GRAPH THEORY AND COMBINATORIAL OPTIMIZATION

GERAD 25th Anniversary Series

- **Essays and Surveys in Global Optimization**
  Charles Audet, Pierre Hansen, and Gilles Savard, editors
- **Graph Theory and Combinatorial Optimization**
  David Avis, Alain Hertz, and Odile Marcotte, editors
- **Numerical Methods in Finance**
  Hatem Ben-Ameur and Michèle Breton, editors
- **Analysis, Control and Optimization of Complex Dynamic Systems**
  El-Kébir Boukas and Roland Malhamé, editors
- **Column Generation**
  Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, editors
- **Statistical Modeling and Analysis for Complex Data Problems**
  Pierre Duchesne and Bruno Rémillard, editors
- **Performance Evaluation and Planning Methods for the Next Generation Internet**
  André Girard, Brunilde Sansò, and Félisa Vázquez-Abad, editors
- **Dynamic Games: Theory and Applications**
  Alain Haurie and Georges Zaccour, editors
- **Logistics Systems: Design and Optimization**
  André Langevin and Diane Riopel, editors
- **Energy and Environment**
  Richard Loulou, Jean-Philippe Waaub, and Georges Zaccour, editors

GRAPH THEORY AND COMBINATORIAL OPTIMIZATION

Edited by
DAVID AVIS
McGill University and GERAD

ALAIN HERTZ
École Polytechnique de Montréal and GERAD

ODILE MARCOTTE
Université du Québec à Montréal and GERAD

David Avis
McGill University & GERAD
Montréal, Canada

Alain Hertz
École Polytechnique de Montréal & GERAD
Montréal, Canada

Odile Marcotte
Université du Québec a Montréal and GERAD
Montréal, Canada

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

ISBN-10: 0-387-25591-5 ISBN 0-387-25592-3 (e-book) Printed on acid-free paper.
ISBN-13: 978-0387-25591-0

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11053149

springeronline.com

Foreword

GERAD celebrates this year its 25th anniversary. The Center was created in 1980 by a small group of professors and researchers of HEC Montréal, McGill University and of the École Polytechnique de Montréal. GERAD's activities achieved sufficient scope to justify its conversion in June 1988 into a Joint Research Centre of HEC Montréal, the École Polytechnique de Montréal and McGill University. In 1996, the Université du Québec à Montréal joined these three institutions. GERAD has fifty members (professors), more than twenty research associates and post doctoral students and more than two hundreds master and Ph.D. students.

GERAD is a multi-university center and a vital forum for the development of operations research. Its mission is defined around the following four complementarily objectives:

- The original and expert contribution to all research fields in GERAD's area of expertise;
- The dissemination of research results in the best scientific outlets as well as in the society in general;
- The training of graduate students and post doctoral researchers;
- The contribution to the economic community by solving important problems and providing transferable tools.

GERAD's research thrusts and fields of expertise are as follows:

- Development of mathematical analysis tools and techniques to solve the complex problems that arise in management sciences and engineering;
- Development of algorithms to resolve such problems efficiently;
- Application of these techniques and tools to problems posed in related disciplines, such as statistics, financial engineering, game theory and artificial intelligence;
- Application of advanced tools to optimization and planning of large technical and economic systems, such as energy systems, transportation/communication networks, and production systems;
- Integration of scientific findings into software, expert systems and decision-support systems that can be used by industry.

One of the marking events of the celebrations of the 25th anniversary of GERAD is the publication of ten volumes covering most of the Center's research areas of expertise. The list follows: **Essays and Surveys in Global Optimization**, edited by C. Audet, P. Hansen and G. Savard; **Graph Theory and Combinatorial Optimization**,

edited by D. Avis, A. Hertz and O. Marcotte; **Numerical Methods in Finance**, edited by H. Ben-Ameur and M. Breton; **Analysis, Control and Optimization of Complex Dynamic Systems**, edited by E.K. Boukas and R. Malhamé; **Column Generation**, edited by G. Desaulniers, J. Desrosiers and M.M. Solomon; **Statistical Modeling and Analysis for Complex Data Problems**, edited by P. Duchesne and B. Rémillard; **Performance Evaluation and Planning Methods for the Next Generation Internet**, edited by A. Girard, B. Sansò and F. Vázquez-Abad; **Dynamic Games: Theory and Applications**, edited by A. Haurie and G. Zaccour; **Logistics Systems: Design and Optimization**, edited by A. Langevin and D. Riopel; **Energy and Environment**, edited by R. Loulou, J.-P. Waaub and G. Zaccour.

I would like to express my gratitude to the Editors of the ten volumes, to the authors who accepted with great enthusiasm to submit their work and to the reviewers for their benevolent work and timely response. I would also like to thank Mrs. Nicole Paradis, Francine Benoît and Louise Letendre and Mr. André Montpetit for their excellent editing work.

The GERAD group has earned its reputation as a worldwide leader in its field. This is certainly due to the enthusiasm and motivation of GERAD's researchers and students, but also to the funding and the infrastructures available. I would like to seize the opportunity to thank the organizations that, from the beginning, believed in the potential and the value of GERAD and have supported it over the years. These are HEC Montréal, École Polytechnique de Montréal, McGill University, Université du Québec à Montréal and, of course, the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds québécois de la recherche sur la nature et les technologies (FQRNT).

Georges Zaccour
Director of GERAD

Avant-propos

Le Groupe d'études et de recherche en analyse des décisions (GERAD) fête cette année son vingt-cinquième anniversaire. Fondé en 1980 par une poignée de professeurs et chercheurs de HEC Montréal engagés dans des recherches en équipe avec des collègues de l'Université McGill et de l'École Polytechnique de Montréal, le Centre comporte maintenant une cinquantaine de membres, plus d'une vingtaine de professionnels de recherche et stagiaires post-doctoraux et plus de 200 étudiants des cycles supérieurs. Les activités du GERAD ont pris suffisamment d'ampleur pour justifier en juin 1988 sa transformation en un Centre de recherche conjoint de HEC Montréal, de l'École Polytechnique de Montréal et de l'Université McGill. En 1996, l'Université du Québec à Montréal s'est jointe à ces institutions pour parrainer le GERAD.

Le GERAD est un regroupement de chercheurs autour de la discipline de la recherche opérationnelle. Sa mission s'articule autour des objectifs complémentaires suivants :

- la contribution originale et experte dans tous les axes de recherche de ses champs de compétence ;
- la diffusion des résultats dans les plus grandes revues du domaine ainsi qu'auprès des différents publics qui forment l'environnement du Centre ;
- la formation d'étudiants des cycles supérieurs et de stagiaires post-doctoraux ;
- la contribution à la communauté économique à travers la résolution de problèmes et le développement de coffres d'outils transférables.

Les principaux axes de recherche du GERAD, en allant du plus théorique au plus appliqué, sont les suivants :

- le développement d'outils et de techniques d'analyse mathématiques de la recherche opérationnelle pour la résolution de problèmes complexes qui se posent dans les sciences de la gestion et du génie ;
- la confection d'algorithmes permettant la résolution efficace de ces problèmes ;
- l'application de ces outils à des problèmes posés dans des disciplines connexes à la recherche opérationnelle telles que la statistique, l'ingénierie financière, la théorie des jeux et l'intelligence artificielle ;
- l'application de ces outils à l'optimisation et à la planification de grands systèmes technico-économiques comme les systèmes énergétiques, les réseaux de télécommunication et de transport, la logistique et la distributique dans les industries manufacturières et de service ;

- l'intégration des résultats scientifiques dans des logiciels, des systèmes experts et dans des systèmes d'aide à la décision transférables à l'industrie.

