) = l(e) - l(J)/|J|$ for all $e \in E$. If $l'(J') < 0$, then $|J'| < |J|$, since

$$0 > l'(J') = l(J') - \frac{l(J)}{|J|}|J'| \geq l(J) - \frac{l(J)}{|J|}|J'| = l(J)(1 - \frac{|J'|}{|J|}) \tag{29.44}$$

(note that $l(J) < 0$ and $l(J') \geq l(J)$). This shows that the algorithm stops after at most $|E| + 1$ iterations.

As throughout the iterations, the collection of minimum-mean length circuits is invariant, a minimum-mean length circuit for the final length function, is also a minimum-mean length circuit for the initial length function. Hence the output is correct.

Finally, for the 0-length circuit C in the final iteration we can take any circuit contained in the $\emptyset$-join J found in the one but last iteration (as J has length 0 in the last iteration). ∎

Barahona [1993b] also showed that, conversely, the minimum-length T-join problem can be solved by solving $O(m^2 \log n)$ minimum-mean length circuit problems, as follows. Let $l \in \mathbb{Q}^E$ be a length function. Start with any T-join J. Find a minimum-mean length circuit C for the length function l' given by: $l'(e) := -l(e)$ if $e \in T$ and $l'(e) := l(e)$ otherwise. If $l'(C) \geq 0$, then J is a T-join minimizing $l(J)$. Otherwise, reset $T := T \triangle C$, and iterate.

(We note here that Guan [1960] proposed to find a circuit C minimizing $l'(C)$ and iteratively reset T as above, until $l'(C) \geq 0$. It is however NP-complete to find such a circuit, and moreover, no polynomial upper bound on the number of iterations is known.)

Barahona [1993b] also observed that the minimum-mean length circuit problem can be solved by solving a 'compact' linear programming problem (that is, one in which the number of variables and constraints is bounded by a polynomial in the size of the graph).

This follows from the fact that, for any graph $G = (V, E)$, the convex hull of

$$\text{(29.45)} \qquad \{\frac{1}{|C|}\chi^C \mid C \text{ circuit}\}$$

(where χ^C is the incidence vector of C in $\mathbb{R}^E$) consists of all vectors x in the circuit cone of G satisfying $\mathbf{1}^\mathsf{T} x = 1$; moreover, by Corollary 29.2f, x belongs to the circuit cone of G if and only if for each edge $e = st$ there exists an $s-t$ flow $y \leq x$ in $G-e$ of value x_e. Here the flow is described on the directed graph obtained from $G-e$ by replacing each edge uv by two arcs (u, v) and (v, u). As the flows are determined by flow conservation constraints (next to the negativity and capacity constraints), this yields a compact linear program.

A minimum mean-weight circuit therefore can be found in polynomial time with any polynomial-time LP-algorithm.

29.11b. Packing T-cuts

System (29.9) generally is not TDI, as is shown by taking $G = K_4$ and $T = VK_4$. This example is the critical example, since Seymour [1977b] showed that if system (29.9) is not TDI, then G, T contains K_4, VK_4 as a 'minor' — see Corollary 29.9b below. To prove this, we follow the approach of Frank and Szigeti [1994] using the results of Sebő [1988b].

Each polyhedron is determined by a TDI-system, albeit not necessarily the minimal system defining the polyhedron. Sebő [1988b] showed that system (29.9) can be extended as follows to a TDI-system defining the up hull of the T-join polytope.

Let $G = (V, E)$ be a graph and let T be an even-size subset of V. Call a set B of edges a *T-border* if there exists a partition $\mathcal{P} = (U_1, \ldots, U_k)$ of V such that

$|U_i \cap T|$ is odd for each i and such that B is equal to the set of edges connecting distinct classes of $\mathcal{P}$. The *value* $\mathrm{val}(B)$ of the T-border B is, by definition, half of the number of components K of $G - B$ with $|K \cap T|$ odd. (This is at least $\frac{1}{2}k$.) So a T-border is a T-cut if and only if $\mathrm{val}(B) = 1$. Moreover, each T-join intersects any T-border B in at least $\mathrm{val}(B)$ edges. Hence the minimum size of any T-join is at least the maximum total value of any packing of T-borders. (The *total value* of a collection of T-borders is the sum of the values of the T-borders in the collection.) Sebő [1988b] showed that the minimum and maximum are equal:

Theorem 29.8. *Let $G = (V, E)$ be a graph and let $T \subseteq V$. Then the minimum size of a T-join is equal to the maximum total value of a packing of T-borders.*

Proof. Choose a counterexample with $|V|$ as small as possible. Then G is connected.

By Corollary 29.2a, it suffices to show that the maximum total value of a packing of T-borders is at least half of the maximum size of a 2-packing of T-cuts[18] Choose a maximum-size 2-packing of T-cuts $\delta(U_1), \ldots, \delta(U_k)$, which by Corollary 29.2a we may assume to be cross-free. We must find a packing of T-borders of total value $\frac{1}{2}k$.

We choose the U_i such that

$$(29.46) \qquad \sum_{i=1}^{k} |U_i|$$

is as small as possible. In particular, $|U_i| \leq |V \setminus U_i|$ for each i.

For each such 2-packing we have

$$(29.47) \qquad \delta(U_i) \neq \delta(U_j) \text{ if } i \neq j,$$

since otherwise we can contract the edges in $\delta(U_i)$ to obtain G', T' and apply induction. We obtain a packing of T'-borders in G', of total value $\frac{1}{2}(k-2)$. Together with the T-border $B := \delta(U_i)$ this gives a packing of T-borders in G of total value $\frac{1}{2}k$. This shows (29.47).

We next show

$$(29.48) \qquad |U_i| = 1 \text{ for each } i.$$

Suppose not. Choose an inclusionwise minimal set U_i with $|U_i| > 1$. So for any j, if $U_j \subset U_i$, then $U_j = \{t\}$ for some $t \in T \cap U_i$. Moreover, for each $t \in T \cap U_i$, there is a j with $U_j = \{t\}$, since otherwise we could reset $U_i := \{t\}$, contradicting the minimality of the sum (29.46). Then $U_i \subseteq T$, since otherwise we can replace U_i by $T \cap U_i$, again contradicting the minimality of the sum (29.46). It follows that the union of the $\delta(t)$ for $t \in U_i$ forms a T-border B of value $\frac{1}{2}(|U_i| + 1)$. Contracting the edges in B gives G', T' say. Applying induction to G', T' (in which there exists a 2-packing of T'-cuts of size $k - (|U_i| + 1)$), gives a packing of T'-borders in G' of total value $\frac{1}{2}(k - |U_i| - 1)$. Adding B, gives a packing of T-borders in G of total value $\frac{1}{2}k$.

So we can assume that $|U_i| = 1$ for each i. Then the union of the $\delta(U_i)$ for $i = 2, \ldots, k$ forms a T-border of value $\frac{1}{2}k$. ∎

This theorem bears upon the system

[18] A 2-*packing* is a family of sets such that no element is in more than two of them.

(29.49) (i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(B) \geq \text{val}(B)$ for each T-border B.

Since each inequality (29.9)(ii) occurs among (29.49), and since, conversely, each inequality (29.49)(ii) is a half-integer sum of inequalities (29.9)(ii), the two systems (29.9) and (29.49) define the same polyhedron — namely $P^{\uparrow}_{T\text{-join}}(G)$. In fact:

Corollary 29.8a. *System* (29.49) *is TDI.*

Proof. For any weight function $w \in \mathbb{Z}_+^E$ we can replace any edge $e = uv$ by a $u - v$ path of length $w(e)$, contracting e if $w(e) = 0$. Applying Theorem 29.8 to the new graph gives an integer optimum dual solution to the problem of minimizing $w^{\mathsf{T}}x$ subject to (29.49). ∎

We next use Theorem 29.8 to show that system (29.9) is TDI if G, T contains no K_4, VK_4 as a 'minor'. We follow the line of proof given by Frank and Szigeti [1994]. We first prove the following.

Call a graph $G = (V, E)$ *bicritical* if $G - u - v$ has a perfect matching for each pair of distinct vertices u and v. Call a graph $G = (V, E)$ *oddly contractible* to K_4 if V can be partitioned into four odd sets V_1, V_2, V_3, V_4 such that $G[V_i \cup V_j]$ is connected for all i, j (also if $i = j$). The following result is due to A. Sebő (cf. Frank and Szigeti [1994]):

Theorem 29.9. *A bicritical graph with at least four vertices is oddly contractible to* K_4.

Proof. Let $G = (V, E)$ be a bicritical graph with $|V| \geq 4$. This immediately implies that G is connected and has a perfect matching M. Moreover,

(29.50) for all $u, v \in V$ with $u \neq v$ there is an odd-length M-alternating $u - v$ path $P_{u,v}$ with first and last edge not in M.

To see this, first assume that $uv \in M$. Then there is a perfect matching N not containing uv (since there exists an edge uw with $w \neq v$ (by the connectedness of G), and hence the perfect matching of $G = \{u, w\}$ together with uw forms a perfect matching). Let C be the circuit in $M \cup N$ containing uv. Then $C - uv$ is a path as required in (29.50).

If $uv \notin M$, let u' and v' be such that $uu' \in M$ and $vv' \in M$. Let N be a perfect matching in $G - u' - v'$. Then $(M \cup N) \setminus \{uu', vv'\}$ contains a $u - v$ path as required. This shows (29.50).

Now (29.50) implies:

(29.51) there exists an odd-length M-alternating circuit $C = (v_0, v_1, \ldots, v_t)$.

(So t is odd, and $v_i v_{i+1} \in M$ if and only if i is odd.) To see (29.51), choose edges $uv \in M$ and $vw \notin M$. Then $P_{u,w}$ does not traverse v (otherwise uv is on $P_{u,w}$). So $C := EP_{u,w} \cup \{uv, vw\}$ is a circuit as required in (29.51).

Let w be such that $wv_0 \in M$. Let K be the component of $G - VC$ containing w. So $N(K) \subseteq VC$. We first show that $|N(K)| \geq 3$. Indeed, first we have $v_0 \in N(K)$. Let s be the first vertex in P_{w,v_1} contained in VC. Then $s \neq v_0$, since otherwise $v_0 w \in EP_{w,v_1}$. Let s' be such that $ss' \in M$. So $s' \neq v_0$. Let r be the first vertex in

$P_{w,s'}$ contained in VC. Again $r \neq v_0$. Moreover, $r \neq s$, since otherwise $ss' \in EP_{w,s'}$ (implying that the last edge of $P_{w,s'}$ is in M, a contradiction). As $v_0, s, r \in N(K)$, we have $|N(K)| \geq 3$.

As K is the union of w with a number of edges in M, $|K|$ is odd. Similarly, any other component of $G - VC$ is even. As C is an odd circuit, VC can be partitioned into three paths with an odd number of vertices, each containing a neighbour of K. Hence G is oddly contractible to K_4. ∎

We define deletion, contraction, and minor for pairs G, T. Let $G = (V, E)$ be a graph, $T \subseteq V$, and $e = uv \in E$. We say that $G - e, T$ arises from G, T by *deleting* e. Let G/e be the graph obtained from G by contracting e. Denote the new vertex to which e is contracted by v^e. Define $T' := T \setminus \{u, v\}$ if $|T \cap \{u, v\}|$ is even, and $T' := (T \setminus \{u, v\}) \cup \{v^e\}$ if $|T \cap \{u, v\}|$ is odd. Then we say that $G/e, T'$ arises from G, T by *contracting* e.

We say that the pair G', T' is a *minor* of the pair G, T if G', T' arises from G, T by a series of deletions and contractions of edges, and of deletions of vertices not in T. Then the following is a special case of a more general hypergraph theorem of Seymour [1977b] (Theorem 80.1):

Corollary 29.9a. *Let $G = (V, E)$ be a graph and let $T \subseteq V$ with $|T|$ even, such that K_4, VK_4 is not a minor of G, T. Then the minimum size of a T-join is equal to the maximum number of disjoint T-cuts.*

Proof. By Theorem 29.8, the minimum size of a T-join is equal to the maximum total value of a packing of T-borders. Consider such an optimum packing, with the number of T-borders as large as possible. If each T-border is a T-cut, we are done. So assume that one of the T-borders, B say, has value at least 2. Let $\mathcal{P} = (U_1, \ldots, U_k)$ be a partition of V with $|U_i \cap T|$ odd for each i and such that B is the union of the $\delta(U_i)$.

Let $G' = (V', E'), T'$ be obtained from G, T by contracting each U_i to one vertex. So $T' = V'$. As G', T' contains no K_4, VK_4 as a minor, G is not bicritical, by Theorem 29.9. Hence there are distinct $u, v \in V'$ such that $G' - u - v$ has no perfect matching. By Tutte's 1-factor theorem this implies that there is a subset U of V' with $u, v \in U$ and with $o(G' - U) \geq |U|$. Take such a U with $|U|$ maximal. Then each component of $G' - U$ is odd. (Otherwise, we can add an element of some even component to U, contradicting the maximality of $|U|$.)

For each component K of $G' - U$, the set of edges of G' incident with K form a V'-border in G' of value $\frac{1}{2}(|K|+1)$. So G' has a packing of V'-borders of total value $|V' \setminus U| + o(G' - U) \geq |V'| = k$. Since $|U| \geq 2$ (as $u, v \in U$), we have $o(G' - U) \geq 2$, so there are at least two such components. Hence the packing contains at least two V'-borders. Decontracting the U_i gives a decomposition of B into a packing of at least two T-borders, of total value k. This contradicts the maximality of the number of T-borders in the original packing. ∎

This can be formulated equivalently in terms of total dual integrality. Note that total dual integrality of system (29.9) is closed under taking minors: deletion of an edge e corresponds to intersection with the hyperplane $H := \{x \mid x_e = 0\}$, while contracting e corresponds to projecting on H. Hence total dual integrality of (29.9) can be characterized by forbidden minors; in fact, there is only one forbidden minor:

Corollary 29.9b. *System* (29.9) *is totally dual integral if and only if* G, T *has no minor* K_4, VK_4.

Proof. To see necessity, it suffices to show that if $G = K_4$ and $T = VK_4$, then (29.9) is not TDI. Taking $w_e := 1$ for each $e \in EG$, the minimum weight of a T-join equals 2, while each two T-cuts intersect, implying that there is no integer optimum dual solution.

To see sufficiency, let K_4, VK_4 not be a minor of $G = (V, E), T$. Let $w \in \mathbb{Z}_+^E$. Let G', T' arise from G, T by replacing each edge e by a path of length w_e, contracting e if $w_e = 0$. Then also G', T' has no minor K_4, VK_4. Moreover, the minimum weight k of a T-join in G is equal to the minimum size of a T'-join in G'. By Corollary 29.9a, G' contains a T'-cut packing of size k. So G contains k T-cuts such that each edge e of G is in at most $w(e)$ of them. This gives an integer optimum dual solution to the problem of minimizing $w^{\mathsf{T}}x$ subject to (29.9). ∎

This implies a characterization of series-parallel graphs:

Corollary 29.9c. *The following are equivalent for any graph* $G = (V, E)$*:*

(29.52)
(i) *G is series-parallel;*
(ii) (29.9) *is TDI for each choice of T;*
(iii) (29.11) *is TDI for each choice of T;*
(iv) (29.11) *is TDI for some choice of T;*
(v) (29.17) *is TDI.*

Proof. The equivalence of (i) and (ii) follows from Corollary 29.9b, since a graph G is series-parallel if and only if G has no K_4 minor. The implication (iii)⇒(iv) is direct.

We next show (v)⇒(ii). Let (29.17) be TDI. Choose $T \subseteq V$ and $w \in \mathbb{Z}_+^E$. Let J be a T-join minimizing $w(J)$. Define $\tilde{w}(e) := w(e)$ if $e \in E \setminus J$ and $\tilde{w}(e) := -w(e)$ if $e \in J$. Then $\emptyset$ is a $\tilde{w}$-minimal $\emptyset$-join. Since (29.17) is TDI, there exist $\lambda_{U,e} \in \mathbb{Z}_+$ for $U \subseteq V$ and $e \in \delta(U)$ with

(29.53) $$\tilde{w} \geq \sum_{U,e} \lambda_{U,e}(\chi^{\delta(U)\setminus\{e\}} - \chi^e).$$

Choose the $\lambda_{U,e}$ such that $\sum_{U,e} \lambda_{U,e}$ is minimized. Then

(29.54) if $\lambda_{U,e} \geq 1$ and $\lambda_{U',e'} \geq 1$, then $e' \notin \delta(U) \setminus \{e\}$.

Otherwise, if $e \in \delta(U')$, then

(29.55) $$(\chi^{\delta(U)\setminus\{e\}} - \chi^e) + (\chi^{\delta(U')\setminus\{e'\}} - \chi^{e'})$$

is nonnegative, and hence we can decrease $\lambda_{U,e}$ and $\lambda_{U',e'}$ by 1, without violating (29.53), contradicting our minimality assumption.

If $e \notin \delta(U')$, then $e \in \delta(U \triangle U')$. Also, (29.55) is at least

(29.56) $$\chi^{\delta(U\triangle U')\setminus\{e\}} - \chi^e,$$

and hence we can decrease $\lambda_{U,e}$ and $\lambda_{U',e'}$ by 1, and increase $\lambda_{U\triangle U',e}$ by 1, without violating (29.53), again contradicting our minimality assumption.

This shows (29.54). So there are no two $\lambda_{U,e} \geq 1$ and $\lambda_{U',e'} \geq 1$ such that the vectors $\chi^{\delta(U)\setminus\{e\}} - \chi^e$ and $\chi^{\delta(U')\setminus\{e'\}} - \chi^{e'}$ have opposite signs in some position. The minimality of $\sum \lambda_{U,e}$ then implies that $\sum \lambda_{U,e} = -\tilde{w}(J) = w(J)$ and that $J \cap \delta(U) = \{e\}$ for each U, e with $\lambda_{U,e} \geq 1$. So each such $\delta(U)$ is a T-cut. Moreover,

(29.57) $$w \geq \sum_{U,e} \lambda_{U,e}\chi^{\delta(U)}.$$

So we have an integral dual solution for the problem of minimizing $w^{\mathsf{T}}x$ over (29.9). This proves (v)⇒(ii).

We next show the reverse implication (ii)⇒(v). Let (29.9) be TDI for each choice of T. To prove that (29.17) is TDI, choose $w \in \mathbb{Z}^E$, such that minimizing $w^{\mathsf{T}}x$ over (29.17) is finite — that is (as (29.17) determines the circuit cone) $w(C) \geq 0$ for each circuit C.

Define $J := \{e \in E \mid w(e) < 0\}$ and $T := \{v \in V \mid \deg_J(v) \text{ is odd}\}$. Moreover, $\tilde{w}(e) := |w(e)|$ for $e \in E$. Then J is a T-join minimizing $\tilde{w}(J)$ (as $w(C) \geq 0$ for each circuit C). Hence, as (29.9) is TDI, there exist $\lambda_U \in \mathbb{Z}_+$ for U with $T \cap U$ odd, such that

(29.58) $$\sum_U \lambda_U \chi^{\delta(U)} \leq \tilde{w} \text{ and } \sum_U \lambda_U = \tilde{w}(J).$$

For each U with $\lambda_U \geq 1$ one has $|J \cap \delta(U)| = 1$; let e_U be the edge in $J \cap \delta(U)$. Then

(29.59) $$w \geq \sum_U \lambda_U(\chi^{\delta(U)\setminus e_U} - \chi^{e_U}),$$

proving total dual integrality of (29.17).

Finally we show (v)⇔(iii)⇔(iv). Consider any $T \subseteq V$ and any vertex χ^J of the T-join polytope, determined by T-join J. Total dual integrality of (29.11) in χ^J means that the following system is TDI:

(29.60) $$\begin{array}{ll} x_e \geq 0 & \text{for each } e \in E \setminus J, \\ x_e \leq 1 & \text{for each } e \in J, \\ x(H) - x(F) \geq 1 - |F|, & \text{for each } U \subseteq V \text{ and partition } F, H \text{ of} \\ & \delta(U) \text{ with } |U \cap T| + |F| \text{ odd and} \\ & |H \cap J| + |F \setminus J| = 1. \end{array}$$

The condition $|H \cap J| + |F \setminus J| = 1$ implies that there exists an edge $e \in \delta(U)$ with $F = (\delta(U) \cap J) \triangle \{e\}$ and $H = (\delta(U) \setminus J) \triangle \{e\}$.

Setting $\tilde{x}_e := 1 - x_e$ if $e \in J$ and $\tilde{x}_e := x_e$ if $e \in E \setminus J$, (29.60) is equivalent to:

(29.61) $$\begin{array}{l} \tilde{x}_e \geq 0 \text{ for } e \in E, \\ \tilde{x}(H \setminus J) + |H \cap J| - \tilde{x}(H \cap J) - \tilde{x}(F \setminus J) - |F \cap J| + \tilde{x}(F \cap J) \geq 1 - |F| \end{array}$$

for each U, F, H as described in (29.60). The second line in (29.61) is equivalent to:

(29.62) $$\tilde{x}(H \triangle (J \cap \delta(U))) - \tilde{x}(F \triangle (J \cap \delta(U))) \geq 1 - |F \triangle (J \cap \delta(U))|.$$

and hence to

(29.63) $$\tilde{x}(\delta(U) \setminus \{e\}) - \tilde{x}_e \geq 0,$$

where $\{e\} := F\triangle(J \cap \delta(U))$. As this equivalence holds for any fixed T, this proves both (iv)$\Rightarrow$(v) and (v)$\Rightarrow$(iii). ∎

(Korach [1982] gave an algorithmic proof of this corollary.)

Sebő [1988b] also characterized the minimal TDI-system for the polyhedron $P^{\uparrow}_{T\text{-join}}(G)$. Call a T-border B *reduced* if $B = \delta(U_1) \cup \cdots \cup \delta(U_k)$ for some partition $\mathcal{P} = (U_1, \ldots, U_k)$ of V such that $|U_i \cap T|$ is odd and $G[U_i]$ is connected for each i and such that the graph obtained by contracting each U_i to one vertex is bicritical. Then the following is a minimal TDI-system for connected graphs:

(29.64) (i) $x_e \geq 0$ for each edge e for which $\{e\}$ is not a T-cut,
(ii) $x(B) \geq \mathrm{val}(B)$ for each reduced T-border B.

Sebő [1993c] showed that for each fixed k, the problem of finding a maximum integer packing of T-cuts subject to a capacity constraint is polynomial-time solvable if $|T| = k$. The method uses that integer linear programming is polynomial-time solvable in fixed dimension (Lenstra [1983]).

29.11c. Packing T-joins

In the previous section we considered packing T-cuts, which relates to the total dual integrality of system (29.9). We now consider packing T-joins, which relates to the total dual integrality of system (29.41).

System (29.41) generally is not TDI. Indeed, let G be the graph $K_{2,3}$ and let $T_0 := VK_{2,3} \setminus \{v_0\}$, where v_0 is one of the two vertices of degree 3 in $K_{2,3}$. Then the minimum size of a T_0-cut in $K_{2,3}$ is equal to 2, while there are no two disjoint T_0-joins. This again is the critical example, as follows again from a more general hypergraph theorem of Seymour [1977b] (Theorem 80.1). For this special case, we follow the line of proof given by Codato, Conforti, and Serafini [1996].

Theorem 29.10. *Let $G = (V, E)$ be a graph and let $T \subseteq V$, such that $K_{2,3}, T_0$ is not a minor of G, T. Then the minimum size of a T-cut is equal to the maximum number of disjoint T-joins.*

Proof. Let G, T form a counterexample, with $|V| + |E|$ as small as possible. Let k be the minimum size of a T-cut. Then trivially G is connected. Moreover:

(29.65) any T-cut C of size k satisfies $C = \delta(t)$ for some $t \in T$.

Indeed, let $C = \delta(U)$ for $U \subseteq V$ with $|U \cap T|$ odd and $|C| = k$. Assume that $1 < |U| < |V| - 1$. Then $G[U]$ is connected, since otherwise there would exist a T-cut smaller than k. Similarly, $G - U$ is connected.

Now contract U to one vertex v', yielding minor G', T' of G, T. The minimum size of a T'-cut in G' equals k. As $|VG'| < |VG|$, we know that G' has k disjoint T'-joins. Each of them intersects $\delta_{G'}(v')$ in exactly one edge (as it is a T'-cut of size k).

We can contract $V \setminus U$ to one vertex v'', yielding minor G'', T'' of G, T. Again, G'' has k disjoint T''-joins, each intersecting $\delta_{G''}(v'')$ in exactly one edge.

Using the one-to-one correspondence between $\delta_{G'}(v')$ and $\delta_{G''}(v'')$, we can glue the two collections of joins together, to obtain k disjoint T-joins in G, contradicting our assumption. This gives (29.65).

Let $T' := \{t \in T \mid \deg(t) = k\}$. Then (29.65) implies that

(29.66) each edge of G intersects T'.

Otherwise we could delete the edge without decreasing the minimum size of a T-cut, by (29.65). This would give a smaller counterexample, contradicting our assumption.

We next have:

(29.67) $|V \setminus T'| \geq 2$.

For suppose $|V \setminus T'| \leq 1$. We know that G has $k-1$ disjoint T-joins (by the minimality of $|V|+|E|$ — otherwise deleting any edge would give a smaller counterexample). Let F be the union of these T-joins. Then $\deg_F(v)$ is even if $v \notin T$ while $\deg_F(v) \equiv k-1 \pmod 2$ if $v \in T$. Hence $\deg_{E\setminus F}(v)$ is odd for each $v \in T'$. As $|V \setminus T'| \leq 1$ it follows that $E \setminus F$ is a T-join, and hence G would have k disjoint T-joins. This contradicts our assumption, and proves (29.67).

Then

(29.68) there is no subset U of T' with $|U| \leq 2$ and $G-U$ disconnected.

Suppose not. If $|U|=1$, let $U=\{t\}$ for $t \in T'$. Then $|K \cap T|$ is odd for some component K of $G-t$. As $G-t$ is disconnected, $|\delta(K)| < \deg(t) = k$, contradicting the fact that $\delta(K)$ is a T-cut.

If $|U|=2$, let $U=\{t,t'\}$ for $t,t' \in T'$. Choose a component K of $G-U$ not contained in T'. Let l (l', respectively) be the number of edges connecting K and t (K and t', respectively). If $|K \cap T|$ is odd, then $l+l' = |\delta(K)| > k$ and hence $|\delta(K \cup \{t,t'\})| \leq (k-l)+(k-l') < k$, a contradiction. If $|K \cap T|$ is even, then $l' + (k-l) = |\delta(K \cup \{t\})| > k$, and similarly $l+(k-l') > k$, a contradiction. This proves (29.68).

Now choose $u \in V \setminus T'$. As $N(u) \subseteq T'$ (by (29.66)), by (29.68) we know $|N(u)| \geq 3$. Choose a component K of $G' := G - (\{u\} \cup N(u))$, with $|N(K)|$ as small as possible. (K exists by (29.67).) If possible, we take K such that moreover $|K \cap T|$ is odd.

Again by (29.68), $|N(K)| \geq 3$. Choose $t_1,t_2,t_3 \in N(K)$. Then

(29.69) for any component $L \neq K$ of G' with $N(L)=\{t_1,t_2,t_3\}$ one has $|L \cap T|$ even.

For suppose that $|L\cap T|$ is odd. By the minimality of $|N(K)|$, we know $N(K) = \{t_1,t_2,t_3\}$. Moreover, $|K \cap T|$ is odd. Let k_i be the number of edges connecting K and t_i and let l_i be the number of edges connecting L and t_i, for $i=1,2,3$. Then $k_1+k_2+k_3 = |\delta(K)| \geq k$, and similarly $l_1+l_2+l_3 \geq k$. This gives the contradiction

(29.70) $k < |\delta(K \cup L \cup \{t_1,t_2,t_3\})| \leq (k-k_1-l_1)+(k-k_2-l_2)+(k-k_3-l_3) \leq k$

(the first inequality follows from (29.65)). This shows (29.69).

Now contract the union of $\{u\} \cup (N(u) \setminus \{t_1,t_2,t_3\})$ and all components $L \neq K$ of G' with $N(L) \neq \{t_1,t_2,t_3\}$ to one vertex u'. Moreover, contract the union of $\{t_1\}$ and all components $L \neq K$ of G' with $N(L) = \{t_1,t_2,t_3\}$ to one vertex t_1'. Finally contract K to one vertex u''. This gives minor G'',T'' of G,T.

So G'' has vertices u',u'',t_1',t_2,t_3, with each of u',u'' adjacent to each of t_1',t_2,t_3 (possibly there are more adjacencies). Each of t_1',t_2,t_3 belongs to T''. As $|T''|$ is even, exactly one of u',u'' belongs to T''. Hence G,T has minor $K_{2,3},T_0$, a contradiction. ∎

This implies the characterization:

Corollary 29.10a. *System* (29.41) *is TDI if and only if* $K_{2,3}, T_0$ *is not a minor of* G, T.

Proof. Necessity follows from the fact that total dual integrality of (29.41) is maintained under taking minors (contraction of an edge e corresponds to intersecting the polytope with the hyperplane $x_e = 0$, and deletion of e corresponds to projecting on it), while the minimum size of a T_0-cut in $K_{2,3}$ is 2, and $K_{2,3}$ has no two disjoint T_0-joins.

To see sufficiency, let $w \in \mathbb{Z}_+^E$. Replace any edge $e = uv$ of G by $w(e)$ parallel edges connecting u and v, yielding the graph G'. Then the minimum weight of a T-cut in G is equal to the minimum size of a T-cut in G'. By Theorem 29.10, this minimum size is equal to the maximum number of disjoint T-joins in G'. These T-joins give an integer optimum dual solution to the problem of minimizing $w^{\mathsf{T}}x$ subject to (29.41). ∎

Generally, system (29.41) is not totally dual half-integral, as is shown by the following example of Seymour [1979a]. Let $G' = (V', E')$ be a connected bridgeless cubic graph with $\chi'(G) = 4$ and with an even number of edges. (For instance, G' is the Petersen graph with one vertex replaced by a triangle (in such a way that the three vertices adjacent to it in the Petersen graph, now each are adjacent to one of the vertices in the triangle).)

Let $G = (V, E)$ be obtained from G' by replacing each edge by a path of length 2. So $|V|$ is even.

Then trivially the minimum size of a V-cut is equal to 2. However, the maximum number of V-joins covering each edge at most twice is equal to 3. For suppose that there exist four V-joins $J_1, \ldots, J_4$ covering each edge at most twice. Since each edge of G is incident with a vertex of degree two, each edge of G is covered exactly twice by the J_i. For $i = 1, 2, 3$, let $C_i := J_i \triangle J_4$. Then each C_i is a vertex-disjoint union of circuits, and each edge of G is in exactly two of the C_i. Then the complements of the C_i form edge-disjoint V'-joins in G. This would yield a 3-edge-colouring of G' — a contradiction.

If we replace each edge of G by two parallel edges, thus obtaining an Eulerian graph, the minimum size of a V-cut equals 4, whereas the maximum number of disjoint V-joins is 3.

If Seymour's 'generalized Fulkerson conjecture' (see Section 28.5) is true, there exists a $\frac{1}{4}$-integer packing (that is, the minimum size of a T-cut is equal to one quarter of the maximum size of a 4-packing of T-joins); in other words, the total dual quarter-integrality of the T-join constraints (29.41) follows — we give the proof of Seymour [1979a] of this derivation.

Proof that the generalized Fulkerson conjecture implies the total dual quarter-integrality of the T-join constraints. Let $G = (V, E)$ be a graph and let $T \subseteq V$. Let k be the minimum size of a T-cut. We must show that the generalized Fulkerson conjecture implies:

(29.71) there exist T-joins $J_1, \ldots, J_{4k}$ covering each edge of G at most four times.

First assume that $T = V$. We show:

(29.72) if each vertex of G has even degree, then there exist V-joins $J_1, \ldots, J_{2k}$ covering each edge of G at most twice.

To see this, assume that each vertex of G has even degree. So k is even. If $k \leq 2$, (29.72) is trivial. (If $k = 2$ there exists a V-join J; then the complement $E \setminus J$ is a V-join again.) So we can assume that $k \geq 4$.

For each $v \in V$, let G_v be a $(k-1)$-edge-connected graph with $\deg_G(v) + 1$ vertices, one of degree k and all other vertices of degree $k-1$. (Such graphs G_v exist: If $k = 4$, take any cubic 3-edge-connected graph on $\deg_G(v) + 2$ vertices (for instance, by taking a circuit on $\deg_G(v) + 2$ vertices, and making opposite vertices adjacent), and contract an arbitrary edge of it. If $k \geq 6$, add a Hamiltonian circuit to the graph for the case $k - 2$.)

We take the G_v vertex-disjoint. Now transform G to a graph H, by replacing each vertex v by G_v, and making each edge of G which was incident with v, incident instead with one of the $\deg_G(v)$ vertices of G_v of degree $k-1$, in such a way that the resulting graph H is k-regular.

We show that H is a k-graph, by showing

(29.73) $|\delta_H(U)| \geq k$ for each $U \subseteq VH$ with $|U|$ odd.

To see this, assume $|\delta_H(U)| < k$. Observe that $|\delta_H(U)|$ is even, as k is even and H is k-regular. Hence $|\delta_H(U)| \leq k-2$. Since each G_v is $(k-1)$-edge-connected, for each $v \in V$ we know that either $VG_v \subseteq U$ or $VG_v \cap U = \emptyset$. Define

(29.74) $X := \{v \in V \mid VG_v \subseteq U\}$.

Then $|\delta_H(U)| = |\delta_G(X)|$. Moreover, $|X|$ is odd as $|VG_v|$ is odd for each $v \in V$. Therefore $|\delta_G(X)| \geq k$ and hence $|\delta_H(U)| \geq k$. This shows (29.73).

Then by the generalized Fulkerson conjecture, there exist perfect matchings $M_1, \ldots, M_{2k}$ in H covering each edge of H exactly twice. Projecting these matchings to the original edges of G, gives V-joins as required in (29.72).

Now, for $T = V$, (29.71) follows from (29.72) by replacing each edge of G by two parallel edges. The case of general T can be reduced to the case $T = V$ as follows. Let T be arbitrary. For each vertex $v \in V \setminus T$, make a new vertex v', connected by k parallel edges with v. This gives the graph $G' = (V', E')$. Then the minimum size of a V'-cut in G' is equal to k. Hence by (29.71) there exist V'-joins $J'_1, \ldots, J'_{4k}$ in G' covering each edge of G' at most four times. Restricting the J'_i to the edges of G, gives V-joins in G as required.

Cohen and Lucchesi [1997] showed that conjecture (29.72) is equivalent to: if all T-cuts have the same parity, then the maximum size of a 2-packing of T-joins is equal to twice the minimum size of a T-cut. They also showed that this is true if $|T| \leq 8$; more strongly, that if $|T| \leq 8$ and all T-cuts have the same parity, then the maximum number of disjoint T-joins is equal to the minimum size of a T-cut.

29.11d. Maximum joins

Let $G = (V, E)$ be a graph. Call a subset J of E a *join* if $|J \cap C| \leq \frac{1}{2}|C|$ for each circuit C; that is, $|J \triangle C| \geq |C|$ for each circuit C. This can be expressed in terms of the length function $l_J : E \to \{-1, +1\}$, defined by

(29.75) $$l_J(e) := \begin{cases} -1 & \text{if } e \in J, \\ +1 & \text{if } e \notin J. \end{cases}$$

So

(29.76) $l_J(F) = |F \triangle J| - |J|$

for each $F \subseteq E$. Then a set J is a join if and only if $l_J(C) \geq 0$ for each circuit C. Note also that

(29.77) a set J is a join if and only if it is a minimum-size T-join for $T := \{v \in V \mid \deg_J(v) \text{ odd}\}$.

Frank [1990b,1993b] gave a min-max relation for the maximum size of a join. By Corollary 29.2a and (29.77), the maximum size of a join is equal to the maximum size of a fractional packing of T-cuts, taken over $T \subseteq V$ with $|T \cap K|$ even for each component K of G. This, however, is not a min-max relation.

A min-max relation can be described in terms of ear-decomposition. Let $G = (V, E)$ be an undirected graph. An *ear* of G is a path or circuit P in G, of length ≥ 1, such that all internal vertices of P have degree 2 in G. The path may consist of a single edge — so any edge of G is an ear.

If I is the set of internal vertices of an ear P, we say that G arises from $G - I$ by *adding ear*. An *ear-decomposition* of G is a series of graphs $G_0, G_1, \ldots, G_k$, where $G_0 = K_1$, $G_k = G$, and G_i arises from G_{i-1} by adding an ear ($i = 1, \ldots, k$).

A graph $G = (V, E)$ has an ear-decomposition if and only if G is 2-edge-connected (see Theorem 15.17). Moreover, the number of ears in any ear-decomposition is equal to $|E| - |V| + 1$. Then the min-max relation for maximum-size join in 2-connected graphs is formulated as:

Theorem 29.11. *Let $G = (V, E)$ be a 2-edge-connected graph. Then the maximum size of a join is equal to the minimum value of*

(29.78) $$\sum_{i=1}^{k} \lfloor \tfrac{1}{2}|EP_i| \rfloor$$

taken over all ear-decompositions $(P_1, \ldots, P_k)$ of G.

Proof. We first show that the maximum is not more than the minimum. Let J be a join in G and let $\Pi = (P_1, \ldots, P_k)$ be an ear-decomposition of G. Let $G' = (V', E')$ be the graph made by $P_1, \ldots, P_{k-1}$ and let $J' := J \cap E'$. By induction we know

(29.79) $$|J'| \leq \sum_{i=1}^{k-1} \lfloor \tfrac{1}{2}|EP_i| \rfloor.$$

If $|J \cap EP_k| \leq \lfloor \frac{1}{2}|EP_k| \rfloor$ we are done. So assume that $|J \cap EP_k| > \lfloor \frac{1}{2}|EP_k| \rfloor$; that is, $l_J(P_k) < 0$. Let u and v be the end vertices of P_k. Let Q be a $u - v$ path in G' minimizing $l_{J'}(Q)$. So $l_J(P_k) + l_J(Q) \geq 0$ (since J is a maximum-size join). Since $l_{J'}(Q) = |J' \triangle EQ| - |J'|$, Q minimizes $|J' \triangle EQ|$.

Then $J'' := J' \triangle EQ$ is again a join in G', since for any circuit C in G':

(29.80) $|J'' \triangle C| = |J' \triangle (EQ \triangle C)| \geq |J' \triangle EQ| = |J''|$

(since Q minimizes $|J' \triangle EQ|$). Moreover,

(29.81) $$\begin{aligned} &|J''| - |J'| = |J' \triangle EQ| - |J'| = l_J(Q) \geq -l_J(P_k) \\ &= |J \cap EP_k| - |EP_k \setminus J| \geq |J \cap EP_k| - \lfloor \tfrac{1}{2}|EP_k| \rfloor. \end{aligned}$$

Hence, by induction applied to J'',

(29.82) $$|J| = |J'| + |J \cap EP_k| \leq |J''| + \lfloor \tfrac{1}{2}|EP_k| \rfloor \leq \sum_{i=1}^{k} \lfloor \tfrac{1}{2}|EP_i| \rfloor.$$

This shows that the maximum is not more than the minimum. To see equality, for any graph G let $\beta(G)$ be the maximum size of a join in G. For any ear-decomposition $\Pi = (P_1, \ldots, P_k)$, let $\sigma(\Pi) := \sum_{i=1}^{k} \lfloor \frac{1}{2}|EP_i| \rfloor$. Let $\pi(G)$ be the minimum of $\sigma(\Pi)$ over all ear-decompositions Π of G. So we must prove $\beta(G) = \pi(G)$. Call an ear-decomposition Π *optimum* if it minimizes $\sigma(\Pi)$.

We first show:

(29.83) Let $U \subseteq V$ with $G[U]$ 2-edge-connected. Then $\pi(G) \leq \pi(G[U]) + \pi(G/U)$.

To see this, first observe that if $G[U]$ has a Hamiltonian circuit C, then an optimum ear-decomposition Π' of $G[U]$ is obtained by first taking C, and next adding the remaining edges as ears. Now in any optimum ear-decomposition Π'' of G/U, we can insert Π' at the first ear of Π'' containing the vertex into which U is contracted (by splitting C appropriately). In this way we obtain an ear-decomposition Π of G with $\sigma(\Pi) \leq \sigma(\Pi') + \sigma(\Pi'')$.

If $G[U]$ has no Hamiltonian circuit, let Π' be an optimum ear-decomposition of $G[U]$. Let C be its first ear. By the above, $\pi(G) \leq \pi(G[VC]) + \pi(G/VC)$. Also, by induction, $\pi(G/VC) \leq \pi((G[U])/VC) + \pi(G/U)$. As C is the first ear of Π', we have $\pi(G[VC]) + \pi((G[U])/VC) = \pi(G[U])$. Combining, we get $\pi(G) \leq \pi(G[U]) + \pi(G/U)$, showing (29.83).

Next we state:

(29.84) if G is factor-critical, then $\pi(G) \leq \lfloor \frac{1}{2}|VG| \rfloor$.

This follows directly from Theorem 24.9, since $\lfloor \frac{1}{2}|EP_i| \rfloor$ is at most $\frac{1}{2}$ the number of internal vertices of P_i.

In particular, it follows that if G is factor-critical, then $\beta(G) = \pi(G)$, as G has a join of size $\lfloor \frac{1}{2}|VG| \rfloor$, namely a matching. So we can assume that G is not factor-critical.

A graph G is called *matching-covered* if each edge of G is contained in a perfect matching. By Theorem 24.10,

(29.85) if G is matching-covered and 2-edge-connected, then $\pi(G) \leq \frac{1}{2}|VG|$.

For any subset W of V let H_W be the graph obtained from $G[W \cup N(W)]$ by deleting all edges in $N(W)$ and contracting all edges in W. (H_W may have parallel edges.) So H_W is a bipartite graph with colour classes $N(W)$ and $\kappa(W) :=$ the set of components of $G[W]$.

(29.86) There is a nonempty subset W of V such that each component of $G[W]$ is factor-critical and such that H_W is 2-edge-connected and matching-covered.

To see this, we first observe that there is a nonempty subset X of V such that each component of $G[X]$ is factor-critical and such that H_X has a matching M covering $N(X)$. Indeed, if G has no perfect matching, then we can take $X := D(G)$ (= the set of vertices v for which G has a maximum-size matching missing v). By Corollary

24.7a, X has the required properties. If G has a perfect matching, call it M. Choose $u \in V$, and let $X := D(G-u)$. Then X has the required properties (note that the vertex matched in M to u belongs to $D(G-u)$).

Having X and M, orient the edges in M in the direction from $\kappa(X)$ to $N(X)$, and all other edges of H_X in the direction from $N(X)$ to $\kappa(X)$. This gives a directed graph, that has (like any directed graph) a strong component L such that no arc enters L. Let W be the union of those components of $G[X]$ whose contraction belong to L. Since no arc leaves L, for any edge $e = uv \in M$, if $u \in N(X)$ and $u \in L$, then $v \in W$. Conversely, if $v \in W$, then $u \in L$. For let $v \in K \in L$. As G is 2-edge-connected, there exists an edge $f \neq e$ leaving K. As $K \in L$ and no arc enters L, both ends of f belong to L. As L is strongly connected, f belongs to a directed circuit. Necessarily, e is in this directed circuit. So both ends of e are in L.

Hence the edges of M intersecting W, form a perfect matching M' in H_W, and so $|N(W)| = |\kappa(W)|$. Moreover, consider any edge e of H_W not in M. In H_X, e is oriented from $N(W)$ to $\kappa(W)$, and hence, as L is a strong component, it is contained in a directed circuit. This directed circuit forms an M'-alternating circuit in H_W, implying that e belongs to a perfect matching in H_W. So H_W is matching-covered. Finally H_W is 2-edge-connected, as it has a strongly connected orientation, since L is a strong component. This shows (29.86).

Define $U := W \cup N(W)$. Then (29.83), (29.84), (29.85), and (29.86) imply

$$\begin{aligned}(29.87)\qquad \pi(G) &\leq \pi(G/U) + \pi(G[U]) \leq \pi(G/U) + \pi(H_W) + \sum_{K\in\kappa(W)} \pi(G[K]) \\ &\leq \pi(G/U) + \tfrac{1}{2}|VH_W| + \sum_{K\in\kappa(W)} \lfloor \tfrac{1}{2}|K| \rfloor \leq \pi(G/U) + \tfrac{1}{2}|U|.\end{aligned}$$

On the other hand, we have

$$(29.88)\qquad \beta(G) \geq \beta(G/U) + \tfrac{1}{2}|U|.$$

Indeed, let $G' := G/N(W)$. Then trivially, $\beta(G) \geq \beta(G')$. The contracted $N(W)$ forms a cut vertex v_0 in G', and so $\beta(G')$ is equal to the sum of the $\beta(G'[K \cup \{v_0\}])$ over all components K of $G - v_0$. Now for each component K of $G[W]$ we have $\beta(G'[K \cup \{v_0\}]) \geq \frac{1}{2}(|K| + 1)$, since $G'[K \cup \{v_0\}]$ has a perfect matching (as K is factor-critical), which is a join. Since $G[W]$ has $|N(W)|$ components, this proves (29.88).

Hence the theorem follows by induction. ∎

The proof gives a polynomial-time algorithm to find a maximum-size join and an ear-decomposition minimizing (29.78).

In Section 24.4d we saw that a graph is factor-critical if and only if it has an ear-decomposition with odd ears only. This can be generalized to (where G/F arises from G by contracting all edges in F):

Theorem 29.12. *Let $G = (V, E)$ be a 2-edge-connected graph. Then the minimum number of even ears in an ear-decomposition of G is equal to the minimum size of a subset F of E with G/F factor-critical.*

Proof. First let $P_1, \ldots, P_k$ be an ear-decomposition of G. Choose one edge from each even ear. This gives a set F with G/F factor-critical, by Theorem 24.9.

Conversely, let $F \subseteq E$ with G/F factor-critical and $|F|$ minimum. By Theorem 24.9, G/F has an ear-decomposition $(P_1, \ldots, P_k)$ with odd ears only. Then we can partition F into $F_1, \ldots, F_k$ such that $P_1 \cup F_1, \ldots, P_k \cup F_k$ is an ear-decomposition of G. This ear-decomposition has at most $|F|$ even ears. ∎

We can derive from this a characterization of the maximum size of a join in any graph:

Corollary 29.12a. *Let $G = (V, E)$ be a connected graph. Then the maximum size $\beta(G)$ of a join is equal to*

(29.89) $$\tfrac{1}{2}(\phi(G) + |V| - 1).$$

where $\phi(G)$ is the minimum size of a subset F of E with G/F factor-critical.

Proof. If G has a cut edge e, the corollary follows by applying induction to G/e, since $\beta(G) = \beta(G/e) + 1$ and $\phi(G) = \phi(G/e) + 1$.

So we can assume that G is 2-edge-connected, and then the corollary follows from Theorem 29.11, with Theorem 29.12. Note that

(29.90) $$\begin{aligned}&\sum_{i=1}^{k} \lfloor \tfrac{1}{2}|EP_i| \rfloor = \tfrac{1}{2}(\text{number of even ears} + \sum_{i=1}^{k}(|EP_i| - 1))\\ &= \tfrac{1}{2}(\text{number of even ears} + |V| - 1).\end{aligned}$$ ∎

For 2-edge-connected bipartite graphs we have:

Corollary 29.12b. *Let $G = (V, E)$ be a 2-edge-connected bipartite graph, with colour classes U and W. Then the maximum size of a join is equal to the minimum number of edges oriented towards U in any strongly connected orientation of G.*

Proof. To see that the maximum is not more than the minimum, consider any strongly connected orientation of G, yielding the directed graph D. By Theorem 6.9, D has an ear-decomposition $(P_1, \ldots, P_k)$. Any ear P_i contains at least $\lfloor \frac{1}{2}|EP_i| \rfloor$ edges oriented towards U. So the sum (29.78) is at most the total number of edges oriented towards U. Hence by Theorem 29.11, the maximum is not more than the minimum.

To see equality, consider an ear-decomposition $P_1, \ldots, P_k$ of G minimizing (29.78). In any ear P_i, we can orient the edges so as to obtain a directed path, with exactly $\lfloor \frac{1}{2}|EP_i| \rfloor$ edges oriented towards U. This gives a strongly connected orientation with $\sum_i \lfloor \frac{1}{2}|EP_i| \rfloor$ edges oriented towards U. So Theorem 29.11 gives equality. ∎

We can derive some more min-max relations for bipartite graphs. Seymour [1981d] observed that Theorem 29.2 is equivalent to:

Theorem 29.13. *Let $G = (V, E)$ be bipartite and let $J \subseteq E$. Then J is a join if and only if there exist $|J|$ disjoint cuts each intersecting J in exactly one edge.*

Proof. By Theorem 29.2, using (29.77). ∎

This implies a max-max relation for the maximum size of a join in bipartite graphs:

Corollary 29.13a. *Let G be bipartite. Then the maximum size of a join is equal to the maximum number of disjoint nonempty cuts.*

Proof. Directly from Theorem 29.13. ∎

Hence, with Corollary 29.12b, a result of D.H. Younger follows (cf. Frank [1993b]):

Corollary 29.13b. *Let G be a 2-edge-connected bipartite graph, with colour classes U and W. Then the minimum number of edges oriented towards U in any strongly connected orientation of G is equal to the maximum number of disjoint nonempty cuts in G.*

Proof. From Corollaries 29.13a and 29.12b. ∎

Frank, Tardos, and Sebő [1984] showed the following. Let G be a 2-edge-connected bipartite graph, with colour classes U and W. Then the minimum number of edges oriented towards U in any strongly connected orientation of G is equal to the maximum value of

$$\sum_{S \in \Pi} \kappa(G - S), \tag{29.91}$$

ranging over all partitions Π of U, where $\kappa(H)$ denotes the number of components of H.

For an extension, see Kostochka [1994]. Szigeti [1996] gave a weighted version, based on matroids. Fraenkel and Loebl [1995] showed that it is NP-complete to find the maximum size of a subset J of the edge set E of a graph G with $l_J(C) < \frac{1}{2}|EC|$ for each circuit C (even if G is planar and bipartite). Connected joins were investigated by Sebő and Tannier [2001].

29.11e. Odd paths

We saw in Section 29.2 that the problem of finding a shortest $s-t$ path in an undirected graph $G = (V, E)$, with length function $l : E \to \mathbb{Q}$ can be solved in polynomial time, if each circuit has nonnegative length. This is by reduction to the weighted matching problem.

As J. Edmonds (cf. Grötschel and Pulleyblank [1981]) observed, another problem reducible to the weighted matching problem is: given a graph $G = (V, E)$ and a length function $l : E \to \mathbb{Q}_+$, find a shortest odd $s-t$ path. Here a path is *odd* if it has an odd number of edges.

This reduction is as follows: make a copy $G' = (V', E')$ of G, and a copy $l' : E' \to \mathbb{Q}_+$ of l, add edges vv' for each $v \in V$ (where v' is the copy of v), each of length 0. Call the extended graph H. Then a minimum-length odd $s-t$ path in G can be found by finding a minimum-length perfect matching M in $H - s' - t'$: let N be the perfect matching $\{vv' \mid v \in V\}$ in H; then the component of $M \cup N$ containing s and t gives a shortest odd $s-t$ path in G.

Next consider the following polyhedron Q in $\mathbb{R}^E$:

(29.92) $$Q := \text{conv.hull}\{\chi^P \mid P \text{ odd } s-t \text{ path}\} + \mathbb{R}_+^E$$

and its blocking polyhedron

(29.93) $$B(Q) = \{x \in \mathbb{R}_+^E \mid x(P) \geq 1 \text{ for each odd } s-t \text{ path } P\}.$$

By the above method, one can optimize over Q in polynomial time. Hence, with the ellipsoid method, one can decide if a given $x \in \mathbb{Q}^E$ belongs to Q or not, and if not, find a separating facet. This also implies that for given capacity function $c : E \to \mathbb{Q}_+$, one can find in polynomial time a fractional packing of odd $s-t$ paths subject to c, of maximum value (by minimizing $c^\mathsf{T} x$ over $B(Q)$).

Schrijver and Seymour [1994] considered the problem (raised by Grötschel [1984]) of finding an explicit system of inequalities describing Q; equivalently, of describing the vertices of $B(Q)$.

Call a subset F of E *odd-blocking* if each odd $s-t$ path contains an edge in F. For each $F \subseteq E$, define $h_F \in \mathbb{Z}_+^E$ as follows, where $e = uv \in E$ and $W_F := \{s,t\} \cup \{v \in V \mid v \text{ is incident with at least one edge in } E \setminus F\}$:

(29.94) $$h_F(e) := \begin{cases} 2 & \text{if } u, v \in W_F \text{ and } e \in F, \\ 1 & \text{if exactly one of } u, v \text{ belongs to } W_F, \\ 0 & \text{otherwise.} \end{cases}$$

In other words,

(29.95) $$h_F = \sum_{v \in W_F} \chi^{\delta(v) \cap F}.$$

In particular, $h_F(e) = 0$ if $e \notin F$.

Note that for each $x \in \mathbb{Z}_+^E$ one has:

(29.96) $h_F^\mathsf{T} x \geq 1$ for each odd-blocking $F \iff$ there exists an odd $s-t$ path P with $\chi^P \leq x \iff h_F^\mathsf{T} x \geq 2$ for each odd-blocking F.

Then Schrijver and Seymour [1994] proved:

(29.97) Let $l : E \to \mathbb{Z}_+$ be a length function such that each circuit and each $s-t$ path has even length. Then the minimum length of an odd $s-t$ path is equal to the maximum value of $2k$ for which there exist odd-blocking sets $F_1, \ldots, F_k$ with $h_{F_1} + \cdots + h_{F_k} \leq l$.

This implies:

(29.98) Let $l : E \to \mathbb{Z}_+$ be a length function. Then the minimum length of an odd $s-t$ path is equal to the maximum value of k for which there exist odd-blocking $F_1, \ldots, F_k$ with $\frac{1}{2}h_{F_1} + \cdots + \frac{1}{2}h_{F_k} \leq l$.

This can be formulated in terms of LP-duality. Let $\mathcal{F}$ be the collection of odd-blocking sets and let H be the $\mathcal{F} \times E$ matrix whose Fth row equals h_F (for $F \in \mathcal{F}$). Then (29.98) states that for $l : E \to \mathbb{Z}_+$:

(29.99) $$\min\{l^\mathsf{T} x \mid x \in \mathbb{Z}_+^E, (\tfrac{1}{2}H)x \geq \mathbf{1}\} = \max\{y^\mathsf{T}\mathbf{1} \mid y \in \mathbb{Z}_+^{\mathcal{F}}, y^\mathsf{T}(\tfrac{1}{2}H) \leq l^\mathsf{T}\}.$$

Equivalently, the system

(29.100) $$\begin{array}{ll} x_e \geq 0 & e \in E, \\ \frac{1}{2}h_F^\mathsf{T} x \geq 1 & F \text{ odd-blocking,} \end{array}$$

determines Q and is TDI. Hence:

(29.101) each vertex of $B(Q)$ is equal to $\frac{1}{2}h_F$ for some odd-blocking $F \subseteq E$

(this implies the conjecture of W.J. Cook and A. Sebő that the vertices of $B(Q)$ are half-integer).

Minimizing $c^{\mathsf{T}}x$ over $B(Q)$ then gives the following. Let $G = (V, E)$ be an undirected graph, let $s, t \in V$, and let $c : E \to \mathbb{R}_+$. Then the maximum value of a fractional packing of odd $s - t$ paths subject to c is equal to the minimum value of

(29.102) $$\frac{1}{2} \sum_{v \in W_F} c(\delta(v) \cap F),$$

taken over odd-blocking $F \subseteq E$.

L. Lovász asked for the complexity of the following combination of two of the problems above: given a graph $G = (V, E)$, vertices $s, t \in V$, and a length function $l : E \to \mathbb{Q}$, such that each circuit has nonnegative length, find a shortest odd $s - t$ path.

29.11f. Further notes

Complexity survey for all-pairs shortest paths in undirected graphs without negative-length circuits ($*$ indicates an asymptotically best bound in the table):

$*$	$O(nm \log n)$	Gabow [1983a]
$*$	$O(n^3)$	Gabow [1983a]

(The algorithm proposed by Bernstein [1984] fails (for instance, for a graph with four vertices).)

Karzanov [1986] gave an $O(|T|m \log n + |T|^3 \log |T|)$-time algorithm to find a shortest T-join and a maximum fractional packing of T-cuts.

It is easy to see that the vertices of $P^{\uparrow}_{T\text{-join}}(G)$ are the incidence vectors of the inclusionwise minimal T-joins (that is, those T-joins that are a forest). Indeed, consider a T-join J. If J contains another T-join J' as subset, then $\chi^{J'} \leq \chi^J$, and hence χ^J is not a vertex of $P^{\uparrow}_{T\text{-join}}(G)$. Conversely, if χ^J is not a vertex, then $\chi^J \geq x$ for some convex combination x of incidence vectors T-joins. Each of these T-joins J' satisfies $\chi^{J'} \leq \chi^J$, and hence $J' \subseteq J$.

Similarly, an inequality $x(C) \geq 1$ for a T-cut C determines a facet if and only if C is an inclusionwise minimal T-cut.

Giles [1981] showed that two inclusionwise minimal T-joins J and J' give adjacent vertices of the polyhedron $P^{\uparrow}_{T\text{-join}}(G)$ if and only if $J \cup J'$ contains exactly one circuit. It implies that the distance of J and J' in $P^{\uparrow}_{T\text{-join}}(G)$ is at most $|J \setminus J'|$ — this implies the Hirsch conjecture for $P^{\uparrow}_{T\text{-join}}(G)$.

Gerards [1992b] showed the following. For any graph H, an *odd-H* is a subdivision of H such that each odd circuit of H becomes an odd circuit in the subdivision. In other words, the edges of H that become an even-length path form a cut in H. The *prism* is the complement of the 6-circuit C_6. Let $G = (V, E)$ be a graph not containing an odd-K_4 or an odd-prism as subgraph. Then for each $T \subseteq V$, the minimum size of a T-join is equal to the maximum number of disjoint T-cuts. This generalizes Corollary 29.9c and Theorem 29.2.

Call a graph $G = (V, E)$ a *Seymour graph* if for each subset T of V for which there exists a T-join, the minimum-size of a T-join is equal to the maximum number of disjoint T-cuts. Ageev, Kostochka, and Szigeti [1995,1997] showed that G is a Seymour graph if and only if for each length function $l \in \mathbb{Z}^E$ with $l(C) \geq 0$ for each circuit C, and for each pair of circuits C_1 and C_2 with $l(C_1) = 0$ and $l(C_2) = 0$, the graph formed by $C_1 \cup C_2$ is neither an odd-K_4 nor an odd-prism. (Here sufficiency was proved by A. Sebő.)

Seymour [1981d] characterized for which pairs G, T with $|T| = 4$, the minimum size of a T-join is equal to the maximum number of disjoint T-cuts. In fact, let $T = \{t_1, t_2, t_3, t_4\}$ and let $k \in \mathbb{Z}_+$. Then there is a packing of T-cuts of size k if and only if

(29.103)
$$\begin{aligned}
&\mathrm{dist}(t_1, t_2) + \mathrm{dist}(t_3, t_4) \geq k,\\
&\mathrm{dist}(t_1, t_3) + \mathrm{dist}(t_2, t_4) \geq k,\\
&\mathrm{dist}(t_1, t_4) + \mathrm{dist}(t_2, t_3) \geq k,
\end{aligned}$$

such that if equality holds in each of these inequalities, then $\mathrm{dist}(t_1, t_2) + \mathrm{dist}(t_1, t_3) + \mathrm{dist}(t_2, t_3)$ is even.

Korach [1982] characterized such pairs for $|T| = 6$, and gave a polynomial-time algorithm recognizing them.

The existence of T-joins satisfying given upper bounds on the degrees can be characterized by reduction to Tutte's 1-factor theorem (cf. Ning [1987]).

Middendorf and Pfeiffer [1990b,1993] showed that it is NP-complete to decide, for given planar graph $G = (V, E)$ and $T \subseteq V$, if the minimum size of a T-join is equal to the maximum number of disjoint T-cuts. As a minimum-size T-join can be found in polynomial time, it follows that it is NP-complete to determine a maximum packing of T-cuts. (Related results are given by Korach and Penn [1992], Korach [1994], and Granot and Penn [1995].)

The *directed* Chinese postman problem can be solved as a minimum-cost circulation problem (see Section 12.5b). The *mixed* Chinese postman problem (with directed and undirected edges) however is NP-complete (Papadimitriou [1976]). Guan [1984] derived from this that the *windy* (or *asymmetric*) *postman problem* (where the length of an edge may depend on the direction in which it is traversed) is NP-complete.

Edmonds and Johnson [1973] showed that the mixed Chinese postman problem in which each vertex has even total degree is polynomial-time solvable. (The *total degree* of a vertex v is the total number of edges (directed and undirected) incident with v.) Similarly, Guan and Pulleyblank [1985] and Win [1989] showed that the windy postman problem is solvable in polynomial time if the graph is Eulerian (by reduction to a minimum-cost circulation problem). More on the windy postman can be found in Grötschel and Win [1992], Pearn and Li [1994], and Raghavachari and Veerasamy [1999b].

For approximation algorithms for the mixed postman problem, see Frederickson [1979] and Raghavachari and Veerasamy [1998,1999a]. Further work on the mixed postman problem is reported in Kappauf and Koehler [1979], Minieka [1979], Brucker [1981], Christofides, Benavent, Campos, Corberán, and Mota [1984], Ralphs [1993], and Nobert and Picard [1996].

An extension of the Edmonds-Gallai decomposition to T-joins was given by Sebő [1990b] (cf. Sebő [1986,1997]). Goemans and Williamson [1992,1995a] gave a fast 2-approximative algorithm for finding a shortest T-join.

Benczúr and Fülöp [2000] give fast algorithms for finding minimum-size T-cut, with generalization to directed graphs.

Tobin [1975] studied finding a negative-length circuit with Edmonds' algorithm. For more on packing T-joins, see Rizzi [1997]. For surveys on T-joins and T-cuts, see Sebő [1988a] and Frank [1996a].

29.11g. On the history of the Chinese postman problem

In a paper in Chinese in *Acta Mathematica Sinica*, entitled (in translation) 'Graphic programming using odd or even points', Guan [1960] introduced the problem of finding a shortest postman route:

> When the author was plotting a diagram for a mailman's route, he discovered the following problem: "A mailman has to cover his assigned segment before returning to the post office. The problem is to find the shortest walking distance for the mailman."

(In a footnote it is mentioned that 'In postal service, a mailman's route is called a segment'.) Next:

> This problem can be reduced to the following: "Given a connected graph in the plane, we are to draw a continuous graph (repetition permitted) from a given point and back minimizing the number of repeated arcs."

So Guan restricted himself to planar graphs. He observed that a postman never has to traverse any edge more than twice. Hence the problem amounts to finding a minimum-length set J of edges such that adding a parallel edge to each of them, gives an Eulerian graph. He next gave an algorithm, which consist of starting with any such set J, and next iteratively improving it by finding a circuit C such that the length of $J \cap C$ is larger than half of the length of C, and replacing J by $J \triangle C$. As in each iteration the length of J decreases, the method finds a shortest route after a finite number of steps.

In a review in Mathematical Reviews of the article of Guan [1960], Fulkerson [1964a] observed:

> Unfortunately, the construction involves examining all simple cycles to see whether the minimality test is met or not, and this is easier said than done.

Therefore, Edmonds [1965e] announced a better method in an abstract for the 27th National Meeting of the Operations Research Society of America (May 1965 in Boston):

> We present an algorithm which does not involve examining simple cycles. It is "good" in the sense that the amount of work in applying it is at worst moderately algebraic, relative to the size of the graph, rather than exponential. It combines two earlier known algorithms: (1) the well-known "shortest path" algorithm, (2) a recent algorithm for "maximum matching".

The name of the problem seems to occur first in the title of this abstract: 'The Chinese Postman's Problem' (where 'The Chinese's Postman Problem' would be more appropriate).

Chapter 30

2-matchings, 2-covers, and 2-factors

The results on matchings are strongly self-refining, as was pointed out by Tutte [1952,1954b] and Edmonds and Johnson [1970,1973]. In this chapter we see a first instance of this phenomenon. By splitting vertices, results on 2-matchings can be derived from those on ordinary matchings. 2-matchings are of interest for the traveling salesman problem.

30.1. 2-matchings and 2-vertex covers

Let $G = (V, E)$ be an undirected graph. A 2-*matching* is a vector $x \in \mathbb{Z}_+^E$ satisfying $x(\delta(v)) \leq 2$ for each vertex v. A 2-*vertex cover* is a vector $y \in \mathbb{Z}_+^V$ such that $y_u + y_v \geq 2$ for each edge uv of G. Defining the *size* of a vector as the sum of its entries, we denote:

(30.1) $\nu_2(G) :=$ the maximum size of a 2-matching in G,
$\tau_2(G) :=$ the minimum size of a 2-vertex cover in G.

Note that

(30.2) $\tau_2(G) = \min\{|V \setminus S| + |N(S)| \mid S \subseteq V, S \text{ stable set}\}$,

since for a minimum-size 2-vertex cover y, the set $S := \{v \in V \mid y_v = 0\}$ is a stable set, while $N(S) = \{v \in V \mid y_v = 2\}$, and since $\chi^{V \setminus S} + \chi^{N(S)}$ is a 2-vertex cover for each stable set S.

Note also that

(30.3) $\nu(G) \leq \frac{1}{2}\nu_2(G) \leq \frac{1}{2}\tau_2(G) \leq \tau(G)$.

The following is a special case of a theorem of Gallai [1957,1958a,1958b] (cf. Theorem 31.7), and can be derived from Kőnig's matching theorem.

Theorem 30.1. $\nu_2(G) = \tau_2(G)$ *for any graph G. That is, the maximum size of a 2-matching is equal to the minimum size of a 2-vertex cover.*

Proof. Make for each vertex v of G a new vertex v', and replace each edge uv of G by two edges $u'v$ and uv'. This makes the bipartite graph H. By Kőnig's matching theorem (Theorem 16.2), H has a vertex cover C and a matching M with $|C| = |M|$. For any edge $e = uv$ of G let $x_e := |\{u'v, uv'\} \cap M|$ and for any vertex v of G let $y_v := |\{v, v'\} \cap C|$. Then x is a 2-matching and y is a 2-vertex cover with $x(E) = |M| = |C| = y(V)$. ∎

This construction was given by Nemhauser and Trotter [1975]. It also yields a polynomial-time reduction of the problems of finding a maximum-size 2-matching and a minimum-size 2-vertex cover to the problems of finding a minimum-size matching and a maximum-size vertex cover in a bipartite graph — hence these problems are polynomial-time solvable.

Call a 2-matching x *perfect* if $x(\delta(v)) = 2$ for each vertex v. So a 2-matching x is perfect if and only if $x(E) = |V|$. Theorem 30.1 implies a characterization of the existence of a perfect 2-matching (Tutte [1952]):

Corollary 30.1a. *Let $G = (V, E)$ be a graph. Then G has a perfect 2-matching if and only if $|N(S)| \geq |S|$ for each stable set S.*

Proof. Directly from Theorem 30.1, since G has a perfect 2-matching $\iff$ $\nu_2(G) \geq |V| \iff \tau_2(G) \geq |V|$. With (30.2), this last is equivalent to the condition of the present corollary. ∎

As finding a perfect 2-matching can be reduced to finding a maximum-size 2-matching, it is polynomial-time solvable.

30.2. Fractional matchings and vertex covers

Any vector $x \in \mathbb{R}^E$ satisfying

(30.4)
$$\begin{array}{lll} \text{(i)} & x_e \geq 0 & \text{for } e \in E, \\ \text{(ii)} & x(\delta(v)) \leq 1 & \text{for } v \in V, \end{array}$$

is called a *fractional matching.* The maximum size $x(E)$ of a fractional matching is called the *fractional matching number*, denoted by $\nu^*(G)$. By linear programming duality, $\nu^*(G)$ is equal to the *fractional vertex cover number* $\tau^*(G)$ — the minimum size of a *fractional vertex cover*, which is any solution $y \in \mathbb{R}^V$ of

(30.5)
$$\begin{array}{lll} \text{(i)} & 0 \leq y_v \leq 1 & \text{for } v \in V, \\ \text{(ii)} & y_u + y_v \geq 1 & \text{for } uv \in E. \end{array}$$

The equality $\nu^*(G) = \tau^*(G)$ also follows from Theorem 30.1, since trivially

(30.6)
$$\tfrac{1}{2}\nu_2(G) \leq \nu^*(G) \leq \tau^*(G) \leq \tfrac{1}{2}\tau_2(G).$$

(An extension to infinite graphs was given by Aharoni and Ziv [1990].)

30.3. The fractional matching polytope

Let $G = (V, E)$ be a graph. The *fractional matching polytope* of G is the polytope determined by (30.4). Balinski [1965] showed:

Theorem 30.2. *Each vertex of the fractional matching polytope of G is half-integer.*

Proof. Let x be a vertex of the fractional matching polytope. We can assume that $x_e > 0$ for each edge e, since if $x_e = 0$ we can apply induction to $G - e$. Hence we can assume also that $x_e < 1$ for each edge e; equivalently, that each vertex of G has degree at least two.

As x is a vertex, there are $|E|$ constraints among (30.4)(ii) satisfied with equality. So $|E| \leq |V|$, implying that G is 2-regular. Then $x_e = \frac{1}{2}$ for each $e \in E$, as it is a solution to setting (30.4)(ii) to equality, and as the solution must be unique (as x is a vertex). ∎

Balinski [1965] also observed that the support of any vertex x of the fractional matching polytope can be partitioned into a matching M, with $x_e = 1$ for $e \in M$, and a set of odd circuits, vertex-disjoint and disjoint from M, with $x_e = \frac{1}{2}$ for each edge e in any of the odd circuits.

30.4. The 2-matching polytope

The 2-*matching polytope* of G is the convex hull of the 2-matchings in G. Theorem 30.2 implies a characterization of the 2-matching polytope (Edmonds [1965b]):

Corollary 30.2a. *The* 2*-matching polytope is determined by:*

(30.7)
$$\begin{array}{lll} \text{(i)} & x_e \geq 0 & \text{for } e \in E, \\ \text{(ii)} & x(\delta(v)) \leq 2 & \text{for } v \in V. \end{array}$$

Proof. Directly from Theorem 30.2, since it implies that the vertices of the polytope determined by (30.7) are integer, and hence are 2-matchings. ∎

Given a graph $G = (V, E)$, the *perfect* 2*-matching polytope* of G is the convex hull of the perfect 2-matchings in G. As the perfect 2-matching polytope is a face of the 2-matching polytope (if nonempty), Corollary 30.2a implies (Edmonds [1965b]):

Corollary 30.2b. *The perfect* 2*-matching polytope is determined by*

(30.8)
$$\begin{array}{lll} \text{(i)} & x_e \geq 0 & \text{for } e \in E, \\ \text{(ii)} & x(\delta(v)) = 2 & \text{for } v \in V. \end{array}$$

Proof. Directly from Corollary 30.2a. ∎

Pulleyblank [1987] related the vertices of the 2-matching polytope with the Edmonds-Gallai decomposition of the graph.

Similar results as for fractional matchings and 2-matchings hold for fractional vertex covers and 2-vertex covers. We discuss them in Section 64.6.

30.5. The weighted 2-matching problem

Given a graph $G = (V, E)$ and a weight function $w \in \mathbb{Q}^E$, the *weight* of a 2-matching x is $w^{\mathsf{T}}x$. The weighted 2-matching problem is strongly polynomial-time solvable:

Theorem 30.3. *A maximum-weight* 2*-matching can be found in time* $O(n^3)$.

Proof. Make the bipartite graph H as in the proof of Theorem 30.1, with weight function $w'(u'v) := w'(uv') := w(uv)$ for each edge uv of G. Then a maximum-weight matching in the new graph gives a maximum-weight 2-matching in the original graph. So Theorem 17.4 gives the present theorem. ∎

One can derive similarly from Egerváry's theorem a characterization of the maximum weight of a 2-matching, given by Gallai [1957,1958a,1958b]. Given $w : E \to \mathbb{Z}_+$, call a vector $y : V \to \mathbb{Z}_+$ a *w-vertex cover* if $y_u + y_v \geq w(e)$ for each edge $e = uv$.

Theorem 30.4. *Let* $G = (V, E)$ *be a graph and let* $w \in \mathbb{Z}_+^E$. *Then the maximum weight* $w^{\mathsf{T}}x$ *of a* 2*-matching* x *is equal to the minimum size of a* 2w*-vertex cover.*

Proof. It is easy to see that the maximum cannot be larger than the minimum. To see equality, make the bipartite graph H as in the proof of Theorem 30.1, with weight $w'(u'v) := w'(uv') := w(uv)$ for each edge uv of G. Then the maximum w-weight of a 2-matching in G is equal to the maximum w'-weight of a matching in H. By Theorem 17.1, the latter is equal to the minimum of $y'(V \cup V')$ where $y' : V \cup V' \to \mathbb{Z}_+$ with $y'(u) + y'(v') \geq w(uv)$ and $y'(u') + y'(v) \geq w(uv)$ for each edge uv of G. Defining $y_v := y'_v + y'_{v'}$ for each $v \in V$, we obtain y as required. ∎

System (30.7) is generally not totally dual integral: if $G = (V, E)$ is the complete graph K_3 on three vertices, and $w(e) := 1$ for each $e \in E$, then the maximum weight of a 2-matching is equal to 3, while there is no integer dual solution of odd value (when considering the dual of maximizing $w^{\mathsf{T}}x$ subject to (30.7)).

However, *half*-integrality holds:

Corollary 30.4a. *System* (30.7) *is totally dual half-integral.*

Proof. This is equivalent to Theorem 30.4. ∎

Pulleyblank [1973,1980] showed that (30.7) can be extended to a TDI system as follows:

Corollary 30.4b. *The following system is totally dual integral:*

$$\text{(30.9)} \qquad \begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for } e \in E, \\ \text{(ii)} & x(\delta(v)) \leq 2 & \text{for } v \in V, \\ \text{(iii)} & x(E[U]) \leq |U| & \text{for } U \subseteq V. \end{array}$$

Proof. Choose $w \in \mathbb{Z}_+^E$. By Corollary 30.4a, the problem of maximizing $w^\mathsf{T} x$ over (30.7) has an optimum dual solution $y \in \frac{1}{2}\mathbb{Z}_+^V$. Let $y'_v := \lfloor y_v \rfloor$ and $T := \{v \in V \mid y_v \notin \mathbb{Z}\}$. Let $z_T := 1$ and $z_U := 0$ for each $U \subseteq V$ with $U \neq T$. Then y', z is an integer optimum dual solution of the problem of maximizing $w^\mathsf{T} x$ over (30.9). ∎

Corollary 30.4a gives the total dual half-integrality of the perfect 2-matching constraints (30.8):

Corollary 30.4c. *System* (30.8) *is totally dual half-integral.*

Proof. Directly from Corollary 30.4a. ∎

More strongly, one has:

Corollary 30.4d. *Let $w \in \mathbb{Z}^E$ with $w(C)$ even for each circuit C. Then the problem of minimizing $w^\mathsf{T} x$ subject to* (30.8) *has an integer optimum dual solution.*

Proof. As $w(C)$ is even for each circuit, there is a subset U of V with $\{e \in E \mid w(e) \text{ odd}\} = \delta(U)$. Now replace w by $w' := w + \sum_{v \in U} \chi^{\delta(v)}$. Then $w'(e)$ is an even integer for each edge e. Hence by Corollary 30.4c there is an integer optimum dual solution y'_v ($v \in V$) for the problem of minimizing ${w'}^\mathsf{T} x$ subject to (30.8). Now setting $y_v := y'_v - 1$ if $v \in U$ and $y_v := y'_v$ if $v \notin U$ gives an integer optimum dual solution y for w. ∎

30.5a. Maximum-size 2-matchings and maximum-size matchings

Uhry [1975] gave the following relation between maximum-size 2-matchings and maximum-size matchings:

Theorem 30.5. *For each maximum-size* 2*-matching* x *in a graph* G*, there exists a maximum-size matching* M *missing each vertex* v *with* $x(\delta(v)) = 0$.

Proof. Let x be a maximum-size 2-matching in G and let M be a maximum-size matching covering a minimum number of vertices v with $x(\delta(v)) = 0$. Suppose that M covers a vertex u with $x(\delta(u)) = 0$. To prove the theorem, we can assume that x has inclusionwise minimal support. This implies that the edges e with $x_e = 1$ form a collection of vertex-disjoint odd circuits.

Let N be the matching consisting of those edges e with $x_e = 2$. Let P be the component of $M \cup N$ containing u. Then P is a path starting at u, and ending at, say, w. If P has even length, then $M \triangle P$ is a maximum-size matching covering fewer vertices v with $x(\delta(v)) = 0$ than M does — a contradiction. So P has odd length, and hence, since x is a maximum-size 2-matching, w belongs to the vertex set of some odd circuit C consisting of edges e with $x_e = 1$. However, in that case we can augment x, by redefining $x_e := 0$ if $e \in P \cap N$, $x_e := 2$ if $e \in P \cap M$, and $x_e := 0$ or 2 alternatingly on the edges of C. ∎

Uhry [1975] (cf. Pulleyblank [1987]) related maximum-size 2-matchings and maximum-size matchings further by:

Theorem 30.6. *Let* x *be a maximum-size* 2*-matching with the set* $\{e \mid x_e = 1\}$ *inclusionwise minimal. Then the support of* x *contains a maximum-size matching* M *of* G.

Proof. As the set $F := \{e \mid x_e = 1\}$ is inclusionwise minimal, it forms a collection $\mathcal{C}$ of vertex-disjoint odd circuits. So $x(\delta(v)) = 0$ or 2 for each vertex v. By Theorem 30.5, we can assume that $x(\delta(v)) = 2$ for each $v \in V$, since deleting all vertices v with $x(\delta(v)) = 0$ does not decrease the maximum size of a matching.

Let M be a maximum-size matching containing a minimum number of edges e with $x_e = 0$. Let N be the matching consisting of those edges e with $x_e = 2$. Consider any component P of $M \cup N$. Then P is not a circuit or an even path of positive length, since otherwise $M \triangle P$ is a maximum-size matching having fewer edges e with $x_e = 0$ than M has — a contradiction. So if P is not a singleton, it is a path of odd length; let it connect vertices u and w. Since P is not M-augmenting, both u and w are vertices on odd circuits in $\mathcal{C}$, say on C_u and C_w respectively. If $C_u \neq C_w$, we can modify x so as to decrease the set of edges e with $x_e = 1$. So $C_u = C_w$.

It follows that each $C \in \mathcal{C}$ contains an even number of vertices covered by M, and hence an odd number of vertices missed by M. Hence

$$(30.10) \qquad 2|M| \leq |V| - |\mathcal{C}| = 2|N| + \sum_{C \in \mathcal{C}} (|C| - 1).$$

Therefore, by augmenting N with a matching of size $\frac{1}{2}(|C| - 1)$ contained in C, for each circuit $C \in \mathcal{C}$, we obtain a matching M' with $|M'| \geq |M|$ contained in the support of x. ∎

(Theorem 30.6 was generalized in (30.88).) Related results were obtained by Balas [1981].

Mühlbacher, Steinparz, and Tinhofer [1984] showed that if x is a vertex of the 2-matching polytope maximizing $|\{e \in E \mid x_e = 2\}|$, then the vector $(i_3(x), i_5(x), \ldots)$ is lexicographically maximal, where $i_k(x)$ is the number of circuits in the support of x of size k. For related work, see Mühlbacher [1979] and Hell and Kirkpatrick [1981].

30.6. Simple 2-matchings and 2-factors

Call a 2-matching x *simple* if x is a 0,1 vector. So we can identify simple 2-matchings with subsets F of E satisfying $\deg_F(v) \leq 2$ for each $v \in V$.

A construction of Tutte [1954b] gives the following characterization of the maximum size of a simple 2-matching, with the help of the Tutte-Berge formula ($E[K,S]$ denotes the set of edges connecting K and S):

Theorem 30.7. *Let $G = (V,E)$ be a graph. The maximum size of a simple 2-matching is equal to the minimum value of*

$$(30.11) \qquad |V| + |U| - |S| + \sum_K \lfloor \tfrac{1}{2}|E[K,S]| \rfloor,$$

where U and S are disjoint subsets of V, with S a stable set, and where K ranges over the components of $G - U - S$.

Proof. To see that the maximum is not more than the minimum, let F be a simple 2-matching and let U and S be disjoint subsets of V, with S a stable set. Then F has at most $2|U|$ edges incident with U. Moreover, for each component K of $G-U-S$, the number of edges in F spanned by $K \cup S$ is at most $|K| + \lfloor \frac{1}{2}|E[K,S]| \rfloor$, since

$$(30.12) \qquad \begin{aligned} &2|F \cap E[K \cup S]| = 2|F \cap E[K]| + 2|F \cap E[K,S]| \\ &\leq 2|F \cap E[K]| + |F \cap E[K,S]| + |E[K,S]| \leq 2|K| + |E[K,S]|. \end{aligned}$$

Hence

$$(30.13) \qquad |F| \leq 2|U| + \sum_K (|K| + \lfloor \tfrac{1}{2}|E[K,S]| \rfloor)$$

(where K ranges over the components of $G-U-S$), giving that F is at most (30.11).

To see the reverse inequality, make a graph $G' = (V', E')$ as follows. For each vertex v of G, introduce vertices v' and v'' of G'. For each edge $e = uv$ of G, introduce vertices $p_{e,u}$ and $p_{e,v}$ and edges

$$(30.14) \qquad u'p_{e,u}, u'p_{e,u}, p_{e,u}p_{e,v}, v'p_{e,v}, v''p_{e,v}.$$

This defines all vertices and edges of G'.

Now:

$$(30.15) \qquad \nu_2^{\mathrm{s}}(G) = \nu(G') - |E|,$$

where $\nu(G')$ denotes the maximum size of a matching in G' and $\nu_2^{s}(G)$ denotes the maximum size of a simple 2-matching in G. In this proof we only need $\geq$ in (30.15). This inequality holds as there is a maximum-size matching M in G' with the property that for each edge $e = uv$ of G, both vertices $p_{e,u}$ and $p_{e,v}$ of G' are covered by M. Then the edges e of G for which edge $p_{e,u}p_{e,v}$ does not belong to M, form a simple 2-matching N in G with $|N| = |M|-|E|$. So we have $\geq$ in (30.15).

By the Tutte-Berge formula (Theorem 24.1), there is a subset X of V' such that the number $o(G'-X)$ of odd components of $G'-X$ is at least $|V'| - 2\nu(G') + |X|$. We take X inclusionwise minimal with this property.

Then for each $v \in V$, if one of v', v'' does not belong to X, then both do not belong to X. For suppose $v' \in X$ and $v'' \notin X$. As v' and v'' have the same set of neighbours in G', removing v' from X, decreases X by 1 and decreases the number of odd components of $o(G'-X)$ by at most one. So we would obtain a smaller set X as required, contradicting the minimality assumption.

Consider any vertex v of G and any edge $e = uv$ of G with $p_{e,v} \in X$. Then the three neighbours of $p_{e,v}$ in G' belong to three different odd components of $G'-X$. (Otherwise, removing $p_{e,v}$ from X decreases X by 1, and decreases $o(G'-X)$ be at most 1, contradicting the minimality of X.) Hence $p_{e,u}, v', v'' \notin X$, and moreover $p_{f,v} \in X$ for each edge f of G incident with v.

Let U be the set of $v \in V$ for which $v', v'' \in X$ and let S be the set of $v \in V$ for which $p_{e,v} \in X$ for each edge e of G incident with v. So U and S are disjoint, and S is a stable set.

Then $|X| = 2|U| + |\delta(S)|$. Let κ denote the number of components K of $G-U-S$ with $|E[K,S]|$ odd. Then

(30.16) $$o(G'-X) = 2|S| + |E[U,S]| + \kappa.$$

Hence we have

(30.17) $$\begin{aligned}\nu_2^{s}(G) &\geq \nu(G') - |E| \geq \tfrac{1}{2}(|V'| + |X| - o(G'-X)) - |E| \\ &= |V| + |U| + \tfrac{1}{2}|\delta(S)| - |S| - \tfrac{1}{2}|E[U,S]| - \tfrac{1}{2}\kappa \\ &= |V| + |U| - |S| + \sum_{K} \lfloor \tfrac{1}{2}|E[K,S]| \rfloor\end{aligned}$$

(where K ranges over the components of $G-U-S$), as required. ∎

A 2-*factor* is a simple perfect 2-matching. Equivalently, it is a subset F of E with $\deg_F(v) = 2$ for each $v \in V$.

Theorem 30.7 implies the following result of Belck [1950] (also Gallai [1950] announced a characterization of the existence of a 2-factor):

Corollary 30.7a. *A graph $G = (V,E)$ has a 2-factor if and only if*

(30.18) $$|S| \leq |U| + \sum_{K} \lfloor \tfrac{1}{2}|E[K,S]| \rfloor$$

for each pair of disjoint subsets U, S of V, with S a stable set, where K ranges over the components of $G - U - S$.

Proof. Directly from Theorem 30.7. ∎

This implies a classical result of Petersen [1891]:

Corollary 30.7b. *Each $2k$-regular graph has a 2-factor.*

Proof. Let $G = (V, E)$ be $2k$-regular. We check (30.18). Let U and S be disjoint subsets of V, with S a stable set. Let l be the number of components K of $G-U-S$ with $|E[K,S]|$ odd. Then for each such component K we have $|E[K,U]| \geq 1$ (since G is Eulerian). Hence $|E[U,S]| \leq 2k|U| - l$. Therefore,

$$\begin{aligned} (30.19)\qquad 2k|S| &= |\delta(S)| = |E[U,S]| + \sum_K |E[K,S]| \\ &\leq 2k|U| - l + \sum_K |E[K,S]| = 2k|U| + \sum_K 2\lfloor \tfrac{1}{2}|E[K,S]|\rfloor \\ &\leq 2k(|U| + \sum_K \lfloor \tfrac{1}{2}|E[K,S]|\rfloor) \end{aligned}$$

(where K ranges over the components of $G - U - S$), and (30.18) follows. ∎

The construction above gives also a reduction of finding a maximum-weight simple 2-matching to finding a maximum-weight matching — hence it can be done in strongly polynomial time. This implies that also a minimum-weight 2-factor can be found in strongly polynomial time.

(Grötschel and Holland [1987] gave computational results on a cutting plane method to find a minimum-weight 2-factor.)

30.7. The simple 2-matching polytope and the 2-factor polytope

Given a graph $G = (V, E)$, the *simple 2-matching polytope* is the convex hull of the simple 2-matchings in G. It can be characterized as follows (Edmonds [1965b]):

Theorem 30.8. *The simple 2-matching polytope is determined by*

$$\begin{array}{llll} (30.20) & \text{(i)} & 0 \leq x_e \leq 1 & (e \in E), \\ & \text{(ii)} & x(\delta(v)) \leq 2 & (v \in V), \\ & \text{(iii)} & x(E[U]) + x(F) \leq |U| + \lfloor \tfrac{1}{2}|F|\rfloor & (U \subseteq V, F \subseteq \delta(U), \\ & & & F \text{ matching}, |F| \text{ odd}). \end{array}$$

Proof. It is easy to show that each simple 2-matching x satisfies (30.20). Condition (iii) follows from

(30.21) $$x(E[U]) + x(F) \leq x(E[U]) + \tfrac{1}{2}x(\delta(U)) + \tfrac{1}{2}x(F) \leq |U| + \tfrac{1}{2}|F|$$

if x is a simple 2-matching.

To show that (30.20) is enough to determine the simple 2-matching polytope, we first show that (30.20) implies an extended version of (30.20)(iii), where we delete the condition that F be a matching. This can be seen by induction on $|F|$. Indeed, suppose that F contains edges f_1, f_2 incident with a vertex v. Let $F' := F \setminus \{f_1, f_2\}$. Then, if $v \in U$, setting $U' := U \setminus \{v\}$:

(30.22) $$x(E[U]) + x(F) \leq x(E[U']) + x(F') + x(\delta(v)) \leq |U'| + \tfrac{1}{2}|F'| + 2 = |U| + \tfrac{1}{2}|F|.$$

If $v \notin U$, setting $U' := U \cup \{v\}$:

(30.23) $$x(E[U]) + x(F) \leq x(E[U']) + x(F') \leq |U'| + \tfrac{1}{2}|F'| = |U| + \tfrac{1}{2}|F|.$$

So we can delete in (iii) the requirement that F be a matching.

We now prove that the conditions determine the simple 2-matching polytope. Let $G' = (V', E')$ be as in the proof of Theorem 30.7. Let x satisfy (30.20). Define $x' \in \mathbb{R}^{E'}$ by

(30.24) $$x'(u'p_{e,u}) := x'(u''p_{e,u}) := x'(v'p_{e,v}) := x'(v''p_{e,v}) := \tfrac{1}{2}x_e \text{ and } x'(p_{e,u}p_{e,v}) := 1 - x_e,$$

for any edge $e = uv$ of G. We show that x' belongs to the matching polytope of G'.

That is, by Edmonds' matching polytope theorem (Corollary 25.1a), we should check

(30.25)
$$\begin{array}{rll} \text{(i)} & x'(e') \geq 0 & \text{for } e' \in E', \\ \text{(ii)} & x'(\delta'(v')) \leq 1 & \text{for } v' \in V', \\ \text{(iii)} & x'(E'[Y]) \leq \lfloor \tfrac{1}{2}|Y| \rfloor & \text{for } Y \subseteq V' \text{ with } |Y| \text{ odd,} \end{array}$$

where $\delta' := \delta_{G'}$ and where $E'[Y]$ is the set of edges in E' spanned by Y.

Trivially we have (30.25)(i) and (ii) by (30.20)(i) and (ii). To prove (30.25)(iii), let Y violate (30.25)(iii). We first show that if one of v', v'' belongs to Y, then both belong to Y. For suppose that $v' \in Y$ and $v'' \notin Y$. Let $Y_1 := Y \setminus \{v'\}$ and $Y_2 := Y \cup \{v''\}$. Then

(30.26) $$x'(E'[Y]) = \tfrac{1}{2}(x'(E'[Y_1]) + x'(E'[Y_2])) \leq x'(E'[Y_1]) + \tfrac{1}{2}x'(\delta'(Y_1)) = \tfrac{1}{2}\sum_{u \in Y_1} x'(\delta'(u)) \leq \tfrac{1}{2}|Y_1| = \lfloor \tfrac{1}{2}|Y| \rfloor,$$

a contradiction.

We choose Y with $|Y| + |\delta'(Y)|$ minimal. Then:

(30.27) (i) if $u', v' \in Y$, then $p_{e,u} \in Y$ and $p_{e,v} \in Y$,
(ii) if $p_{e,u} \in Y$, then $u' \in Y$.

To see (30.27)(i), first suppose that $u', v' \in Y$ and $p_{e,u} \notin Y$. Define $Y' := Y \cup \{p_{e,u}, p_{e,v}\}$. Then $|Y'| + |\delta'(Y')| < |Y| + |\delta'(Y)|$, and hence Y' satisfies inequality (30.25)(iii). Therefore, (30.28) $\quad x'(E'[Y]) \leq x'(E'[Y']) - x'(\delta'(p_{e,u})) \leq \lfloor \frac{1}{2}|Y'| \rfloor - 1 \leq \lfloor \frac{1}{2}|Y| \rfloor.$

This contradicts our assumption that Y violates (30.25)(iii).

To see (30.27)(ii), let $p_{e,u} \in Y$ and $u' \notin Y$. Define $Y' := Y \setminus \{p_{e,u}, p_{e,v}\}$. Again $|Y'| + |\delta'(Y')| < |Y| + |\delta'(Y)|$, and hence Y' satisfies inequality (30.25)(iii). If $p_{e,v} \notin Y$, then

(30.29) $\quad x'(E'[Y]) = x'(E'[Y']) \leq \lfloor \frac{1}{2}|Y'| \rfloor \leq \lfloor \frac{1}{2}|Y| \rfloor.$

If $p_{e,v} \in Y$, then

(30.30) $\quad x'(E'[Y]) \leq x'(E'[Y']) + x'(\delta'(p_{e,v})) \leq \lfloor \frac{1}{2}|Y'| \rfloor + 1 = \lfloor \frac{1}{2}|Y| \rfloor.$

Both (30.29) and (30.30) contradict our assumption that Y does not satisfy (30.25)(iii). This proves (30.27).

Let $U := \{v \in V \mid v', v'' \in Y\}$ and let F be the set of those edges $e = uv$ in $\delta(U)$ with $u \in U$, $v \notin U$, and $p_{e,u} \in Y$. Then $x'(E'[Y]) = x(E[U]) + |E[U]| + x(F)$ and $|Y| = 2|U| + 2|E[U]| + |F|$. Hence (30.20)(iii) implies (30.25)(iii).

So x' is a convex combination of incidence vectors of matchings in G'. Each such vector y satisfies $y(\delta'(v')) = 1$ for each vertex $v' = p_{e,u}$ (as x' satisfies this equality). Hence each such matching corresponds to a simple 2-matching in G, and we obtain x as convex combination of simple 2-matchings in G. ∎

Given a graph $G = (V, E)$, the 2-*factor polytope* is the convex hull of (the incidence vectors of) 2-factors in G. Then:

Corollary 30.8a. *The* 2*-factor polytope is determined by*

(30.31)

(i)	$0 \leq x_e \leq 1$	$(e \in E)$,
(ii)	$x(\delta(v)) = 2$	$(v \in V)$,
(iii)	$x(\delta(U) \setminus F) - x(F) \geq 1 - \lvert F\rvert$	$(U \subseteq V,\ F \subseteq \delta(U)$, F matching, $\lvert F\rvert$ odd).

Proof. Directly from Theorem 30.8, since (30.31)(ii) implies $x(E[U]) = |U| - \frac{1}{2}x(\delta(U))$. ∎

Notes. Grötschel [1977a] characterized the facets of the simple 2-matching polytope and of the 2-factor polytope of the complete graph K_n. Rispoli and Cosares [1998] showed that the diameter of the 2-factor polytope of a complete graph is at most 6. Rispoli [1994] showed that the 'monotonic diameter' of the 2-factor polytope is equal to $\lfloor \frac{1}{2}n \rfloor$ if $n \geq 5$ and $n \neq 8, 9$, and to $\lfloor \frac{1}{2}n \rfloor - 1$ if $n = 3, 4, 8, 9$.

Boyd and Carr [1999] showed that if $G = (V, E)$ is a complete graph and $l : E \to \mathbb{R}_+$ satisfies the triangle inequality, then the minimum value of $l^{\mathsf{T}}x$ over (30.31) is at most $\frac{4}{3}$ times the minimum value of $l^{\mathsf{T}}x$ over (30.31)(i)(ii). They also show that the factor $\frac{4}{3}$ is best possible.

30.8. Total dual integrality

Consider the system

$$(30.32) \quad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq 1 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \leq 2 & (v \in V), \\ \text{(iii)} & x(E[U]) + x(F) \leq |U| + \lfloor \frac{1}{2}|F| \rfloor & (U \subseteq V, F \subseteq \delta(U), \\ & & F \text{ matching}). \end{array}$$

(So $|F|$ is not required to be odd.)

It is a special case of Theorem 32.3 (cf. Cook [1983b]) that system (30.32) is TDI (the restriction in (30.32) that F is a matching follows from (30.22) and (30.23)). This implies that (30.31) is totally dual half-integral. This also gives:

(30.33) Let $w \in \mathbb{Z}^E$ with $w(C)$ even for each circuit C. Then the problem of minimizing $w^{\mathsf{T}} x$ subject to (30.31) has an integer optimum dual solution.

To see this, notice that if $w(C)$ is even for each circuit, there is a subset U of V with $\{e \in E \mid w(e) \text{ odd}\} = \delta(U)$. Now replace w by $w' := w + \sum_{v \in U} \chi^{\delta(v)}$. Then $w'(e)$ is an even integer for each edge e. Hence there is an integer optimum dual solution y'_v $(v \in V)$, for the problem of minimizing ${w'}^{\mathsf{T}} x$ subject to (30.31). Now setting $y_v := y'_v - 1$ if $v \in U$ and $y_v := y'_v$ if $v \notin U$ gives an integer optimum dual solution for w.

30.9. 2-edge covers and 2-stable sets

Let $G = (V, E)$ be an undirected graph. A 2-*edge cover* is a vector $x \in \mathbb{Z}_+^E$ satisfying $x(\delta(v)) \geq 2$ for each vertex v. A 2-*stable set* is a vector $y \in \mathbb{Z}_+^V$ such that $y_u + y_v \leq 2$ for each edge uv of G. Defining the *size* of a vector as the sum of its entries, we denote:

(30.34) $\rho_2(G) :=$ the minimum size of a 2-edge cover in G,
$\alpha_2(G) :=$ the maximum size of a 2-stable set in G.

Note that if G has no isolated vertices, then:

$$(30.35) \quad \alpha_2(G) = \max\{|V| + |U| - |N(U)| \;\big|\; U \subseteq V, U \text{ stable set}\}$$

and that

$$(30.36) \quad \alpha(G) \leq \tfrac{1}{2}\alpha_2(G) \leq \tfrac{1}{2}\rho_2(G) \leq \rho(G).$$

Gallai's theorem (Theorem 19.1) can be extended to 2-matchings and 2-stable sets, which was published also in Gallai [1959a]:

Theorem 30.9. *For any graph $G = (V, E)$ without isolated vertices:*

(30.37) $\alpha_2(G) + \tau_2(G) = \nu_2(G) + \rho_2(G) = 2|V|.$

Proof. Let x be a minimum-size 2-vertex cover. Then $x_v \leq 2$ for each vertex v. Define $y_v := 2 - x_v$ for each vertex v. Then y is a 2-stable set, and hence $\alpha_2(G) \geq y(V) = 2|V| - x(V) = 2|V| - \tau_2(G)$.

Conversely, let y be a maximum-size 2-stable set. Then $y_v \leq 2$ for each vertex v. Define $x_v := 2 - y_v$ for each vertex v. Then x is a 2-vertex cover, and hence $\tau_2(G) \leq x(V) = 2|V| - y(V) = 2|V| - \alpha_2(G)$. This shows that $\alpha_2(G) + \tau_2(G) = 2|V|$.

To see that $\nu_2(G) + \rho_2(G) = 2|V|$, let x be a minimum-size 2-edge cover. For each $v \in V$, reduce $x(\delta(v))$ by $x(\delta(v)) - 2$, by reducing x_e on edges $e \in \delta(v)$. We obtain a 2-matching y of size

(30.38) $$y(E) \geq x(E) - \sum_{v \in V}(x(\delta(v)) - 2) = 2|V| - x(E) = 2|V| - \rho_2(G).$$

Hence $\nu_2(G) \geq 2|V| - \rho_2(G)$.

Conversely, let y be a maximum-size 2-matching. For each $v \in V$, increase $y(\delta(v))$ by $2 - y(\delta(v))$, by increasing y_e on edges $e \in \delta(v)$. We obtain a 2-edge cover x of size

(30.39) $$x(E) \leq y(E) + \sum_{v \in V}(2 - y(\delta(v))) = 2|V| - y(E) = 2|V| - \nu_2(G).$$

Hence $\rho_2(G) \leq 2|V| - \nu_2(G)$. ∎

This implies the following, which is a special case of a theorem of Gallai [1957,1958a,1958b] (cf. Theorem 30.11) (and can be derived alternatively from the Kőnig-Rado edge cover theorem):

Corollary 30.9a. *$\alpha_2(G) = \rho_2(G)$ for any graph G without isolated vertices. That is, the maximum size of a 2-stable set is equal to the minimum size of a 2-edge cover.*

Proof. Directly from Theorems 30.1 and 30.9. ∎

These reductions also imply the polynomial-time solvability of the problems of finding a minimum-size 2-edge cover and a maximum-size 2-stable set.

30.10. Fractional edge covers and stable sets

Any vector $x \in \mathbb{R}^E$ satisfying

(30.40)
(i) $x_e \geq 0$ for $e \in E$,
(ii) $x(\delta(v)) \geq 1$ for $v \in V$,

is called a *fractional edge cover*. The minimum size $x(E)$ of a fractional edge cover is called the *fractional edge cover number* and is denoted by $\rho^*(G)$. By linear programming duality, $\rho^*(G)$ is equal to the *fractional stable set number* $\alpha^*(G)$ — the maximum size of a *fractional stable set*, which is any solution $y \in \mathbb{R}^V$ of

(30.41) (i) $0 \leq y_v \leq 1$ for $v \in V$,
(ii) $y_u + y_v \leq 1$ for $uv \in E$.

The equality $\rho^*(G) = \alpha^*(G)$ also follows from Corollary 30.9a, since trivially

(30.42) $\frac{1}{2}\rho_2(G) \geq \rho^*(G) \geq \alpha^*(G) \geq \frac{1}{2}\alpha_2(G)$.

30.11. The fractional edge cover polyhedron

Let $G = (V, E)$ be a graph. The *fractional edge cover polyhedron* of G is the polyhedron determined by (30.40). Balinski [1965] showed:

Theorem 30.10. *Each vertex of the fractional edge cover polyhedron of G is half-integer.*

Proof. Let x be a vertex of the fractional edge cover polyhedron. We can assume that $x_e > 0$ for each edge e, since if $x_e = 0$ we can apply induction to $G - e$. Moreover, we can assume that G is connected and has at least three vertices.

As x is a vertex, there are $|E|$ constraints among (30.40)(ii) satisfied with equality. Define $U := \{v \mid x(\delta(v)) = 1\}$. So $|E| \leq |V|$. If there exists an end vertex v in U, with neighbour u say, then $u \in U$ and there is no other edge incident with u (otherwise it would have $x_e = 0$), implying the theorem. So no such end vertex exists.

If G is a tree, then there is at most one vertex w with $x(\delta(w)) \neq 1$, implying the existence of an end vertex v and a neighbour u of v with $u, v \in U$.

So G is not a tree, and hence $|E| = |V|$ and $U = V$. Since G has no end vertex, G is a circuit. Then $\frac{1}{2} \cdot \mathbf{1}$ satisfies all constraints that x satisfies. So $x = \frac{1}{2} \cdot \mathbf{1}$, as x is a vertex. ∎

30.12. The 2-edge cover polyhedron

Theorem 30.10 implies a characterization of the 2-*edge cover polyhedron* of G, which is, by definition, the convex hull of the 2-edge covers in G:

Corollary 30.10a. *The* 2*-edge cover polyhedron is determined by*

(30.43) (i) $x_e \geq 0$ for $e \in E$,
(ii) $x(\delta(v)) \geq 2$ for $v \in V$.

Proof. Directly from Theorem 30.10, since it implies that the vertices of the polyhedron determined by (30.43) are integer, and hence 2-edge covers. ∎

Similar results as for fractional edge covers and 2-edge covers hold for fractional stable sets and 2-stable sets. We discuss them in Section 64.5.

30.13. Total dual integrality of the 2-edge cover constraints

Finding a minimum-weight 2-edge cover is easily reduced to the minimum-weight edge cover problem, by splitting vertices. Gallai [1957,1958a,1958b] characterized the minimum weight as follows. Given $w : E \to \mathbb{Z}_+$, a *w-stable set* is a function $y : V \to \mathbb{Z}_+$ with $y_u + y_v \leq w(e)$ for each edge $e = uv$.

Theorem 30.11. *Let $G = (V, E)$ be a graph without isolated vertices and let $w \in \mathbb{Z}_+^E$. Then the minimum weight $w^\mathsf{T} x$ of a 2-edge cover x is equal to the maximum size of a $2w$-stable set.*

Proof. From Egerváry's theorem (Theorem 17.1). ∎

This is equivalent to the following result:

Corollary 30.11a. *System* (30.43) *is totally dual half-integral.*

Proof. Choose $w \in \mathbb{Z}_+^E$. Then the minimum weight $w^\mathsf{T} x$ of a 2-edge cover is equal to

(30.44) $\max\{2y(V) \mid y \in \frac{1}{2}\mathbb{Z}_+^V, y_u + y_v \leq w(e) \text{ for each } e = uv \in E\}$,

by Theorem 30.11. ∎

System (30.43) can be extended to a TDI system as follows:

Corollary 30.11b. *The following system is totally dual integral:*

(30.45) (i) $x_e \geq 0$ for $e \in E$,
(ii) $x(\delta(v)) \geq 2$ for $v \in V$,
(ii) $x(E[U] \cup \delta(U)) \geq |U|$ for $U \subseteq V$.

Proof. Choose $w \in \mathbb{Z}_+^E$. By Corollary 30.11a, the problem of minimizing $w^\mathsf{T} x$ over (30.43) has an optimum dual solution $y \in \frac{1}{2}\mathbb{Z}_+^V$. Define $y'_v := \lfloor y_v \rfloor$ for $v \in V$, and $T := \{v \in V \mid y_v \notin \mathbb{Z}\}$. Define $z_T := 1$ and $z_U := 0$ for each $U \subseteq V$ with $U \neq T$. Then y', z is an integer optimum dual solution for the problem of minimizing $w^\mathsf{T} x$ over (30.45). ∎

30.14. Simple 2-edge covers

Call a 2-edge cover x *simple* if x is a 0,1 vector. Thus we can identify simple 2-edge covers with subsets F of E satisfying $\deg_F(v) \geq 2$ for each $v \in V$. A 2-edge cover exists if and only if all degrees are at least 2. Define

(30.46) $\nu_2^{\mathrm{s}}(G) :=$ the maximum size of a simple 2-matching, $\rho_2^{\mathrm{s}}(G) :=$ the minimum size of a simple 2-edge cover.

Again there is a relation between $\nu_2^{\mathrm{s}}(G)$ and $\rho_2^{\mathrm{s}}(G)$ similar to Gallai's theorem (Theorem 19.1):

Theorem 30.12. *For any graph $G = (V, E)$ of minimum degree at least 2 one has:*

(30.47) $$\nu_2^{\mathrm{s}}(G) + \rho_2^{\mathrm{s}}(G) = 2|V|.$$

Proof. Let M be a maximum-size simple 2-matching. For each $v \in V$, add to M $2 - \deg_M(v)$ edges incident with v. We can do this in such a way that we obtain a simple 2-edge cover F with

(30.48) $$|F| \leq |M| + \sum_{v \in V}(2 - \deg_M(v)) = 2|V| - |M|.$$

So $\rho_2^{\mathrm{s}}(G) \leq 2|V| - |M| = 2|V| - \nu_2^{\mathrm{s}}(G)$.