Le fait marquant des célébrations du 25^{e} du GERAD est la publication de dix volumes couvrant les champs d'expertise du Centre. La liste suit : **Essays and Surveys in Global Optimization**, édité par C. Audet, P. Hansen et G. Savard ; **Graph Theory and Combinatorial Optimization**, édité par D. Avis, A. Hertz et O. Marcotte ; **Numerical Methods in Finance**, édité par H. Ben-Ameur et M. Breton ; **Analysis, Control and Optimization of Complex Dynamic Systems**, édité par E.K. Boukas et R. Malhamé ; **Column Generation**, édité par G. Desaulniers, J. Desrosiers et M.M. Solomon ; **Statistical Modeling and Analysis for Complex Data Problems**, édité par P. Duchesne et B. Rémillard ; **Performance Evaluation and Planning Methods for the Next Generation Internet**, édité par A. Girard, B. Sansò et F. Vázquez-Abad ; **Dynamic Games : Theory and Applications**, édité par A. Haurie et G. Zaccour ; **Logistics Systems : Design and Optimization**, édité par A. Langevin et D. Riopel ; **Energy and Environment**, édité par R. Loulou, J.-P. Waaub et G. Zaccour.

Je voudrais remercier très sincèrement les éditeurs de ces volumes, les nombreux auteurs qui ont très volontiers répondu à l'invitation des éditeurs à soumettre leurs travaux, et les évaluateurs pour leur bénévolat et ponctualité. Je voudrais aussi remercier Mmes Nicole Paradis, Francine Benoît et Louise Letendre ainsi que M. André Montpetit pour leur travail expert d'édition.

La place de premier plan qu'occupe le GERAD sur l'échiquier mondial est certes due à la passion qui anime ses chercheurs et ses étudiants, mais aussi au financement et à l'infrastructure disponibles. Je voudrais profiter de cette occasion pour remercier les organisations qui ont cru dès le départ au potentiel et à la valeur du GERAD et nous ont soutenus durant ces années. Il s'agit de HEC Montréal, l'École Polytechnique de Montréal, l'Université McGill, l'Université du Québec à Montréal et, bien sûr, le Conseil de recherche en sciences naturelles et en génie du Canada (CRSNG) et le Fonds québécois de la recherche sur la nature et les technologies (FQRNT).

Georges Zaccour
Directeur du GERAD

Contents

Contributing Authors

Nair Maria Maia de Abreu
Universidade Federal do Rio de Janeiro, Brasil
nair@pep.ufrj.br

Slim Belhaiza
École Polytechnique de Montréal, Canada
Slim.Belhaiza@polymtl.ca

Peter Brass
City College, City University of New York, USA
peter@cs.ccny.cuny.edu

Komei Fukuda
ETH Zurich, Switzerland
komei.fukuda@ifor.math.ethz.ch

Pierre Hansen
HEC Montréal and GERAD, Canada
Pierre.Hansen@gerad.ca

Alain Hertz
École Polytechnique de Montréal and GERAD, Canada
alain.hertz@gerad.ca

Kartik Krishnan
McMaster University, Canada
kartik@optlab.mcmaster.ca

Wieslaw Kubiak
Memorial University of Newfoundland, Canada
wkubiak@mun.ca

Vadim V. Lozin
Rutgers University, USA
lozin@rutcor.rutgers.edu

Patrice Marcotte
Université de Montréal, Canada
marcotte@iro.umontreal.ca

Carla Silva Oliveira
Escola Nacional de Ciências Estatísticas, Brasil
carlasilva@ibge.gov.br

János Pach
City College, City University of New York, USA
pach@cims.nyu.edu

Vera Rosta
Alfréd Rényi Institute of Mathematics, Hungary & McGill University, Canada
rosta@renyi.hu

Gilles Savard
École Polytechnique de Montréal and GERAD, Canada
gilles.savard@polymtl.ca

F.B. Shepherd
Bell Laboratories, USA
bshep@research.bell-labs.com

Tamás Terlaky
McMaster University, Canada
terlaky@mcmaster.ca

A. Vetta
McGill University, Canada
vetta@math.mcgill.ca

Dominique de Werra
École Polytechnique Fédérale de Lausanne, Switzerland
dewerra@dma.epfl.ch

Preface

Combinatorial optimization is at the heart of the research interests of many members of GERAD. To solve problems arising in the fields of transportation and telecommunication, the operations research analyst often has to use techniques that were first designed to solve classical problems from combinatorial optimization such as the maximum flow problem, the independent set problem and the traveling salesman problem. Most (if not all) of these problems are also closely related to graph theory. The present volume contains nine chapters covering many aspects of combinatorial optimization and graph theory, from well-known graph theoretical problems to heuristics and novel approaches to combinatorial optimization.

In Chapter 1, Belhaiza, de Abreu, Hansen and Oliveira study several conjectures on the algebraic connectivity of graphs. Given an undirected graph G, the algebraic connectivity of G (denoted $a(G)$) is the smallest eigenvalue of the Laplacian matrix of G. The authors use the AutoGraphiX (AGX) system to generate connected graphs that are not complete and minimize (resp. maximize) $a(G)$ as a function of n (the order of G) and m (its number of edges). They formulate several conjectures on the structure of these extremal graphs and prove some of them.

In Chapter 2, Brass and Pach survey the results in the theory of geometric patterns and give an overview of the many interesting problems in this theory. Given a set S of n points in d-dimensional space, and an equivalence relation between subsets of S, one is interested in the equivalence classes of subsets (i.e., *patterns*) occurring in S. For instance, two subsets can be deemed equivalent if and only if one is the translate of the other. Then a *Turán-type* question is the following: "What is the maximum number of occurrences of a given pattern in S?" A *Ramsey-type* question is the following: "Is it possible to color space so that there is no monochromatic occurrence of a given pattern?" Brass and Pach investigate these and other questions for several equivalence relations (translation, congruence, similarity, affine transformations, etc.), present the results for each relation and discuss the outstanding problems.

In Chapter 3, Fukuda and Rosta survey various data depth measures, first introduced in nonparametric statistics as multidimensional generalizations of ranks and the median. These data depth measures have been studied independently by researchers working in statistics, political science, optimization and discrete and computational geometry. Fukuda and Rosta show that computing data depth measures often reduces to

finding a maximum feasible subsystem of linear inequalities, that is, a solution satisfying as many constraints as possible. Thus they provide a unified framework for the main data depth measures, such as the half-space depth, the regression depth and the simplicial depth. They survey the related results from nonparametric statistics, computational geometry, discrete geometry and linear optimization.

In Chapter 4, Hertz and Lozin survey the method of augmenting graphs for solving the maximum independent set problem. It is well known that the maximum matching problem can be solved by looking for augmenting paths and using them to increase the size of the current matching. In the case of the maximum independent set problem, however, finding an augmenting graph is much more difficult. Hertz and Lozin show that for special classes of graphs, all the families of augmenting graphs can be characterized and the problem solved in polynomial time. They present the main results of the theory of augmenting graphs and propose new contributions to this theory.

In Chapter 5, Krishnan and Terlaky present a survey of semidefinite and interior point methods for solving NP-hard combinatorial optimization problems to optimality and designing approximation algorithms for some of these problems. The approaches described in this chapter include non-convex potential reduction methods, interior point cutting plane methods, primal-dual interior point methods and first-order algorithms for solving semidefinite programs, branch-and-cut approaches based on semidefinite programming formulations and finally methods for solving combinatorial optimization problems by means of successive convex approximations.

In Chapter 6, Kubiak presents a study of balancing mixed-model supply chains. A *mixed-model supply chain* is designed to deliver a wide range of customized models of a product to customers. The main objective of the model is to keep the supply of each model as close to its demand as possible. Kubiak reviews algorithms for the model variation problem and introduces and explores the link between model delivery sequences and balanced words. He also shows that the extended problem (obtained by including the suppliers' capacity constraints into the model) is NP-hard in the strong sense, and reviews algorithms for the extended problem. Finally he addresses the problem of minimizing the number of setups in delivery feasible supplier production sequences.

In Chapter 7, Marcotte and Savard present an overview of two classes of bilevel programs and their relationship to well-known combinatorial optimization problems, in particular the traveling salesman problem. In a bilevel program, a subset of variables is constrained to lie in the optimal set of an auxiliary mathematical program. Bilevel programs are hard to

solve, because they are generically non-convex and non-differentiable. Thus research on bilevel programs has followed two main avenues, the continuous approach and the combinatorial approach. The combinatorial approach aims to develop algorithms providing a guarantee of global optimality. The authors consider two classes of programs amenable to this approach, that is, the bilevel programs with linear or bilinear objectives.