To see the reverse inequality, let F be a minimum-size simple 2-edge cover. For each $v \in V$, delete from F $\deg_F(v) - 2$ edges incident with v. We obtain a simple 2-matching M with

(30.49) $$|M| \geq |F| - \sum_{v \in V}(\deg_F(v) - 2) = 2|V| - |F|.$$

So $\nu_2^{\mathrm{s}}(G) \geq 2|V| - |F| = 2|V| - \rho_2^{\mathrm{s}}(G)$, which shows (30.47). ∎

This implies a min-max relation for minimum-size simple 2-edge cover:

Corollary 30.12a. *Let $G = (V, E)$ be a graph of minimum degree at least 2. Then the minimum size of a simple 2-edge cover is equal to the maximum value of*

(30.50) $$|V| - |U| + |S| - \sum_K \lfloor \tfrac{1}{2}|E[K, S]| \rfloor,$$

where U and S are disjoint subsets of V, with S a stable set, and where K ranges over the components of $G - U - S$.

Proof. Directly from Theorems 30.7 and 30.12. ∎

These reductions also imply the polynomial-time solvability of the problem of finding a minimum-size simple 2-edge cover.

Given a graph $G = (V, E)$, the *simple 2-edge cover polytope* is the convex hull of the simple 2-edge covers in G. A special case of Theorem 34.9 below is that the simple 2-edge cover polytope is determined by

$$\text{(30.51)} \quad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq 1 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \geq 2 & (v \in V), \\ \text{(iii)} & x(E[U]) + x(F) \geq |U| + \lceil \frac{1}{2}|F| \rceil & (U \subseteq V, F \subseteq \delta(U), \\ & & |F| \text{ odd}). \end{array}$$

We refer to Theorem 34.10 for the total dual integrality of the following system (Cook [1983b]):

$$\text{(30.52)} \quad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq 1 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \geq 2 & (v \in V), \\ \text{(iii)} & x(E[U]) + x(F) \geq |U| + \lceil \frac{1}{2}|F| \rceil & (U \subseteq V, F \subseteq \delta(U)). \end{array}$$

Theorem 34.11 implies that a minimum-weight simple 2-edge-cover can be found in strongly polynomial time.

30.15. Graphs with $\nu(G) = \tau(G)$ and $\alpha(G) = \rho(G)$

Kőnig's matching theorem states that the matching number $\nu(G)$ is equal to the vertex cover number $\tau(G)$ for each bipartite graph G. A graph G therefore is said to have the *Kőnig property* if $\nu(G) = \tau(G)$. Deming [1979b] and Sterboul [1979] characterized the class of graphs with the Kőnig property.

Note that by Gallai's theorem (Theorem 19.1), for any graph G without isolated vertices:

$$\text{(30.53)} \quad \nu(G) = \tau(G) \iff \alpha(G) = \rho(G)$$

(where $\alpha(G)$ and $\rho(G)$ denote the stable set and edge cover number of G, respectively).

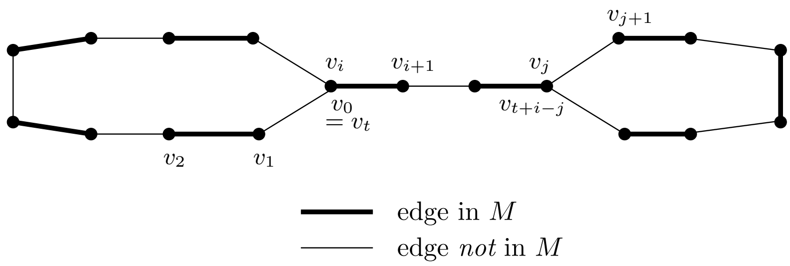


Figure 30.1
An M-posy
The two circuits may intersect.

To characterize graphs G with $\nu(G) = \tau(G)$, Sterboul defined, for any graph $G = (V, E)$ and any matching M in G, an *M-posy* to be an even-length M-alternating closed walk $(v_0, v_1, \ldots, v_t)$, with $v_{i-1}v_i \in M$ if i is even, such that there exist $i < j$ with i odd and j even, $v_1, \ldots, v_j$ all distinct, $v_{j+1}, \ldots, v_t$ all distinct, and

(30.54) $\qquad v_i = v_t, v_{i+1} = v_{t-1}, \ldots, v_j = v_{t+i-j}.$

Lemma 30.13α. *If there exists an even-length M-alternating closed walk $C = (v_0, v_1, \ldots, v_t)$ with $v_i = v_j$ for i, j of different parity, then there exists an M-posy.*

Proof. Let C be a shortest such closed walk, covering a minimum number of edges (in this order of priority). Then

(30.55) $\qquad$ there exist no three distinct $h, i, k \geq 1$ with $v_h = v_i = v_k$,

since otherwise we may assume that h and i have the same parity. Leaving out one of the $v_h - v_i$ parts of C gives a shorter such closed walk.

Choose h, i of different parity with $v_h = v_i$ and with $|h - i|$ minimal. We may assume that $h = 0$ and that $v_0v_1 \notin M$. Choose $j, k \geq i$ of different parity with $v_j = v_k$ and $j < k$, and with $k - j$ minimal. (Such j, k exist, as $v_i = v_t$.) Then j is even and k is odd, since otherwise $v_{j+1} = v_{k-1}$ (as it is the vertex matched to $v_j = v_k$). Moreover, $j - i = t - k$ and

(30.56) $\qquad v_i = v_t, v_{i+1} = v_{t-1}, \ldots, v_j = v_k.$

Otherwise, resetting the $v_k - v_t$ part of C to the $v_j - v_i$ part of C^{-1} or conversely, we obtain again a shortest such closed walk, however covering a fewer number of edges, a contradiction.

Then C is an M-posy, since $v_1, \ldots, v_j$ are all distinct and $v_{j+1}, \ldots, v_t$ are all distinct. If say $v_a = v_b$ with $1 \leq a < b \leq j$, then $b \leq i$ (since otherwise $v_a = v_b = v_l$ for some $l > b$, contradicting (30.55)). So by the minimality of $|h - i|$, $a \equiv b \pmod 2$. Hence, deleting the $v_a - v_b$ part from C gives a shortest such walk, a contradiction. ∎

This is used in proving:

Theorem 30.13. *Let $G = (V, E)$ be a graph. Then the following are equivalent:*

(30.57)
(i) *G has the Kőnig property, that is $\nu(G) = \tau(G)$;*
(ii) *for some maximum-size matching M there is no M-flower and no M-posy;*
(iii) *for each maximum-size matching M there is no M-flower and no M-posy.*

Proof. The implication (iii)⇒(ii) is trivial, and the implication (i)⇒(iii) is easy: suppose $\nu(G) = \tau(G)$, let M be a maximum-size matching and let U

be a minimum-size vertex cover. Then each edge in M has exactly one vertex in U. Suppose that $P = (v_0, \ldots, v_t)$ is an M-flower or an M-posy. Then for each odd k, exactly one of v_k and v_{k+1} belongs to U, while for each even k at least one of v_k and v_{k+1} belongs to U. If $v_t \notin U$, then $v_k \in U$ for each even k. Since $v_t = v_j$ for some even j, it follows that $v_t \in U$. If $v_0 \notin U$, then $v_k \in U$ for each odd k. Since $v_j \in U$ and j is even, we have $v_0 \in U$. So v_0 is covered by M, and hence P is an M-posy. So $v_0 = v_i$ for some odd i. So $v_i \in U$ for some odd i, a contradiction.

It remains to prove (ii)⇒(i). Let M be a maximum-size matching in G and let X be the set of vertices missed by M. Then there is no M-alternating $X - X$ walk (since M has maximum size and since there is no M-flower (cf. Theorem 24.3)). Let U be the set of vertices v for which there is an M-alternating $X - v$ walk and let Z be the set of vertices v for which there exists an odd-length M-alternating $X - v$ walk. Then Z intersects each edge intersecting U, while $|Z|$ is equal to the number of edges in M contained in U.

So we can apply induction to $G - U$ if $U \neq \emptyset$. Hence we may assume that $U = \emptyset$. Equivalently, $X = \emptyset$, that is, M is a perfect matching. Choose $e = uv \in M$. By Lemma 30.13α, $G - u$ has no $M \setminus \{e\}$-flower or $G - v$ has no $M \setminus \{e\}$-flower. By symmetry, we may assume that $G - v$ has no $M \setminus \{e\}$-flower. Since G has no M-posy, $G - v$ has no $M \setminus \{e\}$-posy. Hence, by induction:

$$\nu(G) = \nu(G - v) + 1 = \tau(G - v) + 1 \geq \tau(G). \tag{30.58}$$

Hence $\nu(G) = \tau(G)$. ∎

This implies a characterization due to Lovász [1974] ((i)⇔(ii) below) and Lovász and Plummer [1986] ((i)⇔(iii) below), based on the minimum size $\tau_2(G)$ of a 2-vertex cover studied in Section 30.1:

Corollary 30.13a. *For any graph G, the following are equivalent:*

(30.59)
- (i) $\nu(G) = \tau(G)$,
- (ii) $\tau_2(G) = 2\tau(G)$,
- (iii) *the edges e for which there exists a maximum-size 2-matching x with $x_e \geq 1$, form a bipartite graph.*

Proof. The implication (i)⇒(ii) follows from (30.3). To see (ii)⇒(iii), let U be a minimum-size vertex cover and let x be a maximum-size 2-matching. Then, using Theorem 30.1,

$$\tau_2(G) = \nu_2(G) = \sum_{e \in E} x_e \leq \sum_{e \in E} x_e |e \cap U| = \sum_{v \in U} x(\delta(v)) \leq 2|U| = 2\tau(G), \tag{30.60}$$

and hence we have equality throughout. So $e \in \delta(U)$ if $x_e \geq 1$. As this is true for each maximum-size 2-matching x, we have (iii).

We finally show (iii)$\Rightarrow$(i), which we derive from Theorem 30.13. Suppose that (iii) holds, and let M be a maximum-size matching. If there would exist any M-flower or M-posy, then we can find a 2-matching of size at least $2|M|$ such that M and the support of the 2-matching contains an odd circuit. For an M-flower this is trivial. For an M-posy $(v_0, \ldots, v_t)$, let

$$(30.61) \qquad x := 2\chi^M - \sum_{h=1}^{t}(-1)^h \chi^{v_{h-1}v_h}.$$

Then x is a 2-matching of size $2|M|$. However, the support of x together with M contains an odd circuit. This contradicts (iii). ∎

Note that characterization (iii) can be checked in polynomial time. By Theorem 30.9 and its proof method, we know that (i), (ii), and (iii) are also equivalent to each of:

(30.62) (iv) $\alpha(G) = \rho(G)$,
(v) $\alpha_2(G) = 2\alpha(G)$,
(vi) the edges e for which there exists a minimum-size 2-edge cover x with $x_e \geq 1$, form a bipartite graph.

More on the Kőnig property can be found in Korach [1982], Bourjolly, Hammer, and Simeone [1984], and Bourjolly and Pulleyblank [1989], and related results in Tipnis and Trotter [1989].

30.16. Excluding triangles

Let $G = (V, E)$ be a graph. Call a 2-matching x *triangle-free* if $x(ET) \leq 2$ for each triangle T in G. (A *triangle* is a subgraph isomorphic to K_3.) The *triangle-free 2-matching polytope* is the convex hull of the triangle-free 2-matchings.

In order to characterize the triangle-free 2-matching polytope, Cornuéjols and Pulleyblank [1980a] (cf. Cook [1983b], Cook and Pulleyblank [1987]) showed the following:

Theorem 30.14. *Let $G = (V, E)$ be a simple graph and let $\mathcal{T}$ be a collection of triangles in G. Then the following system is totally dual integral:*

$$(30.63) \qquad \begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for each } e \in E, \\ \text{(ii)} & \tfrac{1}{2}x(\delta(v)) \leq 1 & \text{for each } v \in V, \\ \text{(iii)} & x(ET) \leq 2 & \text{for each } T \in \mathcal{T}. \end{array}$$

Proof. Let $w \in \mathbb{Z}_+^E$ and consider the problem dual to maximizing $w^{\mathsf{T}}x$ over (30.63):

(30.64) $$\begin{aligned} &\text{minimize } \sum_{v \in V} y_v + 2 \sum_{T \in \mathcal{T}} z_T \\ &\text{subject to } \tfrac{1}{2} \sum_{v \in V} y_v \chi^{\delta(v)} + \sum_{T \in \mathcal{T}} z_T \chi^{ET} \geq w, \end{aligned}$$

with $y \in \mathbb{R}_+^V$ and $z \in \mathbb{R}_+^{\mathcal{T}}$. We must show that there exists an integer optimum solution y, z. We take a counterexample with $|E|+w(E)$ minimal. This implies that G is connected. Moreover, $w(e) \geq 1$ for each edge e, since otherwise we could delete e.

On the other hand, $w(e) \leq 2$ for each edge e. To see this, let y, z be any optimum solution. If $y_u \geq 2$ for some vertex u, we can reset $w(e) := w(e) - 1$ for each $e \in \delta(u)$. By resetting, the optimum value decreases by at least 2 (since resetting $y_u := y_u - 2$ gives a feasible solution for the new w, with objective value 1 less than the original objective value). By the minimality of G, w, for the new w there is an integer optimum solution y, z. Resetting $y_u := y_u + 2$ then gives an integer optimum solution for the original w.

So we can assume that $y_v < 2$ for each vertex v, and similarly, that $z_T < 1$ for each $T \in \mathcal{T}$.

Choose an optimum solution y, z with $\sum_{T \in \mathcal{T}} z_T$ minimal. Let $\mathcal{T}_+ := \{T \in \mathcal{T} \mid z_T > 0\}$. Then:

(30.65) no two triangles in $\mathcal{T}_+$ have an edge in common.

For suppose that $T_1, T_2 \in \mathcal{T}_+$ have $ET_1 \cap ET_2 = \{e\}$, say $e = v_1 v_2$. Resetting $z_{T_i} := z_{T_i} - \varepsilon$ and $y_{v_i} := y_{v_i} + 2\varepsilon$ for $i = 1, 2$, for $\varepsilon > 0$ small enough, gives again an optimum solution. However, $\sum_{T \in \mathcal{T}} z_T$ decreases, contradicting our assumption. This proves (30.65).

This implies that $w(e) \leq 2$ for each edge e, since $y_v < 2$ and $z_T < 1$.

Next:

(30.66) for any $T \in \mathcal{T}_+$ and any $v \in VT$ one has either $0 < y_v < 1$ for each $v \in VT$ and $w(e) = 1$ for each $e \in ET$, or $1 < y_v < 2$ for each $v \in VT$ and $w(e) = 2$ for each $e \in ET$.

Let $VT = \{v_1, v_2, v_3\}$. First assume that $\frac{1}{2} y_{v_1} + \frac{1}{2} y_{v_2} + z_T > w(v_1 v_2)$. Then after resetting $y_{v_3} := y_{v_3} + 2\varepsilon$ and $z_T := z_T - \varepsilon$ we obtain again an optimum solution, for $\varepsilon > 0$ small enough. However, $\sum_{T \in \mathcal{T}} z_T$ decreases, contradicting our assumption. So $\frac{1}{2} y_{v_1} + \frac{1}{2} y_{v_2} + z_T = w(v_1 v_2)$, and similarly for any other pair from v_1, v_2, v_3. This implies

(30.67) $$y_{v_1} = w(v_1 v_2) + w(v_1 v_3) - w(v_2 v_3) - z_T,$$

and similarly for v_2 and v_3. So if $w(e) = 1$ for each $e \in ET$, then $0 < y_v < 1$ for each $v \in VT$. Similarly, if $w(e) = 2$ for each $e \in ET$, then $1 < y_v < 2$ for each $v \in VT$. If not all three edges of T have the same weight, (30.67) implies that there is a vertex v in T with $y_v > 2$ or $y_v < 0$, a contradiction. This proves (30.66).

Now consider resetting

$$
\begin{array}{llll}
(30.68) & y_v & := y_v - \varepsilon & \text{if } 0 < y_v < 1, \\
& y_v & := y_v + \varepsilon & \text{if } 1 < y_v < 2, \\
& z_T & := z_T + \varepsilon & \text{if } T \in \mathcal{T}_+ \text{ and } w(e) = 1 \text{ for each edge } e \text{ in } T, \\
& z_T & := z_T - \varepsilon & \text{if } T \in \mathcal{T}_+ \text{ and } w(e) = 2 \text{ for each edge } e \text{ in } T.
\end{array}
$$

If we choose ε close enough to 0 (positive or negative), we obtain again a feasible solution of (30.64), by (30.65) and (30.66), using the integrality of w. Moreover, the objective value changes linearly in ε. However, as y, z is an optimum solution, the objective value cannot decrease. Hence there is no change in the objective value at all. That is, for any ε close enough to 0, we obtain again an optimum solution. Therefore, by choosing ε appropriately, we can decrease the number of noninteger values of y_v, z_T. ∎

This theorem implies (in fact, is equivalent to) the following TDI result:

Corollary 30.14a. *Let $G = (V, E)$ be a simple graph and let $\mathcal{T}$ be a collection of triangles in G. Then the following system is totally dual integral:*

$$
\begin{array}{llll}
(30.69) & \text{(i)} & x_e \geq 0 & \text{for each } e \in E, \\
& \text{(ii)} & x(\delta(v)) \leq 2 & \text{for each } v \in V, \\
& \text{(iii)} & x(ET) \leq 2 & \text{for each } T \in \mathcal{T}, \\
& \text{(iv)} & x(E[U]) \leq |U| & \text{for each } U \subseteq V.
\end{array}
$$

Proof. Let $w \in \mathbb{Z}_+^E$. Let μ be the maximum value of $w^\mathsf{T} x$ over (30.69), This is equal to the maximum value of $w^\mathsf{T} x$ over (30.63) (since (30.69)(iv) follows from (i) and (ii)).

Consider an integer optimum solution y_v ($v \in V$), z_T ($T \in \mathcal{T}$) of the problem dual to maximizing $w^\mathsf{T} x$ over (30.63). Define $y'_v := \lfloor \frac{1}{2} y_v \rfloor$ for $v \in V$ and $T := \{v \in V \mid \frac{1}{2} y_v \notin \mathbb{Z}\}$. Define $a_U := 1$ if $U = T$ and $a_U := 0$ for any other subset U of V.

Then y', a, z is an integer feasible solution of the problem dual to maximizing $w^\mathsf{T} x$ over (30.69), as w is integer. Moreover, it is optimum, since

$$
(30.70) \qquad \sum_{v \in V} 2y'_v + \sum_{U \subseteq V} a_U |U| + \sum_{T \in \mathcal{T}} 2z_T = \sum_{v \in V} y_v + \sum_{T \in \mathcal{T}} 2z_T = \mu. \qquad ∎
$$

The theorem implies the following characterization of the triangle-free 2-matching polytope, given by Cornuéjols and Pulleyblank [1980a] and J.F. Maurras (cf. Cornuéjols and Pulleyblank [1980b]):

Corollary 30.14b. *Let $G = (V, E)$ be a graph. The triangle-free 2-matching polytope is determined by:*

$$
\begin{array}{llll}
(30.71) & \text{(i)} & x_e \geq 0 & \text{for each } e \in E, \\
& \text{(ii)} & x(\delta(v)) \leq 2 & \text{for each } v \in V, \\
& \text{(iii)} & x(ET) \leq 2 & \text{for each triangle } T \text{ in } G.
\end{array}
$$

Proof. Theorem 30.14 implies that the polytope determined by (30.63) is integer (as the right-hand sides are integer). Since (30.71) determines the same polytope, the corollary follows. ∎

In fact, there is a sharper consequence, where we just consider an arbitrary subcollection $\mathcal{T}$ of the triangles:

Corollary 30.14c. *Let $G = (V, E)$ be a graph and let $\mathcal{T}$ be a collection of triangles in G. Then the following inequalities determine an integer polytope:*

(30.72)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(\delta(v)) \leq 2$ for each $v \in V$,
(iii) $x(ET) \leq 2$ for each triangle $T \in \mathcal{T}$.

Proof. Similar to the proof of the previous corollary. ∎

Cornuéjols and Pulleyblank [1980a] also showed that the inequalities (30.72)(i) and (iii) are all necessary, while (ii) is necessary unless $\deg_G(v) = 2$ and v is in a triangle in $\mathcal{T}$ (assuming that G is connected and has at least three vertices). They also gave a polynomial-time algorithm to find a maximum-weight triangle-free 2-matching.

Moreover, they showed the following. A *triangle cluster* is a graph defined recursively as follows: any one-vertex graph is a triangle cluster; if G is a triangle cluster and v is a vertex of G, then by introducing two new vertices u, u' and adding edges vu, vu' and uu', we obtain again a triangle cluster.

For any graph G, let $\beta(G)$ denote the number of components of G that are triangle clusters. This is used in the following min-max relation for maximum-size triangle-free 2-matching (Cornuéjols and Pulleyblank [1980a]):

Theorem 30.15. *The maximum size of a triangle-free 2-matching in a graph $G = (V, E)$ is equal to the minimum value of $|V| + |U| - \beta(G - U)$ taken over $U \subseteq V$.*

Proof. To see that the maximum is not more than the minimum, let x be a maximum-size triangle-free 2-matching in G. Let $U \subseteq V$ and let W be the set of vertices of $G - U$ that are in triangle cluster components. Consider any component K of $G - U$ that is a triangle cluster. Then the edges of K can be partitioned into $\frac{1}{2}(|K| - 1)$ triangles. Hence $x(E[K]) \leq |K| - 1$, and therefore

$$\sum_{v \in K} x(\delta(v)) = 2x(E[K]) + x(\delta(K)) \leq 2(|K| - 1) + x(\delta(K)). \tag{30.73}$$

Summing over all components K that are triangle cluster, we see that

$$\sum_{v \in W} x(\delta(v)) \leq 2|W| - 2\beta(G - U) + x(\delta(W)). \tag{30.74}$$

Moreover,

$$x(\delta(W)) \leq x(\delta(U)) \leq \sum_{v \in U} x(\delta(v)) \leq 2|U|. \tag{30.75}$$

This implies

$$\begin{aligned} 2x(E) &= \sum_{v \in W} x(\delta(v)) + \sum_{v \in V \setminus W} x(\delta(v)) \\ &\leq 2|W| - 2\beta(G-U) + 2|U| + 2|V \setminus W| \\ &= 2(|V| + |U| - \beta(G-U)). \end{aligned} \tag{30.76}$$

This shows that the maximum is not more than the minimum.

To see the reverse inequality, let $\mathcal{T}$ denote the set of triangles in G. By Theorem 30.14, the maximum size of a triangle-free 2-matching is equal to the minimum value of

$$\sum_{v \in V} y_v + 2 \sum_{T \in \mathcal{T}} z_T \tag{30.77}$$

where $y_v \in \mathbb{Z}_+$ (for $v \in V$) and $z_T \in \mathbb{Z}_+$ (for $T \in \mathcal{T}$) such that

$$\tfrac{1}{2} \sum_{v \in V} y_v \chi^{\delta(v)} + \sum_{T \in \mathcal{T}} z_T \chi^{ET} \geq \mathbf{1}. \tag{30.78}$$

Choose y, z attaining this minimum, with

$$\sum_{T \in \mathcal{T}} z_T \text{ as small as possible.} \tag{30.79}$$

Clearly, $y_v \leq 2$ for each $v \in V$ and $z_T \leq 1$ for each $T \in \mathcal{T}$. Let $\mathcal{T}_+ := \{T \in \mathcal{T} \mid z_T = 1\}$.

Then we have:

$$\text{if } T \in \mathcal{T}_+ \text{ and } v \in T, \text{ then } y_v = 0. \tag{30.80}$$

Indeed, suppose $y_v \geq 1$, and let u and u' be the two other vertices in T. Then resetting $z_T := 0$, $y_u := y_u + 1$, and $y_{u'} := y_{u'} + 1$, we obtain y, z again attaining the minimum value (30.77), contradicting our minimality assumption (30.79). This shows (30.80).

Let F be the set of edges contained in some $T \in \mathcal{T}_+$. Then

$$\text{each component of the graph } (V, F) \text{ is a triangle cluster.} \tag{30.81}$$

If not, there exist distinct $T_1, \ldots, T_k \in \mathcal{T}_+$ and distinct $v_1, \ldots, v_k \in V$, such that, taking $v_0 := v_k$,

$$v_{i-1} v_i \in T_i \tag{30.82}$$

for $i = 1, \ldots, k$, and such that $k > 1$. Then resetting $z_{T_i} := 0$ and $y_{v_i} := 2$ for $i = 1, \ldots, k$, we obtain y, z again attaining the minimum value (30.77), contradicting our minimality assumption (30.79). This shows (30.81).

Now let $W := \{v \in V \mid y_v = 0\}$. Then each edge contained in W is contained in some $T \in \mathcal{T}_+$, and hence, by (30.81), each component of $G[W]$

is a triangle cluster. Let k be the number of components of $G[W]$. Then $\sum_{T\in\mathcal{T}} z_T = \frac{1}{2}(|W| - k)$.

Define $U := N(W)$. Then $y_v = 2$ for each $v \in U$, since each edge e connecting W and U should satisfy (30.78). Therefore, (30.77) is at least

(30.83) $$|V|-|W|+|U|+2\cdot\tfrac{1}{2}(|W|-k) = |V|+|U|-k \geq |V|+|U|-\beta(G-U),$$

proving the theorem. ∎

This characterizes the existence of a triangle-free perfect 2-matching:

Corollary 30.15a. *A graph $G = (V, E)$ has a triangle-free perfect 2-matching if and only if $G - U$ has at most $|U|$ components that are triangle clusters, for each $U \subseteq V$.*

Proof. Directly from Theorem 30.15. ∎

Cornuéjols and Pulleyblank [1980b] gave a polynomial-time algorithm to find a triangle-free perfect b-matching. Cook [1983b] and Cook and Pulleyblank [1987] characterized the facets and the minimal TDI-system for the triangle-free 2-matching polytope.

30.16a. Excluding higher polygons

Cornuéjols and Pulleyblank [1983] considered excluding higher polygons. For any collection P of graphs, call a graph G *P-critical* if $G \notin P$ while $G - v \in P$ for each vertex v of G. Let P_k be the collection of graphs that have a perfect 2-matching in which each circuit has length larger than k. Then for each k and each graph $G = (V, E)$:

(30.84) If G is P_k-critical, then G is factor-critical,

and

(30.85) V can be partitioned into edges and subsets U with $G[U]$ P_k-critical if and only if for each $S \subseteq V$, the graph $G - S$ has at most $|S|$ P_k-critical components.

This generalizes Theorem 24.8 and (30.86) below.

Corollary 30.14b does not extend to 2-matchings excluding triangles and pentagons, as is shown by the example given in Figure 30.2. (The sum of the values is at most 4 on each pentagon, but it does not belong to the convex hull of the 2-matchings without pentagons, since the sum of the values is equal to $\frac{20}{3}$, but there is no pentagon-free 2-matching of size ≥ 7.)

30.16b. Packing edges and factor-critical subgraphs

Cornuéjols, Hartvigsen, and Pulleyblank [1982] and Cornuéjols and Hartvigsen [1986] discovered an interesting direction of extensions of the results on matchings. Let $G = (V, E)$ be a graph. Call a subset U of V *factor-critical* if $G[U]$ is factor-critical; that is, if for each $v \in U$, the set $U \setminus \{v\}$ is matchable.

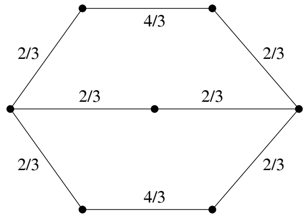


Figure 30.2

Let $\mathcal{F}$ be a collection of factor-critical subsets of V. An *$\mathcal{F}$-matching* is a collection of disjoint subsets from $E \cup \mathcal{F}$. It is *perfect* if it covers V. Call a subset U of V *$\mathcal{F}$-critical* if $G[U]$ has no perfect $\mathcal{F}$-matching but for each $v \in U$, the graph $G[U] - v$ has one. Cornuéjols, Hartvigsen, and Pulleyblank [1982] showed that

(30.86) if U is $\mathcal{F}$-critical, then U is factor-critical.

Then Cornuéjols and Hartvigsen [1986] proved the following extension of Tutte's 1-factor theorem (Theorem 24.1a):

(30.87) G has a perfect $\mathcal{F}$-matching if and only if for each $U \subseteq V$, the graph $G - U$ has at most $|U|$ $\mathcal{F}$-critical components.

Call an $\mathcal{F}$-matching $\mathcal{M}$ *maximum* if it maximizes $\sum_{U \in \mathcal{M}} |U|$. Cornuéjols and Hartvigsen [1986] also showed:

(30.88) Let $\mathcal{M}$ be a maximum $\mathcal{F}$-matching containing a minimum number of sets in $\mathcal{F}$. Let M be a matching containing $\mathcal{M} \cap E$ and having $\lfloor \frac{1}{2}|U| \rfloor$ edges in any $U \in \mathcal{M} \cap \mathcal{F}$. Then M is a maximum-size matching in G.

They also described an extension of the Edmonds-Gallai decomposition theorem. Cornuéjols, Hartvigsen, and Pulleyblank [1982] gave a polynomial-time algorithm to find a maximum $\mathcal{F}$-matching. Related results were obtained by Kirkpatrick and Hell [1978,1983] and Hell and Kirkpatrick [1984,1986].

30.16c. 2-factors without short circuits

Hartvigsen [1984] showed that a maximum size simple 2-matching without triangles can be found in polynomial time. He also gave good characterization for the existence of a 2-factor without triangles.

On the other hand, Cornuéjols and Pulleyblank [1980a] showed with a method of C.H. Papadimitriou that the problem of finding a 2-factor without circuits of length at most 5, is NP-complete. The complexity of deciding if a 2-factor exists without circuits of length at most 4 is not known.

Vornberger [1980] showed the NP-completeness of finding a maximum-weight 2-factor without circuits of length at most 4. The complexity status of finding a maximum-weight 2-factor without circuits of length at most 3 is unknown. Hell, Kirkpatrick, Kratochvíl, and Kříž [1988] and Cunningham and Wang [2000] give related results.

Chapter 31

b-matchings

b-matchings form an extension of 2-matchings and can be handled again by applying splitting techniques to ordinary matchings.

31.1. *b*-matchings

Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. A *b-matching* is a function $x \in \mathbb{Z}_+^E$ satisfying

$$(31.1) \qquad x(\delta(v)) \leq b(v)$$

for each $v \in V$. This is equivalent to: $Mx \leq b$, where M is the $V \times E$ incidence matrix of G.

In (31.1), we count multiplicities: if e is a loop at v, then x_e is added twice at v. (This is consistent with our definition of $\delta(v)$ as a *family* of edges, in which each loop at v occurs twice.)

It is convenient to consider the graph G_b arising from G by splitting each vertex v into $b(v)$ copies, and by replacing any edge uv by $b(u)b(v)$ edges connecting the $b(u)$ copies of u with the $b(v)$ copies of v. More formally, $G_b = (V_b, E_b)$, where

$$(31.2) \qquad \begin{aligned} V_b &:= \{q_{v,i} \mid v \in V, 1 \leq i \leq b(v)\}, \\ E_b &:= \{q_{u,j}q_{v,i} \mid uv \in E, 1 \leq j \leq b(u), 1 \leq i \leq b(v), q_{u,j} \neq q_{v,i}\}. \end{aligned}$$

The condition $q_{u,j} \neq q_{v,i}$ is relevant only if $u = v$, that is, if there is a loop at u.

This construction was given by Tutte [1954b], and yields a min-max relation for maximum-size b-matching (where again the *size* of a vector is the sum of its components):

Theorem 31.1. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the maximum size of a b-matching is equal to the minimum value of*

$$(31.3) \qquad b(U) + \sum_K \lfloor \tfrac{1}{2} b(K) \rfloor$$

taken over $U \subseteq V$, where K ranges over the components of $G - U$ spanning at least one edge[19].

Proof. To see that the maximum is not more than the minimum, consider a b-matching x and a subset U of V. Then the sum of x_e over the edges e intersecting U is at most $b(U)$. The sum of x_e over the edges e contained in some component K of $G - U$ is at most $\lfloor \frac{1}{2} b(K) \rfloor$.

Equality is derived from the Tutte-Berge formula (Theorem 24.1). Let G_b be the graph described in (31.2). Then the maximum size of a b-matching in G is equal to the maximum size of a matching in G_b. By the Tutte-Berge formula, this is equal to the minimum value of

$$\text{(31.4)} \qquad \tfrac{1}{2}(|V_b| + |U'| - o(G_b - U'))$$

over $U' \subseteq V_b$ (where $o(H)$ denotes the number of odd components of a graph H).

Let U' attain this minimum. We may assume that if U' misses at least one copy of some vertex v of G, it misses all copies of v (since deleting all copies does not increase (31.4)). Hence there is a subset U of V such that U' is equal to the set of copies of vertices in U. We take $v \in U$ if $b(v) = 0$.

Let I_U be the set of isolated (hence loopless) vertices of $G - U$. Then $o(G_b - U')$ is equal to $b(I_U)$ plus the number of components K of $G - U$ that span at least one edge and have $b(K)$ odd. Setting k to the number of such components, (31.4) is equal to

$$\text{(31.5)} \qquad \begin{aligned} &\tfrac{1}{2}(b(V) + b(U) - o(G_b - U')) = b(U) + \tfrac{1}{2}(b(V \setminus U) - o(G_b - U')) \\ &= b(U) + \tfrac{1}{2}(b(V \setminus U) - b(I_U) - k), \end{aligned}$$

which is equal to (31.3). ∎

This theorem directly gives a characterization of the existence of a *perfect b-matching*, that is a b-matching having equality in (31.1) for each $v \in V$. This characterization is due to Tutte [1952]. By I_U we denote the set of isolated, loopless vertices of $G - U$.

Corollary 31.1a. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then there exists a perfect b-matching if and only if for each $U \subseteq V$, $G - U - I_U$ has at most $b(U) - b(I_U)$ components K with $b(K)$ odd.*

Proof. Directly from Theorem 31.1, by observing that a perfect b-matching exists if and only if the minimum value of (31.3) is at least $\frac{1}{2} b(V)$. ∎

31.2. The *b*-matching polytope

By a similar construction we can derive a characterization of the b-matching polytope. Given a graph $G = (V, E)$ and $b \in \mathbb{Z}_+^V$, the *b-matching polytope* is

[19] So K may consist of one vertex with a loop attached.

the convex hull of the b-matchings. The inequalities describing the b-matching polytope were announced by Edmonds [1965b] (cf. Pulleyblank [1973], Edmonds [1975]):

Theorem 31.2. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the b-matching polytope is determined by the inequalities*

$$\text{(31.6)} \qquad \begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for } e \in E, \\ \text{(ii)} & x(\delta(v)) \leq b(v) & \text{for } v \in V, \\ \text{(iii)} & x(E[U]) \leq \lfloor \frac{1}{2} b(U) \rfloor & \text{for } U \subseteq V \text{ with } b(U) \text{ odd.} \end{array}$$

Proof. The inequalities (31.6) are trivially valid for the vectors in the b-matching polytope. To see that they determine the b-matching polytope, let x satisfy (31.6). We may assume that $b \geq \mathbf{1}$.

Again consider the graph $G_b = (V_b, E_b)$ obtained by splitting each vertex v into $b(v)$ copies (cf. (31.2)). For any edge $e' = u'v'$ of G_b, with u' and v' copies of u and v in G, define $x'(e') := x_e / b(u)b(v)$, where $e := uv$. We show that x' belongs to the matching polytope of G_b, which implies the theorem.

By Edmonds' matching polytope theorem, it suffices to show that x' satisfies:

$$\text{(31.7)} \qquad \begin{array}{rll} \text{(i)} & x'(e') \geq 0 & \text{for each edge } e' \in E_b, \\ \text{(ii)} & x'(\delta'(u')) \leq 1 & \text{for each vertex } u' \in V_b, \\ \text{(iii)} & x'(E'[U']) \leq \lfloor \frac{1}{2} |U'| \rfloor & \text{for each } U' \subseteq V_b \text{ with } |U'| \text{ odd.} \end{array}$$

Clearly (i) holds. To see (31.7)(ii), let u' be a vertex of G_b, being a copy of vertex u of G. Then

$$\text{(31.8)} \qquad x'(\delta'(u')) = x(\delta(u))/b(u) \leq 1,$$

since for any edge $e = uv$ of G one has that

$$\text{(31.9)} \qquad \sum_{v'} x'(u'v') = \sum_{v'} x(uv)/b(u)b(v) = x(uv)/b(u),$$

where v' ranges over the copies of v in G_b. So summing over all neighbours v' of u' gives $x(\delta(u))/b(u)$.

To see (31.7)(iii), choose $U' \subseteq V_b$ with $|U'|$ odd. Note that x satisfies (31.6)(iii) for *all* subsets U of V, since if $b(U)$ is even, then $x(E[U]) \leq \frac{1}{2} \sum_{v \in U} x(\delta(v)) \leq \frac{1}{2} b(U)$ by (31.6)(ii).

For any vertex v of G let B_v denote the set of copies of v in G_b. We show (31.7)(iii) by induction on the number of $v \in V$ for which U' 'splits' B_v, that is, for which

$$\text{(31.10)} \qquad B_v \cap U' \neq \emptyset \text{ and } B_v \not\subseteq U'.$$

If this number is 0, (31.7)(iii) follows from (31.6)(iii). If this number is nonzero, choose a vertex v satisfying (31.10). Let $U_1 := U' \setminus B_v$ and $U_2 := U' \cup B_v$. So by induction we know

(31.11) $\qquad x'(E'[U_1]) \leq \frac{1}{2}|U_1|$ and $x'(E'[U_2]) \leq \frac{1}{2}|U_2|$.

Moreover, (31.8) implies:

(31.12) $$x'(E'[U_1]) + x'(E'[U_2]) \leq \sum_{u' \in U_1} x'(\delta'(u')) \leq |U_1|.$$

(This uses the fact that $B_v = U_2 \setminus U_1$ is a stable set in G_b.) Now define $\lambda := |B_v \cap U'|/b(v)$ and $\mu := |B_v \setminus U'|/b(v)$. So $\lambda + \mu = 1$ and

(31.13) $$x'(E'[U']) = \lambda x'(E'[U_2]) + \mu x'(E'[U_1]).$$

If $\lambda \leq \frac{1}{2}$, then, by (31.11) and (31.12):

(31.14) $$\begin{aligned} x'(E'[U']) &= (\mu - \lambda)x'(E'[U_1]) + \lambda(x'(E'[U_1]) + x'(E'[U_2])) \\ &\leq \tfrac{1}{2}(\mu - \lambda)|U_1| + \lambda|U_1| = \tfrac{1}{2}|U_1| \leq \lfloor \tfrac{1}{2}|U'| \rfloor. \end{aligned}$$

(The last inequality holds as $U_1 \subset U'$.)

If $\lambda > \frac{1}{2}$, then, by (31.11) and (31.12):

(31.15) $$\begin{aligned} x'(E'[U']) &= (\lambda - \mu)x'(E'[U_2]) + \mu(x'(E'[U_1]) + x'(E'[U_2])) \\ &\leq (\lambda - \mu)\tfrac{1}{2}|U_2| + \mu|U_1| = \tfrac{1}{2}|U_1| + \tfrac{1}{2}(\lambda - \mu)|U_2 \setminus U_1| \\ &= \tfrac{1}{2}|U_1| + \tfrac{1}{2}(\lambda - \mu)b_v = \tfrac{1}{2}|U_1| + \tfrac{1}{2}(|B_v \cap U'| - |B_v \setminus U'|) \\ &\leq \tfrac{1}{2}|U_1| + \tfrac{1}{2}(|B_v \cap U'| - 1) \leq \lfloor \tfrac{1}{2}|U'| \rfloor. \end{aligned}$$

(The last inequality holds as $U' = U_1 \cup (B_v \cap U')$.)

Thus we have (31.7)(iii). ∎

(This theorem follows also from the proof of the total dual integrality of the constraints (31.17) in Theorem 31.3 below.)

Given a graph $G = (V, E)$ and $b \in \mathbb{Z}_+^V$, the *perfect b-matching polytope* is the convex hull of the perfect b-matchings in G. As it is a face of the b-matching polytope (if nonempty), the previous theorem implies:

Corollary 31.2a. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the perfect b-matching polytope is determined by the inequalities*

(31.16)
$$\begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for } e \in E, \\ \text{(ii)} & x(\delta(v)) = b(v) & \text{for } v \in V, \\ \text{(iii)} & x(\delta(U)) \geq 1 & \text{for } U \subseteq V \text{ with } b(U) \text{ odd.} \end{array}$$

Proof. Directly from Theorem 31.2. ∎

(For a direct proof of this Corollary also based on considering the graph G_b obtained from G by splitting each vertex v into $b(v)$ copies, see Aráoz, Cunningham, Edmonds, and Green-Krótki [1983].)

Hurkens [1988] characterized adjacency on the b-matching polytope and showed that the diameter of the b-matching polytope is equal to the maximum size of a b-matching.

31.3. Total dual integrality

System (31.6) generally is not totally dual integral: if $G = (V, E)$ is the complete graph K_3 on three vertices, and $b(v) := 2$ for each $v \in V$ and $w(e) := 1$ for each $e \in E$, then the maximum weight of a b-matching is equal to 3, while there is no integer dual solution of odd value (when considering the dual of optimizing $w^{\mathsf{T}}x$ subject to (31.6)).

However, if we extend (31.6)(iii) to *all* subsets U of V, the system is totally dual integral, as was shown by Pulleyblank [1980]. So the system becomes:

(31.17)
$$\begin{array}{rll} \text{(i)} & x_e \geq 0 & \text{for } e \in E, \\ \text{(ii)} & x(\delta(v)) \leq b(v) & \text{for } v \in V, \\ \text{(iii)} & x(E[U]) \leq \lfloor \frac{1}{2} b(U) \rfloor & \text{for } U \subseteq V. \end{array}$$

It is equivalent to the following result:

Theorem 31.3. *Let $G = (V, E)$ be a graph, let $b \in \mathbb{Z}_+^V$ and let $w \in \mathbb{Z}_+^E$. Then the maximum weight $w^{\mathsf{T}}x$ of a b-matching x is equal to the minimum value of*

(31.18)
$$\sum_{v \in V} y_v b(v) + \sum_{U \subseteq V} z(U) \lfloor \tfrac{1}{2} b(U) \rfloor,$$

where $y \in \mathbb{Z}_+^V$ and $z \in \mathbb{Z}_+^{\mathcal{P}(V)}$ satisfy

(31.19)
$$\sum_{v \in V} y_v \chi^{\delta(v)} + \sum_{U \subseteq V} \chi^{E[U]} \geq w.$$

Proof. By Theorem 31.2 and LP-duality, the maximum weight of a b-matching is equal to the minimum of (31.18) over $y \in \mathbb{R}_+^V$ and $z \in \mathbb{R}_+^{\mathcal{P}(V)}$ satisfying (31.19). Suppose that this minimum is strictly smaller than if we restrict y and z to integer-valued functions. Then there exists a $t \in \mathbb{Z}_+$ such that the minimum with y and z restricted to values in $2^{-t}\mathbb{Z}_+$ is strictly smaller than when restricting y and z to values in $\mathbb{Z}_+$, because we can slightly increase any value of y_v and $z(U)$ to a dyadic vector. Choose t with this property as small as possible. By replacing w by $2^{t-1}w$, we may assume that $t = 1$.

It therefore is enough to show that for each $y \in \frac{1}{2}\mathbb{Z}_+^V$ and $z \in \frac{1}{2}\mathbb{Z}_+^{\mathcal{P}(V)}$ satisfying (31.19), there exist $y' \in \mathbb{Z}_+^V$ and $z' \in \mathbb{Z}_+^{\mathcal{P}(V)}$ satisfying (31.19) such that

(31.20)
$$\sum_{v \in V} y'_v(v) + \sum_{U \subseteq V} z'(U) \lfloor \tfrac{1}{2} b(U) \rfloor \leq \sum_{v \in V} y_v b(v) + \sum_{U \subseteq V} z(U) \lfloor \tfrac{1}{2} b(U) \rfloor.$$

We show this by induction on $w(E)$. More precisely, we consider a counterexample $y \in \frac{1}{2}\mathbb{Z}_+^V$ and $z \in \frac{1}{2}\mathbb{Z}_+^{\mathcal{P}(V)}$ with smallest $w(E)$. Then necessarily

(31.21)
$$y \in \{0, \tfrac{1}{2}\}^V \text{ and } z \in \{0, \tfrac{1}{2}\}^{\mathcal{P}(V)},$$

since if $y_v \geq 1$ for some vertex v we can reduce $w(e)$ by 1 for each $e \in \delta(v)$ and reduce y_v by 1, to obtain a counterexample with smaller $w(E)$. Similarly, if $z(U) \geq 1$ for some $U \subseteq V$ we can reduce $w(e)$ by 1 for each $e \in E[U]$ and reduce $z(U)$ by 1, to obtain a counterexample with smaller $w(E)$.

Put on $y \in \{0, \frac{1}{2}\}^V$ and $z \in \{0, \frac{1}{2}\}^{\mathcal{P}(V)}$ the additional requirements that, first, $y(V)$ is as large as possible, and, second, that

$$(31.22) \qquad \sum_{U \subseteq V} z(U)|U||V \setminus U|$$

is as small as possible.

Let $S := \{v \in V \mid y_v = \frac{1}{2}\}$ and $\mathcal{F} := \{U \subseteq V \mid z(U) = \frac{1}{2}\}$. We first show that $\mathcal{F}$ is laminar; that is,

$$(31.23) \qquad \text{if } U, W \in \mathcal{F}, \text{ then } U \cap W = \emptyset \text{ or } U \subseteq W \text{ or } W \subseteq U.$$

Indeed, suppose that $U \cap W \neq \emptyset$, $U \not\subseteq W$, and $W \not\subseteq U$ for some $U, W \in \mathcal{F}$.

If $b(U \cap W)$ is odd, then decreasing $z(U)$ and $z(W)$ by $\frac{1}{2}$, and increasing $z(U \cap W)$ and $z(U \cup W)$ by $\frac{1}{2}$, would not increase (31.18) (since $\lfloor \frac{1}{2} b(U \cap W) \rfloor + \lfloor \frac{1}{2} b(U \cup W) \rfloor \leq \lfloor \frac{1}{2} b(U) \rfloor + \lfloor \frac{1}{2} b(W) \rfloor$), would maintain (31.19) (since $\chi^{E[U \cap W]} + \chi^{E[U \cup W]} \geq \chi^{E[U]} + \chi^{E[W]}$), would leave $y(V)$ unchanged, but would decrease (31.22), contradicting the minimality of (31.22).

If $b(U \cap W)$ is even, then resetting

$$(31.24) \qquad \begin{aligned} &z(U) := z(U) - \tfrac{1}{2},\ z(W) := z(W) - \tfrac{1}{2},\ z(U \setminus W) := z(U \setminus W) + \tfrac{1}{2}, \\ &z(W \setminus U) := z(W \setminus U) + \tfrac{1}{2}, \text{ and } y_v := y_v + \tfrac{1}{2} \text{ for each } v \in U \cap W, \end{aligned}$$

would not increase (31.18) (since $\lfloor \frac{1}{2} b(U \setminus W) \rfloor + \lfloor \frac{1}{2} b(W \setminus U) \rfloor + b(U \cap W) \leq \lfloor \frac{1}{2} b(U) \rfloor + \lfloor \frac{1}{2} b(W) \rfloor$), would maintain (31.19) (since $\chi^{E[U \setminus W]} + \chi^{E[W \setminus U]} + \sum_{v \in U \cap W} \chi^{\delta(v)} \geq \chi^{E[U]} + \chi^{E[W]}$), but would increase $y(V)$, contradicting the maximality of $y(V)$.

This shows (31.23). Suppose $\mathcal{F} \neq \emptyset$. Then choose an inclusionwise minimal set $U \in \mathcal{F}$ with the property that there exist an even number of sets $W \in \mathcal{F}$ with $W \supset U$. Let $U_1, \ldots, U_k$ be the inclusionwise maximal proper subsets of U with $U_i \in \mathcal{F}$ (possibly $k = 0$). By the choice of U, none of the U_i contain properly a set in $\mathcal{F}$. Then

$$(31.25) \qquad \lfloor \tfrac{1}{2} b(U) \rfloor + \sum_{i=1}^{k} \lfloor \tfrac{1}{2} b(U_i) \rfloor \geq b(U \cap S) + \sum_{i=1}^{k} 2 \lfloor \tfrac{1}{2} b(U_i \setminus S) \rfloor$$

or

$$(31.26) \qquad \lfloor \tfrac{1}{2} b(U) \rfloor + \sum_{i=1}^{k} \lfloor \tfrac{1}{2} b(U_i) \rfloor \geq b(U \setminus S) + \sum_{i=1}^{k} 2 \lfloor \tfrac{1}{2} b(U_i \cap S) \rfloor,$$

as

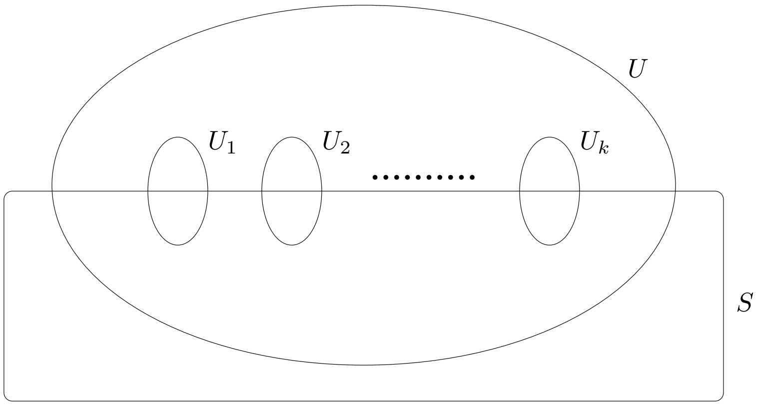


Figure 31.1

(31.27) $$b(U) + 2\sum_{i=1}^{k}\lfloor\tfrac{1}{2}b(U_i)\rfloor \geq b(U\cap S) + b(U\setminus S) + 2\sum_{i=1}^{k}\lfloor\tfrac{1}{2}b(U_i\setminus S)\rfloor + 2\sum_{i=1}^{k}\lfloor\tfrac{1}{2}b(U_i\cap S)\rfloor.$$

If (31.25) holds, then resetting $y_v := y_v + \frac{1}{2}$ for each $v \in U \cap S$, $z(U) := z(U) - \frac{1}{2}$, and $z(U_i) := z(U_i) - \frac{1}{2}$, $z(U_i \setminus S) := z(U_i \setminus S) + 1$ for each $i = 1, \ldots, k$ would not increase (31.18) (by (31.25)) and would maintain (31.19): on edges not spanned by U, the left-hand side of (31.19) does not decrease; on edges spanned by U the contribution of the nonmodified variables is integer, and

(31.28) $$\lfloor\tfrac{1}{2}(\sum_{v\in U\cap S}\chi^{\delta(v)} + \chi^{E[U]} + \sum_{i=1}^{k}\chi^{E[U_i]})\rfloor \leq \sum_{v\in U\cap S}\chi^{\delta(v)} + \sum_{i=1}^{k}\chi^{E[U_i\setminus S]}.$$

By the maximality of $y(V)$ it follows that $U \cap S = \emptyset$. Hence, after resetting we have $z(U_i) = 1$ for each $i = 1, \ldots, k$. If $k > 0$ we contradict (31.21). So $k = 0$, and therefore (as $z(U)$ decreases) (31.18) decreases, contradicting the minimality of (31.18).

If (31.26) holds, then resetting $y_v := y_v + \frac{1}{2}$ for each $v \in U \setminus S$, $z(U) := z(U) - \frac{1}{2}$, and $z(U_i) := z(U_i) - \frac{1}{2}$, $z(U_i \cap S) := z(U_i \cap S) + 1$ for each $i = 1, \ldots, k$ would not increase (31.18) (by (31.26)) and would maintain (31.19), since now

(31.29) $$\lfloor\tfrac{1}{2}(\sum_{v\in U\cap S}\chi^{\delta(v)} + \chi^{E[U]} + \sum_{i=1}^{k}\chi^{E[U_i]})\rfloor \leq \tfrac{1}{2}\sum_{v\in U}\chi^{\delta(v)} + \sum_{i=1}^{k}\chi^{E[U_i\cap S]}.$$

By the maximality of $y(V)$ it follows that $U \setminus S = \emptyset$, that is, $U \subseteq S$. Hence, after resetting we have $z(U_i) = 1$ for each $i = 1, \ldots, k$. If $k > 0$ we again contradict (31.21). So $k = 0$, and therefore (as $z(U)$ decreases) (31.18) decreases, again contradicting the minimality of (31.18).

So $\mathcal{F} = \emptyset$. Now setting $z'_S := 1$ and $y' := \mathbf{0}$ gives (31.20). ∎

(This is the proof method followed by Schrijver and Seymour [1977]. For a related proof, see Hoffman and Oppenheim [1978]. See also Cook [1983b].)

This theorem can be formulated equivalently in terms of total dual integrality:

Corollary 31.3a. *System* (31.17) *is TDI.*

Proof. Directly from Theorem 31.3. ∎

If we restrict the subsets U to odd-size subsets, the system is totally dual half-integral — a result stated by Pulleyblank [1973] and Edmonds [1975]:

Corollary 31.3b. *System* (31.6) *is totally dual half-integral.*

Proof. This follows from Corollary 31.3a, by using the fact that inequality (31.17)(iii) for $|U|$ even, is a half-integer sum of inequalities (31.6)(i) and (ii). ∎

Next considering the perfect b-matching polytope, generally (31.16) is not TDI. However:

Corollary 31.3c. *System* (31.16) *with* (31.16)(iii) *replaced by* (31.17)(iii) *is TDI.*

Proof. Directly from Corollary 31.3a with Theorem 5.25. ∎

This implies for the original system (Edmonds and Johnson [1970]):

Corollary 31.3d. *System* (31.16) *is totally dual half-integral.*

Proof. Consider an inequality $x(E[U]) \leq \lfloor \frac{1}{2} b(U) \rfloor$ in (31.17). If $b(U)$ is odd, this inequality is half of the sum of the inequalities $x(\delta(v)) \leq b(v)$ for $v \in U$ and of $-x(\delta(U)) \leq -1$. If $b(U)$ is even, this inequality is half of the sum of the inequalities $x(\delta(v)) \leq b(v)$ for $v \in U$ and of $-x_e \leq 0$ for $e \in \delta(U)$. ∎

In fact (Barahona and Cunningham [1989]):

Corollary 31.3e. *Let $w \in \mathbb{Z}^E$ with $w(C)$ even for each circuit C. Then the problem of minimizing $w^{\mathsf{T}}x$ subject to* (31.16) *has an integer optimum dual solution.*

Proof. As $w(C)$ is even for each circuit, there is a subset U of V with $\{e \in E \mid w(e) \text{ odd}\} = \delta(U)$. Now replace w by $w' := w + \sum_{v \in U} \chi^{\delta(v)}$. Then $w'(e)$ is an even integer for each edge e. Hence by Corollary 31.3d there is an integer optimum dual solution y'_v ($v \in V$), z_U ($U \subseteq V$, $b(U)$ odd) for the problem of minimizing ${w'}^{\mathsf{T}}x$ subject to (31.16). Now setting $y_v := y'_v - 1$ if $v \in U$ and $y_v := y'_v$ if $v \notin U$ gives an integer optimum dual solution for w. ∎

31.4. The weighted b-matching problem

We now consider the problem of finding a maximum-weight b-matching. Here, for a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, and a weight function $w \in \mathbb{Q}^E$, the *weight* of a b-matching x is $w^{\mathsf{T}}x$.

It should be noted that the method of reducing a b-matching problem to a matching problem by replacing each vertex v by $b(v)$ copies, does not yield a polynomial-time algorithm for the weighted b-matching problem. W.H. Cunningham and A.B. Marsh, III (with suggestions of W.R. Pulleyblank, K. Truemper, and M.R. Rao — cf. Marsh [1979]) and Gabow [1983a] gave polynomial-time algorithms for the weighted b-matching problem. Padberg and Rao [1982] showed, with a method similar to that described in Section 25.5c, that one can test the constraints (31.16) in polynomial time, thus yielding the polynomial-time solvability of the maximum-weight b-matching problem (with the ellipsoid method).

Gerards [1995a] attributed the following method, leading to a strongly polynomial-time algorithm, to J. Edmonds. It extends a similar approach of Anstee [1987], and amounts to reducing the b-matching problem to a bipartite b-matching problem and a nonbipartite $\mathbf{1}$-matching problem.

First there is the following observation.

Lemma 31.4α. *Let $G = (V, E)$ be a graph and let $b, b' \in \mathbb{Z}_+^V$ with $\|b - b'\|_1 = 1$. Let x be a b-matching and let x' be a b'-matching. Then there exists a $y \in \mathbb{Z}^E$ such that $\|y\|_\infty \leq 2$ and such that $x + y$ is a b'-matching and $x' - y$ is a b-matching.*

Proof. By symmetry we may assume that there exists a $u \in V$ such that $b'(u) = b(u) + 1$ and $b'(v) = b(v)$ if $v \neq u$. Hence x is a b'-matching. If x' is a b'-matching, we are done (taking $y = \mathbf{0}$). So we may assume that x' is not a b-matching, that is, $x'_u = b'(u)$. Then there exists a walk $P = (v_0, e_1, v_1, \ldots, e_t, v_t)$ in G such that

(31.30)
(i) $v_0 = u$, $x'_{e_i} > x_{e_i}$ if i is odd, $x'_{e_i} < x_{e_i}$ if i is even, and each edge e is traversed at most $|x'_e - x_e|$ times,
(ii) $x'(\delta(v_t)) < x(\delta(v_t))$ if t is even, and $x'(\delta(v_t)) > x(\delta(v_t))$ if t is odd (if $v_t = v_0$ and t is odd, then $x'(\delta(v_t)) \geq x(\delta(v_t)) + 2$).

The existence of such a path follows by taking a longest path satisfying (31.30)(i).

We now assume that P is a shortest path satisfying (31.30). Then no vertex is traversed more than twice (otherwise we can shortcut P), hence no edge is traversed more than twice. Let y_e be the number of times P traverses e, if $x'_e \geq x_e$, and let y_e be *minus* the number of times P traverses e, if $x'_e < x_e$. Then $x + y$ is a b'-matching, $x' - y$ is a b-matching, and $\|y\|_\infty \leq 2$. ∎

This implies a sensitivity result for maximum-weight b-matchings if we vary b:

Lemma 31.4β. *Let $G = (V, E)$, let $b, b' \in \mathbb{Z}_+^V$ and let a weight function $w \in \mathbb{R}^E$ be given. Then for any maximum-weight b-matching x there exists a maximum-weight b'-matching x' satisfying*

(31.31) $\qquad \|x - x'\|_\infty \leq 2\|b - b'\|_1.$

Proof. We may assume that $\|b - b'\|_1 = 1$. Let x be a maximum-weight b-matching and let x' be a maximum-weight b'-matching. By Lemma 31.4α, we know that there exists an integer vector y with $x + y$ a b'-matching, $x' - y$ a b-matching, and $\|y\|_\infty \leq 2$. Since $x' - y$ is a b-matching and since x is a maximum-weight b-matching, we have $w^{\mathsf{T}} x \geq w^{\mathsf{T}}(x' - y)$, and hence $w^{\mathsf{T}}(x + y) \geq w^{\mathsf{T}} x'$. Since x' is a maximum-weight b'-matching, it follows that $x'' := x + y$ is a maximum-weight b'-matching with $\|x'' - x\|_\infty = \|y\|_\infty \leq 2$. ∎

This is used in showing the strong polynomial-time solvability of the weighted b-matching problem:

Theorem 31.4. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, and a weight function $w \in \mathbb{Q}^E$, a maximum-weight b-matching can be found in strongly polynomial time.*

Proof. I. First consider the case that b is even. Make a bipartite graph H as follows. Make a new vertex v' for each $v \in V$. Let H have edges $u'v$ and uv' for each edge uv of G. Define $\tilde{b}(v) := \tilde{b}(v') := \frac{1}{2}b(v)$ for each $v \in V$. Define a weight $\tilde{w}(u'v) := \tilde{w}(uv') := w(uv)$ for each edge uv of G.

Find a maximum-weight $\tilde{b}$-matching $\tilde{x}$ in H. This can be done in strongly polynomial time by Theorem 21.9. Defining $x(uv) := \tilde{x}(u'v) + \tilde{x}(uv')$ for each edge uv of G, gives a maximum-weight b-matching x in G. Indeed, if there would be a b-matching in G of larger weight than that of x, then there is a half-integer $\tilde{b}$-matching in H of larger weight than that of $\tilde{x}$. This contradicts the fact that in a bipartite graph a maximum-weight b-matching is also a maximum-weight fractional b-matching (by Theorem 21.1).

II. Next consider the case of arbitrary b. Define $b' := 2\lfloor \frac{1}{2} b \rfloor$. Since b' is even, by part I of this proof we can find a maximum-weight b'-matching x' in G in strongly polynomial time. Now b arises from b' by at most $|V|$ resettings of b' to $b' + \chi^u$ for some $u \in V$. So it suffices to give a strongly polynomial-time method to obtain a maximum-weight b'-matching from a maximum-weight b-matching x, where $b' = b + \chi^u$ for some $u \in U$.

To this end, define

$$(31.32) \qquad z := \max\{\mathbf{0}, x - \mathbf{2}\} \text{ and } b'' := \min\{b' - Mz, M\mathbf{4}\}$$

(taking the maximum componentwise), where M is the $V \times E$ incidence matrix of G. ($\mathbf{0}$, $\mathbf{2}$, and $\mathbf{4}$ denote the all-0, all-2, and all-4 vector.)

Now we can find a maximum-weight b''-matching x'' in strongly polynomial time. This follows from the fact that $b''(v) \leq 4\deg(v)$ for each vertex v. So we can consider the graph $G_{b''}$ obtained by splitting each vertex v of G into $b''(v)$ copies, and replacing any edge uv by $b''(u)b''(v)$ edges connecting the $b''(u)$ copies of u by the $b''(v)$ copies of v. Then a maximum-weight matching in $G_{b''}$ gives a maximum-weight b''-matching x'' in G''.

Then $x''+z$ is a b'-matching, since $x''+z \geq \mathbf{0}$ and $M(x''+z) \leq b''+Mz \leq b'$. Moreover, $x''+z$ is a maximum-weight b'-matching, since by Lemma 31.4β, there exists a maximum-weight b'-matching x' satisfying $x - \mathbf{2} \leq x' \leq x + \mathbf{2}$. Then $x'-z$ is a b''-matching (since $x'-z \leq \mathbf{4}$), and hence $w^{\mathsf{T}}x'' \geq w^{\mathsf{T}}(x'-z)$. Therefore $w^{\mathsf{T}}(x'' + z) \geq w^{\mathsf{T}}x'$. ∎

Elaboration of this method gives an $O(n^2m(n^2+m\log n))$-time algorithm. A similar approach of Anstee [1987] gives $O((m+n\log n)n\log\|b\|_\infty + n^2m)$- and $O(n^2\log n(m+n\log n))$-time algorithms.

For weighted *perfect* b-matching, a similar result follows:

Corollary 31.4a. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, and a weight function $w \in \mathbb{Q}^E$, a minimum-weight perfect b-matching can be found in strongly polynomial time.*

Proof. By flipping signs, it suffices to describe a method finding a maximum-weight perfect b-matching in strongly polynomial time.

We can increase each weight by a constant $C := BW + W$, where $W := \|w\|_\infty + 1$ and $B := \|b\|_1$. So each weight becomes $\geq C - W$ and $\leq C + W$. Then each perfect b-matching has weight at least $\frac{1}{2}B(C-W) = \frac{1}{2}B^2W$, while each nonperfect b-matching has weight at most

$$(31.33) \qquad \begin{aligned}(\tfrac{1}{2}B - 1)(C + W) &= \tfrac{1}{2}BC + \tfrac{1}{2}BW - C - W \\ &= \tfrac{1}{2}B^2W + \tfrac{1}{2}BW + \tfrac{1}{2}BW - BW - W - W < \tfrac{1}{2}B^2W.\end{aligned}$$

So each maximum-weight b-matching is perfect. Therefore, Theorem 31.4 applies. (Alternatively, we could repeat the above reduction process.) ∎

31.5. If b is even

The results on b-matchings can be simplified if b is even. In that case, the proofs can be reduced to the bipartite case. The maximum size of a $2b$-

matching is equal to the minimum weight of a 2-vertex cover, taking b as weight:

Theorem 31.5. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the maximum size of a $2b$-matching is equal to the minimum value of $y^\mathsf{T} b$ taken over 2-vertex covers y; equivalently, the minimum value of*

$$b(V) + b(N(S)) - b(S), \tag{31.34}$$

taken over stable sets S.

Proof. Make a bipartite graph H as follows. Make a new vertex v' for each $v \in V$, and let $V' := \{v' \mid v \in V\}$. H has vertex set $V \cup V'$ and edges all $u'v$ and uv' for $uv \in E$.

Define $b' : V \cup V' \to \mathbb{Z}_+$ by $b'(v) := b'(v') := b(v)$ for all $v \in V$. Then the maximum size of a $2b$-matching in G is equal to the maximum size of a b'-matching in H. By Corollary 21.1a, this is equal to the minimum b'-weight of a vertex cover in H, which is equal to the minimum of $y^\mathsf{T} b$ over 2-vertex covers y. ∎

It implies the following characterization of the existence of perfect b-matchings for even b:

Corollary 31.5a. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ with b even. Then there exists a perfect b-matching if and only if $b(N(S)) \geq b(S)$ for each stable set S of G.*

Proof. Directly from Theorem 31.5. ∎

This can also be derived directly from Corollary 31.1a. The following two theorems can be derived from the bipartite case in a way similar to the proof of Theorem 31.5, but they also are special cases of results in this chapter.

First we have a characterization of the $2b$-matching polytope:

Theorem 31.6. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the $2b$-matching polytope is determined by*

$$\begin{array}{lll} \text{(i)} & x_e \geq 0 & \text{for each } e \in E, \\ \text{(ii)} & x(\delta(v)) \leq 2b(v) & \text{for each } v \in V. \end{array} \tag{31.35}$$

Proof. This is a special case of Theorem 31.2. ∎

Second, we mention a result of Gallai [1957,1958a,1958b]. For a graph $G = (V, E)$ and $w : E \to \mathbb{Z}_+$, a *w-vertex cover* is a function $y : V \to \mathbb{Z}_+$ satisfying $y_u + y_v \geq w(uv)$ for each edge uv.

Theorem 31.7. *Let $G = (V, E)$ be a graph and let $w \in \mathbb{Z}_+^E$ and $b \in \mathbb{Z}_+^V$. Then the maximum weight $w^\mathsf{T} x$ of a $2b$-matching x is equal to the minimum value of $y^\mathsf{T} b$ taken over $2w$-vertex covers y.*

Proof. This follows from Theorem 31.3. ∎

31.6. If b is constant

The results on b-matchings can be specialized to 'k-matchings'. Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. A *k-matching* is a function $x \in \mathbb{Z}_+^E$ with $x(\delta(v)) \leq k$ for each vertex v. Thus if we identify k with the all-k vector in $\mathbb{Z}_+^V$, we have a k-matching as before. Therefore, Theorem 31.1 gives a min-max relation for maximum-size k-matching:

Theorem 31.8. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. Then the maximum size of a k-matching is equal to the minimum value of*

$$k|U| + \sum_K \lfloor \tfrac{1}{2} k|K| \rfloor, \tag{31.36}$$

taken over $U \subseteq V$, where K ranges over the components of $G - U$ spanning at least one edge.

Proof. Directly from Theorem 31.1. ∎

Note that it follows that if k is even, we need not round, and hence the maximum size of a k-matching is equal to $\frac{1}{2}k$ times the maximum-size of a 2-matching. This maximum size is described in Theorem 30.1.

Again, a k-matching x is *perfect* if $x_v = k$ for each vertex v. In characterizing the existence, it is convenient to distinguish between the cases of k odd and k even. Let I_U denote the set of isolated (hence loopless) vertices of $G - U$.

Corollary 31.8a. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$ be odd. Then G has a perfect k-matching if and only if for each $U \subseteq V$, $G - U - I_U$ has at most $k(|U| - |I_U|)$ odd components K.*

Proof. Directly from Corollary 31.1a. ∎

For even k, there is the following result due to Tutte [1952]:

Corollary 31.8b. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$ be even. Then G has a perfect k-matching if and only if $|N(S)| \geq |S|$ for each stable set S.*

Proof. Directly from Corollary 31.5a. ∎

So if k is even, there exists a perfect k-matching if and only if there exists a perfect 2-matching.

We also give the characterization of the k-*matching polytope* (the convex hull of k-matchings):

Theorem 31.9. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. Then the k-matching polytope is determined by*

(31.37)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $x(\delta(v)) \leq k$ for each $v \in V$,
(iii) $x(E[U]) \leq \lfloor \frac{1}{2}k|U| \rfloor$ for each $U \subseteq V$ with $k|U|$ odd.

Proof. This is a special case of Theorem 31.2. ∎

31.7. Further results and notes

31.7a. Complexity survey for the b-matching problem

Complexity survey for the maximum-weight b-matching problem:

*	$O(n^2 B)$	Pulleyblank [1973]
	$O(n^2 m \log B)$	W.H. Cunningham and A.B. Marsh, III (cf. Marsh [1979])
*	$O(m^2 \log n \log B)$	Gabow [1983a]
*	$O(n^2 m + n \log B(m + n \log n))$	Anstee [1987]
*	$O(n^2 \log n(m + n \log n))$	Anstee [1987]

Here $B := \|b\|_\infty$, and $*$ indicates an asymptotically best bound in the table.

Johnson [1965] extended Edmonds' matching algorithm to an algorithm (not based on splitting vertices) finding a maximum-size b-matching, with running time polynomially bounded in n, m, and B. Gabow [1983a] gave an $O(nm \log n)$-time algorithm to find a maximum-size b-matching.

31.7b. Facets and minimal systems for the b-matching polytope

Edmonds and Pulleyblank (see Pulleyblank [1973]) described the facets of the b-matching polytope. Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Call G b-*critical* if for each $u \in V$ there exists a b-matching x such that $x(\delta(u)) = b(u) - 1$ and $x(\delta(v)) = b(v)$ for each $v \neq u$.

Let G be simple and connected with at least three vertices and let $b > \mathbf{0}$. Then an inequality $x(\delta(v)) \leq b(v)$ determines a facet of the b-matching polytope if and only if $b(N(v)) > b(v)$, and if $b(N(v)) = b(v) + 1$, then $E[N(v)] \neq \emptyset$.

Moreover, an inequality $x(E[U]) \leq \lfloor \frac{1}{2}b(U) \rfloor$ determines a facet if and only if $G[U]$ is b-critical and has no cut vertex v with $b(v) = 1$.

Unlike in the matching case, the facet-inducing inequalities do not form a totally dual integral system. The minimal TDI-system for the b-matching polytope was characterized by Cook [1983a] and Pulleyblank [1981]. To describe this, call a graph $G = (V, E)$ *b-bicritical* if G is connected and for each $u \in V$ there is a b-matching x with $x(\delta(u)) = b(u) - 2$ and $x(\delta(v)) = b(v)$ for each $v \neq u$. Then a minimal TDI-system for the b-matching polytope (if G is simple and connected and has at least three vertices and if $b > \mathbf{0}$) is obtained by adding the following to the facet-inducing inequalities:

(31.38) $x(E[U]) \leq \frac{1}{2}b(U)$ for each $U \subseteq V$ with $|U| \geq 3$, $G[U]$ b-bicritical and $b(v) \geq 2$ for each $v \in N(U)$.

(The facets of the 2-matching polytope of a complete graph were also given by Grötschel [1977b].)

The vertices of the 2-matching polytope are characterized by:

Theorem 31.10. *Let $G = (V, E)$ be a graph. Then a 2-matching x is a vertex of the 2-matching polytope P if and only if the edges e with $x_e = 1$ form vertex-disjoint odd circuits.*

Proof. Let x be a 2-matching. Define $F := \{e \in E \mid x_e = 1\}$. Clearly, $\deg_F(v) \leq 2$ for each $v \in V$. So F forms a vertex-disjoint set of paths and circuits.

To see necessity in the theorem, let x be a vertex of P. Suppose that K is a component of F that forms a path or an even circuit. Then we can split K into matchings M and N. Then both $x + \chi^M - \chi^N$ and $x - \chi^M + \chi^N$ belong to P, contradicting the fact that x is a vertex of P.

To see sufficiency, suppose that x is not a vertex of P. Then there exists a nonzero vector y such that $x + y$ and $x - y$ belong to P. If $x_e = 0$ or $x_e = 2$, then $y_e = 0$, as $0 \leq x_e \pm y_e \leq 2$. If e and f are two edges in F incident with a vertex v, then $y_e = -y_f$, since $(x_e + x_f) \pm (y_e + y_f) \leq 2$. Hence, if each component of F is an odd circuit, we have $y = \mathbf{0}$, contradicting our assumption. ∎

31.7c. Regularizable graphs

A graph $G = (V, E)$ is called *regularizable* if there exists a k and a perfect k-matching x with $x \geq \mathbf{1}$. So we obtain a k-regular graph by replacing each edge e by x_e parallel edges. Berge [1978c] characterized regularizability as follows:

Theorem 31.11. *Let $G = (V, E)$ be connected and nonbipartite. Then G is regularizable if and only if $|N(U)| > |U|$ for each nonempty stable set U.*

Proof. Necessity being easy, we show sufficiency. Make a bipartite graph H by making for each vertex v a copy v', and replacing any edge uv by two edges uv' and $u'v$. Then every edge of H belongs to some perfect matching of H. To see this, suppose that edge uv' belongs to no perfect matching. Then by Frobenius' theorem (Corollary 16.2a), there exists a subset X of $V \setminus \{u\}$ such that X has less than $|X|$ neighbours in $V' \setminus \{v'\}$ (in the graph H; here $V' := \{v' \mid v \in V\}$). That is, defining $N'(X) := \cup_{u \in X} N_G(u)$,

(31.39) $|N'(X) \setminus \{v\}| < |X|.$

Let $U := X \setminus N'(X)$. Then U is a stable set. Moreover, $N(U) \subseteq N'(X) \setminus X$. By (31.39), $|N'(X)| \leq |X|$, and therefore $|N(U)| \leq |U|$. So by the condition given in the theorem, $U = \emptyset$; that is, $X \subseteq N'(X)$, and so, by (31.39), $X = N'(X)$. However, as G is connected and nonbipartite, H is connected. This contradicts the fact that $X = N'(X)$ and $X \neq V$.

So each edge of H belongs to a perfect matching. Hence each edge of G belongs to a perfect 2-matching. Adding up these perfect 2-matchings gives a perfect k-matching $x \geq \mathbf{1}$ for some k. ∎

Berge [1978b] remarked that this theorem is equivalent to: a connected nonbipartite graph G is regularizable if and only if the only 2-vertex cover of size $\tau_2(G)$ is the all-1 vector (this follows with (30.2)).

With the help of b-matchings, one can also characterize k-regularizable graphs — graphs that can become k-regular by adding edges parallel to existing edges. Let I_U denote the set of isolated (hence loopless) vertices of $G - U$.

Theorem 31.12. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. Then G is k-regularizable if and only if for each $U \subseteq V$, $G - U - I_U$ has at most*

(31.40) $$k|U| - k|I_U| - 2|E[U]| - |\delta(U \cup I_U)|$$

components K with $k|K| + |\delta(K)|$ odd.

Proof. From Corollary 31.1a applied to $b : V \to \mathbb{Z}_+$ defined by $b(v) := k - \deg(v)$ for $v \in V$.

Note that the condition implies $b(v) \geq 0$ for each vertex v. For suppose $\deg(v) > k$. If $k = 0$, then (31.40) is negative for $U := V$, a contradiction. So $k > 0$. Taking $U := \{v\}$, the condition implies that $k - k|I_U| - 2|E[U]| - |\delta(\{v\} \cup I_U)| \geq 0$. As $|\delta(v)| > k$, it follows that $I_U \neq \emptyset$, hence $|I_U| = 1$, say $I_U = \{w\}$. So $|E[U]| = 0$, that is, v is loopless. Moreover, $\delta(U \cup I_U) = \emptyset$, that is, $\{v, w\}$ is a component of G. But then the nonnegativity of (31.40) for $U' := \{v, w\}$ implies $2k \geq 2|E[U']| \geq 2\deg(v)$ (as v is loopless), a contradiction. ∎

See also Berge [1978b,1978d,1981].

31.7d. Further notes

Hoffman and Oppenheim [1978] showed that system (31.17) is 'locally strongly modular'; that is, each vertex of the b-matching polytope is determined by a linearly independent set of inequalities among (31.17) (set to equality), where the matrix in the system has determinant ± 1.

Johnson [1965] characterized the vertices of the fractional b-matching polytope. Koch [1979] studied bases (in the sense of the simplex method) for the linear programming problem of finding a maximum-weight b-matching.

Padberg and Wolsey [1984] described a strongly polynomial-time algorithm to find for any vector x the largest λ such that $\lambda \cdot x$ belongs to the b-matching polytope, and to describe $\lambda \cdot x$ as a convex combination of b-matchings.

b-matching algorithms are studied in the books by Gondran and Minoux [1984] and Derigs [1988a].

Chapter 32

Capacitated b-matchings

In the previous chapter we studied b-matchings, without upper bound given on the values of the edges. In this chapter we refine the results to the case where each edge has a prescribed 'capacity' that bounds the value on the edge. This can be reduced to uncapacitated b-matching.

32.1. Capacitated b-matchings

The capacitated b-matching problem considers b-matchings x satisfying a prescribed capacity constraint $x \leq c$. By a construction of Tutte [1954b], results on capacitated b-matchings can be derived from the results for the uncapacitated case as follows. Denote

$$E[X, Y] := \{e \in E \mid \exists x \in X, y \in Y : e = \{x, y\}\}. \tag{32.1}$$

Theorem 32.1. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$ with $c > \mathbf{0}$. Then the maximum size of a b-matching $x \leq c$ is equal to the minimum value of*

$$b(U) + c(E[W]) + \sum_K \lfloor \tfrac{1}{2}(b(K) + c(E[K, W])) \rfloor, \tag{32.2}$$

taken over disjoint subsets U, W of V, where K ranges over the components of $G - U - W$.

Proof. To see that the maximum is not more than the minimum, let x be a b-matching with $x \leq c$ and let U, W be disjoint subsets of V. Then $x(E[U] \cup \delta(U)) \leq b(U)$ and $x(E[W]) \leq c(E[W])$. Consider next a component K of $G - U - W$. Then $2x(E[K]) + x(E[K, W]) \leq b(K)$ and $x(E[K, W]) \leq c(E[K, W])$. Hence $x(E[K] \cup E[K, W]) \leq \lfloor \frac{1}{2}(b(K) + c(E[K, W])) \rfloor$, and the inequality follows.

The reverse inequality is proved by reduction to Theorem 31.1. Make a new graph $G' = (V', E')$ by replacing each edge of G by a path of length three. That is, for each edge $e = uv$ introduce two new vertices $p_{e,u}$ and $p_{e,v}$ and three edges: $up_{e,u}$, $p_{e,u}p_{e,v}$, and $p_{e,v}v$.

Define $b' \in \mathbb{Z}_+^{V'}$ by $b'(v) := b(v)$ if $v \in V$ and $b'(p_{e,v}) := c(e)$ for any new vertex $p_{e,v}$. Then the maximum size of a b-matching x in G with $x \leq c$ is equal to the maximum size of a b'-matching in G', minus $c(E)$. By Theorem 31.1, there exists a subset U' of V' such that the maximum size of a b'-matching in G' equals

(32.3) $$b'(U') + \sum_{K'} \lfloor \tfrac{1}{2} b'(K') \rfloor,$$

where K' ranges over the components of $G' - U'$ with $|K'| \geq 2$. (Note that G' has no loops.) We choose U' with $|U'|$ as small as possible.

Let $U := V \cap U'$ and let W be the set of isolated vertices of $G' - U'$ that belong to V. We show that (32.2) is at most (32.3) minus $c(E)$, which proves the theorem.

First observe that

(32.4) if $p_{e,v} \in U'$, then $v \in W$.

Otherwise, deleting $p_{e,v}$ from U' does not increase (32.3), contradicting the minimality of $|U'|$. (Here we use that $p_{e,v}$ has degree 2 and that $b'(p_{e,v}) > 0$, that is, $c(e) > 0$. Then $b'(U')$ decreases by $c(e)$ while the sum in (32.3) increases by at most $\lfloor \frac{1}{2} c(e) + 1 \rfloor$, which is at most $c(e)$.)

Hence

(32.5) $$\begin{aligned} b'(U') &= b(U) + b'(U' \setminus V) = b(U) + \sum_{v \in W} c(\delta(v)) \\ &= b(U) + 2c(E[W]) + c(\delta(W)). \end{aligned}$$

Consider a component K' of $G' - U'$ with $|K'| \geq 2$. If K' does not intersect V, then it is equal to $\{p_{e,u}, p_{e,v}\}$ for some edge $e = uv$ of G with $u, v \in U$. So $b'(K') = 2c(e)$. If K' intersect V, let $K := K' \cap V$. Then K is a component of $G - U - W$. Indeed, any edge spanned by K gives a path of length 3 in K' (by (32.4)), and any path in K' between vertices in K gives a path in K. Any edge of G leaving K gives a path of length 3 in G' connecting K' and $U \cup W$. So

(32.6) $$K' = K \cup \{p_{e,u} \mid e = uv \in E, u \in K\} \cup \{p_{e,v} \mid e = uv \in E, u \in K, v \in U\}.$$

Hence

(32.7) $$b'(K') = b(K) + c(E[K, W]) + 2c(E[K]) + 2c(E[K, U]).$$

Therefore, (32.3) is equal to

(32.8) $$\begin{aligned} &b(U) + 2c(E[W]) + c(\delta(W)) + c(E[U]) \\ &+ \sum_K \big(\lfloor \tfrac{1}{2}(b(K) + c(E[K, W])) \rfloor + c(E[K]) + c(E[K, U]) \big), \end{aligned}$$

where K ranges over the components of $G - U - W$. Since

(32.9) $$c(E) = c(E[W]) + c(\delta(W)) + c(E[U]) + \sum_K (c(E[K]) + c(E[K, U])),$$

(32.3) minus $c(E)$ is equal to (32.2). ∎

This implies for *perfect* b-matchings:

Corollary 32.1a. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$ with $c > \mathbf{0}$. Then G has a perfect b-matching $x \leq c$ if and only if for each partition T, U, W of V, $G[T]$ has at most*

(32.10) $$b(U) - b(W) + 2c(E[W]) + c(E[T, W])$$

components K with $b(K) + c(E[K, W])$ odd.

Proof. Directly from Theorem 32.1. ∎

32.2. The capacitated b-matching polytope

Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$. The *c-capacitated b-matching polytope* is the convex hull of the b-matchings x satisfying $x \leq c$. A description of this polytope follows again from that for the uncapacitated b-matching polytope.

Theorem 32.2. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$. The c-capacitated b-matching polytope is determined by*

(32.11)
$$\begin{array}{rll} \text{(i)} & 0 \leq x_e \leq c(e) & (e \in E), \\ \text{(ii)} & x(\delta(v)) \leq b(v) & (v \in V), \\ \text{(iii)} & x(E[U]) + x(F) \leq \lfloor \tfrac{1}{2}(b(U) + c(F)) \rfloor & (U \subseteq V, F \subseteq \delta(U), \\ & & b(U) + c(F) \text{ odd}). \end{array}$$

Proof. It is easy to show that each b-matching $x \leq c$ satisfies (32.11). To show that the inequalities (32.11) completely determine the c-capacitated b-matching polytope, let $x \in \mathbb{R}^E$ satisfy (32.11). Let $G' = (V', E')$ and $b' \in \mathbb{Z}_+^{V'}$ be as in the proof of Theorem 32.1. Define $x' \in \mathbb{R}^{E'}$ by $x'(up_{e,u}) := x'(vp_{e,v}) := x_e$ and $x'(p_{e,u}p_{e,v}) := c(e) - x_e$, for any edge $e = uv$ of G. We show that x' belongs to the b'-matching polytope with respect to G'.

By Theorem 31.2, it suffices to check the constraints (31.6) for x' with respect to G' and b'. That is, we should check (where $\delta' := \delta_{G'}$ and $E'[U']$ is the set of edges in E' spanned by U'):

(32.12)
$$\begin{array}{rll} \text{(i)} & x'(e') \geq 0 & (e' \in E'), \\ \text{(ii)} & x'(\delta'(v')) \leq b'(v') & (v' \in V'), \\ \text{(iii)} & x'(E'[U']) \leq \lfloor \tfrac{1}{2} b'(U') \rfloor & (U' \subseteq V' \text{ with } b'(U') \text{ odd}). \end{array}$$

Trivially we have (32.12)(i) by (32.11)(i). Moreover, for each vertex $v \in V$ one has $x'(\delta'(v)) \leq b'(v)$ by (32.11)(ii). For any vertex $p_{e,u}$ of G', with $e = uv \in E$, one has $x'(\delta'(p_{e,u})) = c(e) = b'(p_{e,u})$.

To prove (32.12)(iii), we first show that it suffices to prove it for those $U' \subseteq V'$ satisfying for each edge $e = uv \in E$:

(32.13) (i) if $u, v \in U'$, then $p_{e,u} \in U'$ and $p_{e,v} \in U'$,
(ii) if $p_{e,u} \in U'$, then $u \in U'$.

To see (32.13)(i), first let $u, v \in U'$ and $p_{e,u} \notin U'$. Define $U'' := U' \cup \{p_{e,u}, p_{e,v}\}$. Then

(32.14)
$$x'(E'[U']) \leq x'(E'[U'']) - x'(\delta'(p_{e,u})) \leq \lfloor \tfrac{1}{2} b'(U'') \rfloor - b'(p_{e,u}) \leq \lfloor \tfrac{1}{2} b'(U') \rfloor.$$

To see (32.13)(ii), let $p_{e,u} \in U'$ and $u \notin U'$. Define $U'' := U' \setminus \{p_{e,u}, p_{e,v}\}$. If $p_{e,v} \notin U'$, then

(32.15)
$$x'(E'[U']) = x'(E'[U'']) \leq \lfloor \tfrac{1}{2} b'(U'') \rfloor \leq \lfloor \tfrac{1}{2} b'(U') \rfloor.$$

If $p_{e,v} \in U'$, then

(32.16)
$$x'(E'[U']) = x'(E'[U'']) + x'(\delta'(p_{e,v})) \leq \lfloor \tfrac{1}{2} b'(U'') \rfloor + b'(p_{e,v}) = \lfloor \tfrac{1}{2} b'(U') \rfloor.$$

This proves that we may assume (32.13) (as repeated application of these modifications gives finally (32.13)). Let $U := U' \cap V$ and let F be the set of those edges $e = uv$ in $\delta(U)$ with $u \in U$, $v \notin U$, and $p_{e,u} \in U'$. Then $x'(E'[U']) = x(E[U]) + c(E[U]) + x(F)$ and $b'(U') = b(U) + 2c(E[U]) + c(F)$. Hence (32.11)(iii) implies (32.12)(iii).

So x' is a convex combination of b'-matchings in G'. Each such b'-matching y satisfies $y(\delta'(v')) = b'(v')$ for each 'new' vertex $v' = p_{e,u}$ (as x' satisfies this equality). Hence each such b'-matching corresponds to a b-matching subject to c in G, and we obtain x as convex combination of b-matchings subject to c in G. ∎

Similarly, the *c-capacitated perfect b-matching polytope* is the convex hull of the perfect b-matchings x satisfying $x \leq c$. Theorem 32.2 implies the following (announced by Edmonds and Johnson [1970] (cf. Green-Krótki [1980], Aráoz, Cunningham, Edmonds, and Green-Krótki [1983])):

Corollary 32.2a. *The c-capacitated perfect b-matching polytope is determined by*

(32.17)
(i)	$0 \leq x_e \leq c(e)$	$(e \in E)$,
(ii)	$x(\delta(v)) = b(v)$	$(v \in V)$,
(iii)	$x(\delta(U) \setminus F) - x(F) \geq 1 - c(F)$	$(U \subseteq V, F \subseteq \delta(U),$ $b(U) + c(F)$ odd).

Proof. Directly from Theorem 32.2, as (32.17)(ii) implies that $x(E[U]) = \frac{1}{2}b(U) - \frac{1}{2}x(\delta(U))$. ∎

32.3. Total dual integrality

System (32.11) generally is not TDI (cf. the example in Section 30.5). To obtain a TDI-system, one should delete the restriction in (32.11)(iii) that $b(U) + c(F)$ is odd. Thus we obtain:

(32.18)

$$\begin{array}{rll} \text{(i)} & 0 \le x_e \le c(e) & (e \in E), \\ \text{(ii)} & x(\delta(v)) \le b(v) & (v \in V), \\ \text{(iii)} & x(E[U]) + x(F) \le \lfloor \frac{1}{2}(b(U) + c(F)) \rfloor & (U \subseteq V, F \subseteq \delta(U)). \end{array}$$

Theorem 32.3. *System* (32.18) *is TDI.*

Proof. Let $G' = (V', E')$ and $b' \in \mathbb{Z}_+^{V'}$ be as in the proof of Theorem 32.1. By Corollary 31.3a, the following system is TDI:

(32.19)

$$\begin{array}{rll} \text{(i)} & x'(e') \ge 0 & (e' \in E'), \\ \text{(ii)} & x'(\delta'(v')) \le b'(v') & (v' \in V'), \\ \text{(iii)} & x'(E'[U']) \le \lfloor \frac{1}{2} b'(U') \rfloor & (U' \subseteq V'). \end{array}$$

Since setting inequalities to equalities maintains total dual integrality (Theorem 5.25), the following system is TDI:

(32.20)

$$\begin{array}{rll} \text{(i)} & x'(e') \ge 0 & (e' \in E'), \\ \text{(ii)} & x'(\delta'(v)) \le b(v) & (v \in V), \\ \text{(iii)} & x'(up_{e,u}) + x'(p_{e,u}p_{e,v}) = c(e) & (u \in e = uv \in E), \\ \text{(iv)} & x'(E'[U']) \le \lfloor \frac{1}{2} b'(U') \rfloor & (U' \subseteq V'). \end{array}$$

The inequalities (32.14), (32.15), and (32.16) show that in (32.20)(iv) we may restrict the U' to those satisfying (32.13). So U' is determined by $U := U' \cap V$ and $F := \{e = uv \in E \mid u, p_{e,u} \in U', v \notin U'\}$.

Moreover, with Theorem 5.27 we can eliminate the variables $x'(up_{e,u})$ for $e \in E$ and $u \in e$ with the equalities (32.20)(iii). That is, we replace $x'(up_{e,u})$ by $c(e) - y_e$, where we set $y_e := x'(p_{e,u}p_{e,v})$ for $e = uv \in E$. Then:

(32.21) $x'(E'[U']) = y(E[U]) + 2(c(E[U]) - y(E[U])) + c(F) - y(F)$ and $b'(U') = b(U) + 2c(E[U]) + c(F)$.

Hence the system becomes:

(32.22)

$$\begin{array}{rll} \text{(i)} & y_e \ge 0 & (e \in E), \\ \text{(ii)} & y_e \le c(e) & (e \in E), \\ \text{(iii)} & -y(\delta(v)) \le b(v) - c(\delta(v)) & (v \in V), \\ \text{(iv)} & -y(E[U]) - y(F) \le \lfloor \frac{1}{2} b(U) + c(F) \rfloor - c(E[U]) - c(F) & \\ & & (U \subseteq V, F \subseteq \delta(U)). \end{array}$$

Setting y_e to $c(e) - x_e$, the system becomes (32.18) and remains TDI. ∎

32.4. The weighted capacitated b-matching problem

By the construction given in the proof of Theorem 32.1, the weighted capacitated b-matching problem can easily be reduced to the uncapacitated variant:

Theorem 32.4. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, $c \in \mathbb{Z}_+^E$, and a weight function $w \in \mathbb{Q}^E$, a maximum-weight b-matching $x \leq c$ can be found in strongly polynomial time.*

Proof. We may assume that $w \geq \mathbf{0}$. Make $G' = (V', E')$ and $b' \in \mathbb{Z}_+^{V'}$ as in the proof of Theorem 32.1. Moreover, define a weight function w' on the edges of G' by $w'(up_{e,u}) := w'(p_{e,u}p_{e,v}) := w'(p_{e,v}v) := w(e)$ for any edge $e = uv$ of G.

Let x' be a maximum-weight b'-matching in G'. Then we may assume that for each edge $e = uv$ of G one has $x'(up_{e,u}) = c(e) - x'(p_{e,u}p_{e,v}) = x'(p_{e,v}v)$. (This follows from the fact that we can assume that $x'(up_{e,u}) = x'(p_{e,v}v)$, since if say $x'(up_{e,u}) = x'(p_{e,v}v) + \tau$ with $\tau > 0$, we can decrease $x'(up_{e,u})$ by τ and increase $x'(p_{e,u}p_{e,v})$ by τ. Next we can reset $x'(p_{e,u}p_{e,v}) := c(e) - x'(up_{e,u})$.)

Now define $x_e := x'(up_{e,u})$ for each edge $e = uv$ of G. Then x is a maximum-weight b-matching with $x \leq c$. ∎

Similarly, for the weighted capacitated perfect b-matching problem:

Theorem 32.5. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, $c \in \mathbb{Z}_+^E$, and a weight function $w \in \mathbb{Q}^E$, a minimum-weight perfect b-matching $x \leq c$ can be found in strongly polynomial time.*

Proof. As in the previous proof replace each edge by a path of length three, yielding the graph G', and define b', and w' similarly. Let x' be a maximum-weight perfect b'-matching in G'. Then for each edge $e = uv$ of G one has $x'(up_{e,u}) = c(e) - x'(p_{e,u}p_{e,v}) = x'(p_{e,v}v)$. Defining $x_e := x'(up_{e,u})$ for each edge $e = uv$ of G, gives a maximum-weight b-matching $x \leq c$. ∎

32.4a. Further notes

Cook [1983b] and Cook and Pulleyblank [1987] determined the facets and the minimal TDI-system for the capacitated b-matching polytope.

Johnson [1965] extended Edmonds' matching algorithm to an algorithm (not based on reduction to matching) that finds a maximum-size capacitated b-matching, with running time bounded by a polynomial in n, m, and $\|b\|_\infty$. Gabow [1983a] gave an $O(nm \log n)$-time algorithm for this.

Cunningham and Green-Krótki [1991] showed the following. Let $G = (V, E)$ be a graph, let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$. Then the convex hull of the integer vectors $y \leq b$ for which there is a perfect y-matching $x \leq c$ is determined by the inequalities

(32.23)
$$\mathbf{0} \leq y \leq b,$$
$$y(\bigcup_{i=0}^{k} A_i) - y(B) \leq \sum_{i=1}^{k}(b(A_i) - 1) + c(E[A_0]) + c(E[A_0, V \setminus B]),$$

where A_0 and B are disjoint subsets of V and where $A_1, \ldots, A_k$ are some of the components of $G - A_0 - B$ such that $b(A_i) + c(E[A_0, A_i])$ is odd for each $i = 1, \ldots, k$.

This characterizes the convex hull of degree-sequences of capacitated b-matchings, where the *degree-sequence* of $x \in \mathbb{Z}^E$ is the vector $y \in \mathbb{Z}^E$ defined by $y_v = x(\delta(v))$ for $v \in V$.

This generalizes the results of Balas and Pulleyblank [1989] on the matchable set polytope (Section 25.5d) and of Koren [1973] on the convex hull of degree-sequences of simple graphs (Section 33.6c below). See also Cunningham and Green-Krótki [1994] and Cunningham and Zhang [1992].

Chapter 33

Simple b-matchings and b-factors

A special case of capacitated b-matchings is obtained when we take capacity 1 on every edge. So the b-matching takes values 0 and 1 only. Such a b-matching is called *simple*. A simple b-matching is the incidence vector of some set of edges. If the b-matching is simple and perfect it is called a *b-factor*.
In this chapter we derive results on simple b-matchings and b-factors in a straightforward way from those on capacitated b-matchings obtained in the previous chapter.

33.1. Simple b-matchings and b-factors

Call a b-matching x *simple* if x is a 0,1 vector. We can identify simple b-matchings with subsets F of E with $\deg_F(v) \leq b(v)$ for each $v \in V$.

Simple b-matchings are special cases of capacitated b-matchings, namely by taking capacity function $c = \mathbf{1}$. Hence a min-max relation for maximum-size simple b-matching follows from the more general capacitated version:

Theorem 33.1. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the maximum size of a simple b-matching is equal to the minimum value of*

$$b(U) + |E[W]| + \sum_K \lfloor \tfrac{1}{2}(b(K) + |E[K, W]|) \rfloor, \tag{33.1}$$

taken over all disjoint subsets U, W of V, where K ranges over the components of $G - U - W$.

Proof. The theorem is the special case $c = \mathbf{1}$ of Theorem 32.1. ∎

A *b-factor* is a simple perfect b-matching. In other words, it is a subset F of E with $\deg_F(v) = b(v)$ for each $v \in V$. The existence of a b-factor was characterized by Tutte [1952,1974] (cf. Ore [1957]):

Corollary 33.1a. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then G has a b-factor if and only if for each partition T, U, W of V, the graph $G[T]$ has at most*

(33.2) $\qquad b(U) - b(W) + 2|E[W]| + |E[T, W]|$

components K with $b(K) + |E[K, W]|$ odd.

Proof. Directly from Theorem 33.1 (or Corollary 32.1a). ∎

(An algorithmic proof was given by Anstee [1985], yielding an $O(n^3)$-time algorithm to find a b-factor. Tutte [1981] gave another proof and a sharpening.)

33.2. The simple b-matching polytope and the b-factor polytope

Given a graph $G = (V, E)$ and a vector $b \in \mathbb{Z}_+^V$, the *simple b-matching polytope* is the convex hull of the simple b-matchings in G. It can be characterized by (Edmonds [1965b]):

Theorem 33.2. *The simple b-matching polytope is determined by*

$$\text{(33.3)}\qquad \begin{array}{rll} \text{(i)} & 0 \le x_e \le 1 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \le b(v) & (v \in V), \\ \text{(iii)} & x(E[U]) + x(F) \le \lfloor \frac{1}{2}(b(U) + |F|) \rfloor & (U \subseteq V, F \subseteq \delta(U), \\ & & b(U) + |F| \text{ odd}). \end{array}$$

Proof. The theorem is a special case of Theorem 32.2. ∎

Given a graph $G = (V, E)$ and a vector $b \in \mathbb{Z}_+^V$, the *b-factor polytope* is the convex hull of (the incidence vectors of) b-factors in G. As it is a face of the simple b-matching polytope (if nonempty), we have:

Corollary 33.2a. *The b-factor polytope is determined by*

$$\text{(33.4)}\qquad \begin{array}{rll} \text{(i)} & 0 \le x_e \le 1 & (e \in E), \\ \text{(ii)} & x(\delta(v)) = b(v) & (v \in V), \\ \text{(iii)} & x(\delta(U) \setminus F) - x(F) \ge 1 - |F| & (U \subseteq V, F \subseteq \delta(U), \\ & & b(U) + |F| \text{ odd}). \end{array}$$

Proof. Directly from Theorem 33.2. ∎

33.3. Total dual integrality

Consider the system (extending (33.3)):

$$\text{(33.5)} \quad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq 1 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \leq b(v) & (v \in V), \\ \text{(iii)} & x(E[U]) + x(F) \leq \lfloor \frac{1}{2}(b(U) + |F|) \rfloor & (U \subseteq V, F \subseteq \delta(U)). \end{array}$$

A special case of Theorem 32.3 is (cf. Cook [1983b]):

Theorem 33.3. *System* (33.5) *is TDI.*

Proof. Directly from Theorem 32.3. ∎

It implies for the b-factor polytope:

Corollary 33.3a. *System* (33.4) *is totally dual half-integral.*

Proof. By Theorems 33.3 and 5.25, the system obtained from (33.5) by setting (33.5)(ii) to equality, is TDI. Then each inequality (33.5) is a half-integer sum of inequalities (33.4), and the theorem follows. ∎

This can be extended to:

Corollary 33.3b. *Let $w \in \mathbb{Z}^E$ with $w(C)$ even for each circuit C. Then the problem of minimizing $w^{\mathsf{T}}x$ subject to* (33.4) *has an integer optimum dual solution.*

Proof. If $w(C)$ is even for each circuit, there is a subset U of V with $\{e \in E \mid w(e) \text{ odd}\} = \delta(U)$. Now replace w by $w' := w + \sum_{v \in U} \chi^{\delta(v)}$. Then $w'(e)$ is an even integer for each edge e. Hence by Corollary 33.3a there is an integer optimum dual solution y'_v $(v \in V)$, z_U $(U \subseteq V$, $b(U)$ odd) for the problem of minimizing $w'^{\mathsf{T}}x$ subject to (33.4). Now setting $y_v := y'_v - 1$ if $v \in U$ and $y_v := y'_v$ if $v \notin U$ gives an integer optimum dual solution for w. ∎

33.4. The weighted simple b-matching and b-factor problem

Also algorithmic results can be derived from the general capacity case, but some arguments can be simplified. While finding a minimum-weight b-factor can be reduced to finding a minimum-weight perfect b-matching, there is a more direct construction, since we can assume that b is not too large. We give the precise arguments in the proofs below.

Theorem 33.4. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, and a weight function $w \in \mathbb{Q}^E$, a maximum-weight simple b-matching can be found in strongly polynomial time.*

Proof. We may assume that $b(v) \leq \deg_G(v)$ for each $v \in V$, since replacing $b(v)$ by $\min\{b(v), \deg_G(v)\}$ for each v does not change the problem.

Now the techniques described in Chapters 31 and 32 (replacing each vertex by $b(v)$ vertices, and next each edge by a path of length three), yield a strongly polynomial reduction to the maximum-weight matching problem. ∎

So a maximum-size simple b-matching and a b-factor (if any) can be found in polynomial time.

A similar construction applies to the weighted b-factor problem:

Theorem 33.5. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, and a weight function $w \in \mathbb{Q}^E$, a minimum-weight b-factor can be found in strongly polynomial time.*

Proof. We may assume that $b(v) \leq \deg_G(v)$ for each $v \in V$, since otherwise there is no b-factor. Now the reduction techniques described in Chapters 31 and 32 yield a strongly polynomial reduction to the minimum-weight perfect matching problem. ∎

33.5. If b is constant

Again we can specialize the results above to k-matchings and k-factors, for $k \in \mathbb{Z}_+$. First we have for the maximum size of a simple k-matching:

Theorem 33.6. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. The maximum size of a simple k-matching is equal to the minimum value of*

$$k|U| + |E[W]| + \sum_K \lfloor \tfrac{1}{2}(k|K| + |E[K, W]|) \rfloor, \tag{33.6}$$

taken over all disjoint subsets U, W of V, where K ranges over the components of $G - U - W$.

Proof. Directly from Theorem 33.1. ∎

A *k-factor* is a simple perfect k-matching. In other words, it is a subset F of E with (V, F) k-regular. Theorem 33.6 implies a classical theorem of Belck [1950]:

Corollary 33.6a. *A graph $G = (V, E)$ has a k-factor if and only if for each partition T, U, W of V, $G[T]$ has at most*

$$k(|U| - |W|) + 2|E[W]| + |E[T, W]| \tag{33.7}$$

components K with $k|K| + |E[K, W]|$ odd.

Proof. Directly from Theorem 33.6. ∎

Petersen [1891] showed that the following is easy:

Theorem 33.7. *Each connected $2k$-regular graph G with an even number of edges has a k-factor.*

Proof. Make an Eulerian tour in G, and colour the edges alternatingly red and blue. Then the red edges form a k-factor. ∎

33.6. Further results and notes

33.6a. Complexity results

Urquhart [1967] gave an $O(b(V)n^3)$-time algorithm for finding a maximum-weight simple b-matching. This was improved by Gabow [1983a] to $O(b(V)m\log n)$ (by reduction to the $O(nm\log n)$-time algorithm of Galil, Micali, and Gabow [1982, 1986] for maximum-weight matching) and to $O(b(V)n^2)$. For maximum-*size* simple b-matching, Gabow [1983a] gave algorithms of running time $O(\sqrt{b(V)}\,m)$ (by reduction to Micali and Vazirani [1980]) and to $O(nm\log n)$.

33.6b. Degree-sequences

A sequence $d_1,\ldots,d_n$ is called a *degree-sequence* of a graph $G=(V,E)$ if we can order the vertices as $v_1,\ldots,v_n$ such that $\deg_G(v_i)=d_i$ for $i=1,\ldots,n$.

From Corollary 33.1a one can derive the characterization of degree-sequences of simple graphs due to Erdős and Gallai [1960]: there exists a simple graph with degrees $d_1\geq d_2\geq\cdots\geq d_n\geq 0$ if and only if $\sum_{i=1}^n d_i$ is even and

$$\text{(33.8)}\qquad \sum_{i=1}^{k} d_i \leq k(k-1)+\sum_{i=k+1}^{n}\min\{k,d_i\}$$

for $k=1,\ldots,n$.

Havel [1955] gave the following recursive algorithm to decide if a sequence is the degree-sequence of a simple graph. A sequence $d_1\geq d_2\geq\cdots\geq d_n$ is the degree-sequence of a simple graph if and only if $0\leq d_n\leq n-1$ and $d_1-1,d_2-1,\ldots,d_{d_n}-1,d_{d_n+1},\ldots,d_{n-1}$ is the degree sequence of a simple graph.

Koren [1973] showed that the convex hull of degree-sequences of simple graphs on a finite vertex set V is determined by:

$$\text{(33.9)}\qquad \begin{array}{lll} \text{(i)} & x_v\geq 0 & \text{for each } v\in V,\\ \text{(ii)} & x(U)-x(W)\leq |U|(|V|-|W|-1) & \text{for disjoint } U,W\subseteq V. \end{array}$$

If the graph need not be simple (but yet is loopless), condition (33.8) can be replaced by $\sum_{i=2}^n d_i\geq d_1$, as can be shown easily (cf. Hakimi [1962a]). Related work was done by Peled and Srinivasan [1989], who showed that system (33.9) is totally dual integral and characterized vertices, facets, and adjacency on the polytope determined by (33.9).