In Chapter 8, Shepherd and Vetta present a study of dijoins. Given a directed graph $G = (V, A)$, a *dijoin* is a set of arcs B such that the graph $(V, A \cup B)$ is strongly connected. Shepherd and Vetta give two results that help to visualize dijoins. They give a simple description of Frank's primal-dual algorithm for finding a minimum dijoin. Then they consider weighted packings of dijoins, that is, multisets of dijoins such that the number of dijoins containing a given arc is at most the weight of the arc. Specifically, they study the cardinality of a weighted packing of dijoins in graphs for which the minimum weight of a directed cut is at least a constant k, and relate this problem to the concept of *skew submodular flow polyhedron.*

In Chapter 9, de Werra generalizes a coloring property of unimodular hypergraphs. A hypergraph H is *unimodular* is its edge-node incidence matrix is totally unimodular. A *k-coloring* of H is a partition of its node set X into subsets $S_1, S_2, \ldots, S_k$ such that no S_i contains an edge E with $|E| \geq 2$. The new version of the coloring property implies that a unimodular hypergraph has an equitable k-coloring satisfying additional constraints. The author also gives an adaptation of this result to balanced hypergraphs.

Acknowledgements The Editors are very grateful to the authors for contributing to this volume and responding to their comments in a timely fashion. They also wish to thank Nicole Paradis, Francine Benoît and André Montpetit for their expert editing of this volume.

DAVID AVIS
ALAIN HERTZ
ODILE MARCOTTE

Chapter 1

VARIABLE NEIGHBORHOOD SEARCH FOR EXTREMAL GRAPHS. XI. BOUNDS ON ALGEBRAIC CONNECTIVITY

Slim Belhaiza
Nair Maria Maia de Abreu
Pierre Hansen
Carla Silva Oliveira

Abstract The algebraic connectivity $a(G)$ of a graph $G = (V, E)$ is the second smallest eigenvalue of its Laplacian matrix. Using the AutoGraphiX (AGX) system, extremal graphs for algebraic connectivity of G in function of its order $n = |V|$ and size $m = |E|$ are studied. Several conjectures on the structure of those graphs, and implied bounds on the algebraic connectivity, are obtained. Some of them are proved, e.g., if $G \neq K_n$

$$a(G) \leq \lfloor -1 + \sqrt{1 + 2m} \rfloor$$

which is sharp for all $m \geq 2$.

1. Introduction

Computers are increasingly used in graph theory. Determining the numerical value of graph invariants has been done extensively since the fifties of last century. Many further tasks have since been explored. Specialized programs helped, often through enumeration of specific families of graphs or subgraphs, to prove important theorems. The prominent example is, of course, the Four-color Theorem (Appel and Haken, 1977a,b, 1989; Robertson et al., 1997). General programs for graph enumeration, susceptible to take into account a variety of constraints and exploit symmetry, were also developped (see, e.g., McKay, 1990, 1998). An interactive approach to graph generation, display, modification and study through many parameters has been pioneered in the system Graph of Cvetković and Kraus (1983), Cvetković et al. (1981), and Cvetković and

Simić (1994) which led to numerous research papers. Several systems for obtaining conjectures in an automated or computer-assisted way have been proposed (see, e.g., Hansen, 2002, for a recent survey). The AutoGraphix (AGX) system, developed at GERAD, Montréal since 1997 (see, e.g., Caporossi and Hansen, 2000, 2004) is designed to address the following tasks: (a) Find a graph satisfying given constraints; (b) Find optimal or near-optimal values for a graph invariant subject to constraints; (c) Refute conjectures (or repair them); (d) Suggest conjectures (or sharpen existing ones); (e) Suggest lines of proof.

The basic idea is to address all those tasks through heuristic search of one or a family of extremal graphs. This can be done in a unified way, i.e., for any formula on one or several invariants and subject to constraints, with the Variable Neighborhood Search (VNS) metaheuristic of Mladenović and Hansen (1997) and Hansen and Mladenović (2001). Given a formula, VNS first searches a local minimum on the family of graphs with possibly some parameters fixed such as the number of vertices n or the number of edges m. This is done by making elementary changes in a greedy way (i.e., decreasing most the objective, in case of minimization) on a given initial graph: rotation of an edge (changing one of its endpoints), removal or addition of one edge, short-cut (i.e., replacing a 2-path by a single edge) detour (the reverse of the previous operation), insertion or removal of a vertex and the like. Once a local minimum is reached, the corresponding graph is perturbed increasingly, by choosing at random another graph in a farther and farther neighborhood. A descent is then performed from this perturbed graph. Three cases may occur: (i) one gets back to the unperturbed local optimum, or (ii) one gets to a new local optimum with an equal or worse value than the unperturbed one, in which case one moves to the next neighborhood, or (iii) one gets to a new local optimum with a better value than the unperturbed one, in which case one recenters the search there. The neighborhoods for perturbation are usually nested and obtained from the unperturbed graph by addition, removal or moving of $1, 2, \ldots, k$ edges.

Refuting conjectures given in inequality form, i.e., $i_1(G) \leq i_2(G)$ where i_1 and i_2 are invariants, is done by minimizing the difference between right and left hand sides; a graph with a negative value then refutes the conjectures. Obtaining new conjectures is done from values of invariants for a family of (presumably) extremal graphs depending on some parameter(s) (usually n and/or m). Three ways are used (Caporossi and Hansen, 2004): (i) a *numerical way*, which exploits the mathematics of Principal Component Analysis to find a basis of affine relations between graph invariants satisfied by those extremal graphs considered; (ii) a *geometric way*, i.e., finding with a "gift-wrapping" algorithm the

convex hull of the set of points corresponding to the extremal graph in invariants space: each facet then gives a linear inequality; (iii) an *algebraic way*, which consists in determining the class to which all extremal graphs belong, if there is one (often it is a simple one such as paths, stars, complete graphs, etc); then formulae giving the value of individual invariants in function of n and/or m are combined. Obtaining possible lines of proof is done by checking if one or just a few of the elementary changes always suffice to get the extremal graphs found; if so, one can try to show that it is possible to apply such changes to any graph of the class under study.

Recall that the Laplacian matrix $L(G)$ of a graph $G = (V, E)$ is the difference of a diagonal matrix with values equal to the degrees of vertices of G, and the adjacency matrix of G. The algebraic connectivity of G is the second smallest eigenvalue of the Laplacian matrix (Fiedler, 1973). In this paper, we apply AGX to get structural conjectures for graphs with minimum and maximum algebraic connectivity given their order $n = |V|$ and size $m = |E|$, as well as implied bounds on the algebraic connectivity.

The paper is organized as follows. Definitions, notation and basic results on algebraic connectivity are recalled in the next section. Graphs with minimum algebraic connectivity are studied in Section 3; it is conjectured that they are path-complete graphs (Harary, 1962; Soltès, 1991); a lower bound on $a(G)$ is proved for one family of such graphs. Graphs with maximum algebraic connectivity are studied in Section 4. Extremal graphs are shown to be complements of disjoint triangles, paths P_3, edges K_2 and isolated vertices K_1. A best possible upper bound on $a(G)$ in function of m is then found and proved.