Kundu [1973] showed that if both sequences $d_1\geq\cdots\geq d_n\geq k$ and $d_1-k\geq\cdots\geq d_n-k\geq 0$ are realizable (as degree-sequence of a simple graph), then the first sequence is realizable by a graph with a k-factor (answering a question of Grünbaum [1970]). See also Edmonds [1964] and Cai, Deng, and Zang [2000].

33.6c. Further notes

Cook [1983b] and Cook and Pulleyblank [1987] determined the facets and the minimal TDI-system for the simple b-matching polytope. Hausmann [1978a,1981] characterized adjacency on the simple b-matching polytope.

Lovász [1972f] extended the Edmonds-Gallai decomposition to b-factors (cf. Lovász [1972e] and Graver and Jurkat [1980]). For a sharpening of Corollary 33.1a by specializing T, U, W, see Tutte [1974,1978].

Fulkerson, Hoffman, and McAndrew [1965] showed the following. Let $G = (V, E)$ be a graph such that any two odd circuits have a vertex in common or are connected by an edge. Let $b \in \mathbb{Z}_+^V$. Then G has a b-factor if and only if $b(V)$ is even and

$$b(U) + 2|E[W]| + |E[T, W]| \geq b(W) \qquad (33.10)$$

for each partition T, U, W of V (cf. Mahmoodian [1977]).

Baebler [1937] showed that any k-regular l-connected graph has an l-factor if k is odd and l is even. Era [1985] proved the following conjecture of Akiyama [1982]: for each k there exists a t such that for each r-regular graph $G = (V, E)$ with $r \geq t$, E can be partitioned into $E_1, \ldots, E_s$ with for each $i = 1, \ldots, s$ one has $k \leq \deg_{E_i}(v) \leq k + 1$ for each vertex v.

Katerinis [1985] showed that if k', k, k'' are odd natural numbers with $k' \leq k \leq k''$, then any graph G having a k'-factor and a k''-factor, also has a k-factor. Related results are reported in Enomoto, Jackson, Katerinis, and Saito [1985].

Goldman [1964] studied augmenting paths for simple b-matchings by reduction to **1**-matchings. More on b-matchings and b-factors can be found in Bollobás [1978], Tutte [1984], and Bollobás, Saito, and Wormald [1985].

Chapter 34

b-edge covers

The covering analogue of a b-matching is the b-edge cover. It is not difficult to derive min-max relations, polyhedral characterizations, and algorithms for b-edge covers from those for b-matchings.

34.1. *b*-edge covers

Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. A *b-edge cover* is a function $x \in \mathbb{Z}_+^E$ satisfying

$$x(\delta(v)) \geq b(v) \tag{34.1}$$

for each $v \in V$.

There is a direct analogue of Gallai's theorem (Theorem 19.1), also given in Gallai [1959a], relating maximum-size b-matchings and minimum-size b-edge covers:

Theorem 34.1. *Let $G = (V, E)$ be a graph without isolated vertices and let $b \in \mathbb{Z}_+^V$. Then the maximum size of a b-matching plus the minimum size of a b-edge cover is equal to $b(V)$.*

Proof. Let x be a minimum-size b-edge cover. For any $v \in V$, reduce $x(\delta(v))$ by $x(\delta(v)) - b(v)$, by reducing x_e on edges $e \in \delta(v)$. We obtain a b-matching y of size

$$y(E) \geq x(E) - \sum_{v \in V}(x(\delta(v)) - b(v)) = b(V) - x(E). \tag{34.2}$$

Hence the maximum-size of a b-matching is at least $b(V) - x(E)$.

Conversely, let y be a maximum-size b-matching. For any $v \in V$, increase $y(\delta(v))$ by $b(v) - y(\delta(v))$, by increasing y_e on edges $e \in \delta(v)$. We obtain a b-edge cover x of size

$$x(E) \leq y(E) + \sum_{v \in V}(b(v) - y(\delta(v))) = b(V) - y(E). \tag{34.3}$$

Hence the minimum-size of a b-edge cover is at most $b(V) - y(E)$. ∎

(An alternative way of proving this is by applying Gallai's theorem for the case $b = \mathbf{1}$ directly to the graph G_b described in (31.2), obtained from G by splitting any vertex v into $b(v)$ vertices.)

With Theorem 34.1, we can derive a min-max relation for minimum-size b-edge cover from that for maximum-size b-matching. Let I_U denote the set of isolated (hence loopless) vertices of $G - U$.

Corollary 34.1a. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the minimum size of a b-edge cover is equal to the maximum value of*

$$(34.4) \qquad b(I_U) + \sum_K \lceil \tfrac{1}{2} b(K) \rceil,$$

taken over $U \subseteq V$, where K ranges over the components of $G - U - I_U$.

Proof. Directly from Theorems 34.1 and 31.1. ∎

The construction in the proof of Theorem 34.1 also implies that a minimum-size b-edge cover can be found in polynomial time.

34.2. The b-edge cover polyhedron

Given a graph $G = (V, E)$ and $b \in \mathbb{Z}_+^V$, the *b-edge cover polyhedron* is the convex hull of the b-edge covers. The inequalities describing the b-edge cover polyhedron can be easily derived from the description of the edge cover polytope, similar to Theorem 31.2.

Theorem 34.2. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the b-edge cover polyhedron is determined by the inequalities*

$$(34.5) \qquad \begin{array}{rll} \text{(i)} & x_e \geq 0 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \geq b(v) & (v \in V), \\ \text{(iii)} & x(E[U] \cup \delta(U)) \geq \lceil \tfrac{1}{2} b(U) \rceil & (U \subseteq V, b(U) \text{ odd}). \end{array}$$

Proof. Similar to the proof of Theorem 31.2, by construction of G_b and reduction to the description of the edge cover polytope (Corollary 27.3a). The theorem also follows from Theorem 34.3 below. ∎

34.3. Total dual integrality

The constraints (34.5) are totally dual integral if we delete the parity condition in (34.5)(iii):

(34.6)
$$\begin{array}{rlr} \text{(i)} & x_e \geq 0 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \geq b(v) & (v \in V), \\ \text{(iii)} & x(E[U] \cup \delta(U)) \geq \lceil \frac{1}{2} b(U) \rceil & (U \subseteq V). \end{array}$$

It is equivalent to the following:

Theorem 34.3. *Let $G = (V, E)$ be a graph, $b \in \mathbb{Z}_+^V$, and $w \in \mathbb{Z}_+^E$. Then the minimum weight $w^\mathsf{T} x$ of a b-edge cover x is equal to the maximum value of*

(34.7)
$$\sum_{v \in V} y_v b(v) + \sum_{U \subseteq V} z_U \lceil \tfrac{1}{2} b(U) \rceil,$$

where $y \in \mathbb{Z}_+^V$ and $z \in \mathbb{Z}_+^{\mathcal{P}(V)}$ satisfy

(34.8)
$$\sum_{v \in V} y_v \chi^{\delta(v)} + \sum_{U \subseteq V} z_U \chi^{E[U] \cup \delta(U)} \leq w.$$

Proof. We derive this from Theorem 32.3. Define $B := b(V) + 1$. Then the minimum is attained by a b-edge cover $x < B \cdot \mathbf{1}$. So adding $x_e \leq B$ for $e \in E$ as inequalities to (34.6) does not make it TDI if it wasn't. Let $\tilde{b}(v) := B \cdot \deg(v) - b(v)$ for each $v \in V$. Then by Theorem 32.3, the following system is TDI:

(34.9)
$$\begin{array}{lr} 0 \leq \tilde{x}_e \leq B & (e \in E), \\ \tilde{x}(\delta(v)) \leq \tilde{b}(v) & (v \in V), \\ \tilde{x}(E[U] \cup F) \leq \lfloor \frac{1}{2}(\tilde{b}(U) + B|F|) \rfloor & (U \subseteq V, F \subseteq \delta(U)). \end{array}$$

Hence also the following system is TDI (by resetting $x_e = B - \tilde{x}_e$ for each $e \in E$):

(34.10)
$$\begin{array}{lr} 0 \leq x_e \leq B & (e \in E), \\ x(\delta(v)) \geq b(v) & (v \in V), \\ x(E[U] \cup F) \geq \lceil \frac{1}{2}(b(U) - B|\delta(U) \setminus F|) \rceil & (U \subseteq V, F \subseteq \delta(U)). \end{array}$$

Now we can restrict ourselves in the last set of inequalities to those with $F = \delta(U)$, as otherwise the right-hand side is negative. So we have system (34.6) added with the superfluous inequalities $x_e \leq B$ for $e \in E$. ∎

Equivalently, in TDI terms:

Corollary 34.3a. *System* (34.6) *is totally dual integral.*

Proof. Directly from Theorem 34.3. ∎

34.4. The weighted b-edge cover problem

A minimum-weight b-edge cover can be found in strongly polynomial time, by reduction to maximum-weight b-matching:

Theorem 34.4. *For any graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, and weight function $w \in \mathbb{Q}^E$, a minimum-weight b-edge cover can be found in strongly polynomial time.*

Proof. Define $B := \|b\|_\infty$. Then we can assume that a minimum-weight b-edge cover x satisfies $x_e \leq B$ for each $e \in E$. Define $\tilde{b}(v) := B \cdot \deg(v) - b(v)$ for each $v \in V$. By Theorem 32.4, we can find a maximum-weight $\tilde{b}$-matching x in strongly polynomial time. Defining $x_e := B - \tilde{x}_e$ for each e then gives a minimum-weight b-edge cover. ∎

34.5. If b is even

The results can be simplified if b is even. In that case, the proofs can be reduced to the bipartite case.

Minimum-size $2b$-edge cover relates to maximum-weight 2-stable set, taking b as weight. Here a 2-*stable set* is a function $y \in \mathbb{Z}_+^V$ with $y_u + y_v \leq 2$ for each edge uv.

Theorem 34.5. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$. Then the minimum size of a $2b$-edge cover is equal to the maximum value of $y^\mathsf{T} b$ where y is a 2-stable set; equivalently, to the maximum value of*

(34.11) $\qquad b(V) + b(S) - b(N(S)),$

taken over stable sets S.

Proof. Similar to the proof of Theorem 31.5. (Alternatively, the present theorem can be derived with Theorem 34.1 from Theorem 31.7.) ∎

For a graph $G = (V, E)$ and $w : E \to \mathbb{Z}_+$, a w-*stable set* is a function $y : V \to \mathbb{Z}_+$ satisfying $y_u + y_v \leq w(uv)$ for each edge uv. Gallai [1957,1958a, 1958b] showed:

Theorem 34.6. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ and $w \in \mathbb{Z}_+^E$. Then the minimum weight $w^\mathsf{T} x$ of a $2b$-edge cover is equal to the maximum value of $y^\mathsf{T} b$ where y is a $2w$-stable set.*

Proof. This follows from Theorem 34.3. ∎

34.6. If b is constant

The above results can also be specialized to k-edge covers, for $k \in \mathbb{Z}_+$. That is, b is constant.

Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. A k-*edge cover* is a function $x \in \mathbb{Z}_+^E$ with $x(\delta(v)) \geq k$ for each vertex v. Thus if we identify k with the all-k

vector in $\mathbb{Z}_+^V$, we have a k-edge cover as before. Therefore, Corollary 34.1a gives the following, where I_U denotes the set of isolated (hence loopless) vertices of $G - U$:

Theorem 34.7. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. Then the minimum size of a k-edge cover is equal to the maximum value of*

$$(34.12) \qquad k|I_U| + \sum_K \lceil \tfrac{1}{2} k|K| \rceil,$$

over $U \subseteq V$, where K ranges over the components of $G - U - I_U$.

Proof. Directly from Corollary 34.1a. ∎

Note that it follows that if k is even, we need not round, and hence the minimum size of a k-edge cover is equal to $\frac{1}{2}k$ times the minimum-size of a 2-edge cover.

34.7. Capacitated b-edge covers

The capacitated b-edge cover problem considers b-edge covers x satisfying a prescribed capacity constraint $x \leq c$. Results on capacitated b-edge covers can be easily derived from the results on capacitated b-matchings.

For minimum-size capacitated b-edge cover, one has:

Theorem 34.8. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$. Then the minimum size of a b-edge cover $x \leq c$ is equal to the maximum value of*

$$(34.13) \qquad b(U) - c(E[U]) + \sum_K \lceil \tfrac{1}{2}(b(K) - c(E[K, U])) \rceil,$$

taken over all pairs T, U of disjoint subsets of V, where K ranges over the components of $G[T]$.

Proof. Define $b'(v) := c(\delta(v)) - b(v)$ for each $v \in V$. Then by Theorem 32.1,

$$\begin{aligned}
(34.14) \qquad & \text{minimum size of a } b\text{-edge cover } x \leq c \\
& = c(E) - \text{maximum size of a } b'\text{-matching } x' \leq c \\
& = c(E) - \min_{T,U,W} (b'(U) + c(E[W]) \\
& + \sum_K \lfloor \tfrac{1}{2}(b'(K) + c(E[K, W])) \rfloor) \\
& = \max_{T,U,W} c(E) - 2c(E[U]) - c(\delta(U)) + b(U) - c(E[W]) \\
& - \sum_K \lfloor \tfrac{1}{2}(2c(E[K]) + c(\delta(K)) - b(K) + c(E[K, W])) \rfloor \\
& = \max_{T,U,W} b(U) - c(E[U]) + \sum_K \lceil \tfrac{1}{2}(b(K) - c(E[K, U])) \rceil
\end{aligned}$$

(since $c(E) = c(E[U]) + c(\delta(U)) + c(E[W]) + c(E[T, W])$), where T, U, W range over partitions of V and where K ranges over the components of $G[T]$. ∎

This reduction also implies that a minimum-size b-edge cover $x \leq c$ can be found in strongly polynomial time.

Let $G = (V, E)$ be a graph, let $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$. The *c-capacitated b-edge cover polytope* is the convex hull of the b-edge covers x satisfying $x \leq c$. The description of the inequalities follows again from that for the capacitated b-matching polytope.

Theorem 34.9. *The c-capacitated b-edge cover polytope is determined by*

$$\text{(34.15)} \quad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq c(e) & (e \in E), \\ \text{(ii)} & x(\delta(v)) \geq b(v) & (v \in V), \\ \text{(iii)} & x(E[U]) + x(\delta(U) \setminus F) \geq \lceil \frac{1}{2}(b(U) - c(F)) \rceil & \\ & \qquad (U \subseteq V, F \subseteq \delta(U),\ b(U) - c(F) \text{ odd}). & \end{array}$$

Proof. From Theorem 32.2, by setting $\tilde{b}(v) := c(\delta(v)) - b(v)$ and $\tilde{x}_e := c(e) - x_e$. ∎

By deleting the parity condition in (34.15)(iii), the system becomes totally dual integral:

Theorem 34.10. *The following system is TDI:*

$$\text{(34.16)} \quad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq c(e) & (e \in E), \\ \text{(ii)} & x(\delta(v)) \geq b(v) & (v \in V), \\ \text{(iii)} & x(E[U]) + x(\delta(U) \setminus F) \geq \lceil \frac{1}{2}(b(U) - c(F)) \rceil & \\ & \qquad (U \subseteq V, F \subseteq \delta(U)). & \end{array}$$

Proof. From Theorem 32.3, with the substitutions as given in the proof of the previous theorem. ∎

The weighted capacitated b-edge cover problem can easily be reduced to the uncapacitated variant:

Theorem 34.11. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, $c \in \mathbb{Z}_+^E$, and a weight function $w \in \mathbb{Q}^E$, a minimum-weight b-edge cover $x \leq c$ can be found in strongly polynomial time.*

Proof. From Theorem 32.4, with the construction given in the proof of Theorem 34.8. ∎

Agarwal, Sharma, and Mittal [1982] showed that a minimum-weight b-edge cover $x \leq c$ can be obtained from a minimum-weight 'fractional' b-edge cover $x' \leq c$ with the help of a minimum-weight $\mathbf{1}$-edge cover algorithm.

34.8. Simple b-edge covers

Call a b-edge cover x *simple* if x is a 0,1 vector. Thus we can identify simple b-edge covers with subsets F of E such that $\deg_F(v) \geq b(v)$ for each $v \in V$.

So defining $\tilde{b}(v) := \deg_G(v) - b(v)$ for $v \in V$, a vector x is a simple b-edge cover if and only if $\mathbf{1} - x$ is a simple $\tilde{b}$-matching. This reduces simple b-edge cover problems to simple $\tilde{b}$-matching problems. With this reduction, Theorem 33.1 gives:

Theorem 34.12. *Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ with $b(v) \leq \deg(v)$ for each $v \in V$. Then the minimum size of a simple b-edge cover is equal to the maximum value of*

$$(34.17) \qquad b(U) - |E[U]| + \sum_K \lceil \tfrac{1}{2}(b(K) - |E[K, U]|) \rceil,$$

taken over all pairs T, U of disjoint subsets of V, where K ranges over the components of $G[T]$.

Proof. From Theorem 33.1 applied to $\tilde{b}$. ∎

The *simple b-edge cover polytope* is the convex hull of the simple b-edge covers in G.

Theorem 34.13. *The simple b-edge cover polytope is determined by*

$$(34.18) \qquad \begin{array}{rll} \text{(i)} & 0 \leq x_e \leq 1 & (e \in E), \\ \text{(ii)} & x(\delta(v)) \geq b(v) & (v \in V), \\ \text{(iii)} & x(E[U]) + x(\delta(U) \setminus F) \geq \lceil \tfrac{1}{2}(b(U) - |F|) \rceil & \\ & & (U \subseteq V, F \subseteq \delta(U),\ b(U) - |F| \text{ odd}). \end{array}$$

Proof. This is a special case of Theorem 34.9. ∎

Again the system is TDI:

Theorem 34.14. *System* (34.18) *is totally dual integral after deleting the parity condition in* (iii)*.*

Proof. The theorem is a special case of Theorem 34.10. ∎

Simple b-matchings are special cases of capacitated b-matchings, namely by taking the capacity function $c = \mathbf{1}$. Hence a minimum-weight simple b-edge cover can be found in strongly polynomial time:

Theorem 34.15. *Given a graph $G = (V, E)$, $b \in \mathbb{Z}_+^V$, and a weight function $w \in \mathbb{Q}^E$, a minimum-weight simple b-edge cover can be found in strongly polynomial time.*

Proof. The theorem is a special case of Theorem 34.11. ∎

We can specialize these results to k-edge covers, for $k \in \mathbb{Z}_+$. A *simple k-edge cover* is a set of edges covering each vertex at least k times. Thus it corresponds to subgraphs of minimum degree at least k. A min-max relation for minimum-size simple k-edge cover reads:

Theorem 34.16. *Let $G = (V, E)$ be a graph and let $k \in \mathbb{Z}_+$. Then the minimum size of a simple k-edge cover is equal to the maximum value of*

(34.19) $$k|U| - |E[U]| + \sum_K \lceil \tfrac{1}{2}(k|K| - |E[K, U]|) \rceil,$$

taken over all pairs T, U of disjoint subsets of V, where K ranges over the components of $G[T]$.

Proof. This is a special case of Theorem 34.12. ∎

34.8a. Simple b-edge covers and b-matchings

Let $G = (V, E)$ be a graph and let $b \in \mathbb{Z}_+^V$ with $b(v) \le \deg_G(v)$ for each $v \in V$. Define

(34.20) $$\nu^{\mathrm{s}}(b) := \text{the maximum size of a simple } b\text{-matching},$$
$$\rho^{\mathrm{s}}(b) := \text{the minimum size of a simple } b\text{-edge cover}.$$

Similar to Theorem 34.1, there is a relation between $\nu^{\mathrm{s}}(b)$ and $\rho^{\mathrm{s}}(b)$, generalizing Gallai's theorem (Theorem 19.1):

(34.21) $$\nu^{\mathrm{s}}(b) + \rho^{\mathrm{s}}(b) = b(V).$$

To see this, let M be a maximum-size simple b-matching. For each $v \in V$, add to M $b(v) - \deg_M(v)$ edges incident with v. We can do this in such a way that we obtain a simple b-edge cover F with $|F| \le |M| + \sum_{v \in V}(b(v) - \deg_M(v)) = b(V) - |M|$. So $\rho^{\mathrm{s}}(b) \le b(V) - |M| = b(V) - \nu^{\mathrm{s}}(b)$.

To see the reverse inequality, let F be a minimum-size simple b-edge cover. For each $v \in V$, delete from F $\deg_F(v) - b(v)$ edges incident with v. We obtain a simple b-matching M with $|M| \ge |F| - \sum_{v \in V}(\deg_F(v) - b(v)) = b(V) - |F|$. So $\nu^{\mathrm{s}}(b) \ge b(V) - |F| = b(V) - \rho^{\mathrm{s}}(b)$, which shows (34.21).

There is a second relation between simple b-matchings and simple b-edge covers. Define $\tilde{b}(v) := \deg_G(v) - b(v)$ for each $v \in V$. Then trivially (by complementing),

(34.22) $$\nu^{\mathrm{s}}(b) + \rho^{\mathrm{s}}(\tilde{b}) = |E|.$$

(34.21) implies

(34.23) $$b(V) - 2\nu^{\mathrm{s}}(b) = \rho^{\mathrm{s}}(b) - \nu^{\mathrm{s}}(b) = 2\rho^{\mathrm{s}}(b) - b(V),$$

and (34.22) implies

(34.24) $$\rho^{\mathrm{s}}(b) - \nu^{\mathrm{s}}(b) = \rho^{\mathrm{s}}(\tilde{b}) - \nu^{\mathrm{s}}(\tilde{b}).$$

Hence

(34.25) $$b(V) - 2\nu^{\mathrm{s}}(b) = \tilde{b}(V) - 2\nu^{\mathrm{s}}(\tilde{b}) = 2\rho^{\mathrm{s}}(b) - b(V) = 2\rho^{\mathrm{s}}(\tilde{b}) - \tilde{b}(V).$$

So the 'deficiency' of a maximum-size b-matching is equal to the 'surplus' of a minimum-size b-edge cover, and this parameter is invariant under replacing b by $\tilde{b} = \deg_G - b$.

34.8b. Capacitated b-edge covers and b-matchings

The results of the previous section hold more generally for capacitated b-matchings. Let $G = (V, E)$ be a graph, let $b \in \mathbb{Z}_+^V$ and let $c \in \mathbb{Z}_+^E$ with $b(v) \leq c(\delta(v))$ for each $v \in V$. Define

(34.26) $\nu^c(b) :=$ the maximum size of a b-matching $x \leq c$,
$\rho^c(b) :=$ the minimum size of a b-edge cover $x \leq c$.

Then:

(34.27) $\nu^c(b) + \rho^c(b) = b(V)$.

To see this, consider a maximum-size b-matching $x \leq c$. We can increase x to obtain a b-edge cover $y \leq c$, in such a way that $y(E) \leq x(E) + \sum_{v \in V}(b(v) - x(\delta(v))) = b(V) - x(E)$. So $\rho^c(b) \leq b(V) - x(E) = b(V) - \nu^c(b)$.

To see the reverse inequality, consider a minimum-size b-edge cover $y \leq c$. We can decrease y to obtain a b-matching $x \leq y$ such that $x(E) \geq y(E) - \sum_{v \in V}(y(\delta(v)) - b(v)) = b(V) - y(E)$. So $\nu^c(b) \geq b(V) - y(E) = b(V) - \rho^c(b)$, which shows (34.27).

Again, there is a second relation between capacitated b-matchings and capacitated b-edge covers. Define $\tilde{b}(v) := c(\delta(v)) - b(v)$ for each $v \in V$. Then trivially (by replacing x by $c - x$),

(34.28) $\nu^c(b) + \rho^c(\tilde{b}) = c(E)$.

Combining (34.27) and (34.28) gives as in (34.25):

(34.29) $b(V) - 2\nu^c(b) = \tilde{b}(V) - 2\nu^c(\tilde{b}) = 2\rho^c(b) - b(V) = 2\rho^c(\tilde{b}) - \tilde{b}(V)$.

So the 'deficiency' of a maximum-size b-matching $x \leq c$ is equal to the 'surplus' of a minimum-size b-edge cover $y \leq c$, and this parameter is invariant under replacing b by $\tilde{b} := c \circ \delta - b$.

Chapter 35

Upper and lower bounds

In the previous chapters we considered nonnegative integer functions satisfying certain lower *or* upper bounds. We now turn over to the more general case where we put both upper *and* lower bounds. We also relax the condition that the functions be nonnegative. Again, the results can be proved by refining the results of previous chapters — thus all results are obtained essentially by reduction to the fundamental results of Tutte and Edmonds.

35.1. Upper and lower bounds

Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}^V$ and $d, c \in \mathbb{Z}^E$. We will consider functions $x \in \mathbb{Z}^E$ satisfying

$$\text{(35.1)} \qquad \begin{array}{rll} \text{(i)} & d(e) \leq x_e \leq c(e) & \text{for all } e \in E, \\ \text{(ii)} & a(v) \leq x(\delta(v)) \leq b(v) & \text{for all } v \in V. \end{array}$$

The existence of such a function is characterized in the following theorem. (As usual, $E[X, Y]$ denotes the set of edges xy in E with $x \in X$ and $y \in Y$.)

Theorem 35.1. *Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}^V$ with $a \leq b$ and $d, c \in \mathbb{Z}^E$ with $d < c$. Then there exists an $x \in \mathbb{Z}^E$ satisfying* (35.1) *if and only if for each partition T, U, W of V, the number of components K of $G[T]$ with $b(K) = a(K)$ and*

$$\text{(35.2)} \qquad b(K) + c(E[K, W]) + d(E[K, U])$$

odd is at most

$$\text{(35.3)} \qquad b(U) - 2d(E[U]) - d(E[T, U]) - a(W) + 2c(E[W]) + c(E[T, W]).$$

Proof. To see necessity, consider a component K of $G[T]$ with $b(K) = a(K)$. Then

$$\text{(35.4)} \qquad 2x(E[K]) = b(K) - x(\delta(K)) = b(K) - x(E[K, U]) - x(E[K, W]).$$

Hence, if (35.2) is odd, we have $x(E[K, U]) \geq d(E[K, U]) + 1$ or $x(E[K, W]) \leq c(E[K, W]) - 1$. So $x(E[T, U]) - d(E[T, U]) + c(E[T, W]) - x(E[T, W])$ is at least the number of such components. On the other hand, (35.5) $$x(E[T,U]) - x(E[T,W]) = x(\delta(U)) - x(\delta(W)) \\ \leq b(U) - 2d(E[U]) - a(W) + 2c(E[W]).$$

This proves necessity.

To see sufficiency, we may assume that $d = \mathbf{0}$, since the theorem is invariant under replacing $a(v)$ by $a(v) - d(\delta(v))$ and $b(v)$ by $b(v) - d(\delta(v))$ for each v, and c by $c - d$ and d by $\mathbf{0}$. (It does not change the parity of (35.2) and does not change (35.3).)

We show sufficiency by application of Corollary 32.1a. Define

(35.6) $$R := \{v \in V \mid a(v) < b(v)\}.$$

Extend G to a graph $G' = (V', E')$, and define $b' \in \mathbb{Z}_+^{V'}$ and $c' \in \mathbb{Z}_+^{E'}$, as follows. For each $v \in V$, let $b'(v) := b(v)$ and for each $e \in E$, let $c'(e) := c(e)$. Introduce a new vertex v_0, with $b'(v_0) := b(V)$, and a loop v_0v_0 at v_0, with $c'(v_0v_0) := \infty$. Moreover, for each $v \in R$ introduce an edge vv_0 with $c'(vv_0) := b(v) - a(v)$.

Now a function x as required exists if and only if there exists a perfect b'-matching $x' \leq c'$ in G'. So it suffices to test the constraints given by Corollary 32.1a for G', b', and c'. Assuming x' does *not* exist, we can partition V' into T', U', and W' such that $G'[T']$ has more than $b'(U') - b'(W') + 2c'(E'[W']) + c'(E'[T', W'])$ components K' with $b'(K') + c'(E'[K', W'])$ odd. By parity, the excess is at least 2. (This follows from the fact that $b'(V') = 2b(V)$ is even.)

Let $T := T' \setminus \{v_0\}$, $U := U' \setminus \{v_0\}$, and $W := W' \setminus \{v_0\}$.

First assume that $v_0 \in U'$; so $T' = T$ and $W' = W$. Then the number of components K of $G'[T'] = G[T]$ with $b(K) + c(E[K, W])$ odd is trivially at most $b(T) + c(E[T, W])$, and hence at most

(35.7) $$b(U) + b(V) - b(W) + 2c(E[W]) + c(E[T, W]) \\ = b'(U') - b'(W') + 2c'(E'[W']) + c'(E'[T', W']),$$

a contradiction.

Second assume that $v_0 \in W'$. Then $c'(E'W') = \infty$, which is again a contradiction.

Hence we may assume that $v_0 \in T'$; so $U' = U$ and $W' = W$. Then $G'[T']$ has exactly one component containing v_0. All other components K are components of $G[T]$ that are disjoint from R (since no vertex in K is adjacent to v_0). So $G[T]$ has more than

(35.8) $$b'(U') - b'(W') + 2c'(E'[W']) + c'(E'[T', W']) \\ = b(U) - a(W) + 2c(E[W]) + c(E[T, W])$$

components K contained in $V \setminus R$ with $b(K) + c(E[K, W])$ odd. This contradicts the condition of the theorem. ∎

By taking $d = \mathbf{0}$ and $c = \boldsymbol{\infty}$ we obtain as special case (where again I_U denotes the set of isolated (hence loopless) vertices of $G - U$):

Corollary 35.1a. *Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}_+^V$ with $a \leq b$. Then there exists a function $x \in \mathbb{Z}_+^E$ satisfying*

(35.9) $$a(v) \leq x(\delta(v)) \leq b(v)$$

for each $v \in V$ if and only if for each $U \subseteq V$, $G - U - I_U$ has at most $b(U) - a(I_U)$ components K with $b(K)$ odd and $a(K) = b(K)$.

Proof. We show sufficiency. Suppose that no such x exists. By Theorem 35.1 (for $d = \mathbf{0}$, $c = \boldsymbol{\infty}$), there exists a partition T, U, W of V with $E[W] = \emptyset$ and $E[T, W] = \emptyset$ such that the number of components K of $G[T]$ with $b(K) = a(K)$ and $b(K)$ odd, is more than $b(U) - a(W)$. We may assume that each component K of $G[T]$ spans at least one edge: otherwise, if $K = \{v\}$, moving v from T to W, decreases the number of such components by at most 1, while $b(U) - a(W)$ decreases by at least 1 (since $b(v) = a(v)$ and $b(v)$ is odd).

So we can assume that $W = I_U$, in which case we have a contradiction with the condition in the present corollary. ∎

Another special case, for $d = \mathbf{0}$, and $c = \mathbf{1}$, is the characterization of Lovász [1970c] of the existence of subgraphs with prescribed degrees:

Corollary 35.1b. *Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}_+^V$ with $a \leq b$. Then E has a subset F such that*

(35.10) $$a(v) \leq \deg_F(v) \leq b(v)$$

for each $v \in V$ if and only if for each partition T, U, W of V, the number of components K of $G[T]$ with $b(K) = a(K)$ and $b(K) + |E[K, W]|$ odd is at most $b(U) - a(W) + 2|E[W]| + |E[T, W]|$.

Proof. This is the case $d = \mathbf{0}$, $c = \mathbf{1}$ of Theorem 35.1. ∎

The construction described in the proof of Theorem 35.1 also implies:

Theorem 35.2. *Given a graph $G = (V, E)$, $a, b \in \mathbb{Z}^V$, $d, c \in \mathbb{Z}^E$, and $w \in \mathbb{Q}^E$, a vector $x \in \mathbb{Z}^E$ satisfying $d \leq x \leq c$ and $a(v) \leq x(\delta(v)) \leq b(v)$ for each $v \in V$, and minimizing $w^{\mathsf{T}} x$, can be found in strongly polynomial time.*

Proof. The construction in the proof of Theorem 35.1 reduces this to Theorem 32.4. ∎

35.2. Convex hull

We now characterize the convex hull of the functions $x \in \mathbb{Z}^E$ satisfying (35.1):

Theorem 35.3. *Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}^V$ and $d, c \in \mathbb{Z}^E$ with $a \leq b$ and $d \leq c$. Then the convex hull of the vectors $x \in \mathbb{Z}^E$ satisfying* (35.1) *is determined by* (35.1) *together with the inequalities*

$$x(E[U]) - x(E[W]) + x(F \cap \delta(U)) - x(H \cap \delta(W)) \le \lfloor \tfrac{1}{2}(b(U) - a(W) + c(F) - d(H)) \rfloor, \tag{35.11}$$

where U and W are two disjoint subsets of V and where F and H partition $\delta(U \cup W)$, with $b(U) - a(W) + c(F) - d(H)$ odd.

Proof. Necessity of (35.11) follows by adding up the following inequalities, each implied by (35.1):

$$\begin{aligned} &x(E[U]) + \tfrac{1}{2}x(\delta(U)) \le \tfrac{1}{2}b(U),\\ &-x(E[W]) - \tfrac{1}{2}x(\delta(W)) \le -\tfrac{1}{2}a(W),\\ &\tfrac{1}{2}x(F) \le \tfrac{1}{2}c(F),\\ &-\tfrac{1}{2}x(H) \le -\tfrac{1}{2}d(H). \end{aligned} \tag{35.12}$$

The left-hand sides add up to the left-hand side of (35.11), and the right-hand side to the unrounded right-hand side of (35.11).

To see sufficiency of (35.11), we may assume that $d = \mathbf{0}$. Indeed, the theorem is invariant under resetting $a(v) := a(v) - d(\delta(v))$, and $b(v) := b(v) - d(\delta(v))$ for all $v \in V$, and $c := c - d$ and $d := \mathbf{0}$. Then, as above, we can reduce the theorem to Corollary 32.2a characterizing the convex hull of capacitated b-matchings.

Let x satisfy (35.1) and (35.11). Let R, G', b', and c' be as in the proof of Theorem 35.1. Define $x'(e) := x_e$ for each $e \in E$, $x'(vv_0) := b(v) - x(\delta(v))$ for each $v \in R$, and $x'(v_0v_0) := 2x(E)$.

We show that x' belongs to the c'-capacitated perfect b'-matching polytope (with respect to G'). This implies that x belongs to the convex hull of vectors $x \in \mathbb{Z}_+^E$ satisfying (35.1).

By Corollary 32.2a, it suffices to check

$$\begin{aligned} &\text{(i)} && 0 \le x'(e') \le c'(e') \quad (e' \in E'),\\ &\text{(ii)} && x'(\delta'(v')) = b'(v') \quad (v' \in V'),\\ &\text{(iii)} && x'(\delta'(U') \setminus F') - x'(F') \ge 1 - c'(F')\\ &&& \qquad (U' \subseteq V',\ F' \subseteq \delta'(U') \text{ with } b'(U') + c'(F') \text{ odd}). \end{aligned} \tag{35.13}$$

(35.13)(i) and (ii) are direct. To see (35.13)(iii), let $U' \subseteq V'$ and $F' \subseteq \delta'(U')$ with $b'(U')+c'(F')$ odd. We may assume that $v_0 \in U'$ (as we can replace U' by its complement, since $b'(V') = 2b(V)$ is even). Let $W := \{v \in V \mid vv_0 \in F\}$ and $U := V \setminus (U' \cup W)$. Let $F := F' \cap E$ and $H := \delta_E(U \cup W) \setminus F$.

Now $b'(U') = b(V) + b(V \setminus (U \cup W))$ and $c'(F') = c(F) + (b - a)(W)$. So $b'(U') + c'(F')$ odd implies that $b(U) - a(W) + c(F)$ is odd. So by (35.11),

$$2x(E[U]) - 2x(E[W]) + 2x(F \cap \delta(U)) - 2x(H \cap \delta(W)) \le b(U) - a(W) + c(F) - 1. \tag{35.14}$$

Hence

$$\begin{aligned} &x'(\delta'(U') \setminus F') - x'(F')\\ &= x(H) + \sum_{v \in U}(b(v) - x(\delta(v))) - x(F) - \sum_{v \in W}(b(v) - x(\delta(v))) \end{aligned} \tag{35.15}$$

$$= x(H) + b(U) - 2x(E[U]) - x(\delta(U)) - x(F) - b(W)$$
$$+ 2x(E[W]) + x(\delta(W)) = -2x(E[U]) + 2x(E[W]) - 2x(F \cap \delta(U))$$
$$+ 2x(H \cap \delta(W)) + (b(U) - b(W))$$
$$\geq 1 - b(U) + a(W) - c(F) + (b(U) - b(W))$$
$$= 1 - c(F) - b(W) + a(W) = 1 - c'(F'),$$

proving (35.13)(iii). ∎

The special case $d = \mathbf{0}$, $c = \boldsymbol{\infty}$ was mentioned by Schrijver and Seymour [1977]:

Corollary 35.3a. *Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}_+^V$. Then the convex hull of those $x \in \mathbb{Z}^E$ satisfying*

(35.16)
(i) $x_e \geq 0$ for each $e \in E$,
(ii) $a(v) \leq x(\delta(v)) \leq b(v)$ for each $v \in V$,

is determined by (35.16) *together with the inequalities:*

(35.17) $$x(E[U]) - x(E[W]) - x(\delta(W) \setminus \delta(U)) \leq \lfloor \tfrac{1}{2}(b(U) - a(W)) \rfloor,$$

where U and W are disjoint subsets of V with $b(U) - a(W)$ odd.

Proof. This is a special case of Theorem 35.3. ∎

Similarly, we can characterize the convex hull of subgraphs with prescribed bounds on the degrees:

Corollary 35.3b. *Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}_+^V$ with $a \leq b$. Then the convex hull of the incidence vectors of subsets F of E satisfying*

(35.18) $$a(v) \leq \deg_F(v) \leq b(v)$$

for each $v \in V$, is determined by

(35.19)
(i) $0 \leq x_e \leq 1$ for each $e \in E$,
(ii) $a(v) \leq x(\delta(v)) \leq b(v)$ for each $v \in V$,

together with the inequalities

(35.20) $$x(E[U]) - x(E[W]) + x(F \cap \delta(U)) - x(H \cap \delta(W)) \leq \lfloor \tfrac{1}{2}(b(U) - a(W) + |F|) \rfloor,$$

where U and W are disjoint subsets of V and where F and H partition $\delta(U \cup W)$, with $b(U) - a(W) + |F|$ odd.

Proof. Again this is a special case of Theorem 35.3. ∎

We note that for the $V \times E$ incidence matrix M of any graph $G = (V, E)$, any $a, b \in \mathbb{Z}^V$, $d, c \in \mathbb{Z}^E$, and any $k, l \in \mathbb{Z}$ one has:

(35.21)
$$\begin{aligned}&\text{conv.hull}\{x \in \mathbb{Z}^E \mid d \leq x \leq c, a \leq Mx \leq b, k \leq x(E) \leq l\}\\ &= \text{conv.hull}\{x \in \mathbb{Z}^E \mid d \leq x \leq c, a \leq Mx \leq b\}\\ &\cap \{x \in \mathbb{R}^E \mid k \leq x(E) \leq l\}.\end{aligned}$$

This can be proved similarly to Corollary 18.10a.

35.3. Total dual integrality

System (35.1) together with the inequalities (35.11) generally is not TDI (cf. the example in Section 30.5). To obtain a totally dual integral system we should delete the restriction in (35.11) that $b(U) - a(W) + c(F) - d(H)$ be odd. Thus we obtain the system:

(35.22)
$$\begin{aligned}&\text{(i)} && d(e) \leq x_e \leq c(e) && (e \in E),\\ &\text{(ii)} && a(v) \leq x(\delta(v)) \leq b(v) && (v \in V),\\ &\text{(iii)} && x(E[U]) - x(E[W]) + x(F \cap \delta(U)) - x(H \cap \delta(W))\\ &&& \quad \leq \lfloor \tfrac{1}{2}(b(U) - a(W) + c(F) - d(H)) \rfloor\\ &&&&& (U, W \subseteq V, U \cap W = \emptyset,\ F, H\\ &&&&& \text{partition } \delta(U \cup W)).\end{aligned}$$

Theorem 35.4. *System* (35.22) *is totally dual integral.*

Proof. Again we may assume $d = \mathbf{0}$. Let R, G', b', and c' be as in the proof of Theorem 35.1. By Theorem 32.3, the following system, in the variable $x' \in \mathbb{R}^{E'}$, is TDI (where $\delta' := \delta_{G'}$ and $E'[U']$ is the set of edges in E' spanned by U'):

(35.23)
$$\begin{aligned}&\text{(i)} && 0 \leq x'(e') \leq c'(e') && (e' \in E'),\\ &\text{(ii)} && x'(\delta'(v')) = b'(v') && (v' \in V'),\\ &\text{(iii)} && x'(E'[U']) + x'(F') \leq \lfloor \tfrac{1}{2}(b'(U') + c'(F')) \rfloor\\ &&&&& (U' \subseteq V', F' \subseteq \delta'(U')).\end{aligned}$$

Adding the equality

(35.24)
$$x'(v_0v_0) - x'(E) = 0$$

to (35.23) maintains total dual integrality (since (35.24) is valid for each vector x' satisfying (35.23)).

We can restrict the inequalities (35.23)(iii) to those with $v_0 \notin U'$. To see this, assume $v_0 \in U'$. Define $U := U' \cap V$ and $U'' := V \setminus U'$. Then

(35.25)
$$\begin{aligned}x'(E'[U']) &= x'(v_0v_0) + x'(E'[U]) + \sum_{v \in U \cap R} x'(vv_0)\\ &= x'(E) + x'(E'[U]) + \sum_{v \in U \cap R} x'(vv_0) = x'(E'[U'']) + \sum_{v \in U} x'(\delta'(v))\end{aligned}$$

and

(35.26) $$\lfloor \tfrac{1}{2}(b'(U') + c'(F')) \rfloor = \lfloor \tfrac{1}{2}(b'(U'') + 2b'(U) + c'(F')) \rfloor = \lfloor \tfrac{1}{2}(b'(U'') + c'(F')) \rfloor + \sum_{v \in U} b'(v),$$

since $b'(U') = b'(U) + b'(v_0) = b'(U) + b'(V)$. So the inequality

(35.27) $$x'(E'[U']) + x'(F') \leq \lfloor \tfrac{1}{2}(b'(U') + c'(F')) \rfloor$$

is a sum of

(35.28) $$x'(E'[U'']) + x'(F') \leq \lfloor \tfrac{1}{2}(b'(U'') + c'(F')) \rfloor$$

and of $x'(\delta'(v)) = b'(v)$ for $v \in U$. So we can assume that $v_0 \notin U'$.

Now adding an integer multiple of a valid equality to another constraint, maintains total dual integrality. So using (35.23)(ii) we can replace (35.23)(i) by:

(35.29) $$\begin{array}{ll} 0 \leq x'(e) \leq c(e) & (e \in E), \\ a(v) \leq x'(\delta(v)) \leq b(v) & (v \in V), \end{array}$$

since for $v \in R$, subtracting $x'(\delta'(v)) = b(v)$ from $0 \leq x'(vv_0) \leq b(v) - a(v)$ gives $-b(v) \leq -x'(\delta(v)) \leq -a(v)$.

For $U' \subseteq V$ and $F' \subseteq \delta'(U')$, let $W := \{v \in V \mid vv_0 \in F'\}$ and $F := F' \cap E$. As

(35.30) $$\begin{aligned} & x'(E'[U']) + x'(F') - \sum_{v \in W} x'(\delta'(v)) \\ & = x'(E'[U']) + x'(F') - 2x'(E[W]) - x'(\delta(W)) \\ & = x'(E'[U' \setminus W]) + x'(F') - x'(E[W]) - x'(\delta(W) \setminus \delta(U' \setminus W)) \end{aligned}$$

and

(35.31) $$\lfloor \tfrac{1}{2}(b'(U') + c'(F')) \rfloor - b(W) = \lfloor \tfrac{1}{2}(b(U' \setminus W) - a(W) + c(F)) \rfloor,$$

we can replace (35.23)(iii) by (taking $U := U' \setminus W$):

(35.32) $$x'(E'[U]) + x'(F) - x'(E[W]) - x'(\delta(W) \setminus \delta(U)) \leq \tfrac{1}{2}\lfloor b(U) - a(W) + c(F) \rfloor \text{ for } U, W \subseteq V \text{ with } U \cap W = \emptyset \text{ and for } F \subseteq \delta(U \cup W).$$

Each of the variables $x'(vv_0)$ $(v \in R)$ and $x'(v_0v_0)$ occurs exactly once in the system, in an equality constraint, with coefficient 1. So we can delete these variables maintaining total dual integrality (Theorem 5.27), and we obtain system (35.22). ∎

As special cases one can derive the total dual integrality of the systems corresponding to $d = \mathbf{0}$, $c = \boldsymbol{\infty}$ and to $d = \mathbf{0}$, $c = \mathbf{1}$ (the subgraph polytope).

35.4. Further results and notes

35.4a. Further results on subgraphs with prescribed degrees

Corollary 35.1b of Lovász [1970c] implies the following. Let $G = (V, E)$ be a graph and let $b, b' \in \mathbb{Z}_+^V$ with $b + b' > \deg_G$. Then E can be partitioned into a simple b-matching and a simple b'-matching if and only if

(35.33) $\qquad |E[U, W]| \leq b(U) + b'(W)$

for each pair of disjoint subsets U and W of V.

This corresponds to the case $a < b$ in Corollary 35.1b, by taking $a := \deg_G - b'$. Then there are no components K with $a(K) = b(K)$.

The condition can be equivalently described as:

(35.34) $$\sum_{v \notin U} \max\{0, a(v) - \deg_{G-U}(v)\} \leq b(u)$$

for $U \subseteq V$.

This implies the following result of Lovász [1970c]:

(35.35) Let $G = (V, E)$ be a graph of maximum degree k and let $k_1, k_2 \geq 0$ with $k_1 + k_2 = k + 1$. Then E can be partitioned into a simple k_1-matching and a simple k_2-matching

(since $|E[U, W]| \leq (k_1 + k_2) \min\{|U|, |W|\} \leq k_1|U| + k_2|W|$). A special case is a result noted by Tutte [1978]: for all $0 \leq r \leq k$, each k-regular graph has a subgraph in which each degree belongs to $\{r, r + 1\}$.

Thomassen [1981a] gave the following short direct proof of (35.35). In fact he proved the following extension of (35.35):

(35.36) Let $G = (V, E)$ be a graph in which each vertex has degree k or $k + 1$ and let $1 \leq k' < k$. Then G has a subgraph $G' = (V, E')$ in which each vertex has degree k' or $k' + 1$.

Note that (35.35) follows from this by embedding G into a k-regular graph.

To prove (35.36), it suffices to prove the case $k' = k - 1$. Let U be the set of vertices of degree $k + 1$ in G. We can assume that deleting any edge of G results in a vertex of degree less than k. Hence no two distinct vertices in U are adjacent. There may be loops at the vertices in U; let W be the set of those vertices in U that are not incident with a loop. Since each vertex in W has degree $k+1$ and each vertex in $V \setminus U$ has degree k, by Hall's marriage theorem, G contains a matching M connecting W to $V \setminus U$. Now deleting the edges in M and deleting, for each vertex $v \in U \setminus W$, one of the loops attached at v, gives a graph G' as required.

A 'dual' consequence was noted by Gupta [1978]:

(35.37) Let $G = (V, E)$ be a graph of minimum degree δ and let $\delta_1, \delta_2 \geq 0$ with $\delta_1 + \delta_2 = \delta - 1$. Then E can be partitioned into E_1 and E_2 such that $G_i = (V, E_i)$ has minimum degree at least δ_i for $i = 1, 2$.

Gupta [1978] mentioned that the following direct derivation from Theorem 20.6 was shown to him by C. Berge:

Apply induction on δ_1, the case $\delta_1 = 0$ being trivial. If $\delta_1 > 0$, by the induction hypothesis E can be partitioned into E_1 and E_2 such that $\delta(G_1) \geq \delta_1 - 1$ and

$\delta(G_2) \geq \delta_2 + 1 = \delta - \delta_1$. (Here $G_i = (V, E_i)$ for $i = 1, 2$.) We choose this partition with $|E_2|$ minimal.

Let S be the set of vertices v with $\deg_{E_1}(v) = \delta_1 - 1$. By the minimality of $|E_2|$, S spans no edge of E_2. Let $F := \delta(S) \cap E_2$. So $\deg_F(v) = \deg_{E_2}(v) = \deg_G(v) - \deg_{E_1}(v) \geq \delta - \delta_1 + 1$ for each $v \in S$. Let $p := \delta - \delta_1 + 1 = \delta_2 + 2$. Now by Theorem 20.6, F can be partitioned into $F_1, \ldots, F_p$ such that each vertex v is covered by at least $\min\{p, \deg_F(v)\}$ of the F_i. Then replacing E_1 by $E_1 \cup F_1$ and E_2 by $E_2 \setminus F_1$ gives a partition as required. Indeed, if $\deg_F(v) \geq p$, then $\deg_{F_i}(v) \geq 1$ for each i, implying

(35.38) $$\deg_{E_1 \cup F_1}(v) = \deg_{E_1}(v) + \deg_{F_1}(v) \geq (\delta_1 - 1) + 1 = \delta_1$$

and

(35.39) $$\deg_{E_2 \setminus F_1}(v) \geq \sum_{i=2}^{p} \deg_{F_i}(v) \geq p - 1 = \delta_2 + 1.$$

If $\deg_F(v) < p$, then $v \notin S$ and $\deg_{F_1}(v) \leq 1$, and hence

(35.40) $$\deg_{E_1 \cup F_1}(v) \geq \deg_{E_1}(v) \geq \delta_1$$

and

(35.41) $$\deg_{E_2 \setminus F_1}(v) \geq \deg_{E_2}(v) - 1 \geq (\delta_2 + 1) - 1 = \delta_2.$$

This proves (35.37).

Las Vergnas [1978] showed that if $a \leq \mathbf{1} \leq b$ holds, a simpler condition can be formulated in Corollary 35.1b:

(35.42) for each $U \subseteq V$, the number of odd components K of $G - U$ with $|K| = 1$ and $a(K) = 1$, or with $|K| \geq 3$ and $a(K) = b(K)$ is at most $b(U)$.

Anstee [1985] gave a proof of Lovász's theorem, with an $O(n^3)$-time algorithm to find the subgraph. Heinrich, Hell, Kirkpatrick, and Liu [1990] gave a simplified proof of Lovász's theorem for $a < b$, implying an $O(\sqrt{a(V)}\, m)$-time algorithm.

Lovász [1970c] also characterized the minimum deviation that subsets can have from prescribed lower and upper bounds on the degrees. In fact, he showed the following (where $\alpha_+ := \max\{0, \alpha\}$ for any $\alpha \in \mathbb{R}$): Let $G = (V, E)$ be a graph and let $a, b \in \mathbb{Z}_+^V$ with $a \leq b$. Then the minimum of

(35.43) $$\sum_{v \in V} ((a(v) - \deg_F(v))_+ + (\deg_F(v) - b(v))_+)$$

over $F \subseteq E$ is equal to the maximum value of

(35.44) $a(W) - b(U) - 2|E[W]| - |E[T, W]|+$number of components K of $G[T]$ with $a(K) = b(K)$ and with $a(K) + |E[K, W]|$ odd,

taken over all partitions T, U, W of V.

Let $B : V \to \mathcal{P}(\mathbb{Z}_+)$. The B-matching problem asks for a subgraph H of G such that $\deg_H(v) \in B(v)$ for each $v \in V$. In general, this is NP-complete, even when $B(v) \in \{\{1\}, \{0, 3\}\}$ for each $v \in V$ (Lovász [1972f]).

If $|\mathbb{Z}_+ \setminus B(v)| = 1$ for each vertex v, Lovász [1973a] gave a characterization. Lovász [1972f] investigated the case where $\mathbb{Z}_+ \setminus B(v)$ contains no two consecutive integers, for which Cornuéjols [1988] gave a polynomial-time algorithm, and Sebő [1993b] a good characterization.

For algorithms to find subgraphs of minimum deviation see Hell and Kirkpatrick [1993]. Other work on subgraphs with prescribed degrees includes Berge and Las Vergnas [1978], Shiloach [1981], Kano and Saito [1983], Akiyama and Kano [1985a], Kano [1985,1986], Anstee [1990], Cai [1991], and Li and Cai [1998]. A survey is given by Akiyama and Kano [1985b].

35.4b. Odd walks

Let $G = (V, E)$ be an undirected graph, let $s, t \in V$, and let $l : E \to \mathbb{Q}$. Call a walk *odd* if it has an odd number of edges. Then a shortest odd $s-t$ walk without repeated edges can be found as follows. For each edge e of G, set $d(e) := 0$ and $c(e) := 1$, and add an edge $\tilde{e}$ parallel to e, of length $l(\tilde{e}) := -l(e)$, and define $d(\tilde{e}) := -1$, $c(\tilde{e}) := 0$. Let M be the $V \times E'$ incidence matrix of the extended graph $G' = (V, E')$. Define $b : V \to \mathbb{Z}$ by $b(s) := b(t) := 1$ and $b(v) := 0$ for each $v \in V \setminus \{s, t\}$. Then a shortest odd $s-t$ walk without repeated edges can be found by finding an $x \in \mathbb{Z}^{E'}$ satisfying $d \leq x \leq c$ and $Mx = b$ and minimizing $l^{\mathsf{T}}x$.