2. Definitions and basic results concerning algebraic connectivity

Consider again a graph $G = \big(V(G), E(G)\big)$ such that $V(G)$ is the set of vertices with cardinality n and $E(G)$ is the set of edges with cardinality m. Each $e \in E(G)$ is represented by $e_{ij} = \{v_i, v_j\}$ and in this case, we say that v_i is *adjacent* to v_j. The *adjacency matrix* $A = [a_{ij}]$ is an $n \times n$ matrix such that $a_{ij} = 1$, when v_i and v_j are adjacent and $a_{ij} = 0$, otherwise. The *degree* of v_i, denoted $d(v_i)$, is the number of edges incident with v_i. The *maximum degree* of G, $\Delta(G)$, is the largest vertex degrees of G. The *minimum degree* of G, $\delta(G)$, is defined analogously. *The vertex (or edge) connectivity* of G, $\kappa(G)$ (or $\kappa'(G)$) is the minimum number of vertices (or edges) whose removal from G results in a disconnected graph or a trivial one. A *path* from v to w

in G is a sequence of distinct vertices starting with v and ending with w such that consecutive vertices are adjacent. Its length is equal to its number of edges. A graph is connected if for every pair of vertices, there is a *path* linking them. The distance $d_G(v, w)$ between two vertices v and w in a connected graph is the length of the shortest path from v to w. The *diameter* of a graph G, d_G, is the maximum distance between two distinct vertices. A path in G from a node to itself is referred to as a *cycle*. A connected acyclic graph is called a *tree*. A complete graph, K_n, is a graph with n vertices such that for every pair of vertices there is an edge. A *clique* of G is an induced subgraph of G which is complete. The size of the largest clique, denoted $\omega(G)$, is called *clique number*. An empty graph, or a trivial one, has an empty edge set. A set of pairwise non adjacent vertices is called an *independent set*. The size of the largest independent set, denoted $\alpha(G)$, is the independence number. For further definitions see Godsil and Royle (2001).

As mentionned above, the *Laplacian* of a graph G is defined as the $n \times n$ matrix

$$L(G) = \triangle - A, \tag{1.1}$$

when A is the adjacency matrix of G and $\triangle$ is the diagonal matrix whose elements are the vertex degrees of G, called the *degree matrix* of G. $L(G)$ can be associated with a positive semidefinite quadratic form, as we can see in the following proposition:

PROPOSITION 1.1 (MERRIS, 1994) *Let G be a graph. If the quadratic form related to $L(G)$ is*

$$q(x) = xL(G)x^t, \quad x \in \Re^n,$$

then q is positive semidefinite.

The polynomial $p_{L(G)}(\lambda) = \det(\lambda I - L(G)) = \lambda^n + q_1\lambda^{n-1} + \cdots + q_{n-1}\lambda + q_n$ is called the *characteristic polynomial* of $L(G)$. Its *spectrum* is

$$\zeta(G) = (\lambda_1, \ldots, \lambda_{n-1}, \lambda_n), \tag{1.2}$$

where $\forall i$, $1 \leq i \leq n$, λ_i is an eigenvalue of $L(G)$ and $\lambda_1 \geq \ldots \geq \lambda_n$.

According to Proposition 1.1, $\forall i$, $1 \leq i \leq n$, λ_i is a non-negative real number. Fiedler (1973) defined λ_{n-1} as the *algebraic connectivity* of G, denoted $a(G)$.

We next recall some inequalities related to algebraic connectivity of graphs. These properties can be found in the surveys of Fiedler (1973) and Merris (1994).

PROPOSITION 1.2 *Let G_1 and G_2 be spanning graphs of G such that $E(G_1) \cap E(G_2) = \phi$. Then $a(G_1) + a(G_2) \leq a(G_1 \cup G_2)$.*

PROPOSITION 1.3 *Let G be a graph and G_1 a subgraph obtained from G by removing k vertices and all adjacent edges in G. Then*

$$a(G_1) \geq a(G) - k.$$

PROPOSITION 1.4 *Let G be a graph. Then,*
(1) $a(G) \leq [n/(n-1)]\delta(G) \leq 2|E|/(n-1)$;
(2) $a(G) \geq 2\delta(G) - n + 2$.

PROPOSITION 1.5 *Let G be a graph with n vertices and $G \neq K_n$. Suppose that G contains an independent set with p vertices. Then,*

$$a(G) \leq n - p.$$

PROPOSITION 1.6 *Let G be a graph with n vertices. If $G \neq K_n$ then $a(G) \leq n - 2$.*

PROPOSITION 1.7 *Let G be a graph with n vertices and m edges. If $G \neq K_n$ then*

$$a(G) \leq \left(\frac{2m}{n-1}\right)^{(n-1)/n}$$

PROPOSITION 1.8 *If $G \neq K_n$ then $a(G) \leq \delta(G) \leq \kappa(G)$. For $G = K_n$, we have $a(K_n) = n$ and $\delta(K_n) = \kappa(K_n) = n - 1$.*

PROPOSITION 1.9 *If G is a connected graph with n vertices and diameter d_G, then $a(G) \geq 4/nd_G$ and $d_G \leq \sqrt{2\Delta(G)/a(G)}\log_2(n^2)$.*

PROPOSITION 1.10 *Let T be a tree with n vertices and diameter d_T. Then,*

$$a(T) \leq 2\left[1 - \cos\left(\frac{\pi}{d_T + 1}\right)\right].$$

A partial graph of G is a graph G_1 such that $V(G_1) = V(G)$ and $E(G_1) \subset E(G)$.

PROPOSITION 1.11 *If G_1 is a partial graph of G then $a(G_1) \leq a(G)$.*

Moreover

PROPOSITION 1.12 *Consider a path P_n and a graph G with n vertices. Then, $a(P_n) \leq a(G)$.*

Consider graphs $G_1 = \big(V(G_1), E(G_1)\big)$ and $G_2 = \big(V(G_2), E(G_2)\big)$. The *Cartesian product* of G_1 and G_2 is a graph $G_1 \times G_2$ such that $V(G_1 \times G_2) = V(G_1) \times V(G_2)$ and $\big((u_1, u_2), (v_1, v_2)\big) \in E(G_1 \times G_2)$ if and only if either $u_1 = v_1$ and $(u_2, v_2) \in E(G_2)$ or $(u_1, v_1) \in E(G_1)$ and $u_2 = v_2$.

PROPOSITION 1.13 *Let G_1 and G_2 be graphs. Then,*

$$a(G_1 \times G_2) = \min\{a(G_1), a(G_2)\}.$$

3. Minimizing $a(G)$

When minimizing $a(G)$ we found systematically graphs belonging to a little-known family, called *path-complete graphs* by Soltès (1991). They were previously considered by Harary (1962) who proved that they are (non-unique) connected graphs with n vertices, m edges and maximum diameter. Soltès (1991) proved that they are the unique connected graphs with n vertices, m edges and maximum average distance between pairs of vertices. Path-complete graphs are defined as follows: they consist of a complete graph, an isolated vertex or a path and one or several edges joining one end vertex of the path (or the isolated vertex) to one or several vertices of the clique, see Figure 1.1 for an illustration. We will need a more precise definition:

For n and $t \in N$ when $1 \leq t \leq n-2$, we consider a new family of connected graphs with n vertices and $m_t(r)$ edges as follows:

$$G\big(n, m_t(r)\big) = \{G \mid \text{for } t \leq r \leq n-2,\ G \text{ has } m_t(r) \text{ edges}, \\ m_t(r) = (n-t)(n-t-1)/2 + r\}.$$

DEFINITION 1.1 Let $n, m, t, p \in \mathbb{N}$, with $1 \leq t \leq n-2$ and $1 \leq p \leq n-t-1$. A graph with n vertices and m edges such that

$$\frac{(n-t)(n-t-1)}{2} + t \leq m \leq \frac{(n-t)(n-t-1)}{2} + n - 2$$

is called (n, p, t) *path-complete graph*, denoted $\mathrm{PC}_{n,p,t}$, if and only if

(1) the maximal clique of $\mathrm{PC}_{n,p,t}$ is K_{n-t};
(2) $\mathrm{PC}_{n,p,t}$ has a t-path $P_{t+1} = [v_0, v_1, v_2, \ldots, v_t]$ such that $v_0 \in K_{n-t} \cap P_{t+1}$ and v_1 is joined to K_{n-t} by p edges;
(3) there are no other edges.

Figure 1.1 displays a (n, p, t) *path-complete graph.*

It is easy to see that all connected graphs with n vertices can be partitioned into the disjoint union of the following subfamilies:

$$G(n, m_1) \oplus G(n, m_2) \oplus \cdots \oplus G(n, m_{n-2}).$$

Besides, for every (n, p, t), $\mathrm{PC}_{n,p,t} \in G(n, m_t)$.

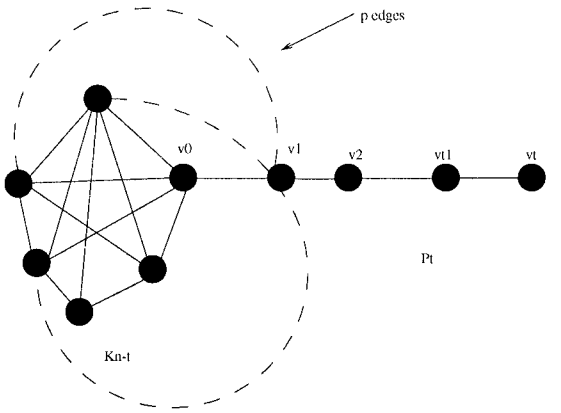


Figure 1.1. A (n, p, t) path-complete graph

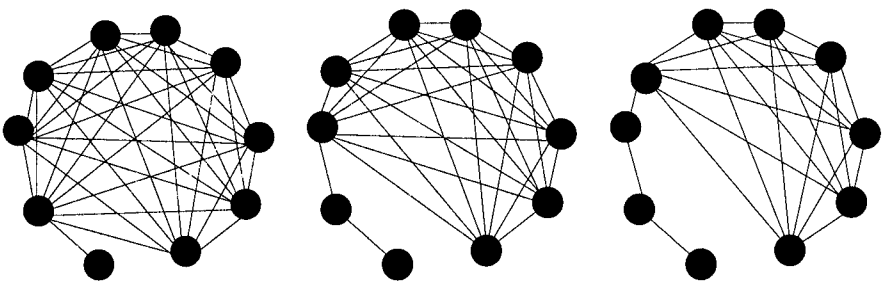


Figure 1.2. Path-complete graphs

3.1 Obtaining conjectures

Using AGX, connected graphs $G \neq K_n$ with (presumably) minimum algebraic connectivity were determined for $3 \leq n \leq 11$ and $n-1 \leq m \leq n(n-1)/2-1$. As all graphs turned out to belong to the same family, a structural conjecture was readily obtained.

CONJECTURE 1.1 *The connected graphs $G \neq K_n$ with minimum algebraic connectivity are all path-complete graphs.*

A few examples are given in Figure 1.2, for $n = 10$.

Numerical values of $a(G)$ for all extremal graphs found are given in Table 1.1, for $n = 10$ and $n-1 \leq m \leq n(n-1)/2-1$.

For each n, a piecewise concave function of m is obtained. From this table and the corresponding Figure 1.3 we obtain:

Table 1.1. $n = 10$; $\min a(G)$ on m

m	9	10	11	12	13	14	15	16	17
$a(G)$	0.097	0.103	0.109	0.115	0.123	0.134	0.137	0.151	0.170
m	18	19	20	21	22	23	24	25	26
$a(G)$	0.175	0.177	0.208	0.238	0.247	0.252	0.256	0.345	0.384
m	27	28	29	30	31	32	33	34	35
$a(G)$	0.406	0.419	0.428	0.435	0.673	0.801	0.876	0.924	0.957
m	36	37	38	39	40	41	42	43	44
$a(G)$	0.981	1	2	3	4	5	6	7	8

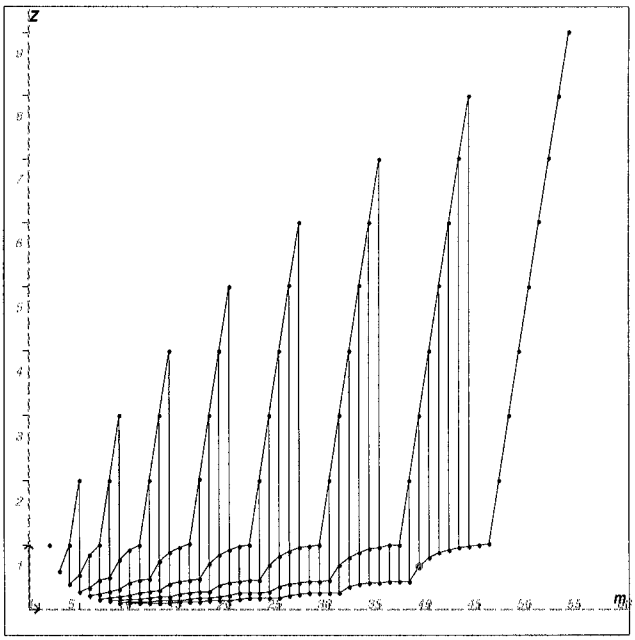


Figure 1.3. $\min a(G)$; $a(G)$ on m

CONJECTURE 1.2 *For each $n \geq 3$, the minimum algebraic connectivity of a graph G with n vertices and m edges is an increasing, piecewise concave function of m. Moreover, each concave piece corresponds to a family $\mathrm{PC}_{n,p,t}$ of path-complete graphs. Finally, for $t = 1$, $a(G) = \delta(G)$, and for $t \geq 2$, $a(G) \leq 1$.*

3.2 Proofs

We do not have a proof of Conjecture 1.1, nor a complete proof of Conjecture 1.2. However, we can prove some of the results of the latter.

We now prove that, under certain conditions, the algebraic connectivity of a path-complete graph minimizes the algebraic connectivity of every graph in $G(n, m_t)$, when $t = 1$ and $t = 2$.

PROPERTY 1.1 *Consider a path-complete graph* $\mathrm{PC}_{n,p,t}$.
(1) *For* $t = 1$, $a(\mathrm{PC}_{n,p,1}) = p$,
(2) *For* $t \geq 2$, $a(\mathrm{PC}_{n,1,t}) \leq a(\mathrm{PC}_{n,p,t}) \leq 1$.

Proof. Let us start with the second statement. According to the definition of path-complete graph, $\delta(R_{n,p,t}) = 1$, when $t \geq 2$. From Propositions 1.8 and 1.11, we obtain the following inequalities

$$a(\mathrm{PC}_{n,1,t}) \leq a(\mathrm{PC}_{n,p,t}) \leq \delta(\mathrm{PC}_{n,p,t}).$$

Therefore, $a(\mathrm{PC}_{n,p,t}) \leq 1$.

Now, consider the first statement. Let $t = 1$ and $\overline{\mathrm{PC}_{n,p,1}}$ be the complement graph of $\mathrm{PC}_{n,p,1}$. Figure 1.4 shows both graphs, $\overline{\mathrm{PC}_{n,p,1}}$ and $\mathrm{PC}_{n,p,1}$.

$\overline{\mathrm{PC}_{n,p,1}}$ has p isolated vertices and one connected component isomorphic to $K_{1,n-p-1}$. Its Laplacian matrix is,

$$L(\overline{\mathrm{PC}_{n,p,1}}) = \begin{bmatrix} L(K_{1,n-p-1}) & 0 \\ 0 & 0 \end{bmatrix}.$$

From Biggs (1993), we have

$$\det[L(K_{1,b}) - \lambda I_{b+1}] = \lambda[\lambda - (b+1)](\lambda - 1)^{b-1}.$$

Then,

$$\begin{aligned} det[L(\overline{\mathrm{PC}_{n,p,1}}) - \lambda I_n] &= (-\lambda)^p \det[L(K_{1,n-p-1}) - \lambda I_{n-p}] \\ &= (-\lambda)^p \lambda[\lambda - (n-p)](\lambda - 1)^{n-p-2}. \end{aligned}$$

According to Merris (1994), if $\zeta(G) = (\lambda_n, \lambda_{n-1}, \ldots, \lambda_2, 0)$ then $\zeta(\overline{G}) = (n - \lambda_2, n - \lambda_3, \ldots, n - \lambda_n, 0)$. So, we have

$$\begin{aligned} \zeta(\overline{\mathrm{PC}_{n,p,1}}) &= (n-p, 1, \ldots, 1, 0, \ldots, 0) \\ \zeta(\mathrm{PC}_{n,p,1}) &= (n, \ldots, n, n-1, \ldots, n-1, p, 0). \end{aligned}$$

Consequently, $a(\mathrm{PC}_{n,p,1}) = p$. □

PROPERTY 1.2 *For* $(n, p, 1)$ *path-complete graphs, we have* $\delta(\mathrm{PC}_{n,p,1}) = \kappa(\mathrm{PC}_{n,p,1}) = p$.