So by Theorem 35.2, this can be solved in strongly polynomial time. Better running times were given by Goldberg and Karzanov [1994,1996]: $O(m)$ for finding such an odd $s-t$ walk, $O(nm \log n)$ and $O(nm\sqrt{\log L})$ for finding a shortest such odd $s-t$ walk, strengthened to $O(m \log n)$ and $O(m\sqrt{\log L})$ for nonnegative lengths. (L is the maximum absolute value of the lengths, assuming they are integer.)

Chapter 36

Bidirected graphs

In the previous chapter we considered integer solutions of $d \le x \le c$, $a \le Mx \le b$ where M is the incidence matrix of an undirected graph. Earlier, in Chapter 12, we considered the same problem if M is the incidence matrix of a directed graph.
Edmonds and Johnson [1970] showed that M can more generally be the incidence matrix of a 'bidirected' graph — a structure that comprises both undirected and directed graphs. That is, M has entries 0, ± 1, and ± 2, such that the sum of the absolute values of the entries in any column is equal to 2. The results are obtained by a simple reduction to the undirected case, although the elaboration takes some effort.
The results could be formulated just in terms of matrices, but the graph-theoretic interpretation is helpful in formulating, visualizing, and proving the results.

36.1. Bidirected graphs

A *bidirected graph* is a triple $G = (V, E, \sigma)$, where (V, E) is an undirected graph and where σ assigns to each $e \in E$ and $v \in e$ a 'sign' $\sigma_{e,v} \in \{+1, -1\}$.

If e is a loop, that is, e is a family $\{v, v\}$, we may assign different signs to the two occurrences of v. However, in the problems discussed in this chapter, loops where the signs are different are irrelevant. So we assume that the signs in a loop are the same, either both $+1$, or both -1.

Clearly, undirected graphs and directed graphs can be considered as special cases of bidirected graphs. Graph terminology for the graph (V, E) extends in an obvious way to the bidirected graph (V, E, σ).

Let $G = (V, E, \sigma)$ be a bidirected graph. The edges e for which $\sigma_{e,v} = 1$ for each $v \in e$ are called the *positive edges*, those with $\sigma_{e,v} = -1$ for each $v \in e$ the *negative edges*, and the remaining edges are called the *directed edges*. The $V \times E$ *incidence matrix* of G is the $V \times E$ matrix M defined by:

(36.1) $M_{v,e} := \sigma_{e,v}$ if e is not a loop,
$M_{v,e} := 2\sigma_{e,v}$ if e is a loop,

setting $\sigma_{e,v} := 0$ if $v \notin e$. It follows that an integer matrix M is the $V \times E$ incidence matrix of a bidirected graph if and only if the sum of the absolute values of the entries in any column of M is equal to 2.

For vectors $a, b \in \mathbb{Z}^V$ and $d, c \in \mathbb{Z}^E$, we consider integer solutions $x \in \mathbb{R}^E$ of

(36.2) (i) $d \leq x \leq c$,
(ii) $a \leq Mx \leq b$.

The related existence and optimization problems can be reduced as follows to the case where G is just an undirected graph. First, we can assume that G has no negative edges, since multiplying the corresponding column of M by -1 gives an equivalent problem. Next, any directed edge $f = su$, with $\sigma_{f,s} = -1$ and $\sigma_{f,u} = +1$, can be handled as follows.

(36.3) Extend G by a new vertex t and replace edge e by two new positive edges st and tu. This makes the bidirected graph $G' = (V', E')$. Define $a', b' \in \mathbb{Z}^{V'}$ by $a'(v) := a(v)$ and $b'(v) := b(v)$ for $v \in V$ and $a'(t) := b'(t) := 0$. Define $d', c' \in \mathbb{Z}^{E'}$ by $d'(e) := d(e)$ and $c'(e) := c(e)$ for $e \in E \setminus \{f\}$, and $d'(st) := -\infty$, $c'(st) := \infty$, $d'(tu) := d(f)$, and $c'(tu) := c(f)$. Let M' be the $V' \times E'$ incidence matrix of G'.

Then there is a one-to-one relation between (integer) solutions of (36.2) and those for the system corresponding to G', M', a', b', c', d': just define $x(tu) := x_f$ and $x(st) := -x_f$.

Algorithmically, this gives a direct reduction to the undirected case:

Theorem 36.1. *For $w \in \mathbb{Q}^E$, an integer vector x maximizing $w^{\mathsf{T}}x$ over* (36.2) *can be found in strongly polynomial time.*

Proof. By multiplying columns of M by -1, we can assume that G has no negative edges. Next, apply (36.3) to each directed edge. This reduces the problem to one on a bidirected graph with all edges positive, that is, on an undirected graph. Hence, the theorem follows from Theorem 35.2. ∎

We next consider characterizations. Let $G = (V, E, \sigma)$ be a bidirected graph. For any $T \subseteq V$, $G[T]$ denotes the bidirected subgraph of G induced by T (that is, $G[T] = (T, E[T], \sigma')$, where σ' is the restriction of σ to pairs e, v with $e \in E[T]$). We set for $U \subseteq V$:

(36.4) $\delta(U) := \delta_E(U)$.

For disjoint $X, Y \subseteq V$, we denote:

(36.5) $E[X, Y^+] := \{e \in \delta(X) \mid \exists v \in Y : \sigma_{e,v} = +1\}$,
$E[X, Y^-] := \{e \in \delta(X) \mid \exists v \in Y : \sigma_{e,v} = -1\}$.

For any vector z, let z_+ be the vector obtained from z by setting each negative entry to 0. Similarly, let z_- be the vector obtained from z by setting each positive entry to 0. So $z = z_+ + z_-$.

In the following theorem the condition that $d < c$ is not really a restriction: if $d_e = c_e$ we know that $x_e = d_e$ and hence we can dispose of e by contracting it appropriately. But if we delete the condition $d < c$, the formulation of the theorem would be more complicated.

Theorem 36.2. *Let $a \le b$ and $d < c$. Then there exists an integer vector x satisfying* (36.2) *if and only if for each partition T, U, W of V, the number of components K of $G[T]$ with $b(K) = a(K)$ and*

$$b(K)+c(E[K,W^+])+c(E[K,U^-])+d(E[K,U^+])+d(E[K,W^-]) \tag{36.6}$$

odd is at most

$$y_+^\mathsf{T} b + y_-^\mathsf{T} a - (y^\mathsf{T} M)_- c - (y^\mathsf{T} M)_+ d, \tag{36.7}$$

where $y := \chi^U - \chi^W$.

Proof. The validity of the theorem is invariant under multiplying a row v of M by -1 and replacing $b(v)$ by $-a(v)$ and $a(v)$ by $-b(v)$ (then if $v \in U$ we move v to W, and if $v \in W$ we move v to U). Similarly, the validity is invariant under multiplying a column e by -1 and replacing $c(e)$ by $-d(e)$ and $d(e)$ by $-c(e)$.

Hence, to see necessity, we can assume that $W = \emptyset$ and $E[T, U^-] = \emptyset$. So $y_+ = y$ and $y_- = \mathbf{0}$ and $E[T, U^+] = \delta(T)$. Then

$$\begin{aligned}(x-d)(\delta(T)) &= (x-d)(E[T,U^+]) \le (y^\mathsf{T} M)_+(x-d) \\ &\le (y^\mathsf{T} M)_+(x-d) - (y^\mathsf{T} M)_-(c-x) \\ &= y^\mathsf{T} Mx - (y^\mathsf{T} M)_+ d - (y^\mathsf{T} M)_- c \\ &= y_+^\mathsf{T} Mx + y_-^\mathsf{T} Mx - (y^\mathsf{T} M)_+ d - (y^\mathsf{T} M)_- c \\ &\le y_+^\mathsf{T} b + y_-^\mathsf{T} a - (y^\mathsf{T} M)_+ d - (y^\mathsf{T} M)_- c.\end{aligned} \tag{36.8}$$

On the other hand, for each component K of $G[T]$ one has $(x-d)(\delta(K)) \ge 0$. Moreover, if $b(K) = a(K)$ and (36.6) is odd, then $(x-d)(\delta(K))$ is odd, since

$$(x-d)(\delta(K)) \equiv (x-d)(\delta(K)) + 2x(E[K]) \equiv b(K) + d(\delta(K)) \pmod 2 \tag{36.9}$$

So $(x-d)(\delta(T))$ is not less than the number of components K of $G[T]$ with $a(K) = b(K)$ and (36.6) odd, showing necessity of the condition.

To show sufficiency, we can assume that G has no negative edges, since we can multiply columns of M by -1. We show sufficiency by induction on the number of directed edges. If this number is 0, the theorem reduces to Theorem 35.1. So we can assume that there is an edge $f = su$ with $\sigma_{f,s} = -1$ and $\sigma_{f,u} = +1$. Then we apply construction (36.3), to obtain $G' = (V', E'), M', a', b', d', c'$.

Now there exists an integer vector x satisfying $d \le x \le c$ and $a \le Mx \le b$ if there exists an integer vector x' satisfying $d' \le x' \le c'$ and $a' \le M'x' \le b'$. So we can assume that no such x' exists. By induction (as G' has fewer directed edges than G), we know that V' can be partitioned into T', U', and W' such that the number of components K' of $G'[T']$ with $b'(K') = a'(K')$ and

$$(36.10)\qquad b'(K') + c'(E'[K', W'^{+}]) + c'(E'[K', U'^{-}]) + d'(E'[K', U'^{+}]) + d'(E'[K', U'^{-}])$$

odd, is more than

$$(36.11)\qquad {y'}^{\mathsf{T}}_{+} b' + {y'}^{\mathsf{T}}_{-} a' - ({y'}^{\mathsf{T}} M')_{-} c' - ({y'}^{\mathsf{T}} M')_{+} d',$$

where $y' := \chi^{U'} - \chi^{W'}$.

Since $c'(st) = \infty$ we know that $({y'}^{\mathsf{T}} M')_{st} \geq 0$, that is, $y'_s + y'_t \geq 0$. Similarly, since $d'(st) = -\infty$ we know that $({y'}^{\mathsf{T}} M')_{st} \leq 0$, that is, $y'_s + y'_t \leq 0$. So $y'_s = -y'_t$, and hence either $s \in U'$, $t \in W'$, or $s \in W'$, $t \in U'$, or $s, t \in T'$.

Let $U := U' \cap V$, $W := W' \cap V$, and $T := T' \cap V$. Then for any component K' of $G'[T']$ with $b'(K') = a'(K')$ and (36.10) odd, $K := K' \cap V$ is a component of $G[T]$ with $b(K) = a(K)$ and (36.6) odd. Moreover, (36.11) is equal to (36.7). Hence we have a contradiction with the condition given in the theorem. ∎

36.2. Convex hull

Also the convex hull of the integer solutions of (36.2) can be characterized (where we do not assume $d < c$):[20]

Theorem 36.3. *The convex hull of the integer solutions of* (36.2) *is determined by*

$$(36.12)\qquad \begin{array}{rl} \text{(i)} & d \leq x \leq c, \\ \text{(ii)} & a \leq Mx \leq b, \\ \text{(iii)} & \frac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x \\ & \quad \leq \lfloor \frac{1}{2}(b(U) - a(W) + c(F) - d(H)) \rfloor \\ & \qquad \text{for } U, W \subseteq V \text{ with } U \cap W = \emptyset, \\ & \qquad \text{and for partitions } F, H \text{ of } \delta(U \cup W) \\ & \qquad \text{with } b(U) - a(W) + c(F) - d(H) \text{ odd.} \end{array}$$

Proof. Necessity of (36.12) follows from the facts that $\frac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)$ is an integer vector and that for each vector x satisfying (36.2) one has $\chi^U Mx \leq \chi^U b = b(U)$, $\chi^W Mx \geq \chi^W a = a(W)$, $\chi^F x \leq \chi^F c = c(F)$, and $\chi^H x \geq \chi^H d = d(H)$.

Again, to show sufficiency, we can assume that G has no negative edges, and we apply induction on the number of directed edges. If this number is 0, the theorem reduces to Theorem 35.3.

So we can assume that there is a directed edge $f = su$. Again, construct $G' = (V', E')$, M', a', b', d', c', as in (36.3). By induction we know that the theorem holds for the new structure.

[20] In order to reduce notation, in this chapter we take incidence vectors χ^U, χ^W, χ^F, and χ^H as *row* vectors.

Let $x \in \mathbb{R}^E$ satisfy (36.12) for G, a, b, c, d. Define $x' \in \mathbb{R}^{E'}$ by $x'(e) := x(e)$ for each $e \in E \setminus \{f\}$, and $x'(st) := -x(f)$ and $x'(tu) := x(f)$. Now it suffices to show that x' satisfies the inequalities for G', a', b', c', d' (since of x' is a convex combination of integer solutions, also x is).

So let U' and W' be disjoint subsets of V' and let F' and H' partition $\delta'(U' \cup W')$, with $b'(U') - a'(W') + c'(F') - d'(H')$ odd. Since $c'(st) = \infty$ and $d'(st) = -\infty$, we know that $st \notin \delta'(U' \cup W')$.

Let $U := U' \cap V$ and $W := W' \cap V$. Moreover, let F and H arise from F' and H' by replacing any occurrence of tu by f. Then

(36.13) $$\tfrac{1}{2}((\chi^{U'} - \chi^{W'})M' + \chi^{F'} - \chi^{H'})x' = \tfrac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x$$

since $x'(F') = x(F)$ and $x'(H') = x(H)$, and moreover, $\chi^{U'}M'x' = \chi^U Mx$ and $\chi^{W'}M'x' = \chi^W Mx$ (as $\chi^t M'x' = x'(st) + x'(tu) = 0$).

Also we have

(36.14) $$\begin{aligned}&\lfloor \tfrac{1}{2}(b'(U') - a'(W') + c'(F') - d'(H'))\rfloor\\ &= \lfloor \tfrac{1}{2}(b(U) - a(W) + c(F) - d(H))\rfloor,\end{aligned}$$

as $a'(t) = b'(t) = 0$. Hence (36.12) gives the required inequality for U', W', F', H'. ∎

The special case $a = b$, $d = \mathbf{0}$ was announced by Edmonds and Johnson [1970] and elaborated by Aráoz, Cunningham, Edmonds, and Green-Krótki [1983]. It amounts to, for $b \in \mathbb{Z}_+^V$ and $c \in \mathbb{Z}_+^E$:

(36.15) $$\mathbf{0} \leq x \leq c, Mx = b.$$

Then:

Corollary 36.3a. *The convex hull of the integer solutions of* (36.15) *is determined by* (36.15) *together with the constraints*

(36.16) $$x(\delta(U) \setminus F) - x(F) \geq 1 - c(F)$$

where $U \subseteq V$ and $F \subseteq \delta(U)$ with $b(U) + c(F)$ odd.

Proof. Directly from Theorem 36.3, by replacing Mx by b in (36.12)(iii). ∎

For undirected graphs we obtain a characterization of the capacitated perfect b-matching polytope as special case — cf. Corollary 32.2a.

36.3. Total dual integrality

System (36.12) generally is not totally dual integral (cf. the example in Section 30.5). However, if we delete the parity condition in (36.12)(iii):

(36.17)
$$\begin{array}{rl} \text{(i)} & d \le x \le c, \\ \text{(ii)} & a \le Mx \le b, \\ \text{(iii)} & \frac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x \\ & \quad \le \lfloor \frac{1}{2}(b(U) - a(W) + c(F) - d(H)) \rfloor \\ & \qquad \text{for } U, W \subseteq V \text{ with } U \cap W = \emptyset \text{ and for partition} \\ & \qquad F, H \text{ of } \delta(U \cup W), \end{array}$$

we obtain a totally dual integral system:

Theorem 36.4. *System* (36.17) *is totally dual integral.*

Proof. Again we can assume that there are no negative edges, and apply induction on the number of directed edges of G. If there is no directed edge, the theorem reduces to Theorem 35.4. If there is a directed edge $f = su$, we again construct $G' = (V', E')$, M', a', b', d', c' as in (36.3).

Let Σ and Σ' be the systems for G, a, b, c, d, and for G', a', b', c', d', respectively. By induction we know that Σ' is totally dual integral. Now the constraint $x'(st) + x'(tu) = 0$ belongs to Σ'. This implies the total dual integrality of the system Σ'' obtained from Σ' by adding an integer multiple of $x'(st)+x'(tu) = 0$ to any other constraint of Σ' so as to make the coefficient of the variable $x'(st)$ equal to 0.

Now deleting the constraints $x'(st) + x'(tu) = 1$ and the variable $x'(st)$ from Σ'', identifying $x'(e) = x(e)$ for all $e \in E \setminus \{f\}$, and identifying $x'(tu) = x(f)$, gives again a totally dual integral system. We show that it is system Σ.

Indeed, $a' \le M'x' \le b'$ becomes $a \le Mx \le b$. Similarly, for each $e \in E \setminus \{f\}$, $d'(e) \le x'(e) \le c'(e)$ becomes $d(e) \le x_e \le c(e)$, and $d'(tu) \le x'(tu) \le c'(tu)$ becomes $d(f) \le x(f) \le c(f)$, while $d'(st) \le x'(st) \le c'(st)$ is void (as the bounds are $-\infty$ and $+\infty$).

Consider next the following inequality of Σ':

(36.18)
$$\begin{array}{l} \frac{1}{2}((\chi^{U'} - \chi^{W'})M + \chi^{F'} - \chi^{H'})x' \\ \le \lfloor \frac{1}{2}(b'(U') - a'(W') + c'(F') - d'(H')) \rfloor, \end{array}$$

where U' and W' are disjoint subsets of V' and where F' and H' partition $\delta'(U' \cup W')$.

Since $c'(st) = \infty$, $d'(s,t) = -\infty$, we know that $st \notin \delta'(U' \cup W')$. Consider the coefficient of $x'(st)$ in (36.18). If this coefficient is 0, (36.18) reduces to (36.17)(iii). If this coefficient is positive, then $s, t \in U'$. Set $U'' := U' \setminus \{t\}$ and $W'' := W' \cup \{t\}$. Then in Σ'', (36.18) becomes (by subtracting $x'(st) + x'(tu) = 0$):

(36.19)
$$\begin{array}{l} \frac{1}{2}((\chi^{U''} - \chi^{W''})M + \chi^{F'} - \chi^{H'})x' \\ \le \lfloor \frac{1}{2}(b'(U'') - a'(W'') + c'(F') - d'(H')) \rfloor \end{array}$$

(since $b'(t) = a'(t) = 0$). In (36.19), the coefficient of $x'(st)$ is 0, and hence (36.19) reduces to (36.17)(iii).

We proceed similarly if the coefficient of $x'(st)$ in (36.18) is negative. ∎

A consequence is the total dual half-integrality of the original system:

Corollary 36.4a. *System* (36.12) *is totally dual half-integral.*

Proof. This follows from the fact that each inequality in (36.17) is a half-integer nonnegative combination of inequalities in (36.12). ∎

A special case is the total dual half-integrality of

(36.20)
$$\begin{aligned} &\text{(i)} && x \geq \mathbf{0}, \\ &\text{(ii)} && Mx = b, \\ &\text{(iii)} && x(\delta(U)) \geq 1 \quad \text{for each } U \subseteq V \text{ with } b(U) \text{ odd} \end{aligned}$$

(Edmonds and Johnson [1970]):

Corollary 36.4b. *System* (36.20) *is totally dual half-integral.*

Proof. This is a special case of Corollary 36.4a. ∎

From this one can derive (Barahona and Cunningham [1989]):

Corollary 36.4c. *Let* $w \in \mathbb{Z}^E$ *with* $w(C)$ *even for each circuit* C*. Then the problem of minimizing* $w^{\mathsf{T}}x$ *subject to* (36.20) *has an integer optimum dual solution.*

Proof. If $w(C)$ is even for each circuit, there is a subset U of V with $\{e \in E \mid w(e) \text{ odd}\} = \delta(U)$. Now replace w by $w' := w + \sum_{v \in U} M_v^{\mathsf{T}}$, where M_v denotes row v of M. Then $w'(e)$ is an even integer for each edge e. Hence by Corollary 36.4b there is an integer optimum dual solution y'_v ($v \in V$), z_U ($U \subseteq V$, $b(U)$ odd) for the problem of minimizing ${w'}^{\mathsf{T}}x$ subject to (36.20). Now setting $y_v := y'_v - 1$ if $v \in U$ and $y_v := y'_v$ if $v \notin U$ gives an integer optimum dual solution for w. ∎

36.4. Including parity conditions

We are not yet at the end of our self-refining trip. As was observed by Edmonds and Johnson [1973], the results can be generalized even further by including parity constraints. This can be reduced to the previous case by adding loops at the vertices at which there is a parity constraint.

Let $G = (V, E, \sigma)$ be a bidirected graph and let M be the $V \times E$ incidence matrix of G. (For definitions and terminology, see Section 36.1.) Let $a, b \in \mathbb{Z}^V$ and $d, c \in \mathbb{Z}^E$ and let S^{odd} and S^{even} be two disjoint subsets of V. We consider integer solutions x of:

(36.21) (i) $d \leq x \leq c$,
(ii) $a \leq Mx \leq b$,
(iii) $(Mx)_v$ is odd if $v \in S^{\text{odd}}$,
(iv) $(Mx)_v$ is even if $v \in S^{\text{even}}$.

The problem of finding a maximum-weight integer vector x satisfying (36.21) can be easily reduced to the special case without parity constraints, discussed in the previous chapter:

Theorem 36.5. *For any $w \in \mathbb{Q}^E$, an integer vector x maximizing $w^{\mathsf{T}}x$ over* (36.21) *can be found in strongly polynomial time.*

Proof. The condition $(Mx)_v$ is odd, can be replaced by $1 \leq (Mx)_v + 2z_v \leq 1$, where z_v is a new integer variable (bounded by $-\infty$ and ∞). Similarly, for the even case. This gives a reduction to the problem of Theorem 36.1, which implies the present theorem. ∎

We next characterize the existence of an integer vector x satisfying (36.21). To this end we make the following assumptions, which can easily be satisfied:

(36.22) (i) $a(v)$ and $b(v)$ are odd (if finite) for each $v \in S^{\text{odd}}$,
(ii) $a(v)$ and $b(v)$ are even (if finite) for each $v \in S^{\text{even}}$,
(iii) if $a(v) = b(v)$, then $v \in S^{\text{odd}} \cup S^{\text{even}}$.

Define $S := S^{\text{odd}} \cup S^{\text{even}}$. Moreover, for any vector z, again let z_+ arise from z by replacing any negative component by 0, and let z_- arise from z by replacing any positive component by 0. So $z = z_+ + z_-$.

Theorem 36.6. *Assume* (36.22) *and that $d < c$. Then there exists an integer vector $x \in \mathbb{Z}^E$ satisfying* (36.21) *if and only if for each partition T, U, W of V, the number of components K of $G[T]$ contained in $S^{\text{odd}} \cup S^{\text{even}}$ and with*

(36.23) $$|K \cap S^{\text{odd}}| + c(E[K, W^+]) + c(E[K, U^-]) + d(E[K, U^+]) + d(E[K, W^-])$$

odd is at most

(36.24) $$y_+^{\mathsf{T}} b + y_-^{\mathsf{T}} a - (y^{\mathsf{T}} M)_- c - (y^{\mathsf{T}} M)_+ d,$$

where $y := \chi^U - \chi^W$.

Proof. Define $L := \{v \in S \mid a(v) < b(v)\}$, $L^{\text{odd}} := L \cap S^{\text{odd}}$, and $L^{\text{even}} := L \cap S^{\text{even}}$.

Extend the bidirected graph G by a loop l at any vertex $v \in L$, where l has two positive ends at v. This makes the bidirected graph $G' = (V, E', \sigma')$, with $V \times E'$ incidence matrix M'. Define $a'(v) := a(v)$ and $b'(v) := b(v)$ for each $v \in V \setminus L$. Moreover, $a'(v) := b'(v) := 1$ for $v \in L^{\text{odd}}$ and $a'(v) := b'(v) := 0$

for $v \in L^{\text{even}}$. Define $d'(e) := d(e)$ and $c'(e) := c(e)$ for each $e \in E$. For each loop l at $v \in L$, define $d'(l) := \frac{1}{2}(b'(v) - b(v))$ and $c'(l) := \frac{1}{2}(a'(v) - a(v))$.

Now there exist an integer vector x satisfying (36.21) if and only if there exists an integer vector $x' \in \mathbb{Z}^{E'}$ satisfying $d' \le x' \le c'$ and $a' \le M'x' \le b'$. So we should show that the conditions given in the present theorem imply those given in Theorem 36.2 (for the modified structure). (Since in Theorem 36.2 the condition $d < c$ is required, we had to exclude loops at vertices in $S \setminus L$.)

To this end, let T, U, W partition V. Then any component K of $G'[T]$ with $b'(K) = a'(K)$ and

(36.25) $$b'(K) + c'(E'[K, W^+]) + c'(E'[K, U^-]) + d'(E'[K, U^+]) + d'(E[K, W^-])$$

odd, is a component of $G[T]$ contained in S, with $|K \cap S^{\text{odd}}| + c(E[K, W^+]) + c(E[K, U^-]) + d(E[K, U^+]) + d(E[K, W^-])$ odd (note that $a'(v) = b'(v) \iff v \in S$, and that $b'(K) \equiv |K \cap S^{\text{odd}}| \bmod 2$). Moreover, for $y := \chi^U - \chi^W$ one has

(36.26) $$\begin{aligned} & y_+^{\mathsf{T}} b' + y_-^{\mathsf{T}} a' - (y^{\mathsf{T}} M')_- c' - (y^{\mathsf{T}} M')_+ d' \\ & = y_+^{\mathsf{T}} b + y_-^{\mathsf{T}} a - (y^{\mathsf{T}} M)_- c - (y^{\mathsf{T}} M)_+ d, \end{aligned}$$

since

(36.27) $$\begin{aligned} & y_+^{\mathsf{T}} b' = b'(U) = b(U \setminus L) + |U \cap L^{\text{odd}}|, \\ & y_-^{\mathsf{T}} a' = -a'(W) = -a(W \setminus L) - |W \cap L^{\text{odd}}|, \\ & (y^{\mathsf{T}} M')_- c' = (y^{\mathsf{T}} M)_- c - 2(\tfrac{1}{2}(a'(W \cap L) - a(W \cap L))) \\ & = (y^{\mathsf{T}} M)_- c - |W \cap L^{\text{odd}}| + a(W \cap L), \\ & (y^{\mathsf{T}} M')_+ d' = (y^{\mathsf{T}} M)_+ d + 2(\tfrac{1}{2}(b'(U \cap L) - b(U \cap L))) \\ & = (y^{\mathsf{T}} M)_+ d + |U \cap L^{\text{odd}}| - b(U \cap L). \end{aligned}$$ ∎

A special case is the following result on orientations by Frank, Tardos, and Sebő [1984].

Corollary 36.6a. *Let $G = (V, E)$ be an undirected graph and let $l, u \in \mathbb{Z}_+^V$ be such that $l(v) \equiv u(v)$ (mod 2) for each $v \in V$. Then G has an orientation $D = (V, A)$ such that*

(36.28) $$l(v) \le \deg_D^{\text{out}}(v) \le u(v) \text{ and } \deg_D^{\text{out}}(v) \equiv u(v) \pmod 2$$

for each $v \in V$ if and only if for each partition T, U, W of V, the number of components K of $G[T]$ with $u(K) + |E[K]| + |E[K, U]|$ odd is at most

(36.29) $$u(U) - l(W) - |E[U]| + |E[W]| + |\delta(W)|.$$

Proof. Let $D' = (V, A')$ be an arbitrary orientation of G. Let $\delta^{\text{out}}(U) := \delta_{D'}^{\text{out}}(U)$ and $\delta^{\text{in}}(U) := \delta_{D'}^{\text{in}}(U)$ for any $U \subseteq V$.

Then G has an orientation as required in the theorem if and only if there exists a vector $x \in \mathbb{Z}^{A'}$ with $\mathbf{0} \le x \le \mathbf{1}$ and

$$(36.30) \qquad l(v) \leq x(\delta^{\text{in}}(v)) + |\delta^{\text{out}}(v)| - x(\delta^{\text{out}}(v)) \leq u(v)$$

and

$$(36.31) \qquad x(\delta^{\text{in}}(v)) + |\delta^{\text{out}}(v)| - x(\delta^{\text{out}}(v)) \equiv u(v) \pmod 2$$

for each $v \in V$. (This can be seen by reversing the orientation if and only if $x_a = 1$.)

Define for each $v \in V$,

$$(36.32) \qquad a(v) := l(v) - |\delta^{\text{out}}(v)| \text{ and } b(v) := u(v) - |\delta^{\text{out}}(v)|.$$

Moreover, let $d, c \in \mathbb{Z}^{A'}$ with $d = \mathbf{0}$ and $c = \mathbf{1}$. Let M be the $V \times A'$ incidence matrix of D' (such that $M_{v,a} = -1$ if a leaves v and $M_{v,a} = +1$ if a enters v). Let S^{odd} and S^{even} be the sets of vertices v with $b(v)$ odd and even, respectively.

Then the existence of an orientation as required is equivalent the existence of an integer vector x satisfying (36.21). Hence, by Theorem 36.6, it is equivalent to the condition that for each partition T, U, W of V the number of components K of $G[T]$ with (for the bidirected graph $G = (V, E, \sigma)$ obtained from M):

$$(36.33) \qquad b(K) + |E[K, W^+]| + |E[K, U^-]|$$

odd is at most

$$(36.34) \qquad u(U) - \sum_{v \in U} |\delta^{\text{out}}(v)| - l(W) + \sum_{v \in W} |\delta^{\text{out}}(v)| + |\delta^{\text{out}}(U)| + |\delta^{\text{in}}(W)|.$$

Now (36.33) is equal to

$$(36.35) \qquad \begin{aligned} & u(K) - \sum_{v \in K} |\delta^{\text{out}}(v)| + |E[K, W^+]| + |E[K, U^-]| \\ & = u(K) - |E[K]| + |\delta^{\text{out}}(K)| + |E[K, W^+]| + |E[K, U^-]| \\ & \equiv u(K) - |E[K]| + |\delta^{\text{out}}(K)| + |E[K, W^+]| + 2|E[K, U^+]| \\ & + |E[K, U^-]| \equiv u(K) + |E[K]| + |E[K, U]| \pmod 2, \end{aligned}$$

since $|\delta^{\text{out}}(K)| = |E[K, U^+]| + |E[K, W^+]|$ and $|E[K, U]| = |E[K, U^+]| + |E[K, U^-]|$. Moreover, (36.34) is equal to (36.29), proving the corollary. ∎

One can similarly derive the following two further orientation results of Frank, Tardos, and Sebő [1984].

Corollary 36.6b. *Let $G = (V, E)$ be an undirected graph and let $u \in \mathbb{Z}_+^V$. Then G has an orientation $D = (V, A)$ such that*

$$(36.36) \qquad \deg_D^{\text{out}}(v) \leq u(v) \text{ and } \deg_D^{\text{out}}(v) \equiv u(v) \pmod 2$$

for each $v \in V$ if and only if for each $U \subseteq V$ the number of components K of $G - U$ with $u(K) + |E[K]| + |\delta(K)|$ odd is at most $u(U) - |E[U]|$.

Proof. Similar to the proof of Corollary 36.6a. ∎

Corollary 36.6c. *Let $G = (V, E)$ be an undirected graph and let $l \in \mathbb{Z}_+^V$. Then G has an orientation $D = (V, A)$ such that*

$$\text{(36.37)} \qquad \deg_D^{\text{out}}(v) \geq l(v) \text{ and } \deg_D^{\text{out}}(v) \equiv l(v) \pmod 2$$

for each $v \in V$ if and only if for each $U \subseteq V$ the number of components K of $G - U$ with $l(K) + |E[K]| + |\delta(K)|$ odd is at most $|E[U]| + |\delta(U)| - l(U)$.

Proof. Similar to the proof of Corollary 36.6a. ∎

36.5. Convex hull

The convex hull of the integer solutions of (36.21) is characterized by:

Theorem 36.7. *Assuming* (36.22)*, the convex hull of the integer solutions of* (36.21) *is determined by* (36.21)(i) *and* (ii)*, together with the constraints*

$$\text{(36.38)} \qquad \tfrac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x \leq \lfloor \tfrac{1}{2}(b(U) - a(W) + c(F) - d(H)) \rfloor,$$

where U and W are disjoint subsets of $V \setminus S$ and where F and H partition $\delta(U \cup W \cup R)$ for some $R \subseteq S$ with $|R \cap S^{\text{odd}}| + b(U) - a(W) + c(F) - d(H)$ odd.

Proof. To see necessity of (36.38), let x be an integer vector satisfying (36.21), and choose U, W, R, F and H as described in the theorem. As x satisfies $d \leq x \leq c$ and $a \leq Mx \leq b$ one directly has $((\chi^U - \chi^W)M + \chi^F - \chi^H)x \leq b(U) - a(W) + c(F) - d(H)$. So it suffices to show that strict inequality holds. Now $(\chi^U + \chi^W + \chi^R)M + \chi^F + \chi^H$ is an even vector. So (using (36.21)(iii) and (iv))

$$\text{(36.39)} \qquad \begin{aligned} &((\chi^U - \chi^W)M + \chi^F - \chi^H)x \equiv \chi^R Mx \equiv |R \cap S^{\text{odd}}| \\ &\not\equiv b(U) - a(W) + c(F) - d(H) \pmod 2 \end{aligned}$$

This shows strict inequality.

We next show that (36.38) determines the convex hull, by reduction to Theorem 36.3. Let L, L^{odd}, L^{even}, $G' = (V, E')$, M', a', b', d', c' be as in the proof of Theorem 36.6. Let $x \in \mathbb{R}^E$ satisfy (36.21)(i) and (ii) and all constraints (36.38). Define $x \in \mathbb{R}^{E'}$ by $x'(e) := x(e)$ for each $e \in E$, and $x'(l) := a'(v) - x(\delta(v))$ for the loop l at any $v \in L$. Then $d' \leq x' \leq c'$ and $a' \leq M'x' \leq b'$. It suffices to show that x' is a convex combination of integer solutions of this system. By Theorem 36.3, it suffices to check condition (36.12)(iii) for G', x'.

Let U' and W' be disjoint subsets of V and let F and H partition $\delta'(U' \cup W')$, with $b'(U') - a'(W') + c'(F) - d'(H)$ odd. Define $U := U' \setminus S$, $W := W' \setminus S$, and $R := (U' \cup W') \cap S$. Then $|R \cap S^{\text{odd}}| + b(U) - a(W) + c(F) - d(H)$ is odd, since $|R \cap S^{\text{odd}}| \equiv b'(U' \cap S) - a'(W' \cap S) \pmod 2$. Moreover,

$$
(36.40)\qquad \begin{aligned}
&\chi^{U'} M'x' = \chi^{U} Mx + b(U' \cap (S \setminus L)) + |U' \cap L^{\text{odd}}|,\\
&\chi^{W'} M'x' = \chi^{W} Mx + a(W' \cap (S \setminus L)) + |W' \cap L^{\text{odd}}|,\\
&b'(U') = b(U) + b(U' \cap (S \setminus L)) + |U' \cap L^{\text{odd}}|, \text{ and}\\
&a'(W') = a(W) + a(W' \cap (S \setminus L)) + |W' \cap L^{\text{odd}}|.
\end{aligned}
$$

Hence (36.38) for x implies (36.12)(iii) for x'. ∎

36.5a. Convex hull of vertex-disjoint circuits

Green-Krótki [1980] and Aráoz, Cunningham, Edmonds, and Green-Krótki [1983] showed that the previous theorem implies a characterization of the convex hull of disjoint sets of circuits:

Corollary 36.7a. *Let $G = (V, E)$ be a graph. Then the convex hull of the vectors χ^F where F is the edge set of the union of a number of vertex-disjoint circuits is given by:*

$$
(36.41)\qquad \begin{array}{rll}
\text{(i)} & 0 \leq x_e \leq 1 & (e \in E),\\
\text{(ii)} & x(\delta(v)) \leq 2 & (v \in V),\\
\text{(iii)} & x(\delta(U) \setminus F) - x(F) \geq 1 - |F| & (U \subseteq V,\ F \subseteq \delta(U), |F| \text{ odd}).
\end{array}
$$

Proof. This follows directly from Theorem 36.7, since x is an incidence vector χ^F of the edge set of a vertex-disjoint union of disjoint circuits if and only if (36.41)(i) and (ii) are satisfied, together with: $x(\delta(v))$ even for each $v \in V$. So we can take $a = \mathbf{0}$, $b = \mathbf{2}$, $d = \mathbf{0}$, $c = \mathbf{1}$, $S^{\text{even}} = V$, and $S^{\text{odd}} = \emptyset$. In particular, U and W are empty in (36.38). ∎

Note that Corollaries 29.2e and 36.7a imply that the polytope described in Corollary 36.7a is obtained from the $\emptyset$-join polytope by adding the constraint (36.41)(ii).

This has as consequence Corollary 29.2f (due to Seymour [1979b]) characterizing the circuit cone. Given a graph $G = (V, E)$, the *circuit cone* is the cone in $\mathbb{R}^E$ generated by the incidence vectors of circuits. This cone is determined by:

$$
(36.42)\qquad \begin{array}{rll}
\text{(i)} & x_e \geq 0 & \text{for each } e \in E,\\
\text{(ii)} & x(D) \geq 2x_e & \text{for each cut } D \text{ and } e \in D.
\end{array}
$$

To prove this, we may assume (by scaling) that $x(E) \leq 1$. Then (36.42)(ii) implies (36.41)(iii), and hence the characterization follows from Corollary 36.7a.

36.6. Total dual integrality

We finally show that the system given by (36.21)(i) and (ii) and (36.38) after deleting the parity constraint on R, is TDI:

Theorem 36.8. *Assuming* (36.22)*, the following system is TDI (setting $T := V \setminus S$):*

(36.43)
$$\begin{array}{rl} \text{(i)} & d \le x \le c, \\ \text{(ii)} & \frac{1}{2}a_v \le \frac{1}{2}(Mx)_v \le \frac{1}{2}b_v, \text{ for } v \in S, \\ \text{(iii)} & a_v \le (Mx)_v \le b_v, \text{ for } v \in T, \\ \text{(iv)} & \frac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x \\ & \qquad \le \frac{1}{2}(b(U) - a(W) + c(F) - d(H) - \varepsilon), \end{array}$$

where U and W are disjoint subsets of T, where F and H partition $\delta(U \cup W \cup R)$ for some $R \subseteq S$, and where $\varepsilon \in \{0,1\}$ such that $\varepsilon \equiv |R \cap S^{\mathrm{odd}}| + b(U) - a(W) + c(F) - d(H) \pmod 2$.

Proof. The partition of V into S and T induces a partition of M, a, b into M_S, a_S, b_S and M_T, a_T, b_T. By Theorem 36.4, the system

(36.44)
$$\begin{array}{rl} \text{(i)} & d \le x \le c, \\ \text{(ii)} & 0 \le z \le \frac{1}{2}(b_S - a_S), \\ \text{(iii)} & M_S x + 2z = b_S, \\ \text{(iv)} & a_T \le M_T x \le b_T \end{array}$$

becomes TDI by adding the inequalities

(36.45)
$$\begin{array}{l} \frac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x + z(U \cap S) - z(W \cap S) \\ \le \lfloor \frac{1}{2}(b(U) - b(W \cap S) - a(W \cap T) + c(F) - d(H)) \rfloor, \end{array}$$

for disjoint subsets U, W of V and partitions F, H of $\delta(U \cup W)$. (36.44) is equivalent to:

(36.46)
$$\begin{array}{l} \frac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x + z(U \cap S) - z(W \cap S) \\ \le \frac{1}{2}(b(U) - b(W \cap S) - a(W \cap T) + c(F) - d(H) - \varepsilon), \end{array}$$

where $\varepsilon \in \{0,1\}$ and

(36.47)
$$\varepsilon \equiv b(U) - b(W \cap S) - a(W \cap T) + c(F) - d(H) \pmod 2.$$

Substituting $z := \frac{1}{2}(b_S - M_S x)$ in (36.44)(ii) gives (36.43)(ii), and in (36.46) gives

(36.48)
$$\begin{array}{l} \frac{1}{2}((\chi^U - \chi^W)M + \chi^F - \chi^H)x + \frac{1}{2}b(U \cap S) - \frac{1}{2}\chi^{U \cap S} M_S x \\ - \frac{1}{2}b(W \cap S) + \frac{1}{2}\chi^{W \cap S} M_S x \\ \le \frac{1}{2}(b(U) - b(W \cap S) - a(W \cap T) + c(F) - c(H) - \varepsilon). \end{array}$$

Equivalently:

(36.49)
$$\begin{array}{l} \frac{1}{2}((\chi^{U \cap T} - \chi^{W \cap T})M + \chi^F - \chi^H)x \\ \le \frac{1}{2}(b(U \cap T) - a(W \cap T) + c(F) - d(H) - \varepsilon). \end{array}$$

This is equivalent to (36.43)(iv), and total dual integrality is maintained by Theorem 5.27. Note that

(36.50)
$$\begin{array}{l} \varepsilon \equiv b(U) - b(W \cap S) - a(W \cap T) + c(F) - d(H) \\ \equiv b(U \cap T) - a(W \cap T) + c(F) - d(H) + b(U \cap S) + b(W \cap S) \\ \equiv b(U \cap T) - a(W \cap T) + c(F) - d(H) + |R \cap S^{\mathrm{odd}}| \pmod 2, \end{array}$$

where $R := (U \cup W) \cap S$. ∎

We remark that the coefficients of the inequalities in (36.43) generally are not all integer.

36.7. Further results and notes

36.7a. The Chvátal rank

The results on the convex hull in this chapter (and in previous chapters) can be interpreted in terms of the so-called 'Chvátal rank' of a system of inequalities or of a matrix. (This relates to the cutting planes reviewed in Section 5.21.)

For any polyhedron P, let P_{I} denote the *integer hull* of P, that is, the convex hull of the integer vectors in P. If P is a rational polyhedron, then P_{I} is again a rational polyhedron. This polyhedron can be approached as follows.

Define for any polyhedron P, the set P' by:

(36.51) $$P' := \bigcap_{H \supseteq P} H_{\mathrm{I}},$$

where H ranges over all rational affine halfspaces containing P as a subset. Here an *affine halfspace* is a set of the form

(36.52) $$H = \{x \in \mathbb{R}^n \mid w^{\mathsf{T}} x \leq \alpha\}$$

for some nonzero $w \in \mathbb{R}^n$ and some $\alpha \in \mathbb{R}$. It is *rational* if w and α are rational. So trivially (since $P \subseteq H \Rightarrow P_{\mathrm{I}} \subseteq H_{\mathrm{I}}$):

(36.53) $$P \supseteq P' \supseteq P_{\mathrm{I}}.$$

Note that if H is as in (36.52) and w is integer, with relatively prime components, then

(36.54) $$H_{\mathrm{I}} = \{x \in \mathbb{R}^n \mid w^{\mathsf{T}} x \leq \lfloor \alpha \rfloor\}.$$

So P' arises from P by adding a 'first round of cuts'. Observe that if $P = \{x \mid Mx \leq b\}$ for some rational $m \times n$ matrix M and some vector $b \in \mathbb{Q}^m$, then in (36.51) we can restrict the affine hyperplanes H to those for which there exists a vector $y \in \mathbb{Q}^m_+$ with $y^{\mathsf{T}} M$ integer and nonzero and

(36.55) $$H = \{x \mid (y^{\mathsf{T}} M)x \leq y^{\mathsf{T}} b\}$$

(by Farkas' lemma).

It can be shown that P' is a rational polyhedron again. To P' we can apply this operation again, and obtain $P'' = (P')'$. We thus obtain a series of polyhedra P, P', P'',..., $P^{(t)}$, ..., satisfying

(36.56) $$P \supseteq P' \supseteq P'' \supseteq \cdots \supseteq P^{(t)} \supseteq \cdots P_I.$$

Now Chvátal [1973a] (cf. Schrijver [1980b]) showed that for each polyhedron P there is a finite t with $P^{(t)} = P_{\mathrm{I}}$. The smallest such t is called the *Chvátal rank* of P.

It can be proved more strongly (Cook, Gerards, Schrijver, and Tardos [1986]) that for each rational matrix M there is a finite value t such that the polyhedron $P := \{x \mid Mx \leq b\}$ has Chvátal rank at most t, for each integer vector b (of appropriate dimension). The smallest such t is called the *Chvátal rank* of M. So each totally unimodular matrix has Chvátal rank 0.

The *strong Chvátal rank* of M is, by definition, the Chvátal rank of the matrix

$$(36.57) \qquad \begin{pmatrix} I \\ -I \\ M \\ -M \end{pmatrix}.$$

So the strong Chvátal rank of M is the smallest t such that for all integer vectors d, c, a, b the polyhedron $\{x \mid d \leq x \leq c, a \leq Mx \leq b\}$ has Chvátal rank at most t. So M is totally unimodular if and only if M is integer and has strong Chvátal rank 0 (this is the Hoffman-Kruskal theorem).

Theorem 36.3 implies that the $V \times E$ incidence matrix of a bidirected graph has strong Chvátal rank at most 1. (Matrices of strong Chvátal rank at most 1 are said in Gerards and Schrijver [1986] to have the *Edmonds-Johnson property.*)

Theorem 36.9. *The $V \times E$ incidence matrix of a bidirected graph has strong Chvátal rank at most* 1.

Proof. We must show that for each integer d, c, a, b, one has $P' = P_{\mathrm{I}}$ for $P := \{x \mid d \leq x \leq c, a \leq Mx \leq b\}$. This follows from

$$(36.58) \qquad \begin{aligned} P' &\subseteq \{x \in P \mid \forall y \in \{0, \tfrac{1}{2}\}^n : y^{\mathsf{T}}M \in \mathbb{Z}^n \Rightarrow y^{\mathsf{T}}Mx \leq \lfloor y^{\mathsf{T}}b \rfloor\} \\ &= P_{\mathrm{I}} \subseteq P', \end{aligned}$$

where the equality follows from Theorem 36.3. ∎

It is generally not true that also the *transpose* M^{T} of these matrices have Chvátal rank at most 1, as is shown by the incidence matrix M of the complete graph K_4. In Section 68.6c we shall study the Chvátal rank of such matrices M^{T}.

36.7b. Further notes

Gabow [1983a] gave an $O(m^{\frac{3}{2}})$-time algorithm for finding a maximum $s-t$ flow in a bidirected graph with unit capacities. Moreover, he gave $O(m^2 \log n)$- and $O(n^2 m)$-time algorithms for finding a minimum-cost bidirected $s-t$ flow of given value, with unit capacities.

Chapter 37

The dimension of the perfect matching polytope

In this chapter the dimension of the perfect matching polytope is characterized. It implies a characterization of the dimension of the perfect matching space — the linear space spanned by the incidence vectors of perfect matchings.
The basis of determining the dimension is formed by the matching-covered graphs without nontrivial tight cuts. For such graphs, there is an easy formula for the dimension.
Key result (needed in characterizing the perfect matching lattice in the next chapter) is a characterization of Lovász of the matching-covered graphs without nontrivial tight cuts: the 'braces' and the 'bricks'.

37.1. The dimension of the perfect matching polytope

Naddef [1982] gave a min-max formula for the dimension of the perfect matching polytope. By the work of Edmonds, Lovász, and Pulleyblank [1982], it is equivalent to the following.

Let $G = (V, E)$ be a graph and let E_0 be the set of edges covered by at least one perfect matching. Defining $G_0 := (V, E_0)$, one trivially has:

(37.1) $$\dim(P_{\text{perfect matching}}(G)) = \dim(P_{\text{perfect matching}}(G_0)).$$

So when investigating the dimension of the perfect matching polytope, we can confine ourselves to *matching-covered* graphs, that is, to graphs in which each edge is contained in at least one perfect matching.

A further reduction can be obtained by considering tight cuts. A cut C is called *odd* if $C = \delta(U)$ for some $U \subseteq V$ with $|U|$ odd. A cut C is called *tight* if it is odd and each perfect matching intersects C in exactly one edge.

Let $G = (V, E)$ be a graph and let $U \subseteq V$. Recall that G/U denotes the graph obtained from G by contracting U to one vertex, which vertex we will call U. In the obvious way, we will consider the edge set of G/U as a subset of the edge set of G. Hence, for any $x \in \mathbb{R}^E$, we can speak of the *projection* of x to the edges of G/U.

Theorem 37.1. *Let $G = (V, E)$ be a matching-covered graph and let $\delta(U)$ be a tight cut. Define $G_1 := G/U$ and $G_2 := G/\overline{U}$ (where $\overline{U} := V \setminus U$). Then*

(37.2) $$\dim(P_{\text{perfect matching}}(G)) = \dim(P_{\text{perfect matching}}(G_1)) + \dim(P_{\text{perfect matching}}(G_2)) - |\delta(U)| + 1.$$

Proof. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Then a vector $x \in \mathbb{R}^E$ belongs to the perfect matching polytope of G if and only if its projections to E_1 and E_2 belong to the perfect matching polytopes of G_1 and G_2 respectively. Moreover, since G is matching-covered and since $\delta(U)$ is tight, the projection of $P_{\text{perfect matching}}(G)$ on $\delta(U)$ has dimension equal to $|\delta(U)| - 1$. ∎

This theorem gives a reduction if there exists a nontrivial tight cut. (A cut C is called *nontrivial* if $C = \delta(U)$ for some U with $1 < |U| < |U| - 1$.) Then:

Theorem 37.2. *Let $G = (V, E)$ be a matching-covered graph without any nontrivial tight cut and with at least one perfect matching. Then*

(37.3) $$\dim(P_{\text{perfect matching}}(G)) = |E| - |V| + k,$$

where k is the number of bipartite components of G.

Proof. We may assume that G is connected. If G is bipartite, the result follows from Theorem 18.6. If G is nonbipartite, consider a vector x in the relative interior of the perfect matching polytope of G. Since G is matching-covered, we know that $x_e > 0$ for each edge e, and since G has no nontrivial tight cut, we know that $x(C) > 1$ for each nontrivial odd cut C. Hence the only constraints in (25.2) satisfied by x with equality are the constraints $x(\delta(v)) = 1$ for $v \in V$. So $\dim(P_{\text{perfect matching}}(G)) \geq |E| - |V|$.

To see equality, we show that the constraints $x(\delta(v)) = 1$ are independent. For let $u \in V$, and choose an odd-length $u - u$ walk $(u, e_1, \ldots, e_t, u)$. For each $e \in E$, let x_e be the number of odd i with $e = e_i$, minus the number of even i with $e = e_i$. Then $x(\delta(u)) = 2$ and $x(\delta(v)) = 0$ for all $v \neq u$. ∎

Theorems 37.1 and 37.2 describe the decomposition of the dimension problem. We now aggregate these results.

For any cut C, any set U with $C = \delta(U)$ is called a *shore* of C. Two cuts C and C' are called *cross-free* if they have shores U and U' that are disjoint. A collection $\mathcal{F}$ of cuts is *cross-free* if each two cuts in $\mathcal{F}$ are cross-free.

Let $\mathcal{F}$ be a cross-free collection of nontrivial cuts. An *$\mathcal{F}$-contraction* of G is a graph obtained from G by choosing a $U_0 \subseteq V$ with $\delta(U_0) \in \mathcal{F}$, contracting U_0, and contracting each maximal proper subset U of $V \setminus U_0$ with $\delta(U) \in \mathcal{F}$.

One easily checks that, if G is connected, there exist precisely $|\mathcal{F}| + 1$ $\mathcal{F}$-contractions. Let $\text{nonbip}_G(\mathcal{F})$ denote the number of $\mathcal{F}$-contractions that are nonbipartite.

Corollary 37.2a. *Let $G = (V, E)$ be a connected matching-covered graph with $|V| \geq 2$. Let $\mathcal{F}$ be any inclusionwise maximal cross-free collection of nontrivial tight cuts. Then*

$$(37.4) \qquad \dim(P_{\text{perfect matching}}(G)) = |E| - |V| - \text{nonbip}_G(\mathcal{F}) + 1.$$

Proof. The corollary follows directly by induction from Theorems 37.1 and 37.2, as follows.

If $\mathcal{F} = \emptyset$, then (37.4) follows from Theorem 37.2. If $\mathcal{F} \neq \emptyset$, choose a cut $\delta(U) \in \mathcal{F}$. Let $G_1 := G/U$ and $G_2 := G/\overline{U}$ (where $\overline{U} := V \setminus U$). Then G_1 and G_2 are connected and matching-covered again.

Let $\mathcal{F}_1$ be the set of cuts in $\mathcal{F}$ that have a shore properly contained in $V \setminus U$ and let $\mathcal{F}_2$ be the set of cuts in $\mathcal{F}$ that have a shore properly contained in U.