Proof. It follows from Definition 1.1, that $\delta(\mathrm{PC}_{n,p,1}) = p$. Applying Proposition 1.8 we obtain $a(\mathrm{PC}_{n,p,1}) \leq k(\mathrm{PC}_{n,p,1}) \leq p$. Since Property 1.1 gives $a(\mathrm{PC}_{n,p,1}) = p$ then $k(\mathrm{PC}_{n,p,1}) = p$. □

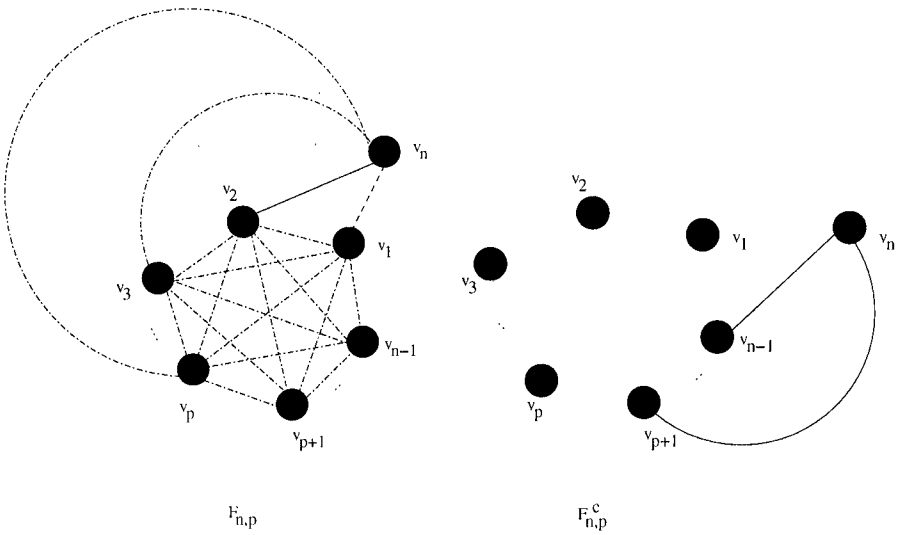


Figure 1.4. $\mathrm{PC}_{n,p,1}$ and its complement $\overline{\mathrm{PC}_{n,p,1}}$

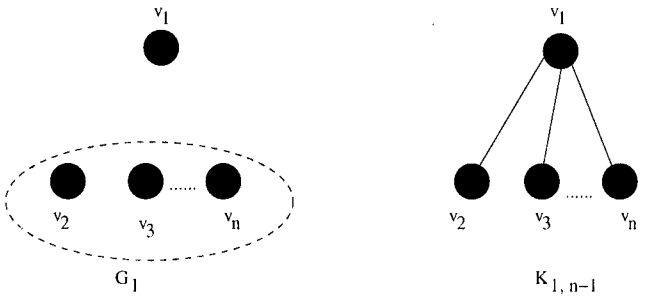


Figure 1.5. Graphs $K_{1,n-1}$ and G_1

PROPOSITION 1.14 *Among all* $G \in G(n, m_1)$ *with maximum degree* $n-1$, $a(G)$ *is minimized by* $\mathrm{PC}_{n,1,1}$.

Proof. Let G be a graph with n vertices. Consider spanning graphs of G $K_{1,n-1}$ and G_1 such that $E(K_{1,n-1}) \cap E(G_1) = \phi$ and G_1 has two connected components, one of them with $n-1$ vertices. Figure 1.5 shows these graphs.

We may consider $G = (V, E)$ where $V(G) = V(K_{1,n-1}) = V(G_1)$ and $E(G) = E(K_{1,n-1}) \cup E(G_1)$. Then, $\Delta(G) = n - 1$. According to Proposition 1.2, we have $a(K_{1,n-1}) + a(G_1) \leq a(G)$. From Biggs (1993), $a(K_{1,n-1}) = 1$. Since G_1 is a disconnected graph then $a(G_1) = 0$. However, $a(\mathrm{PC}_{n,1,1}) = 1$, therefore $a(G) \geq 1$. □

PROPOSITION 1.15 *For every $G \in G(n, m_1)$ such that $\delta(G) \geq (n-2)/2 + p/2$, where $1 \leq p \leq n-2$, we have*

$$a(G) \geq a(\mathrm{PC}_{n,p,1}) = p.$$

Proof. Consider $G \in G(n, m_1)$ with $\delta(G) \geq (n-2)/2 + p/2$. According to Proposition 1.4, we have

$$a(G) \geq 2\delta(G) - n + 2 \geq 2\left\lceil \frac{n-2}{2} + \frac{p}{2} \right\rceil - n + 2 = p.$$

Consequently, $a(G) \geq a(\mathrm{PC}_{n,p,1}) = p$. □

PROPOSITION 1.16 *For every $G \in G(n, m_2)$ such that $\delta(G) \geq (n-1)/2$, we have*

$$a(G) \geq 1 \geq a(\mathrm{PC}_{n,p,2}).$$

Proof. Consider $G \in G(n, m_2)$ with $\delta(G) \geq (n-1)/2$. According to Proposition 1.4, we have

$$a(G) \geq 2\delta(G) - n + 2 \geq 2\left\lceil \frac{n-1}{2} \right\rceil - n + 2 = 1.$$

From Property 1.1, $a(\mathrm{PC}_{n,p,2}) \leq 1$. Then, $a(G) \geq 1 \geq a(\mathrm{PC}_{n,p,2})$. □

To close this section we recall a well-known result.

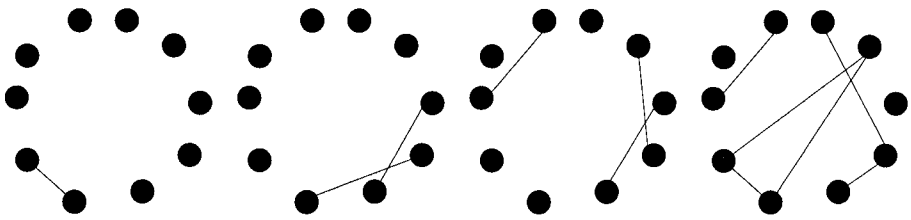
PROPOSITION 1.17 *Let T be a tree with n vertices. For every T, $a(T)$ is minimized by the algebraic connectivity of a single path P_n, where $a(P_n) = 2[1 - \cos(\pi/n)]$. Moreover, for every graph G with n vertices $a(P_n) \leq a(G)$.*

4. Maximizing $a(G)$

4.1 Obtaining conjectures

Using AGX, connected graphs $G \neq K_n$ with (presumably) maximum algebraic connectivity $a(G)$ were determined for $3 \leq n \leq 10$ and $(n-1)(n-2)/2 \leq m \leq n(n-1)/2 - 1$. We then focused on those among them with maximum $a(G)$ for a given m. These graphs having many edges, it is easier to understand their structure by considering their complement $\overline{G}$. It appears that these $\overline{G}$ are composed of disjoint triangles K_3, paths P_3, edges K_2 and isolated vertices K_1.

A representative subset of these graphs $\overline{G}$ is given in Figure 1.6.



$\overline{G}(n = 10; m = 44)$ $\overline{G}(n = 10; m = 43)$ $\overline{G}(n = 10; m = 42)$ $\overline{G}(n = 10; m = 39)$

Figure 1.6.