Then $\mathcal{F}_1$ forms an inclusionwise maximal cross-free collection of nontrivial tight cuts in G_1. So inductively

$$(37.5) \qquad \dim(P_{\text{perfect matching}}(G_1)) = |EG_1| - |VG_1| - \text{nonbip}_{G_1}(\mathcal{F}_1) + 1.$$

Similarly,

$$(37.6) \qquad \dim(P_{\text{perfect matching}}(G_2)) = |EG_2| - |VG_2| - \text{nonbip}_{G_2}(\mathcal{F}_2) + 1.$$

Now each $\mathcal{F}$-contraction of G is an $\mathcal{F}_i$ contraction of G_i for exactly one $i \in \{1, 2\}$. Hence

$$(37.7) \qquad \text{nonbip}_G(\mathcal{F}) = \text{nonbip}_{G_1}(\mathcal{F}_1) + \text{nonbip}_{G_2}(\mathcal{F}_2).$$

Since moreover $|EG| = |EG_1| + |EG_2| - |\delta(U)|$ and $|VG_1| + |VG_2| = |VG| + 2$, we obtain (37.4) with Theorem 37.1. ∎

37.2. The perfect matching space

We derive from Corollary 37.2a a characterization of the perfect matching space and its dimension. The *perfect matching space* of a graph $G = (V, E)$ is the linear hull of the incidence vectors of perfect matchings; that is,

$$(37.8) \qquad S_{\text{perfect matching}}(G) := \text{lin.hull}\{\chi^M \mid M \text{ perfect matching in } G\}.$$

(Here lin.hull denotes linear hull.)

Corollary 37.2a directly gives for the dimension of the perfect matching space:

Corollary 37.2b. *Let $G = (V, E)$ be a connected matching-covered graph with $|V| \geq 2$. Let $\mathcal{F}$ be any inclusionwise maximal cross-free collection of nontrivial tight cuts. Then*

$$(37.9) \qquad \dim(S_{\text{perfect matching}}(G)) = |E| - |V| - \text{nonbip}_G(\mathcal{F}) + 2.$$

Proof. The dimension of the perfect matching space is 1 more than the dimension of the perfect matching polytope. So the corollary follows from Corollary 37.2a. ∎

With the help of the description of the perfect matching polytope we can similarly describe the perfect matching space in terms of equations:

Theorem 37.3. *The perfect matching space of a graph $G = (V, E)$ is equal to the set of vectors $x \in \mathbb{R}^E$ satisfying*

$$(37.10)\quad \begin{array}{rll} \text{(i)} & x_e = 0 & \text{if } e \text{ is contained in no perfect matching,} \\ \text{(ii)} & x(C) = x(\delta(v)) & \text{for each tight cut } C \text{ and each vertex } v. \end{array}$$

Proof. Condition (37.10) clearly is necessary for each vector x in the perfect matching space. To see sufficiency, let $x \in \mathbb{R}^E$ satisfy (37.10). We can assume that G has at least one perfect matching.

By adding sufficiently many incidence vectors of perfect matchings to x, we can achieve that $x_e \geq 0$ for each edge e, and $x_e > 0$ for at least one edge e, and $x(C) \geq x(\delta(v))$ for each odd cut C and each vertex v. By scaling we can achieve that $x(\delta(v)) = 1$ for each $v \in V$. Then x belongs to the perfect matching polytope of G, and hence to the perfect matching space. ∎

37.3. The brick decomposition

For any inclusionwise maximal cross-free collection $\mathcal{F}$ of nontrivial tight cuts, the family of $\mathcal{F}$-contractions is called a *brick decomposition.* (We note here that it does not mean that each $\mathcal{F}$-contraction is a brick as defined in Section 37.6.)

Lovász [1987] showed that a brick decomposition is a unique family of graphs (up to isomorphism), independently of the maximal cross-free collection of tight cuts chosen:

Theorem 37.4. *All brick decompositions of a matching-covered graph $G = (V, E)$ are the same (up to isomorphism).*

Proof. By induction on $|V|$. Consider two maximal cross-free collection $\mathcal{F}$ and $\mathcal{F}'$ of nontrivial tight cuts.

Case 1: $\mathcal{F}$ and $\mathcal{F}'$ have a common member $\delta(U)$. By induction, the result of two decompositions of $G/\overline{U}$ is the same (where $\overline{U} := V \setminus U$). Similarly, the result of two decompositions of G/U is the same. The theorem follows.

Case 2: There exist $C \in \mathcal{F}$ and $C' \in \mathcal{F}'$ with C and C' cross-free. Let $\mathcal{F}''$ be a maximal cross-free collection of nontrivial tight cuts containing C and C'. By Case 1, the decompositions of G by $\mathcal{F}$ and $\mathcal{F}''$ result in the same family of graphs. Similarly, the decompositions of G by $\mathcal{F}'$ and $\mathcal{F}''$ result in the same family of graphs. The theorem follows.

Case 3: There exist $C = \delta(U) \in \mathcal{F}$ and $C' = \delta(U') \in \mathcal{F}'$ with $|U \cap U'|$ odd and at least 3. Then trivially $C'' := \delta(U \cap U')$ is tight again. Let $\mathcal{F}''$ be a maximal cross-free collection of nontrivial tight cuts containing C''. By Case 2, the decompositions of G by $\mathcal{F}$ and $\mathcal{F}''$ result in the same family of graphs. Similarly, the decompositions of G by $\mathcal{F}'$ and $\mathcal{F}''$ result in the same family of graphs. Again, the theorem follows.

Case 4: None of the previous cases applies. Let $C = \delta(U) \in \mathcal{F}$ and $C' = \delta(U') \in \mathcal{F}'$. Then $\mathcal{F} = \{C\}$ and $\mathcal{F}' = \{C'\}$. For suppose that say $\mathcal{F}$ contains another cut $C'' = \delta(U'')$. We can assume that $U \subseteq U''$ and that $U'' \cap U'$ is odd. So $|U'' \cap U'| = 1$ (as Case 3 does not apply), and therefore $|U \cap U'| = 1$ (as Case 2 does not apply). However, $U \cup U'$ is odd and disjoint from $U'' \setminus U$, implying that $U \cup U'$ is at most $|V|-2$, and so Case 3 applies, a contradiction.

So $\mathcal{F} = \{C\}$ and $\mathcal{F}' = \{C'\}$. We can now assume that $U \cap U'$ is odd. Since Case 3 does not apply, $|U \cap U'| = 1$ and $|U \cup U'| = |V| - 1$. Let $U \cap U' = \{u\}$ and $U' \cup U = V \setminus \{v\}$.

Now $\{u, v\}$ is a 2-vertex-cut in G, separating $U \setminus \{u\}$ and $U' \setminus \{u\}$. For suppose that there is an edge e connecting $U \setminus \{u\}$ and $U' \setminus \{u\}$. Let M be a perfect matching containing e. Let f be the edge in M covering u. Then f leaves at least one of U and U'. Since e leaves both U and U', this contradicts the fact that U and U' give tight cuts.

As G has no cut vertices (as G is matching-covered), this implies that $G/\overline{U}$ and G/U' are isomorphic graphs, and similarly that G/U and $G/\overline{U}$ are isomorphic. The theorem follows. ∎

37.4. The brick decomposition of a bipartite graph

All graphs in the brick decomposition of a bipartite graph are bipartite:

Theorem 37.5. *Let G be a matching-covered graph and let $\mathcal{F}$ be an inclusionwise maximal cross-free collection of nontrivial tight cuts. Then G is bipartite if and only if each $\mathcal{F}$-contraction is bipartite.*

Proof. It suffices to prove that for any nontrivial tight cut $\delta(U)$:

(37.11) G is bipartite if and only if G/U and $G/\overline{U}$ are bipartite

(where $\overline{U} := VG \setminus U$). Sufficiency in (37.11) is direct (actually, it holds for any cut). To see necessity in (37.11), note that, since G is matching-covered, U has neighbours only in the largest colour class of the bipartite graph $G-U$. So G/U is bipartite, and similarly, $G/\overline{U}$ is bipartite. ∎

37.5. Braces

A bipartite graph $G = (V, E)$, with colour classes U and W, is called a *brace* if G is matching-covered with $|V| \geq 4$ and for all distinct $u, u' \in U$ and $w, w' \in W$, the graph $G - u - u' - w - w'$ has a perfect matching.

By Hall's marriage theorem (Theorem 22.1), a connected bipartite graph $G = (V, E)$ with equal-sized colour classes U and W is a brace if and only if for each subset X of U with $1 \leq |X| \leq |U| - 2$ one has

$$(37.12) \qquad |N(X)| \geq |X| + 2.$$

Theorem 37.6. *Each tight cut in a brace is trivial.*

Proof. Let $G = (V, E)$ be a brace with colour classes U and W, and suppose that $\delta(T)$ is a nontrivial tight cut. As $|T|$ is odd, by symmetry we can assume that $|U \cap T| < |W \cap T|$.

Then $|U \cap T| = |W \cap T| - 1$, since there exists a perfect matching intersecting $\delta(T)$ in exactly one edge. Since $\delta(T)$ is nontrivial, $1 \leq |U \cap T| \leq |U| - 2$.

Moreover, there is no edge e connecting $U \cap T$ and $W \setminus T$. Otherwise this e would be contained in a perfect matching M. This perfect matching also contains an edge connecting $U \setminus T$ and $W \cap T$, contradicting the tightness of $\delta(T)$.

So $N(U \cap T) \subseteq W \cap T$, and hence $|N(U \cap T)| \leq |U \cap T| + 1$, contradicting (37.12). ∎

37.6. Bricks

A graph G is called a *brick* if G is 3-connected and bicritical, and has at least four vertices. (A graph G is called *bicritical* if $G - u - v$ has a perfect matching for any two distinct vertices u, v.)

The following key result was shown by Edmonds, Lovász, and Pulleyblank [1982]:

Theorem 37.7. *Each tight cut in a brick is trivial.*

Proof. Let $G = (V, E)$ be a brick, and suppose that it has a nontrivial tight cut C_0. Let $\mathcal{C}$ be the collection of odd cuts in G.

For any $b \in \mathbb{Q}^V$, consider the linear program

$$(37.13) \qquad \begin{array}{lll} \text{minimize} & \displaystyle\sum_{e=uv \in E} (b(u) + b(v)) x_e & \\ \text{subject to} & x(C) \geq 1 & (C \in \mathcal{C}), \\ & x_e \geq 0 & (e \in E). \end{array}$$

and its dual

(37.14) $$\begin{aligned} &\text{maximize} && \sum_{C\in\mathcal{C}} y(C) && \\ &\text{subject to} && \sum_{C\ni e} y(C) \leq b(u)+b(v) && (e=uv\in E), \\ & && y(C)\geq 0 && (C\in\mathcal{C}). \end{aligned}$$

We first show:

(37.15) there exist $y \in \mathbb{Q}_+^{\mathcal{C}}$ and $b \in \mathbb{Q}_+^V$ such that $\sum_{C\ni e} y(C) \leq b(u)+b(v)$ for each edge $e=uv$, and such that $y(\mathcal{C}) = b(V)$ and $y(C_0) > 0$.

To prove this, define $w = \chi^{C_0}$ (the incidence vector of C_0 in $\mathbb{R}^E$). As C_0 is tight, the maximum of $w(M)$ over perfect matchings M is equal to 1. Hence, by Edmonds' perfect matching polytope theorem (Theorem 25.1) and by linear programming duality, there exists a vector $z \in \mathbb{Q}^{\mathcal{C}}$ such that

(37.16)
(i) $\sum_{C\ni e} z(C) \leq -w(e)$ for each edge e,
(ii) $z(C) \geq 0$ if C is nontrivial,
(iii) $z(\mathcal{C}) = -1$.

For $v \in V$, define $b(v) := -z(\delta(v))$ if $z(\delta(v)) < 0$, and $b(v) := 0$ otherwise. For $C \in \mathcal{C}$, define $y(C) := z(C)$ if $z(C) > 0$, and $y(C) := 0$ otherwise. Then (37.16) implies:

(37.17)
(i) $b(u)+b(v) \geq \sum_{C\ni e} y(C) + w(e)$ for each edge $e = uv$,
(ii) $b \geq \mathbf{0}$, $y \geq \mathbf{0}$,
(iii) $b(V) = y(\mathcal{C}) + 1$.

So resetting $y(C_0) := y(C_0) + 1$ gives b and y as required in (37.15), proving (37.15).

This implies:

(37.18) for some vector $b \in \mathbb{Z}_+^V$ there exists an integer optimum solution $y \in \mathbb{Z}_+^{\mathcal{C}}$ of (37.14) such that $y(C_0) \geq 1$.

Indeed, in (37.15) we can assume (by scaling) that b and y are integer. Then by the properties described in (37.15), y is a feasible solution of (37.14). Since the maximum in (37.13) is at least $b(V)$ (as any perfect matching M satisfies $w(M) = b(V)$), and since $y(\mathcal{C}) = b(V)$, we know that y is an optimum solution of (37.14). This proves (37.18).

Now fix a b as in (37.18), with $b(v)$ minimal. Then

(37.19) for any optimum solution y of (37.14) and any $C \in \mathcal{C}$ one has that if $y(C) > 0$, then C is tight.

Indeed, any perfect matching M attains the maximum (37.13) (as the maximum value equals $b(V)$). So if $y(C) > 0$, by complementary slackness, $|M \cap C| = 1$. This shows (37.19).

Call a vector $y \in \mathbb{R}_+^{\mathcal{C}}$ *laminar* if the collection $\{C \in \mathcal{C} \mid y(C) > 0\}$ is laminar. Then:

(37.20) there exists a laminar integer optimum solution of (37.14) such that $y(C) \geq 1$ for at least one nontrivial tight cut C.

To see this, choose an integer optimum solution y of (37.14) such that $y(C) \geq 1$ for at least one nontrivial tight cut C, with

(37.21) $$\sum_{C \in \mathcal{C}} y(C)s(C)$$

minimized, where $s(C)$ denotes the number of pairs of vertices separated by C. We show that y is laminar.

Suppose to the contrary that C and C' cross, with $y(C) > 0$ and $y(C') > 0$. We can choose $U', U'' \subseteq V$ such that $C = \delta(U)$, $C' = \delta(U')$, and $|U \cap U'|$ is odd. Let $D := \delta(U \cap U')$ and $D' := \delta(U \cup U')$. Let $\varepsilon := \min\{y(C), y(C')\}$. Decrease $y(C)$ and $y(C')$ by ε, and increase $y(D)$ and $y(D')$ by ε. Then we obtain again a feasible solution of (37.14), while (37.21) is smaller. So both D and D' are trivial. Hence $U \cap U' = \{u\}$ and $U \cup U' = V \setminus \{v\}$ for some vertices u and v. As G is 3-connected, there is an edge e connecting $U \setminus U'$ and $U' \setminus U$. Since G is matching-covered, there is a perfect matching M containing e. So $e \in C \cap C'$. As C and C' are tight, e is the only edge of M intersecting $C \cup C'$. Hence no edge of M intersects $D = \delta(U \cap U')$, a contradiction. This proves (37.20).

Fix y satisfying (37.20). We note that the first set of constraints in (37.14) gives:

(37.22) if $e = uv \in C$ and $y(C) > 0$, then $b(u) > 0$ or $b(v) > 0$.

Moreover,

(37.23) for each $u \in V$, $b(u) = 0$ or $y(\delta(u)) = 0$.

Otherwise, decreasing $b(u)$ and $y(\delta(u))$ by 1 would give b and y with smaller $b(V)$.

We also show:

(37.24) if $y(\delta(U)) > 0$, then $G[U]$ is connected.

If not, let K be an odd component of $G[U]$ and let e be an edge in $\delta(U)$ not incident with K. Let M be a perfect matching containing e. Then M intersects $\delta(U)$ in more than one edge (since K is odd), while $\delta(U)$ is tight since $y(\delta(U)) > 0$. This contradiction proves (37.24).

Now choose an odd cut $C = \delta(U)$ with $y(C) > 0$, an edge $e_0 = u_0v \in C$ with $u_0 \in U$ and $b(u_0) > 0$, such that $|U|$ is as small as possible. (Such U, e_0, u_0 exist by (37.22).)

By (37.23), $|U| > 1$. Let $U_1, \ldots, U_k$ be the maximal proper subsets of U with $y(\delta(U_i)) > 0$. By (37.20), the U_i are pairwise disjoint. Note that $u_0 \notin U_1 \cup \cdots \cup U_k$, by the minimality of $|U|$.

Define

(37.25) $U' := U \setminus (U_1 \cup \cdots \cup U_k)$, $U_+ := \{u \in U' \mid b(u) > 0\}$, and $U_0 := U' \setminus U_+$.

Then

(37.26) there is no edge joining distinct sets among $U_0, U_1, \ldots, U_k$.

Directly from (37.22) and the minimality of $|U|$.

Moreover,

(37.27) there is no edge $e = uv$ with $u \in U_+$ and $v \in U'$.

For suppose that such an edge e exists. Then there is a perfect matching containing e. Hence, by complementary slackness, we have equality in the corresponding constraint of (37.14). As $b(u)+b(v) > 0$, we know that $y(C) > 0$ for some C with $e \in C$. Then $C = \delta(S)$ for some $S \subseteq U$. This contradicts the definition of the U_i, proving (37.27).

As $G[U]$ is connected (by (37.24)), it follows that $U_0 = \emptyset$. Next

(37.28) $|U_+| = k + 1$.

For consider any perfect matching M containing edge e_0. Then M intersects any $\delta(U_i)$ in exactly one edge (as each $\delta(U_i)$ is tight, by (37.19)) and it also intersects $\delta(U)$ in exactly one edge, namely e_0. Since $|M \cap \delta(U)| = 1$, we know with (37.26) that the edge in $M \cap \delta(U_i)$ connects U_i and U_+. Moreover, no edge in M connects two vertices in U_+ (by (37.27)). Hence we have (37.28).

(37.29) No edge connects any U_i with $V \setminus U$.

Otherwise, the same counting as for proving (37.28) gives $|U_+| = k$, a contradiction.

As $|U| > 1$ we know $k > 0$. Choose $s, t \in U_+$. As G is bicritical, $G - s - t$ has a perfect matching M. Then M intersects each $\delta(U_i)$ at least once, and hence (by (37.29)) $|U_+ \setminus \{s, t\}| \geq k$, a contradiction. ∎

37.7. Matching-covered graphs without nontrivial tight cuts

The foregoing is used in obtaining the following basic result of Lovász [1987]:

Theorem 37.8. *Let $G = (V, E)$ be a connected graph with at least four vertices. Then G is matching-covered without nontrivial tight cuts if and only if G is a brick or a brace.*

Proof. If G is a brick or a brace, then trivially G is matching-covered. Moreover, Theorems 37.6 and 37.7 show that braces and bricks have no nontrivial tight cuts.

Conversely, assume that G is matching-covered and has no nontrivial tight cuts.

Case 1: G is not bicritical. We show that G is a brace. As G is not bicritical, by Tutte's 1-factor theorem (Theorem 24.1a) there exists a subset U of V such that $G-U$ has $|U|$ odd components, with $|U|\geq 2$. As G is matching-covered, U is a stable set, and $G-U$ has no even components. For each component K of $G-U$, $\delta(K)$ is tight, and hence trivial, that is $|K|=1$. So G is bipartite, and U is one of its colour classes. If G is not a brace, there exists a subset X of U with $1\leq|X|\leq|U|-2$ and $|N(X)|\leq|X|+1$. Let $Y\subseteq V\setminus U$ with $N(X)\subseteq Y$ and $|Y|=|X|+1$. Then $\delta(X\cup Y)$ is a nontrivial tight cut, a contradiction.

Case 2: G is bicritical. We show that G is a brick. So we must show that G is 3-connected. As G is matching-covered, G is trivially 2-connected. Suppose that $\{u,v\}$ is a 2-vertex-cut. Let K be a component of $G-u-v$. As $G-u-v$ has a perfect matching, $|K|$ is even. Then $\delta(K\cup\{u\})$ is a nontrivial cut which is tight, since the intersection of $\delta(K\cup\{u\})$ with any perfect matching M is odd and at most 2 (as each edge in the intersection is incident with u or v). ∎

Chapter 38

The perfect matching lattice

This chapter is devoted to giving a proof of the deep theorem of Lovász [1987] characterizing the perfect matching lattice of a graph — the lattice generated by the incidence vectors of perfect matchings.

We summarize concepts and results from previous chapters that we need in the proof. Let $G = (V, E)$ be a graph. The following notions will be used:

- A cut C in G is *tight* if each perfect matching intersects C in exactly one edge.
- A cut C is *trivial* if $C = \delta(v)$ for some vertex v.
- G is *matching-covered* if each edge is contained in a perfect matching.
- G is *bicritical* if for each two distinct vertices u and v, the graph $G-u-v$ has a perfect matching.
- G is a *brick* if it is 3-connected and bicritical and has at least 4 vertices.
- A subset B of V is a *barrier* if $G-B$ has at least $|B|$ odd components. A *maximal barrier* is an inclusionwise maximal barrier. A *nontrivial barrier* is a barrier B with $|B| \geq 2$.

Moreover, the following results will be used:

- the perfect matching lattice of a bipartite graph is equal to the set of integer vectors in the perfect matching space (this is an easy consequence of Kőnig's edge-colouring theorem, see Theorem 20.12).
- Any two distinct inclusionwise maximal barriers in a connected matching-covered graph are disjoint (Corollary 24.11a).
- A graph is a brick if and only if it is nonbipartite and matching-covered and has no nontrivial tight cuts (a consequence of Theorem 37.8).
- A graph is bicritical if and only if it has no nontrivial barrier (a consequence of Tutte's 1-factor theorem (Corollary 24.1a)).

Throughout this chapter, $\overline{U}$ denotes the complement of U.

38.1. The perfect matching lattice

The *perfect matching lattice* (usually briefly the *matching lattice*) of a graph $G = (V, E)$ is the lattice generated by the incidence vectors of perfect matchings in G; that is,

$$L_{\text{perfect matching}}(G) := \text{lattice}\{\chi^M \mid M \text{ perfect matching in } G\}. \tag{38.1}$$

So it is a sublattice of $\mathbb{Z}^E$ and is contained in the perfect matching space of G.

In Section 20.8 we saw that the perfect matching lattice of a *bipartite* graph $G = (V, E)$ is equal to the intersection of $\mathbb{Z}^E$ with the perfect matching space of G. This characterization does not hold in general for nonbipartite graphs, as is shown by the Petersen graph. However, as was proved by Lovász [1987], any graph for which the characterization does not hold, contains the Petersen graph in some sense. In particular, for any graph without Petersen graph minor, the characterization remains valid.

In analyzing the perfect matching lattice of G, two initial observations are of interest:

- We can assume that G is matching-covered, since any edge contained in no perfect matching can be deleted;
- If G has a nontrivial tight cut, we can reduce the analysis by considering the two graphs obtained by contracting either of the shores of the cut.

So we can focus the investigations on nonbipartite matching-covered graphs without nontrivial tight cuts; that is, by Theorem 37.8, on bricks.

38.2. The perfect matching lattice of the Petersen graph

We will need a characterization of the perfect matching lattice of the Petersen graph, which is easy to prove:

Theorem 38.1. *Let G be the Petersen graph and let C be a 5-circuit in G. Then the perfect matching lattice consists of all integer vectors x in the perfect matching space with $x(EC)$ even.*

Proof. Inspection of the Petersen graph (cf. Figure 38.1) shows that each edge of G is contained in exactly two perfect matchings, that (hence) G has exactly six perfect matchings, that any two perfect matchings intersect each other in exactly one edge, and that each perfect matching intersects EC in an even number of edges.

Let $M_0 := \delta(VC)$ (the set of edges intersecting VC in one vertex). Then M_0 is a perfect matching of G. Let $M_1, \ldots, M_5$ be the five other perfect matchings of G. So each of the M_i intersects M_0 in one edge.

By adding appropriate integer multiples of $\chi^{M_1}, \ldots, \chi^{M_5}$ to x we can achieve that $x_e = 0$ for each $e \in M_0$. As x is in the perfect matching space, we know that there exists a number t such that $x(\delta(v)) = t$ for each vertex v. Hence, as $|EC|$ is odd, $x_e = \frac{1}{2}t$ for all $e \in EC$; similarly, for each edge e in the 5-circuit vertex-disjoint from C one has $x_e = \frac{1}{2}t$. As $x(EC)$ is even, we know that $\frac{5}{2}t$ is even, hence $\frac{1}{2}t$ is even. Now the vector

(38.2) $$y := \chi^{M_1} + \cdots \chi^{M_5} - \chi^{M_0}$$

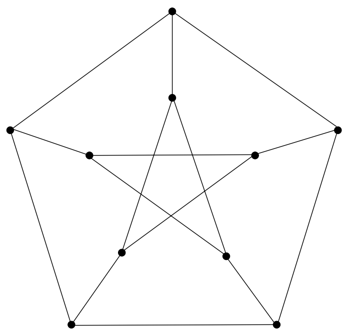


Figure 38.1
The Petersen graph

satisfies $y_e = 0$ for $e \in M_0$ and $y_e = 2$ for $e \notin M_0$. Hence x is an integer multiple of y, proving that x belongs to the perfect matching lattice of G. ∎

38.3. A further fact on the Petersen graph

In the proof of the characterization of the perfect matching lattice, we need a further, technical fact on the Petersen graph.

Let $G = (V, E)$ be a graph and let $b : V \to \mathbb{Z}_+$. Recall that a *b-factor* is a subset F of E with $\deg_F(v) = b(v)$ for each $v \in V$.

Theorem 38.2. *Let $G = (V, E)$ be the Petersen graph and let C be a 5-circuit in G. Let $b : V \to \mathbb{Z}_+$ be such that*

- *either there exists a $u \in V$ with $b(u) = 3$ and $b(v) = 1$ for all $v \neq u$,*
- *or there exist distinct $u, u' \in V$ with $b(u), b(u') \in \{0, 2\}$ and $b(v) = 1$ for all $v \neq u, u'$, such that if $b(u) = b(u') = 0$, then u and u' are nonadjacent.*

Then there exist b-factors F and F' such that $|F \cap EC|$ and $|F' \cap EC|$ have different parities.

Proof. By induction on $b(V)$. If $b(u) = b(u') = 0$ for some distinct $u, u' \in V$, then u and u' are nonadjacent. Let x be the common neighbour of u and u' and let y be the neighbour of x distinct from u and u'. Then $G - x - N(x)$ forms a 6-circuit (by inspection — cf. Figure 38.1), D say. Split ED into two matchings, M and M'. Adding edge xy to M and M' gives b-factors as required since EC intersects ED in an odd number of edges (as $ED = EG \setminus \delta(N(x))$, and $|EC|$ is odd and $|EC \cap \delta(N(x))|$ is even).

If $b(u') = 2$, choose a neighbour u'' of u' with u'' different from and nonadjacent to u. Define $b'(u'') := 0$, $b'(u) := b(u)$, and $b'(v) := 1$ for all other vertices. By induction, there exist b'-factors F and F' such that F and

F' intersect EC in different parities. Adding edge $u'u''$ to F and F' gives b-factors as required.

If $b(u) = 3$ for some $u \in V$, we can choose any neighbour u' of u, define $b'(u) := 2$, $b'(u') := 0$, and $b'(v) := 1$ for all $v \neq u, u'$, and apply induction as above. ∎

38.4. Various useful observations

In this section we prove a few easy facts that turn out to be useful.

Let $G = (V, E)$ be a graph and let $U \subseteq V$. Recall that G/U denotes the graph obtained from G by contracting U to one vertex, which vertex we will call U. In the obvious way, we will consider the edge set of G/U as a subset of the edge set of G. Hence, for any $x \in \mathbb{R}^E$, we can speak of the *projection* of x to the edges of G/U.

We now characterize when G/U is a brick if G is a brick:

Theorem 38.3. *Let $G = (V, E)$ be a brick and let $U \subseteq V$. Then G/U is a brick if and only if $G - U$ is 2-connected and factor-critical.*

Proof. Necessity being easy, we prove sufficiency.

First, let $G - U$ be 2-connected. Then G/U is 3-connected, for suppose that vertices u and u' of G/U form a 2-vertex-cut of G/U. If both u and u' are different from vertex U of G/U, then u, u' would also form a 2-vertex-cut of G, contradicting the 3-connectivity of G. If, say, u' is equal to vertex U of G/U, then u is a cut vertex of $G - U$, contradicting the 2-connectivity of $G - U$.

Second, let $G - U$ be factor-critical. To see that G/U is bicritical, let B be a nontrivial barrier of G/U. If B does not contain vertex U of G/U, then B would also be a nontrivial barrier of G, contradicting the bicriticality of G. If B contains vertex U, then $G - U$ is not factor-critical. ∎

Maximal barriers leave factor-critical components:

Theorem 38.4. *Let $G = (V, E)$ be a graph with a perfect matching and let B be a maximal barrier. Then each component K of $G - B$ is factor-critical.*

Proof. Suppose not. Then K has a nonempty subset B' such that $(G[K]) - B'$ has at least $|B'| + 1$ odd components. Hence $B \cup B'$ is a barrier of G, contradicting the maximality of B. ∎

We note that

(38.3) if $B_1, \ldots, B_k$ are the maximal nontrivial barriers of a graph $G = (V, E)$, having a perfect matching, then for each $u \in V \setminus (B_1 \cup \cdots \cup B_k)$, the graph $G - u$ is factor-critical.

In bicritical graphs, nonempty stable sets have many neighbours (a *neighbour* of S is a vertex not in S adjacent to at least one vertex in S):

Theorem 38.5. *Let $G = (V, E)$ be bicritical with $|V| \geq 4$. Then any nonempty stable set S has at least $|S| + 2$ neighbours.*

Proof. Suppose that $|N(S)| \leq |S| + 1$. Since $|V \setminus S| \geq |S|$ (as G has a perfect matching), we know $|V \setminus S| \geq 2$. Hence we can choose two vertices $v, v' \in V \setminus S$ such that $|N(S) \setminus \{v, v'\}| < |S|$. This however contradicts the fact that $G - v - v'$ has a perfect matching, since each vertex in S should be matched to a vertex in $N(S)$. ∎

It will also be useful to make the following observation:

Theorem 38.6. *Let $G = (V, E)$ be a graph and let U be an odd subset of V, such that for each edge $e \in \delta(U)$ there is a perfect matching M_e with $M_e \cap \delta(U) = \{e\}$. Define $G_1 := G/\overline{U}$ and $G_2 := G/U$, and let $x \in \mathbb{Z}^E$. If, for each $i = 1, 2$, the projection of x to EG_i belongs to the perfect matching lattice of G_i, then x belongs to the perfect matching lattice of G.*

Proof. Let x' and x'' be the projections of x to EG_1 and to EG_2, respectively. Since x' belongs to the perfect matching lattice of G_1, there exist perfect matchings $M'_1, \ldots, M'_{k'}$ and $N'_1, \ldots, N'_{l'}$ of G_1 such that

(38.4) $$x' = \sum_{i=1}^{k'} \chi^{M'_i} - \sum_{j=1}^{l'} \chi^{N'_j}.$$

Similarly, there exist perfect matchings $M''_1, \ldots, M''_{k''}$ and $N''_1, \ldots, N''_{l''}$ of G_2 such that

(38.5) $$x'' = \sum_{i=1}^{k''} \chi^{M''_i} - \sum_{j=1}^{l''} \chi^{N''_j}.$$

Consider any $e \in \delta(U)$. Then $x'_e = x_e = x''_e$. Hence, using the projections of M_e to EG_1 and to EG_2, we can assume that

(38.6) $$|\{i = 1, \ldots, k' \mid e \in M'_i\}| = |\{i = 1, \ldots, k'' \mid e \in M''_i\}| \text{ and}$$
$$|\{j = 1, \ldots, l' \mid e \in N'_j\}| = |\{j = 1, \ldots, l'' \mid e \in N''_j\}|,$$

since we can add the projection of M_e to EG_1 to both sums in (38.4), if the number of i with $e \in M'_i$ is less than the number of i with $e \in M''_i$; similarly, if it would be more.

We can do this for each $e \in \delta(U)$, to obtain (38.6) for each $e \in \delta(U)$. It implies that $k' = k''$ and $l' = l''$. It also implies that we can 'match' the M'_i and M''_i in common edges in $\delta(U)$. That is, by permuting indices, we can assume that M'_i and M''_i have an edge in $\delta(U)$ in common, for each $i = 1, \ldots, k'$. In other words, each $M'_i \cup M''_i$ is a perfect matching of G. Similarly, we can assume that each $N'_j \cup N''_j$ is a perfect matching of G. Then

$$x = \sum_{i=1}^{k'} \chi^{M_i' \cup M_i''} - \sum_{j=1}^{l'} \chi^{N_j' \cup N_j''}. \tag{38.7}$$

So x belongs to the perfect matching lattice of G. ∎

38.5. Simple barriers

In this section, we fix a brick $G = (V, E)$ and an edge e such that $G - e$ is matching-covered, and study barriers of $G - e$. In particular we focus on 'simple' barriers of $G - e$. They play an important role in the proof of the characterization of the perfect matching lattice.

For any $B \subseteq V$, let $I(B)$ denote the set of isolated vertices of $G - e - B$ and let $K(B)$ denote the set of nonisolated vertices of $G - e - B$. Then B is called a *simple barrier* of $G - e$ if $|I(B)| = |B| - 1$. So a simple barrier is a barrier of $G - e$, and hence a stable set (as $G - e$ is matching-covered). Note that each singleton is a simple barrier.

For any simple barrier B of $G-e$, $K(B)$ is an odd component of $G-e-B$, since $G-e$ is matching-covered and connected. (Trivially, $|K(B)|$ is odd, since $|V|$ is even and $|I(B)| = |B| - 1$. If $K(B)$ would not be connected, let L be an odd component of $K(B)$ and let f be an edge connecting $K(B) \setminus L$ and B. Let M be a perfect matching of $G - e$ containing f. Necessarily some edge in M leaves L. But then more that one edge in M connects $K(B)$ and B, and also each vertex in $I(B)$ is matched to B, while $|I(B)| = |B| - 1$, a contradiction.)

Since a barrier B of $G - e$ with $|B| \geq 2$ is not a barrier of G (since G is bicritical), e necessarily connects two odd components of $G - e - B$. If B is a simple barrier of $G - e$ with $|B| \geq 2$, then e connects $K(B)$ with some vertex $v_1 \in I(B)$. (G has a perfect matching M intersecting $\delta(K(B))$ in at least three edges, and hence M contains an edge connecting $K(B)$ and $I(B)$. This edge must be e.)

Then the perfect matchings M of G are of two types:

(38.8) M does not contain e, in which case M matches B with the components of $G - e - B$,
or M contains e, in which case two of the edges in M leaving B are incident with $K(B)$, and the other edges in M leaving B are incident with $I(B) \setminus \{v_1\}$.

We now give some further easy properties of simple barriers. Recall that a subset U of the vertex set V of a graph G is called *matchable* if $G[U]$ has a perfect matching.

Theorem 38.7. *Let $G = (V, E)$ be a brick, let $e \in E$ be such that $G - e$ is matching-covered and let B be a simple barrier of $G - e$. Let $e = v_1 v_2$ with $v_1 \in B \cup I(B)$ and $v_2 \in K(B)$. Then:*

(38.9) (i) *if* $|B| \geq 2$*, then* $v_1 \in I(B)$*;*
(ii) *for any* $u \in B$*, the set* $(B - u) \cup I(B)$ *is matchable;*
(iii) *for any distinct* $u, u' \in B$*, the set* $(B - u - u') \cup (I(B) - v_1)$ *is matchable;*
(iv) $G - e/K(B)$ *is matching-covered;*
(v) $G[B \cup I(B)]$ *is connected;*
(vi) *any cut vertex* v *of* $G[B \cup I(B)]$ *belongs to* $I(B)$*;*
(vii) *if* $Y \subseteq I(B)$ *and* $G[B \cup I(B)] - Y$ *has at least* $|Y| + 1$ *components, then it contains precisely* $|Y| + 1$ *components and any component of* $G[B \cup I(B)] - Y$ *not containing* v_1 *consists of a singleton vertex in* B*.*

Proof. Since all assertions are trivial if $|B| = 1$, we can assume that $|B| \geq 2$. We saw above that then $v_1 \in I(B)$, proving (i).

(ii) follows from the fact that $G - u - v_2$ has a perfect matching. Similarly, (iii) follows from the fact that $G - u - u'$ has a perfect matching, necessarily containing e. (iv) is directly implied by the fact that $G - e$ is matching-covered, and (v) follows from (ii).

To see (vi), assume that $v \in B$. Choose a component K of $G[B \cup I(B)] - v$ not containing v_1. Since $(B - v) \cup I(B)$ is matchable by (ii), $|K \cap B| = |K \cap I(B)|$. Choose $v' \in K \cap B$. Then $(B - v - v') \cup (I(B) - v_1)$ is matchable by (iii). However, $|K \cap B \setminus \{v'\}| < |K \cap I(B)|$, a contradiction. This proves (vi).

To prove (vii), let α be the number of components of $G[B \cup I(B)] - Y$ containing v_1, let β be the number of other components intersecting $I(B)$, and let γ be the number of other components (hence each consisting of a singleton vertex in B). So $\alpha + \beta + \gamma \geq |Y| + 1$. Now by Theorem 38.5, each component K satisfies

$$|K \cap B| \geq |K \cap I(B)| + 1. \tag{38.10}$$

Indeed, if $K \cap I(B) = \emptyset$, this is trivial. If $K \cap I(B) \neq \emptyset$, then by Theorem 38.5, $|K \cap I(B)| + 2 \leq N(K \cap I(B))| \leq |K \cap B| + 1$, as $N(K \cap I(B)) \subseteq (K \cap B) \cup \{v_2\}$. This proves (38.10).

Moreover,

$$\text{if } v_1 \notin K \text{ and } K \cap I(B) \neq \emptyset, \text{ then } |K \cap B| \geq |K \cap I(B)| + 2, \tag{38.11}$$

since then $N(K \cap I(B)) \subseteq K \cap B$.

(38.10) and (38.11) imply

$$\begin{aligned} \alpha + 2\beta + \gamma &\leq \sum_K (|K \cap B| - |K \cap I(B)|) = |B| - |I(B) \setminus Y| \\ &= |Y| + 1 \leq \alpha + \beta + \gamma, \end{aligned} \tag{38.12}$$

where K ranges over the components of $G[B \cup I(B)] - Y$. Hence $\beta = 0$, and (vii) follows. ∎

We next consider the case where v_2 is a cut vertex of $G[K(B)]$.

Theorem 38.8. *Let $G = (V, E)$ be a brick and let $e = v_1v_2$ be an edge such that $G - e$ is matching-covered. Let B be a simple barrier of $G - e$ with $v_1 \in I(B)$ and $v_2 \in K(B)$. Let Z be a union of components of $G[K(B)] - v_2$, with $Z \neq K(B) - v_2$. Then $G/(Z \cup \{v_2\})$ is matching-covered and has exactly one brick in its brick decomposition.*

Proof. Define $U := Z \cup \{v_2\}$ and $L := K(B) \setminus U$. Note that L is matchable, since $G - v - v'$ has a perfect matching for some $v, v' \in B$ (necessarily containing e and containing no edge connecting $K(B)$ and B). So $|L|$ is even.

We first show that

(38.13) G/U is matching-covered.

Consider first any perfect matching M of $G - e$. Then M has exactly one edge leaving $B \cup I(B)$. Hence M has exactly one edge leaving U (since if there were at least three, then at least two of them should leave $B \cup I(B)$). So M gives a perfect matching in G/U. Since $G - e$ is matching-covered, this implies that each edge of G/U except the image of e is contained in a perfect matching of G/U.

As $L \neq \emptyset$ and $|L|$ is even, G has a perfect matching M with at least three edges leaving $L \cup \{v_2\}$. So it contains at least two edges connecting L and B. Hence M contains e, and all other edges leaving $B \cup I(B)$ connect it with L. So the image of M is a perfect matching in G/U containing the image of e. This shows (38.13).

To see that G/U has only one brick in its brick decomposition, choose a counterexample with $|B|$ as small as possible. This implies:

(38.14) $|N(X) \cap I(B)| > |X|$ for each nonempty subset X of $B \setminus N(L)$.

Assume that this is not the case. Since $|B \cap N(L)| \geq 2$ (as G is 3-connected), we know $|X| \leq |B| - 2$, and so $|N(X) \cap I(B)| \leq |B| - 2 = |I(B)| - 1$, implying $I(B) \not\subseteq N(X)$. Each neighbour of $I(B) \setminus N(X)$ belongs to $(B \setminus X) \cup \{v_2\}$, as there is no edge connecting X and $I(B) \setminus N(X)$. So, using Theorem 38.5,

(38.15)
$$|B| - |X| = |B \setminus X| \geq |N(I(B) \setminus N(X))| - 1 \geq |I(B) \setminus N(X)| + 1 = |B| - |N(X) \cap I(B)|,$$

implying $|N(X) \cap I(B)| = |X|$ and $v_1 \notin N(X)$. Define $B' := B \setminus X$. Then B' is a simple barrier of $G - e$ again, with $I(B') = I(B) \setminus N(X)$ and $K(B') = K(B) \cup N(X) \cup X$.

Let S be the union of X, $N(X) \cap I(B)$, and the contracted vertex U of G/U. Then each perfect matching of G/U has exactly one edge leaving S (as X is matched to $(I(B) \cap N(X)) \cup \{U\}$ in G/U, since $X \cap N(L) = \emptyset$). So S determines a tight cut in G/U. As $G/\overline{S}$ is bipartite, it suffices to show that the brick decomposition of $G/U/S$ contains exactly one brick.

Since $X \cap N(L) = \emptyset$, L is a union of components of $G[K(B')] - v_2$. Then

(38.16) $G/U/S = G/(K(B') \setminus L) \cup \{v_2\}$.

Hence, by the minimality of B, G/U has a brick decomposition with exactly one brick. This shows (38.14).

We finally derive from (38.14) that G/U has only one brick in its brick decomposition, in fact, that it *is* a brick — equivalently that $G-U$ is 2-connected and factor-critical (Theorem 38.3).

Assume that $G-U$ is not 2-connected, and let v be a cut vertex of $G-U$. Then each component of $G-U-v$ intersects $B \cup I(B)$ as G is 3-connected. Hence v is a cut vertex of $G[B \cup I(B)]$. So Theorem 38.7(vi) applies. In particular, $v \in I(B)$.

Since each component of $G[L]$ is adjacent to at least two vertices in B (since G is 3-connected), we know by Theorem 38.7(vii) that all vertices in B adjacent to L belong to the same component of $G[B \cup I(B)] - v$ as v_1. Any other component consists of one vertex, w say, in B. But then this contradicts (38.14), taking $X = \{w\}$. So $G-U$ is 2-connected.

To show that $G-U$ is factor-critical, suppose to the contrary that there exists a nonempty subset Y of $\overline{U}$ such that $G-U-Y$ has at least $|Y|+1$ odd components.

Then $Y \subseteq I(B)$. Otherwise choose $v \in Y \setminus I(B)$. So $v \in L \cup B$. Then $G-U-v$ has no perfect matching. However, as G is bicritical, $G-v-v_2$ has a perfect matching M. Then the restriction of M to $\overline{U}$ is a perfect matching of $G-U-v$, a contradiction. So $Y \subseteq I(B)$.

Each component of $G-U-Y$ containing a component of L has at least two elements in B (since G is 3-connected). So $G[B \cup I(B)] - Y$ has precisely $|Y|+1$ components. Hence it has $|Y|$ singleton components in B, without neighbours in L (by Theorem 38.7(vii)). Let X be the union of these components. Each neighbour y of any $x \in X$ with $y \notin U$ belongs to Y. So $|X| \geq |Y| \geq |N(X) \cap I(B)|$, contradicting (38.14). ∎

We next consider *pairs* of simple barriers B_1, B_2. The following auxiliary theorem is of special interest for disjoint simple barriers B_1 and B_2 of $G-e$ where B_2 intersects $I(B_1)$.

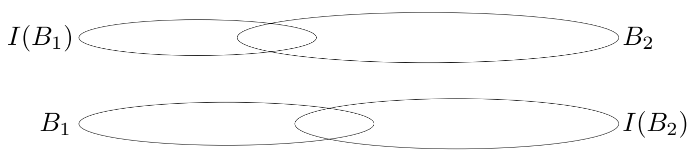


Figure 38.2

Theorem 38.9. *Let $G = (V, E)$ be a brick and let $e = v_1v_2 \in E$ be such that $G-e$ is matching-covered. Let B_1 and B_2 be disjoint simple barriers of $G-e$ with $v_1 \in I(B_1)$ and $v_2 \in I(B_2)$. Then*

(38.17) (i) $I(B_1) \cap I(B_2) = \emptyset$;
(ii) $B_1 \cup I(B_2)$ *and* $B_2 \cup I(B_1)$ *are stable sets;*
(iii) $|B_1 \cap I(B_2)| = |B_2 \cap I(B_1)|$;
(iv) $B_2 \setminus I(B_1)$ *is again a simple barrier of* $G - e$, *with* $I(B_2 \setminus I(B_1)) = I(B_2) \setminus B_1$.

Proof. (i) follows from the fact that all neighbours of any $u \in I(B_1) \cap I(B_2)$ belong to $B_1 \cap B_2 = \emptyset$. Since $N(I(B_2)) \subseteq B_2 \cup \{v_1\}$, which is disjoint from B_1, we have that $B_1 \cup I(B_2)$ is a stable set. Similarly, $B_2 \cup I(B_1)$ is a stable set, implying (ii).

Since $I(B_2) \setminus B_1 \subseteq I(B_2 \setminus I(B_1))$, we know

(38.18)
$$\begin{aligned}&|I(B_2)| - |I(B_2) \cap B_1| = |I(B_2) \setminus B_1| \leq |I(B_2 \setminus I(B_1))|\\ &\leq |B_2 \setminus I(B_1)| - 1 = |B_2| - 1 - |B_2 \cap I(B_1)|\\ &= |I(B_2)| - |B_2 \cap I(B_1)|.\end{aligned}$$

So $|B_2 \cap I(B_1)| \leq |B_1 \cap I(B_2)|$, and hence by symmetry $|B_2 \cap I(B_1)| = |B_1 \cap I(B_2)|$, and we have equality throughout in (38.18). This gives (iii) and (iv). ∎

The last auxiliary theorem in this section reads:

Theorem 38.10. *Let G be a brick and let $e = v_1v_2$ be an edge of G with $G-e$ matching-covered. Let B_1 and B_2 be simple barriers of $G-e$, and define $J_i := B_i \cup I(B_i)$ for $i = 1, 2$, with $v_1 \in J_1$ and $v_2 \in J_2$, and $X := V \setminus (J_1 \cup J_2)$. Assume that $J_1 \cap J_2 = \emptyset$, and that, for each $u \in X$, $G-e-u$ is factor-critical and $G-u/J_1$ and $G-u/J_2$ are 2-connected. Then if $G-e$ has a 2-vertex-cut separating J_1 and J_2, it has a 2-vertex-cut $\{u, u'\}$ separating J_1 and J_2 such that for some component K of $G-e-u-u'$, both $G/(K \cup \{u\})$ and $G/\overline{K \cup \{u\}}$ are bricks.*[21]

Proof. Note that if $\{u, u'\}$ is $J_1 - J_2$ separating in $G-e$ (which by definition implies that $u, u' \notin J_1 \cup J_2$), then $G-e-u-u'$ has a perfect matching (by the assumption in the theorem). Moreover, since G is 3-connected, $G-u-u'$ is connected. Hence $G-e-u-u'$ has exactly two components, one containing J_1 and one containing J_2. We will apply Theorem 38.3.

We first show:

(38.19) Let $\{u, u'\}$ be $J_1 - J_2$ separating in $G-e$ and let K be a component of $G-e-u-u'$. Then the graph $G[K \cup \{u\}]$ is factor-critical.

By symmetry, we may assume that $J_1 \subseteq K$. Define $S := K \cup \{u\}$. Choose a vertex $v \in S$. We prove that $G[S] - v$ has a perfect matching. If $v = u$, then $G[S] - v = G[K]$ has a perfect matching (as K is a component of $G-e-u-u'$). So let $v \neq u$. As $G-e-u'$ is factor-critical by the assumption

[21] It is important to note that it is not concluded that also $G/(K \cup \{u'\})$ and $G/\overline{K \cup \{u'\}}$ are bricks.

in the theorem, $G - e - u' - v$ has a perfect matching M. Since $|K|$ is even, the edge in M incident with u, connects u with K. So M contains a matching spanning $S \setminus \{v\}$. This proves (38.19).

In order to prove that $G[K \cup \{u\}]$ is 2-connected, we need a special kind of 2-vertex-cut and a special order of the components:

(38.20) there exist a pair u, u' separating J_1 and J_2 in $G - e$ and components $K \supseteq J_1$ and $L \supseteq J_2$ of $G - e - u - u'$ such that for each $v \in K \setminus J_1$, $\{u', v\}$ does not separate J_1 and J_2 in $G - e$ and for each $v \in L \setminus J_2$, $\{u, v\}$ does not separate J_1 and J_2 in $G - e$.

To prove this, let $\{u, u'\}$ be a 2-vertex-cut separating J_1 and J_2 in $G-e$. Let K and L be the components of $G-e-u-u'$ containing J_1 and J_2, respectively. We choose u and u' such that L is minimal. Then by the minimality of L, for each $v \in L \setminus J_2$, neither $\{u, v\}$ nor $\{u', v\}$ separates J_1 and J_2 in $G - e$.

If (38.20) does not hold, then there exist $v, v' \in K \setminus J_2$ such that $\{u, v\}$ and $\{u', v'\}$ are vertex-cuts in $G-e$, each separating J_1 and J_2. Let Y be the component of $G-e-u-v$ not containing u'. Since $N_{G-e}(Y) \subseteq \{u, v\}$, we know that $v_1 \in Y$ and hence $J_1 \subseteq Y$. Let Y' be the component of $G-e-u'-v'$ not containing u. Again $J_1 \subseteq Y'$. Hence $J_1 \subseteq Y \cap Y'$. Now $N_{G-e}(Y \cap Y') \subseteq \{v, v'\}$ (since $N_{G-e}(Y \cap Y') \subseteq N_{G-e}(Y) \cup N_{G-e}(Y') \subseteq \{u, u', v, v'\}$; but u' is not a neighbour of $Y \cap Y'$ since u' is not in component Y of $G-e-u-v$; similarly for u). This implies $v \neq v'$. Hence $v' \in Y$.

Let A be the component of $G - e - u - v$ different from Y, and let A' be the component of $G - e - u' - v'$ different from Y'. Then $Y' \cap A = \emptyset$. Indeed, $N(Y' \cap A) \subseteq N(Y') \cup N(A) \subseteq \{u, v, u', v'\}$. Moreover, $u, v' \notin N(Y' \cap A)$, since $u \in A'$ and $v' \in Y$. So $|N(Y' \cap A)| \leq 2$, implying $Y' \cap A = \emptyset$ by the 3-connectivity of G.

Similarly, $K \cap A \cap A' = \emptyset$. Indeed, $N(K \cap A \cap A') \subseteq N(K) \cup N(A) \cup N(A') = \{u, v, u', v'\}$. Moreover, $v, v' \notin N(K \cap A \cap A')$, since $v \in Y'$ and $v' \in Y$. So $|N(K \cap A \cap A')| \leq 2$, implying $K \cap A \cap A' = \emptyset$.

So $K \cap A$ intersects neither Y' nor A', hence $K \cap A \subseteq \{u', v'\}$. However, $u' \notin K$ and $v' \notin A$. So $K \cap A = \emptyset$. Hence $K \subseteq Y \cup \{v\}$. So $Y = K \setminus \{v\}$, implying that $|Y|$ is odd, a contradiction (since $G - e - u - v$ has a perfect matching). This proves (38.20).

Let u, u' be as in (38.20). By symmetry, it suffices to show:

(38.21) $G[K \cup \{u\}]$ is 2-connected.

Let $S := K \cup \{u\}$. Suppose that there exists a $v \in S$ with $G[S \setminus \{v\}]$ disconnected. Let Z be a component of $G[S \setminus \{v\}]$ not containing v_1, and let Y be any other component. If $u \notin Z$, then $N(Z) \subseteq \{v, u'\}$, contradicting the 3-connectivity of G. So $u \in Z$.

So $u \notin Y$, and hence $N(Y) \subseteq \{u', v, v_2\}$, implying by the 3-connectivity of G, that $v_1 \in Y$. So $N_{G-e}(Y) = \{u', v\}$. If $v \notin J_1$, then $J_1 \subseteq Y$ (as $G[J_1]$ is connected), implying that $\{u', v\}$ is $J_1 - J_2$ separating in $G-e$, contradicting the condition in (38.20). So $v \in J_1$.

If $Y \not\subseteq J_1$, then $Y \setminus J_1$ has only two neighbours in G/J_1: J_1 and u', contradicting the fact that $G - u'/J_1$ is 2-connected (by the condition in the theorem). So $Y \subseteq J_1$.

Let M be a perfect matching in $G - u' - v_1$. So M intersects $\delta(J_1)$ in exactly two edges (since $|I(B_1) \setminus \{v_1\}| = |B_1| - 2$ and $u' \in K(B_1)$, as $u' \notin J_1$). If $G[J_1 \setminus \{v\}]$ is connected, then $Y = J_1 \setminus \{v\}$. Then M contains an edge leaving J_1 and not incident with v. This contradicts the fact that $N_{G-e}(Y) \subseteq \{u', v\}$ and that M does not cover u'.