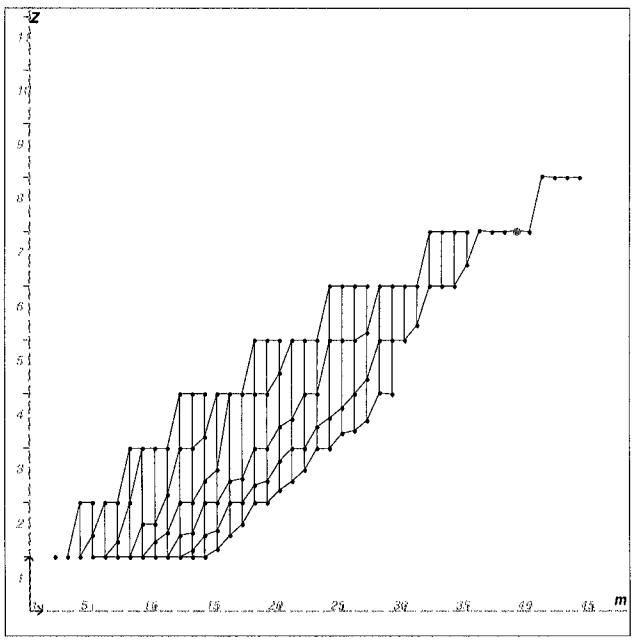


Figure 1.7. $\max a(G)$; $a(G)$ on m

CONJECTURE 1.3 *For all $m \geq 2$ there is a graph $G \neq K_n$ with maximum algebraic connectivity $a(G)$ the complement $\overline{G}$ of which is the disjoint union of triangles K_3, paths P_3, edges K_2 and isolated vertices K_1.*

Values of $a(G)$ for all extremal graphs obtained by AGX are represented in function of m in Figure 1.7.

It appears that the maximum $a(G)$ follow an increasing "staircase" with larger and larger steps. Values of $a(G)$, m and n for the graphs of this staircase (or upper envelope) are listed in Table 1.2.

An examination of Table 1.2 leads to the next conjecture.

Table 1.2. Value of $a(G)$, m and n for graphs, with maximum $a(G)$ for m given, found by AGX

$a(G)$	.	1	1	2	2	2	2	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5
m	.	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
n	.	3	4	4	4	5	5	5	5	6	6	6	6	6	7	7	7	7	7	7	7	7
$a(G)$	5	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	8	8	8	8	8
m	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
n	8	8	8	8	8	8	8	9	9	9	9	9	9	9	9	10	10	10	10	10	10	10

CONJECTURE 1.4 *For all $n \geq 4$ there are $n-1$ consecutive values of m (beginning at 3) for which a graph $G \neq K_n$ with maximum algebraic connectivity $a(G)$ has n vertices. Moreover, for the first $\lfloor (n-1)/2 \rfloor$ of them $a(G) = n-2$ and for the last $\lceil (n-1)/2 \rceil$ of them $a(G) = n-3$.*

Considering the successive values of $a(G)$ for increasing m, it appears that for $a(G) = 2$ onwards their multiplicities are 4, 4, 6, 6, 8, 8, ... After a little fitting, this leads to the following observation:

$$\left\lceil \frac{a(G)(a(G)+2)}{2} \right\rceil \leq m$$

and to our final conjecture:

CONJECTURE 1.5 *If G is a connected graph such that $G \neq K_n$ then*

$$a(G) \leq \lfloor -1 + \sqrt{1+2m} \rfloor$$

and this bound is sharp for all $m \geq 2$.

One can easily see that this conjecture improves the bound already given in Proposition 1.7, i.e., $a(G) \leq \left(2m/(n-1)\right)^{(n-1)/n}$.

4.2 Proofs

We first prove Conjectures 1.3 and 1.4. Then, we present a proof for the last conjecture. The extremal graphs found point the way.

Proof of Conjectures 1.3 and 1.4. From Propositions 1.6 and 1.8 if $G \neq K_n$, $a(G) \leq \delta(G) \leq n-2$. For this last bound to hold as an equality one must have $\delta(G) = n-2$, which implies G must contain all edges except up to $\lfloor n/2 \rfloor$ of them, i.e., $n(n-1)/2 - \lfloor n/2 \rfloor \leq m \leq n(n-1)/2 - 1$. Moreover, the missing edges of G (or edges of $\overline{G}$) must form a matching. Assume there are $1 \leq r \leq \lfloor n/2 \rfloor$ missing edges and that they form a

matching. Then from Merris (1994) $\det[L(\overline{G})-\lambda I_n] = -\lambda^{n-2r}\lambda^r(\lambda-2)^r$. Hence

$$\zeta(\overline{G}) = (\underbrace{2,\ldots,2}_{r \text{ times}},0,\ldots,0), \quad \zeta(G) = (\underbrace{n,\ldots,n}_{n-r-1 \text{ times}},\underbrace{n-2,\ldots,n-2}_{r \text{ times}},0)$$

and $a(G) = n-2$. If there are $r > \lfloor n/2 \rfloor$ missing edges in G, $a(G) \leq \delta(G) \leq n-3$. Several cases must be considered to show that this bound is sharp, in all of which $r \leq n$, as otherwise $\delta(G) < n-3$. Moreover, one may assume $r \leq n-1$ or otherwise there is a smaller n such that all edges can be used and with $\delta(G)$ as large or larger:

(i) $r \bmod 3 = 0$. Then there is a $t \in \mathbb{N}$ such that $r = 3t$. Assume the missing edges of G form disjoint triangles in $\overline{G}$. Then (Biggs, 1993)

$$\det[L(K_3) - \lambda I_3] = \lambda(\lambda-3)^2$$

and

$$\det[L(\overline{G}) - \lambda I_n] = (-\lambda)^{n-r}\lambda^t(\lambda-3)^{2t}.$$

Hence

$$\zeta(\overline{G}) = (\underbrace{3,\ldots,3}_{2t \text{ times}},0,\ldots,0),$$

$$\zeta(\overline{G}) = (\underbrace{n,\ldots,n}_{n-2t-1 \text{ times}},\underbrace{n-3,\ldots,n-3}_{2t \text{ times}},0)$$

and $a(G) = n-3$.

(ii) $r \bmod 3 = 1$. Then there is a $t \in \mathbb{N}$ such that $r = 3t+1$. Assume the missing edges of G form t disjoint triangles and a disjoint edge. Then, as above,

$$\det[L(\overline{G}) - \lambda I_n] = (-\lambda)^{n-r-1}\lambda^{(r+2)/3}(\lambda-2)(\lambda-3)^{(2r-2)/3},$$

and $a(G) = n-3$.

(iii) $r \bmod 3 = 2$. Then there is a $t \in \mathbb{N}$ such that $r = 3t+2$. Assume the missing edges of G form t disjoint triangles and a disjoint path P_3 with 2 edges. From the characteristic polynominal of $L(P_3)$ and similar arguments as above one gets $a(G) = n-3$. □

Proof of Conjecture 1.5. Let $S \neq K_n$ a graph with all edges except up to $\lfloor n/2 \rfloor$ of them. So, $n(n-1)/2 - \lfloor n/2 \rfloor \leq m \leq n(n-1)/2 - 1$.

(i) If n is odd then,

$$\frac{n(n-1)}{2} - \frac{n-1}{2} \leq m \leq \frac{n(n-1)}{2} - 1.$$

Since $n^2 - 2n + 1/2 \geq n(n-2)/2$, $m \geq n(n-2)/2$.

(ii) If n is even, then

$$\frac{n(n-1)}{2} - \frac{n}{2} \leq m \leq \frac{n(n-1)}{2} - 1.$$

So, $2m \geq n(n-2)$ and $n - 2 \leq \lfloor -1 + \sqrt{1+2m} \rfloor$. From Proposition 1.6, $a(G) \leq n-2$. Then, $a(G) \leq \lfloor -1 + \sqrt{1+2m} \rfloor$.