So v is a cut vertex of $G[J_1]$, and hence by Theorem 38.7(vi), v belongs to $I(B_1)$. Now $v \neq v_1$, since $v_1 \in Y$. By Theorem 38.7(vii), $G[J_1 \setminus \{v\}]$ has two components, one containing v_1 and one consisting only of some neighbour, w say, of v. So $Z \cap J_1 = \{w\}$ and $|(Y \setminus \{v_1\}) \cup \{v\}|$ is even. Then M contains a matching with union $(Y \setminus \{v_1\}) \cup \{v\}$. Hence at most one edge in M leaves J_1, a contradiction. This shows (38.21). ∎

38.6. The perfect matching lattice of a brick

We now prove the theorem of Lovász [1987]:

Theorem 38.11. *Let $G = (V, E)$ be a brick different from the Petersen graph. Then the perfect matching lattice of G is equal to the set of integer vectors in the perfect matching space of G.*

Proof. We choose a counterexample with $|V| + |E|$ minimal. Let x be an integer vector in the perfect matching space of G that is not in the perfect matching lattice of G. We can assume that $x(\delta(v)) = 0$ for each vertex v (this can be achieved by adding an appropriate integer multiple of χ^M to x, for some perfect matching M in G).

Claim 1. Let $\delta(U)$ be an odd cut in G such that both G/U and $G/\overline{U}$ are matching-covered and have exactly one brick in their brick decompositions. Then there exist no perfect matchings M and N of G with $|M \cap \delta(U)| - |N \cap \delta(U)| = 2$.

Proof of Claim 1. Suppose to the contrary that such perfect matchings M, N exist. In particular, $|U|, |\overline{U}| \geq 3$. As $x(\delta(U))$ is even (since $x(\delta(v))$ is even for each vertex v), by adding an appropriate integer multiple of $\chi^M - \chi^N$ to x we can achieve that $x(\delta(U)) = 0$.

Let x' and x'' be the projections of x to the edges of $G/\overline{U}$ and G/U, respectively. Let $H := G/\overline{U}$.

Consider any minimal subset W of U, such that $|W| \geq 3$ and such that $\delta(W)$ is a tight cut of H. (Such a set exists, since $\delta(U)$ is tight in H.) Since H has exactly one brick in its brick decomposition, we know that $H/\overline{W}$ or H/W is bipartite and matching-covered. If $H/\overline{W}$ is bipartite and matching-covered, the colour class of $H/\overline{W}$ not containing vertex $\overline{W}$ would be a nontrivial barrier in G. This contradicts the fact that G is a brick.

So H/W is bipartite and matching-covered. Hence the projection of x' to the edges of H/W belongs to the perfect matching lattice of H/W. So (by Theorem 38.6) the projection y of x' to the edges of $I := H/\overline{W}$ is not in the perfect matching lattice of I.

By the minimality of W, I is a brick. Since y is not in the perfect matching lattice of I, by the minimality of $|V| + |E|$, I is the Petersen graph and has a 5-circuit disjoint from vertex $\overline{W}$ of I with $y(EC)$ odd.

As $\delta(W)$ is not tight in G (since G is a brick), G has a perfect matching L satisfying $|L \cap \delta(W)| \geq 3$, and hence $|L \cap \delta(W)| = 3$ (since I is the Petersen graph). Then by Theorem 38.2 (defining $b(\overline{W}) := 3$ and $b(v) := 1$ for each vertex $v \neq \overline{W}$ of I), we can modify L on the edges of I not incident with $\overline{W}$ to obtain a perfect matching L' of G such that the intersections of L and L' with EC have different parities. Resetting $x := x + \chi^L - \chi^{L'}$ we achieve that $x(EC)$, and hence $x'(EC)$, is even.

Hence the projection of the new x on the edges of $G/\overline{U}$ is in the perfect matching lattice of $G/\overline{U}$. We can perform similar resettings to achieve that the projection of the new x on the edges of G/U is in the perfect matching lattice of G/U. Then the new x, and hence also the original x, belongs to the perfect matching lattice of G, by Theorem 38.6. This contradicts our assumption. *End of Proof of Claim 1*

There exists an edge e with $G - e$ matching-covered

To see this, we first show:

Claim 2. There are no edges e and f such that $G - e - f$ is matching-covered and bipartite.

Proof of Claim 2. Suppose that such e and f exist. As $G - e - f$ is matching-covered, the colour classes of $G - e - f$ have the same size, and as G is matching-covered and nonbipartite, e is spanned by one of the colour classes, and f by the other.

Let M be a perfect matching in G containing e and f and let N be a perfect matching in G not containing e and f. By adding an appropriate integer multiple of $\chi^M - \chi^N$ to x we can achieve that $x_e = 0$. Since x is in the perfect matching space of G, this implies that $x_f = 0$. By Corollary 20.12a, the restriction of x to $G - e - f$ is in the perfect matching lattice of $G - e - f$. Hence x belongs to the perfect matching lattice of G, contradicting our assumption. *End of Proof of Claim 2*

This gives:

Claim 3. There is an edge e such that $G - e$ is matching-covered.

Proof of Claim 3. For each edge e, let $\mathcal{M}_e$ denote the collection of perfect matchings of G containing e. Choose any edge e with $\mathcal{M}_e$ inclusionwise minimal. We prove that $G-e$ is matching-covered.

Suppose that $G-e$ is not matching-covered. Hence there is an edge $f \neq e$ such that each perfect matching of G containing f, also contains e; that is, $\mathcal{M}_f \subseteq \mathcal{M}_e$. By the minimality of $\mathcal{M}_e$, $\mathcal{M}_f = \mathcal{M}_e$. Hence there is no perfect matching containing exactly one of e, f. We show that

(38.22) $\quad G-e-f$ is bipartite.

As there is no perfect matching containing e but not containing f, by Tutte's 1-factor theorem, there exists a subset B of V spanning e such that $G-f-B$ has more than $|B|-2$ odd components; hence, by parity, at least $|B|$ odd components. As $|B| \geq 2$ and as G is bicritical, f connects two distinct odd components, K_1 and K_2 say, of $G-f-B$. Moreover, as G is bicritical, each component of $G-f-B$ is odd.

We show that $G-e-f$ is bipartite with colour classes B and $W := V \setminus B$. That is, e is the only edge contained in B, and each component of $G-f-B$ is a singleton.

To see this, first assume that some component K of $G-f-B$ is not a singleton. Then $\delta(K)$ is a nontrivial cut, and hence it is not tight. So there exists a perfect matching M with $|M \cap \delta(K)| \geq 3$. If $f \notin M$, then (adding up over all components of $G-f-B$), $|M \cap \delta(B)| \geq |B|+2$, a contradiction. If $f \in M$, then similarly $|M \cap \delta(B)| \geq |B|$, again a contradiction (since $e \in M$).

Second assume that B spans some edge e' different from e. Let M be a perfect matching containing e'. If $f \notin M$, then $|M \cap \delta(B)| \geq |B|$, contradicting the fact that $e' \in M$. If $f \in M$, then $|M \cap \delta(B)| \geq |B|-2$, contradicting the fact that both e and e' belong to M. This shows (38.22).

In particular, any odd circuit in G contains exactly one of e and f. By Claim 2, $G-e-f$ is not matching-covered. Hence there is an edge g such that each perfect matching containing g contains e or f. Hence $\mathcal{M}_g = \mathcal{M}_e = \mathcal{M}_f$. So, as before, each of $G-e-f$, $G-e-g$, $G-f-g$ is bipartite. Hence each odd circuit in G contains exactly one edge from each pair taken from e, f, g, a contradiction. *End of Proof of Claim 3*

Each maximal barrier of $G - e$ is simple

We fix an edge e with $G-e$ matching-covered. Let e connect vertices v_1 and v_2.

Claim 4. Let B be a maximal barrier of $G-e$. Then B is simple and $G/\overline{K(B)}$ is a brick.

Proof of Claim 4. As the claim is trivial if $|B|=1$, we can assume $|B| \geq 2$; that is, B is nontrivial. Since G has no nontrivial barrier, B is not a barrier of G, and hence e connects two different components of $G-e-B$.

By Theorem 38.4, each component K of $G - e - B$ is factor-critical. So it suffices to show (by Theorem 38.3) that $G[K(B)]$ is 2-connected. In other words, $G - e - B$ has precisely one block[22].

Let $\mathcal{K}$ denote the collection of components of $G - e - B$, and let $\mathcal{L}$ denote the collection of blocks of $G - e - B$. For $K \in \mathcal{K}$, let $\mathcal{L}_K$ denote the set of blocks of $G[K]$.

It is useful to state the following formulas (38.23) and (38.25). For any perfect matching M of G and any $K \in \mathcal{K}$ one has

(38.23) $$\sum_{L \in \mathcal{L}_K} (|M \cap \delta(L)| - 1) = |M \cap \delta(K)| - 1.$$

This can be shown inductively as follows. Consider any subsets U' and U'' of a set U of vertices with $U' \cup U'' = U$, $|U' \cap U''| = 1$, and no edge connecting $U' \setminus U''$ and $U'' \setminus U'$. Then $|M \cap \delta(U)| - 1 = (|M \cap \delta(U')| - 1) + (|M \cap \delta(U'')| - 1)$, since

(38.24) $$|M \cap \delta(U')| + |M \cap \delta(U'')| = |M \cap \delta(U' \cup U'')| + |M \cap \delta(U' \cap U'')| = |M \cap \delta(U)| + 1.$$

One also has

(38.25) $$\sum_{K \in \mathcal{K}} (|M \cap \delta(K)| - 1) = 2|M \cap \{e\}|,$$

since

(38.26) $$\sum_{K \in \mathcal{K}} |M \cap \delta(K)| = |M \cap \delta(B)| + 2|M \cap \{e\}| = |B| + 2|M \cap \{e\}| = |\mathcal{K}| + 2|M \cap \{e\}|.$$

Suppose now that the claim is not true — that is, $|\mathcal{L}| \geq 2$. We derive:

(38.27) for each $L \in \mathcal{L}$ and for each edge $f \in \delta(L)$, G has a perfect matching M with $M \cap \delta(L) = \{f\}$.

Indeed, if $f \neq e$, let M be a perfect matching of $G - e$ containing f. By (38.23) and (38.25), M intersects $\delta(L)$ in exactly one edge. So $M \cap \delta(L) = \{f\}$.

Suppose next that $f = e$. As $|\mathcal{L}| \geq 2$ by assumption, there exists a block $L' \neq L$. As G has no tight nontrivial cuts, G has a perfect matching M with $|M \cap \delta(L')| \geq 3$, and hence by (38.23) and (38.25), $|M \cap \delta(L)| = 1$, that is, $M \cap \delta(L) = \{e\}$. This proves (38.27).

Now for each $L \in \mathcal{L}$ there exists a perfect matching M with $|M \cap \delta(L)| \geq 3$, and hence, by (38.23) and (38.25), $|M \cap \delta(L)| = 3$ and $|M \cap \delta(L')| = 1$ for all other $L' \in \mathcal{L}$. Moreover, let N be a perfect matching not containing e. Then adding an appropriate integer multiple of $\chi^M - \chi^N$ to x we can achieve that $x(\delta(L)) = 0$, while $x(\delta(L'))$ does not change for any other $L' \in \mathcal{L}$.

As we can do this for all $L \in \mathcal{L}$, we can assume that

[22] A *block* of a graph H is an inclusionwise maximal set L of vertices with $|L| \geq 2$ and with $G[L]$ 2-connected.

(38.28) $x(\delta(L)) = 0$ for all $L \in \mathcal{L}$.

Since x is in the perfect matching space, with (38.23) this gives that

(38.29) $x(\delta(K)) = 0$ for all $K \in \mathcal{K}$.

Moreover, $x_e = 0$, since

(38.30) $$2x_e = \sum_{K\in\mathcal{K}} x(\delta(K)) - x(\delta(B)) = -x(\delta(B)) = -\sum_{v\in B} x(\delta(v)) = 0.$$

Let H be the matching-covered bipartite graph obtained from $G - e$ by contracting each $K \in \mathcal{K}$ to a vertex. Since $x(\delta(K)) = 0$ for each $K \in \mathcal{K}$ and $x(\delta(v)) = 0$ for each $v \in B$, and since $x_e = 0$, we know from Corollary 20.12a that $x|EH$ is in the perfect matching lattice of H. Now for each $K \in \mathcal{K}$ and for each $f \in \delta(K)$ with $f \neq e$, there exists a matching M in $G - e$ containing f, and hence there is a matching with union $K \setminus \{v\}$, where v is the vertex in K incident with f. We therefore can extend each perfect matching of H to a perfect matching of $G - e$ intersecting each $\delta(K)$ in one edge. This implies that we may assume that $x_f = 0$ for each $f \in \delta(B)$.

Hence each edge f with $x_f \neq 0$ is spanned by some $L \in \mathcal{L}$. Let $\mathcal{L}'$ be the collection of those blocks $L \in \mathcal{L}$ spanning at least one edge f with $x_f \neq 0$. We choose x satisfying all previous assumptions and such that $|\mathcal{L}'|$ is as small as possible.

As each $K \in \mathcal{K}$ is factor-critical, each $L \in \mathcal{L}$ is factor-critical. Hence, by Theorem 38.3,

(38.31) $G/\overline{L}$ is a brick for each $L \in \mathcal{L}$.

Moreover,

(38.32) we can assume that, for each $L \in \mathcal{L}$ with $G/\overline{L}$ the Petersen graph, there is a 5-circuit C in $G[L]$ with $x(EC)$ even.

Indeed, choose any 5-circuit C in $G[L]$, and suppose that $x(EC)$ is odd. Let M be a perfect matching in G with $|M \cap \delta(L)| = 3$. By Theorem 38.2, we can modify M on the edges spanned by L so as to obtain a perfect matching N with $|N \cap EC|$ having parity different from $|M \cap EC|$, and such that M and N coincide for all edges not spanned by L. Now adding $\chi^M - \chi^N$ to x makes $x(EC)$ even, and does not invalidate our previous assumptions. This shows (38.32).

We show next:

(38.33) for each $\mathcal{L}_0 \subseteq \mathcal{L}$ with $x_f = 0$ for each $f \in \delta(\bigcup\mathcal{L}_0)$, one has $\mathcal{L}_0 \subseteq \mathcal{L}'$.

We show this by induction on $|\mathcal{L}_0|$. If $\mathcal{L}_0 = \emptyset$, this is trivial. If $\mathcal{L}_0 \neq \emptyset$, we can choose an $L \in \mathcal{L}_0$ such that L has a vertex v such that each $L' \in \mathcal{L}_0$ with $L' \neq L$ is disjoint from $L \setminus \{v\}$. Hence each $f \in \delta(L)$ with $x_f \neq 0$ is incident with v. By (38.31) and (38.32), $x|E(G/\overline{L})$ is in the perfect matching lattice of $G/\overline{L}$. So

(38.34) $$x|E(G/\overline{L}) = \sum_{M} \lambda_M \chi^M,$$

where M ranges over perfect matchings of $G/\overline{L}$ and where $\lambda_M \in \mathbb{Z}$. Let $\mathcal{M}$ denote the collection of perfect matchings of $G/\overline{L}$ not containing an edge leaving L at v. So if $f \in M \in \mathcal{M}$ and f is incident with $\overline{L}$, then $x_f = 0$. By (38.27), for each $f \in \delta(L)$ we can choose a perfect matching N_f of G/L containing f. Then for each perfect matching M of $G/\overline{L}$, let $\widetilde{M} := M \cup N_f$ where f is the edge of M leaving L. Then, by replacing x by

(38.35) $$x - \sum_{M \in \mathcal{M}} \lambda_M \chi^{\widetilde{M}},$$

x changes only on edges spanned by L, and we achieve that $x_f = 0$ for each edge $f \in \delta(v)$ spanned by L. Hence for $\mathcal{L}_0' := \mathcal{L}_0 \setminus \{L\}$ we have $x_f = 0$ for each $f \in \delta(\bigcup \mathcal{L}_0')$. Therefore, by the induction hypothesis, $\mathcal{L}_0' \subseteq \mathcal{L}'$. So $x_f = 0$ for each $f \in \delta(L)$. Hence, taking the λ_M as above, by replacing x by

(38.36) $$x - \sum_{M} \lambda_M \chi^{\widetilde{M}},$$

where M ranges over all perfect matchings of $G/\overline{L}$, we achieve that $x|E(G/\overline{L}) = \mathbf{0}$. This proves (38.33).

Applying (38.33) to $\mathcal{L}_0 := \mathcal{L}$, we derive that $x = \mathbf{0}$, a contradiction.

End of Proof of Claim 4

We remind that for each maximal nontrivial barrier B of $G - e$ one has $e \in \delta(K(B))$ and:

(38.37) for each perfect matching M of G: $e \in M \iff |M \cap \delta(K(B))| = 3$.

Pairs of simple barriers of $G - e$

Claim 5. Let B_1 and B_2 be simple barriers of $G - e$ and let $J_i := B_i \cup I(B_i)$ (for $i = 1, 2$), with $J_1 \cap J_2 = \emptyset$ and $v_i \in J_i$ (for $i = 1, 2$). Then $H := G - e/J_1/J_2$ is not a brick.

Proof of Claim 5. Suppose that H is a brick. By adding an appropriate integer multiple of $\chi^M - \chi^N$ to x, where M and N are perfect matchings in G containing e and not containing e, respectively, we can achieve $x_e = 0$. Then, since $x(\delta(v)) = 0$ for each vertex v, we have that $x(\delta(J_1)) = x(\delta(J_2)) = 0$. As $G - e/\overline{J_1}$ and $G - e/\overline{J_2}$ are bipartite and as H is a brick, it follows that the respective projections of x belong to the perfect matching space of $G - e/\overline{J_1}$, $G - e/\overline{J_2}$, and H.

As x is not in the perfect matching lattice of G, by Theorem 38.6 at least one of these projections is not in the corresponding perfect matching lattice. As $G-e/\overline{J_1}$ and $G-e/\overline{J_2}$ are bipartite, it follows (as G is a minimal counterexample to Theorem 38.11) that H is the Petersen graph and that $x(EC)$ is odd for some 5-circuit C in H disjoint from vertices J_1 and J_2 of H. Then it suffices to show:

(38.38) G has perfect matchings M and N, each containing e, such that M and N intersect EC in different parities,

since then adding $\chi^M - \chi^N$ to x turns the parity of $x(EC)$.

To prove (38.38), let

(38.39) $X := VG \setminus (J_1 \cup J_2) = K(B_1) \cap K(B_2)$.

So $VH = X \cup \{J_1, J_2\}$. We first show that for $i = 1, 2$:

(38.40) if $|J_i| \geq 3$, and a and b are distinct neighbours of vertex J_i of H with $a, b \in X$, then $\{aJ_i, bJ_i\}$ is the image of a matching in G.

To see this, we can assume that $i = 1$.

If J_1 and J_2 are adjacent vertices of H, then a and b are the only neighbours of J_1 in X. Choose $z \in B_2$. As G is bicritical, $G - v_1 - z$ has a perfect matching M. Then M matches up all vertices in $J_2 \setminus \{z\}$. Moreover, all but two vertices in B_1 are matched with vertices in $I(B_1) \setminus \{v_1\}$. Hence two edges of M connect B_1 and K. So M contains edges connecting a and b with B_1.

If J_1 and J_2 are nonadjacent vertices of H, let z be the vertex distinct from a, b adjacent in H to J_1. Since G is bicritical, $G - v_1 - z$ has a perfect matching M. All but two vertices in B_1 are matched with vertices in $I(B_1) \setminus \{v_1\}$. Since M misses z, M contains edges connecting a and b with B_1. This shows (38.40).

Moreover, we have:

(38.41) if J_1 and J_2 are adjacent vertices in H, and $|J_1| \geq 3$ and $|J_2| \geq 3$, then J_1 has a neighbour a_1 in X, and J_2 has a neighbour a_2 in X, such that $\{a_1J_1, J_1J_2, J_2a_2\}$ is the image of a matching in G.

Let f be an edge of $G-e$ connecting J_1 and J_2. By (38.40), J_1 has a neighbour a_1 in X such that there exists an edge connecting a_1 and J_1 disjoint from f. Similarly, J_2 has a neighbour a_2 in X such that there exists an edge connecting a_2 and J_2 disjoint from f. This gives the a_1 and a_2 required in (38.41).

By Theorem 38.2, we can find subsets F_1 and F_2 of the edge set of H such that for each $j = 1, 2$,

(38.42)
(i) each vertex in X is incident with exactly one edge in F_j,
(ii) for each $i = 1, 2$, if $|J_i| = 1$, then J_i is incident with none of the edges in F_j, and, if $|J_i| \geq 3$, then J_i is incident with exactly two edges in F_j,
(iii) $|F_1 \cap EC|$ and $|F_2 \cap EC|$ have different parities.

(Note that if $|J_1| = |J_2| = 1$, then J_1 and J_2 are not adjacent, as then $J_1 = \{v_1\}$ and $J_2 = \{v_2\}$, $e = v_1v_2$, and $H = G - e$.)

If J_1 and J_2 are adjacent vertices of H and $|J_1| \geq 3$, $|J_2| \geq 3$, we can choose the F_j such that moreover

(38.43) $\quad a_1J_1$, a_2J_2 belong to both F_1 and F_2,

where a_1 and a_2 are as in (38.41). To see this, note that a_1 and a_2 are nonadjacent (as the Petersen graph has no 4-circuit). Then there exist by Theorem 38.2 subsets F_1' and F_2' of the edge set of H such that for each $j = 1, 2$, each vertex of H different from a_1 and a_2 is incident with exactly one edge in F_j', while a_1 and a_2 are not covered by F_j', and such that $|F_1' \cap EC|$ and $|F_2' \cap EC|$ have different parities. Extending the F_j' with the edges a_1J_1 and a_2J_2 gives F_j as required.

By Theorem 38.7(iii), (38.40) and (38.41), F_1 and F_2 are projections of perfect matchings M and N of G containing e, as required in (38.38).

End of Proof of Claim 5

This claim can be sharpened as follows:

Claim 6. Let B_1 and B_2 be simple barriers of $G - e$ and let $J_i := B_i \cup I(B_i)$ (for $i = 1, 2$), with $J_1 \cap J_2 = \emptyset$ and $v_i \in J_i$ (for $i = 1, 2$). Define $X := V \setminus (J_1 \cup J_2)$. If $G - e - u$ is factor-critical for each $u \in X$ and $H := G - e/J_1/J_2$ is bicritical, then $G/J_1/J_2$ has a 2-vertex-cut intersecting $\{J_1, J_2\}$.

Proof of Claim 6. If $G - u/J_1$ is not 2-connected for some $u \in X$, then $\{u, J_1\}$ is a 2-vertex-cut in G/J_1 (since G is 3-connected), hence in $G/J_1/J_2$, as required. So we may assume that $G - u/J_1$ and $G - u/J_2$ are 2-connected for each $u \in X$.

Let H be bicritical. By Claim 5, H is not a brick. Hence H is not 3-connected. Let $\{u, u'\}$ be a 2-vertex-cut of H. If $\{u, u'\}$ intersects $\{J_1, J_2\}$ we are done. So suppose that $\{u, u'\}$ is disjoint from $\{J_1, J_2\}$. Since G is 3-connected and e connects J_1 and J_2, we know that $\{u, u'\}$ separates J_1 and J_2. Hence, by Theorem 38.10, we may assume that the components K and L of $G - e - u - u'$ are such that $G/(K \cup \{u\})$ and $G/\overline{K \cup \{u\}}$ are bricks.

Define $U := K \cup \{u\}$. Then G has a perfect matching M with $|M \cap \delta(U)| \geq 3$, since G has no nontrivial tight cuts. As each edge in $\delta(U) \setminus \{e\}$ is incident with u or u', we know $|M \cap \delta(U)| = 3$. Let $f \in \delta(U) \setminus \{e\}$ and let N be a perfect matching in $G - e$ containing f. Then $|N \cap \delta(U)| = 1$, contradicting Claim 1. *End of Proof of Claim 6*

$G - e$ has exactly two maximal nontrivial barriers

By Corollary 24.11a, we know:

(38.44) $\quad$ any two distinct maximal barriers of $G - e$ are disjoint.

Since each maximal nontrivial barrier B contains $N(v_1)\setminus\{v_2\}$ or $N(v_2)\setminus\{v_1\}$ (as e connects $I(B)$ and $K(B)$), we know that $G-e$ has at most two maximal nontrivial barriers. In fact:

Claim 7. $G-e$ has exactly two maximal nontrivial barriers B_1 and B_2.

Proof of Claim 7. First assume that $G-e$ has no nontrivial barriers; that is, $G-e$ is bicritical. This contradicts Claim 6 for $B_1 := \{v_1\}$ and $B_2 := \{v_2\}$. ($G-e-u$ is factor-critical for each $u \in V$ by (38.3).) So $G-e$ has at least one maximal nontrivial barrier, B_1 say. Let $J_1 := B_1 \cup I(B_1)$, and assume without loss of generality that $v_1 \in I(B_1)$.

Assume that there is exactly one maximal nontrivial barrier. Then $G-e/J_1$ has no nontrivial barrier; that is, it is bicritical. By Claim 4, G/J_1 is a brick, and hence is 3-connected. This contradicts Claim 6, taking $B_2 := \{v_2\}$. ($G-e-u$ is factor-critical for each $u \in V \setminus J_1$ by (38.3).)

End of Proof of Claim 7

Decomposition of G

Having the two maximal nontrivial barriers B_1 and B_2, assuming $v_1 \in I(B_1)$ and $v_2 \in I(B_2)$, we define

(38.45) $\qquad J_1 := B_1 \cup I(B_1)$ and $J_2 := B_2 \cup I(B_2)$.

Note that J_1 and J_2 might intersect. Define $J_1' := J_1 \setminus J_2$, $J_2' := J_2 \setminus J_1$, $B_1' := B_1 \setminus I(B_2)$, and $B_2' := B_2' \setminus I(B_1)$. By Theorem 38.9, B_1' and B_2' are simple barriers again, with $I(B_1') = I(B_1) \setminus B_2$ and $I(B_2') = I(B_2) \setminus B_1$.

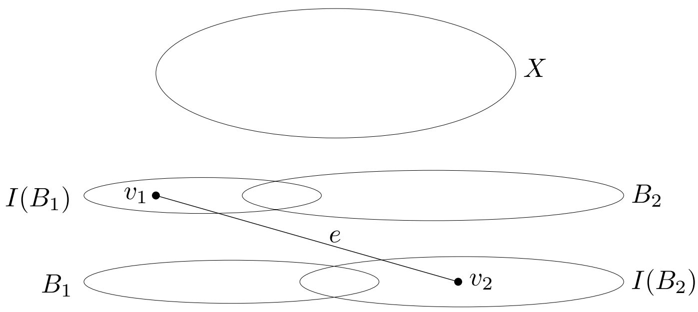


Figure 38.3

Thus we obtain a decomposition of V into

(38.46) $\qquad B_1', B_2', I(B_1'), I(B_2'), B_1 \cap I(B_2), B_2 \cap I(B_1),$
$\qquad X := K(B_1) \cap K(B_2),$

where e connects $I(B_1')$ and $I(B_2')$.

By Theorem 38.9, $G - X$ is bipartite, with colour classes $B_1 \cup I(B_2)$ and $B_2 \cup I(B_1)$.

$G[X]$ has exactly two components

Claim 8. $G[X]$ *is disconnected.*

Proof of Claim 8. Consider $H := G - e/J_1/J_2'$. Note that H is isomorphic to $G - e/J_1'/J_2$, since, if $J_1 \cap J_2 \neq \emptyset$, then $J_1 \cap J_2$ has neighbours both in J_1' and J_2', and nowhere else (by Theorems 38.5 and 38.9).

By Claim 5, H is not a brick. However,

(38.47) H is bicritical.

To see this, choose two distinct vertices v, v' of H. We can assume that $v \neq J_2'$ and $v' \neq J_1$. (If $v = J_2'$ or $v' = J_1$ then exchange v and v'.) Let w be equal to v if $v \neq J_1$ and let w be any vertex in B_1 if $v = J_1$. Similarly, let w' be equal to v' if $v' \neq J_2'$ and let w' be any vertex in B_2' if $v' = J_2'$. Then $G - e - w - w'$ has a perfect matching, since $\{w, w'\}$ is neither contained in B_1 nor in B_2. As B_1 is a simple barrier in $G - e$, each vertex in $I(B_1)$ is matched to a vertex in B_1. Similarly, each vertex in $I(B_2')$ is matched to a vertex in B_2'. Hence this perfect matching gives a perfect matching of $H - v - v'$. This proves (38.47).

By Claim 6, $G/J_1/J_2'$ has a 2-vertex-cut $\{u, u'\}$ intersecting $\{J_1, J_2'\}$. ($G - e - u$ is factor-critical for each $u \in X$ by (38.3).) If $\{u, u'\} = \{J_1, J_2'\}$ we are done. So we can assume that $u' \notin \{J_1, J_2'\}$. If $u = J_1$, then u' is a cut vertex of $G - J_1$, contradicting Claim 4. If $u = J_2'$, observe that $G/J_1/J_2'$ is isomorphic to $G/J_1'/J_2$, where the isomorphism brings vertex J_1 to vertex J_1', and vertex J_2' to vertex J_2. So u' is a cut vertex of $G - J_2$, again contradicting Claim 4.

End of Proof of Claim 8

We have that

(38.48) each component of $G[X]$ is even,

as for any $u \in B_1'$, $G[K(B_2)] - u$ has a perfect matching M. Then trivially no edge in M connects $K(B_2)$ and J_2. Moreover, no edge in M connects $K(B_1)$ and J_1, since $e \notin M$ (as e is not contained in $K(B_2)$) and since each vertex in $I(B_1')$ is matched to a vertex in $B_1' \setminus \{u\}$ (note that $J_1' \subseteq K(B_2)$).

For any subset L of X, any perfect matching M of G, and any $i \in \{1, 2\}$, define

(38.49) $\lambda_i(M, L) :=$ the number of edges in M connecting L and B_i.

Claim 9. For any component L *of* $G[X]$ *and any perfect matching* M *of* G *containing* e *one has* $\{\lambda_1(M, L), \lambda_2(M, L)\} = \{0, 2\}$.

Proof of Claim 9. Since $\lambda_1(M,L)+\lambda_2(M,L)=|M\cap\delta(L)|$ is even (as $|L|$ is even by (38.48)) and since $\lambda_i(M,L)\leq 2$ for $i=1,2$ (since M has two edges connecting $K(B_i)$ and B_i), it suffices to show that $\lambda_1(M,L)\neq\lambda_2(M,L)$.

Suppose that $\lambda_1(M,L)=\lambda_2(M,L)$. Since $e\in M$, $|M\cap\delta(J_1')|=3$. As no edge connects L and $I(B_i)$ (since e is the only edge connecting $K(B_i)$ and $I(B_i)$, but $v_1,v_2\notin L$), we have that M has $\lambda_1(M,L)$ edges connecting L and J_1'. Hence for $U:=J_1'\cup L$,

$$
\begin{aligned}
(38.50)\qquad |M\cap\delta(U)| &= |M\cap\delta(J_1')|+|M\cap\delta(L)|-2\lambda_1(M,L)\\
&= 3+\lambda_1(M,L)+\lambda_2(M,L)-2\lambda_1(M,L)=3.
\end{aligned}
$$

Moreover, any perfect matching N of $G-e$ satisfies $|N\cap\delta(U)|=1$. Indeed, $|N\cap\delta(J_1')|=1$ and $|N\cap\delta(J_2)|=1$. So $|N\cap\delta(X)|\leq 2$. Hence if $|N\cap\delta(U)|\geq 3$, then $N\cap\delta(U)$ contains an edge leaving neither J_1' nor J_2. Hence N has an edge connecting L and $X\setminus L$, a contradiction. So $|N\cap\delta(U)|=1$.

We show that both $G/\overline{U}$ and G/U are matching-covered, AND THat each has a unique brick in its brick decomposition, contradicting Claim 1.

Consider $G':=G/J_2$. Then G' is a brick by Claim 4, and L is a nonempty union of components of $G'-J_1'-\{J_2\}$. Moreover, $G'-e$ is matching-covered (since each perfect matching of $G-e$ has exactly one edge in $\delta(J_2)$) and B_1' is a simple barrier of $G'-e$. So by Theorem 38.8 (taking $Z:=X\setminus L$ and $v_2=J_2$), $G'/\overline{U}=G/\overline{U}$ is matching-covered and has a unique brick in its brick decomposition.

Let $U':=J_2'\cup(X\setminus L)$. Similarly, $G/\overline{U'}$ is matching-covered and has a unique brick in its brick decomposition. Since $\overline{U'}=U\cup(J_1\cap J_2)$, we have $\overline{U\cup U'}=J_1\cap J_2$. So $G/U/U'$ is matching-covered and bipartite. As U' gives a tight cut in G/U, also G/U is matching-covered and has a unique brick in its brick decomposition. *End of Proof of Claim 9*

Claim 10. $G[X]$ has exactly two components.

Proof of Claim 10. Let M be any perfect matching of G containing e. Then $\lambda_i(M,X)\leq 2$ for $i=1,2$, and hence by Claim 9, $G[X]$ has exactly two components. *End of Proof of Claim 10*

Conclusion

Let L_1 and L_2 be the components of $G[X]$. For $j=1,2$, let Z_j be the set of pairs $\{b,b'\}$ with $b\in B_1$, $b'\in B_2'$ such that $L_j\cup\{b,b'\}$ is matchable. In particular, if $b\in B_1$ and $b'\in B_2'$ are adjacent, then $\{b,b'\}\in Z_1\cap Z_2$. Then

Claim 11. For each $j=1,2$, any $b\in N(L_j)$ belongs to some pair in Z_j.

Proof of Claim 11. As $b\in N(L_j)$, there is an edge f joining b and L_j. Let M be a perfect matching of $G-e$ containing f. Then $\lambda_1(M,L_j)=\lambda_2(M,L_j)=1$, and hence $\{b,b'\}\in Z_j$ for some b'. *End of Proof of Claim 11*

Note that if $b \in N(L_j)$ for some j, then $b \in B_1' \cup B_2'$ (since X has no neighbour in $I(B_1) \cup I(B_2)$).

Claim 12. Each pair in Z_1 intersects each pair in Z_2.

Proof of Claim 12. Suppose to the contrary that there exist disjoint pairs $\{b, b'\} \in Z_1$ and $\{c, c'\} \in Z_2$, taking $b, c \in B_1$ and $b', c' \in B_2'$. By definition of Z_j, $L_1 \cup \{b, b'\}$ and $L_2 \cup \{c, c'\}$ are matchable. Moreover, by Theorem 38.7, also $J_1 \setminus \{b, c, v_1\}$ and $J_2' \setminus \{b', c', v_2\}$ are matchable. Together with e, this gives a perfect matching M of G containing e with $\lambda_1(M, L_1) \leq 1$ and $\lambda_2(M, L_1) \leq 1$. This contradicts Claim 9. *End of Proof of Claim 12*

Claim 13. $Z_1 \cap Z_2 = \emptyset$, $|B_1| = |B_2| = 2$, $I(B_1) \cap B_2 = I(B_2) \cap B_1 = \emptyset$, $B_1 \cup B_2$ is a stable set, and Z_1 and Z_2 are perfect matchings on $B_1 \cup B_2$.

Proof of Claim 13. We have $|N(L_j) \cap B_i| \geq 2$ for $j = 1, 2$ and $i = 1, 2$, since (for $j = 1$, $i = 1$, say) L_1 has at least two neighbours in $K(B_2)$ (as $G[K(B_2)]$ is 2-connected), which must belong to B_1.

Assume that $Z_1 \cap Z_2 \neq \emptyset$. Let $\{c, c'\} \in Z_1 \cap Z_2$ with $c \in B_1$ and $c' \in B_2'$. We can choose $b \in N(L_1) \cap B_1$ with $b \neq c$. Then $\{b, c'\} \in Z_1$ (by Claims 11 and 12). We can choose $b' \in N(L_2) \cap B_2'$ with $b' \neq c'$. Again, $\{b', c\} \in Z_2$. As $\{b, c'\}$ and $\{b', c\}$ are disjoint, this contradicts Claim 12. So $Z_1 \cap Z_2 = \emptyset$.

Then $B_1 \cup B_2'$ is a stable set, since if there is an edge connecting $b \in B_1$ and $b' \in B_2'$, then $L_1 \cup \{b, b'\}$ and $L_2 \cup \{b, b'\}$ are matchable, and hence $\{b, b'\} \in Z_1 \cap Z_2$, a contradiction.

This implies $B_1 \cap I(B_2) = \emptyset$, since otherwise there is an edge connecting $b \in B_1 \cap I(B_2)$ and $b' \in B_2' = B_2 \setminus I(B_1)$ (since $B_1 \cap I(B_2)$ has more than $|B_1 \cap I(B_2)| = |B_2 \cap I(B_1)|$ neighbours in B_2, by Theorem 38.5). Hence, by (38.17)(iii), $B_2 \cap I(B_1) = \emptyset$. So $B_2' = B_2$.

Next, for each $j = 1, 2$, no two pairs in Z_j intersect. For assume that $\{b, b'\}$, $\{b, c'\}$ belong to Z_1 with b', c' different vertices in B_2. As $|N(L_2) \cap B_1| \geq 2$, we can choose (by Claim 11) $\{d, d'\} \in Z_2$, with $d \in B_1$ and $d \neq b$. However, then $d' = b'$ and $d' = c'$ by Claim 12, a contradiction, as $b' \neq c'$.

So Z_j consists of disjoint pairs. As each pair in Z_1 intersects each pair in Z_2, we have that each Z_j consists of two disjoint pairs, that Z_1 and Z_2 cover the same set of vertices, and that $Z_1 \cap Z_2 = \emptyset$. In particular,

(38.51) $$|N(X) \cap B_1| = |N(X) \cap B_2| = 2.$$

Finally we show that $|B_i| = 2$ for $i = 1, 2$. Suppose that (say) $|B_1| \geq 3$. Then $|I(B_1)| \geq 2$. Choose $v \in I(B_1) \setminus \{v_1\}$. As G is bicritical, $G - v - v_1$ has a perfect matching M. Necessarily, at least three edges of M connect B_1 and $K(B_1)$, hence (as $B_1 \cup B_2$ is stable) M has at least three edges connecting X and B_1. So $|N(X) \cap B_1| \geq 3$, contradicting (38.51).

End of Proof of Claim 13

This claim in particular implies that

(38.52) v_1 and v_2 have degree 3

(since all neighbours of v_1 belong to $B_1 \cup \{v_2\}$). We can set

(38.53) $B_1 = \{b_1, b'_1\}$, $B_2 = \{b_2, b'_2\}$,
$Z_1 = \{\{b_1, b'_2\}, \{b'_1, b_2\}\}$, $Z_2 = \{\{b_1, b_2\}, \{b'_1, b'_2\}\}$.

Claim 14. $L_j \cup B_i$ is matchable for all $i, j \in \{1, 2\}$.

Proof of Claim 14. We may assume $i = 2$, $j = 1$. Let M and N be matchings spanning $L_1 \cup \{b'_1, b_2\}$ and $L_1 \cup \{b_1, b'_2\}$, respectively. The path P in $M \cup N$ starting at b'_1 ends at b'_2, as if P would end at b_1, then $L_1 \cup \{b_1, b_2\}$ is matchable (while $\{b_1, b_2\} \notin Z_1$), and if it would end at b_2, then $L_1 \cup \{b_1, b'_1, b_2, b'_2\}$ is matchable, implying that G has a perfect matching M' containing e with $\lambda_1(M', L_1) = \lambda_2(M', L_1) = 2$, contradicting Claim 9. So $M \triangle EP$ is a perfect matching on $L_1 \cup \{b_2, b'_2\}$. *End of Proof of Claim 14*

Claim 15. $G - e'$ is matching-covered for each edge e' of G.

Proof of Claim 15. Since G is connected and e is chosen arbitrarily under the condition that $G - e$ is matching-covered, we can assume that e' is incident with e. In particular, we can assume that e' connects v_1 and b_1. Suppose that $G - e'$ is not matching-covered. Then there exists an edge $f \neq e'$ such that each perfect matching of G containing f also contains e'. So f is disjoint from e'.

First assume that f is incident with v_2. We may assume that f connects v_2 with vertex b_2. By definition of Z_1, $L_1 \cup \{b_1, b'_2\}$ is matchable. Since also L_2 is matchable, we can find a perfect matching of G containing f but not e', contradicting our assumption.

So we may assume that f is incident with L_1. Let M' be a perfect matching of G containing f. If M' does not intersect $\delta(L_1)$, we can extend $M'[L_1] \cup \{v_1 b'_1, v_2 b'_2\}$ by a matching spanning $L_2 \cup \{b_1, b_2\}$ to obtain a perfect matching containing f but not e', a contradiction. So M' intersects $\delta(L_1)$. Hence, necessarily, it contains an edge joining L_1 with b'_1 (as $e' \in M'$). So also it contains an edge joining L_1 and b_2. Therefore, M' contains a matching M spanning $L_1 \cup \{b'_1, b_2\}$. Let N be a matching spanning $L_1 \cup \{b_1, b'_2\}$.

Like in Claim 14, the path P in $M \cup N$ starting at b'_1 ends at b'_2. Similarly, the path Q in $M \cup N$ starting at b_2 ends at b_1. At least one of $M \triangle EP$ and $M \triangle EQ$ contains f (since f is in M and on at most one of P, Q). As $L_2 \cup \{b_1, b'_1\}$ and $L_2 \cup \{b_2, b'_2\}$ are matchable (by Claim 14), there is a perfect matching containing f and not e', a contradiction. *End of Proof of Claim 15*

This gives with (38.52) that

(38.54) G is 3-regular,

since by Claim 15 we can take for e any edge of G.

Claim 16. $|L_1| = |L_2| = 2$.

Proof of Claim 16. Since G is 3-regular, each $b \in B_1 \cup B_2$ has a unique neighbour in L_j, for each $j = 1, 2$. In fact, for any $j = 1, 2$,

(38.55) if $b \in B_1$, $b' \in B_2$, and $\{b, b'\} \notin Z_j$, then the neighbours of b and b' in L_j coincide.

For assume that the neighbour c of b in L_j differs from the neighbour c' of b' in L_j. As G is bicritical, $G - c - c'$ has a perfect matching M. Let M' be the set of edges in M intersecting L_j. As $|L_j|$ is even, M' spans either $L_j - c - c'$ or $(L_j - c - c') \cup (B_1 - b) \cup (B_2 - b')$. Extending M' with the edges bc and $b'c'$, we obtain a matching spanning $L_j \cup \{b, b'\}$, contradicting $\{b, b'\} \notin Z_j$, or spanning $L_j \cup B_1 \cup B_2$, contradicting Claim 9. This shows (38.55).

Now (38.55) implies that $N(B_1) \cap L_1 = N(B_2) \cap L_1$. As this set is not a 2-vertex-cut of G, we have $|L_1| = 2$. Similarly, $|L_2| = 2$.

End of Proof of Claim 16

So both L_1 and L_2 consist of a single edge. Therefore, G is the Petersen graph, contradicting our assumption. ∎

38.7. Synthesis and further consequences of the previous results

The previous results imply a characterization of the matching lattice for matching-covered graphs (Lovász [1987]):

Corollary 38.11a. *Let $G = (V, E)$ be a matching-covered graph and let $x \in \mathbb{Z}^E$. Then x belongs to the perfect matching lattice of G if and only if for some maximal cross-free collection $\mathcal{F}$ of nontrivial tight cuts:*

(38.56) (i) *$x(D) = x(\delta(v))$ for each $D \in \mathcal{F}$ and each $v \in V$;*
(ii) *for every Petersen brick resulting from the given tight cut decomposition, and for some 5-circuit C in that brick, the sum of the x_e over edges e mapping to EC, is even.*

Proof. Directly from Theorems 38.6, 38.1, and 38.11. ∎

Corollary 38.11a implies the following (conjectured by Lovász [1985]):

Corollary 38.11b. *Let $G = (V, E)$ be a matching-covered graph and let $x \in 2\mathbb{Z}^E$ be such that $x(C) = x(C')$ for any two tight cuts C and C'. Then x belongs to the perfect matching lattice of G.*

Proof. Directly from Corollary 38.11a. ∎

Moreover, there is the following corollary for regular graphs (recall that a *k-graph* is a k-regular graph with $|C| \geq k$ for each odd cut):

Corollary 38.11c. *Let $G = (V, E)$ be a k-graph. Then the all-2 vector $\mathbf{2}$ belongs to the perfect matching lattice of G. If G has no subgraph homeomorphic to the Petersen graph, then the all-1 vector belongs to the perfect matching lattice of G.*

Proof. Directly from Corollary 38.11a. ∎

A special case is the following result of Seymour [1979a], which also follows from the conjecture of Tutte [1966], proved by Robertson, Seymour, and Thomas [1997], Sanders, Seymour, and Thomas [2000], and Sanders and Thomas [2000], that each bridgeless cubic graph without Petersen graph minor, is 3-edge-colourable.

Corollary 38.11d. *Let $G = (V, E)$ be a bridgeless cubic graph without Petersen graph minor. Then the all-1 vector $\mathbf{1}$ belongs to the perfect matching lattice of G.*

Proof. This is a special case of Corollary 38.11c. ∎

Similarly, the following consequence, a theorem of Seymour [1979a], supports a positive answer to the question of Fulkerson [1971a] whether each cubic graph G satisfies $\chi'(G_2) = 6$:

Corollary 38.11e. *Let $G = (V, E)$ be a bridgeless cubic graph. Then the all-2 vector $\mathbf{2}$ in $\mathbb{R}^E$ belongs to the perfect matching lattice of G.*

Proof. Again, this is a special case of Corollary 38.11c. ∎

38.8. What further might (not) be true

The conjecture that the perfect matchings in any graph would constitute a Hilbert base, is too bold: Let G be the graph obtained from the Petersen graph by adding one additional edge (connecting nonadjacent vertices of the Petersen graph). Let $x_e := 1$ if e is an edge of the Petersen graph, and $x_e := 0$ if e is the new edge. Then x belongs to the perfect matching cone[23] and to the perfect matching lattice (since G is a brick). However, x is not a nonnegative integer combination of perfect matchings, since the Petersen graph is not 3-edge-colourable. (This example was given by Goddyn [1993].)

Two weaker conjectures might yet hold true. The first one is due to L. Lovász (cf. Goddyn [1993]):

[23] The *perfect matching cone* is the cone generated by the incidence vectors of the perfect matchings.

(38.57) (?) for any graph without Petersen graph minor, the incidence vectors of the perfect matchings form a Hilbert base. (?)

The second one was given in Section 28.6 above ((28.28)), and is due to Seymour [1979a] (the *generalized Fulkerson conjecture*):

(38.58) (?) each k-graph contains $2k$ perfect matchings, covering each edge exactly twice. (?)

(A *k-graph* is a k-regular graph $G = (V, E)$ with $d_G(U) \geq k$ for each odd $U \subseteq V$.) For $k = 3$, (38.58) was asked by Fulkerson [1971a]:

(38.59) (?) each bridgeless cubic graph has 6 perfect matchings covering each edge precisely twice. (?)

What has been proved by Robertson, Seymour, and Thomas [1997], Sanders, Seymour, and Thomas [2000], and Sanders and Thomas [2000] is:

(38.60) each bridgeless cubic graph without Petersen graph minor is 3-edge-colourable.

This is a special case of conjecture (38.57), and of the 4-*flow conjecture* of Tutte [1966]:

(38.61) (?) each bridgeless graph without Petersen graph minor has three cycles covering each edge precisely twice. (?)

(A *cycle* is an edge-disjoint union of circuits.) Related is the following theorem of Alspach, Goddyn, and Zhang [1994]:

(38.62) the circuits of a graph G form a Hilbert base $\iff$ G has no Petersen graph minor.

It implies that the *circuit double cover conjecture* (asked by Szekeres [1973], conjectured by Seymour [1979b]):

(38.63) (?) each bridgeless graph has a family of circuits covering each edge precisely twice, (?)

is true for graphs without Petersen graph minor:

(38.64) each bridgeless graph without Petersen graph minor has a family of circuits covering each edge precisely twice.

(For cubic graphs this was shown by Alspach and Zhang [1993].) This is also a special case of the 4-flow conjecture (38.61).

Seymour [1979b] conjectures that

(38.65) (?) each *even* integer vector x in the circuit cone is a nonnegative integer combination of incidence vectors of circuits. (?)

This is more general than the circuit double cover conjecture.

Bermond, Jackson, and Jaeger [1983] have proved that

(38.66) each bridgeless graph has a family of circuits covering each edge precisely four times.

Tarsi [1986] mentioned the following strengthening of the circuit double cover conjecture:

(38.67) (?) in each bridgeless graph there exists a family of at most 5 cycles covering each edge precisely twice. (?)

Finally, the 5-*flow conjecture* of Tutte [1954a]:

(38.68) (?) each bridgeless graph has a nowhere-zero 5-flow, (?)

can be formulated in terms of circuits as follows (by Theorem 28.4):

(38.69) (?) each bridgeless graph can be oriented such that there exist directed circuits, covering each edge at least once and at most four times. (?)

Seymour [1981b] showed that each bridgeless graph has a nowhere-zero 6-flow; equivalently:

(38.70) each bridgeless graph can be oriented such that there exist directed circuits, covering each edge at least once and at most five times.

It improves an earlier result of Jaeger [1976,1979] that each bridgeless graph has a nowhere-zero 8-flow. This is equivalent to: each bridgeless graph contains three cycles covering all edges.

Notes. More on nowhere-zero flows and circuit covers can be found in Itai, Lipton, Papadimitriou, and Rodeh [1981], Bermond, Jackson, and Jaeger [1983], Bouchet [1983], Steinberg [1984], Alon and Tarsi [1985], Fraisse [1985], Jaeger, Khelladi, and Mollard [1985], Tarsi [1986], Khelladi [1987], Möller, Carstens, and Brinkmann [1988], Catlin [1989], Goddyn [1989], Jamshy and Tarsi [1989,1992], Fan [1990,1993, 1995,1998], Jackson [1990], Zhang [1990,1993c], Raspaud [1991], Alspach and Zhang [1993], Fan and Raspaud [1994], Huck and Kochol [1995], Lai [1995], Steffen [1996], and Galluccio and Goddyn [2002]. Surveys were given by Jaeger [1979,1985,1988], Zhang [1993a,1993b], and Seymour [1995a], and a book was devoted to it by Zhang [1997b]. The extension to matroids is discussed in Section 81.10.

38.9. Further results and notes

38.9a. The perfect 2-matching space and lattice

Let $G = (V, E)$ be a graph. The *perfect 2-matching space* of G is the linear hull of the perfect 2-matchings in G. This space is easily characterized with the help of Corollary 30.2b:

Theorem 38.12. *The perfect 2-matching space of G consists of all vectors $x \in \mathbb{R}^E$ such that $x_e = 0$ if e is not in the support of any perfect 2-matching and such that $x(\delta(v)) = x(\delta(u))$ for all $u, v \in V$.*

Proof. Clearly each vector x in the perfect 2-matching space satisfies the condition. To see the reverse, let x satisfy the condition. By adding appropriate multiples of perfect 2-matchings, we can assume that $x \geq \mathbf{0}$. If $x = \mathbf{0}$ we are done, so we can assume $x \neq \mathbf{0}$. Then, by scaling, we can assume that $x(\delta(v)) = 2$ for each vertex v. Hence, by Corollary 30.2b, x belongs to the perfect 2-matching polytope of F, and therefore to the perfect 2-matching space. ∎

The *perfect 2-matching lattice* of G is the lattice generated by the perfect 2-matchings in G. Jungnickel and Leclerc [1989] showed that a characterization of the perfect 2-matching lattice can be easily derived from the theorem of Petersen that the edges of any $2k$-regular graph can be decomposed into k 2-factors (Corollary 30.7b):

Theorem 38.13. *The perfect 2-matching lattice of G consists of all integer vectors x in the perfect 2-matching space of G with $x(\delta(v))$ even for one (hence for each) vertex v.*

Proof. Trivially, each vector x in the perfect 2-matching lattice satisfies the condition. To see the reverse, let x satisfy the condition. By adding integer multiples of perfect 2-matchings, we can assume that $x \geq \mathbf{0}$. Replace each edge e by x_e parallel edges, yielding graph G', of degree $2k$ for some integer $k > 0$. Now by Corollary 30.7b, the edges of G' can be partitioned into k 2-factors. This gives a decomposition of x as a sum of k perfect 2-matchings in G. ∎

38.9b. Further notes

De Carvalho, Lucchesi, and Murty [2002a,2002b] showed that each brick G different from K_4, the prism $\overline{C_6}$, and the Petersen graph, has an edge e such that $G - e$ is a matching-covered graph with precisely one brick in its brick decomposition (conjectured by L. Lovász in 1987). Having this, the proof of Theorem 38.11 can be shortened considerably (de Carvalho, Lucchesi, and Murty [2002c]). (Earlier related work was done by de Carvalho and Lucchesi [1996].)

Naddef and Pulleyblank [1982] study the relation between ear-decompositions and the GF(2)-rank of the incidence vectors of the perfect matchings.

Kilakos [1996] characterized the lattice generated by the matchings M that have a positive coefficient in at least one fractional $\chi'^*(G)$-edge-colouring (these matchings form a face of the matching polytope of G).