Now, consider $(n-1)(n-2)/2 \leq m \leq n(n-1)/2 - (\lfloor n/2 \rfloor + 1)$. This way, $m = n(n-1)/2 - r$, with $\lfloor n/2 \rfloor + 1 \leq r \leq n-1$. So, $r \leq \frac{3}{2}(n-1)$. We can add n^2 to each side of the inequality above. After some algebraic manipulations, we get $(n-2)^2 \leq 2m+1$. So, $n - 3 \leq -1 + \sqrt{2m+1}$.

From the proof of Conjecture 1.4, we have $a(S) \leq n-3$. Then, $a(S) \leq \lfloor -1 + \sqrt{1+2m} \rfloor$. As we can consider every $G \neq K_n$ with n vertices as a partial (spanning) graph of S, from Proposition 1.11, we then have $a(G) \leq a(S) \leq \lfloor -1 + \sqrt{1+2m} \rfloor$. □

References

Appel, K. and Haken, W. (1977a). Every planar map is four colorable. I. Discharging. *Illinois Journal of Mathematics*, 21:429–490.

Appel, K. and Haken, W. (1977b). Every planar map is four colorable. II. Reducibility. *Illinois Journal of Mathematics*, 21:491–567.

Appel, K. and Haken, W. (1989). *Every Planar Map Is Four Colorable.* Contemporary Mathematics, vol. 98. American Mathematical Society, Providence, RI.

Biggs, N. (1993). *Algebraic Graph Theory*, 2 ed. Cambridge University Press.

Caporossi, G. and Hansen, P. (2000). Variable neighborhood search for extremal graphs. I. The AutoGraphiX system. *Discrete Mathematics*, 212:29–44.

Caporossi, G. and Hansen, P. (2004). Variable neighborhood search for extremal graphs. V. Three ways to automate finding conjectures. *Discrete Mathematics*, 276:81–94.

Cvetković, D., Kraus, L., and Simić, S. (1981). *Discussing Graph Theory with a Computer.* I. *Implementation of Graph Theoretic Algorithms.* Univ. Beograd Publ. Elektrotehn. Fak, pp. 100–104.

Cvetković, D. and Kraus, L. (1983). *"Graph" an Expert System for the Classification and Extension of Knowledge in the Field of Graph Theory, User's Manual.* Elektrothen. Fak., Beograd.

Cvetković, D. and Simić, S. (1994). Graph-theoretical results obtained by the support of the expert system "graph." *Bulletin de l'Académie Serbe des Sciences et des Arts*, 19:19–41.

Diestel, R. (1997). *Graph Theory*, Springer.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298 – 305.

Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*, Springer.

Hansen, P. (2002). Computers in graph theory. *Graph Theory Notes of New York*, 43:20 – 34.

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449 – 467.

Harary, F. (1962). The maximum connectivity of a graph. *Proceedings of the National Academy of Sciences of the United States of America*, 48:1142 – 1146.

McKay, B.D. (1990). *Nauty User's Guide (Version 1.5)*. Technical Report, TR-CS-90-02, Department of Computer Science, Australian National University.

McKay, B.D. (1998). Isomorph-free exhaustive generation. *Journal of Algorithms*, 26:306 – 324.

Merris, R. (1994). Laplacian matrices of graphs: A survey. *Linear Algebra and its Applications*, 197/198:143 – 176.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, 24(11):1097 – 1100.

Robertson, N., Sanders, D., Seymour, P., and Thomas, R. (1997). The four-colour theorem. *Journal of Combinatorial Theory, Series B*, 70(1):2 – 44.

Soltès, L. (1991). Transmission in graphs: A bound and vertex removing. *Mathematica Slovaca*, 41(1):11 – 16.

Chapter 2

PROBLEMS AND RESULTS ON GEOMETRIC PATTERNS

Peter Brass
János Pach

Abstract Many interesting problems in combinatorial and computational geometry can be reformulated as questions about occurrences of certain patterns in finite point sets. We illustrate this framework by a few typical results and list a number of unsolved problems.

1. Introduction: Models and problems

We discuss some extremal problems on repeated geometric patterns in finite point sets in Euclidean space. Throughout this paper, a *geometric pattern* is an equivalence class of point sets in d-dimensional space under some fixed geometrically defined equivalence relation. Given such an equivalence relation and the corresponding concept of patterns, one can ask several natural questions:

(1) *What is the maximum number of occurrences of a given pattern among all subsets of an n-point set?*
(2) *How does the answer to the previous question depend on the particular pattern?*
(3) *What is the minimum number of distinct k-element patterns determined by a set of n points?*

These questions make sense for many specific choices of the underlying set and the equivalence relation. Hence it is not surprising that several basic problems of combinatorial geometry can be studied in this framework (Pach and Agarwal, 1995).

In the simplest and historically first examples, due to Erdős (1946), the underlying set consists of point pairs in the plane and the defining equivalence relation is the isometry (congruence). That is, two point pairs, $\{p_1, p_2\}$ and $\{q_1, q_2\}$, determine the same pattern if and only if

$|p_1 - p_2| = |q_1 - q_2|$. In this case, (1) becomes the well-known *Unit Distance Problem*: What is the maximum number of unit distance pairs determined by n points in the plane? It follows by scaling that the answer does not depend on the particular distance (pattern). For most other equivalence relations, this is not the case: different patterns may have different maximal multiplicities. For $k = 2$, question (3) becomes the *Problem of Distinct Distances*: What is the minimum number of distinct distances that must occur among n points in the plane? In spite of many efforts, we have no satisfactory answers to these questions. The best known results are the following.

THEOREM 2.1 (SPENCER ET AL., 1984) *Let $f(n)$ denote the maximum number of times the same distance can be repeated among n points in the plane. We have*

$$ne^{\Omega(\log n/\log\log n)} \leq f(n) \leq O(n^{4/3}).$$

THEOREM 2.2 (KATZ AND TARDOS, 2004) *Let $g(n)$ denote the minimum number of distinct distances determined by n points in the plane. We have*

$$\Omega(n^{0.8641}) \leq g(n) \leq O\left(\frac{n}{\sqrt{\log n}}\right).$$

In Theorems 2.1 and 2.2, the lower and upper bounds, respectively, are conjectured to be asymptotically sharp. See more about these questions in Section 3.

Erdős and Purdy (1971, 1977) initiated the investigation of the analogous problems with the difference that, instead of pairs, we consider *triples* of points, and call two of them *equivalent* if the corresponding triangles have the same angle, or area, or perimeter. This leads to questions about the maximum number of equal angles, or unit-area resp. unit-perimeter triangles, that can occur among n points in the plane, and to questions about the minimum number of distinct angles, triangle areas, and triangle perimeters, respectively. Erdős's Unit Distance Problem and his Problem of Distinct Distances has motivated a great deal of research in extremal graph theory. The questions of Erdős and Purdy mentioned above and, in general, problems (1), (2), and (3) for larger than two-element patterns, require the extension of graph theoretic methods to hypergraphs. This appears to be one of the most important trends in modern combinatorics.

Geometrically, it is most natural to define two sets to be *equivalent* if they are congruent or similar to, or translates, homothets or affine images of each other. This justifies the choice of the word "pattern" for the resulting equivalence classes. Indeed, the algorithmic aspects

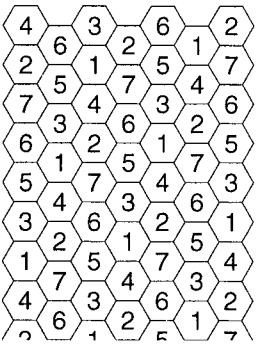


Figure 2.1. Seven coloring of the plane showing that $\chi(\mathbb{R}^2) \leq 7$

of these problems have also been studied in the context of geometric pattern matching (Akutsu et al., 1998; Brass, 2000; Agarwal and Sharir, 2002; Brass, 2002). A typical algorithmic question is the following.

(4) *Design an efficient algorithm for finding all occurrences of a given pattern in a set of n points.*

It is interesting to compare the equivalence classes that correspond to the same relation applied to patterns of different sizes. If A and A' are equivalent under congruence (or under some other group of transformations mentioned above), and a is a point in A, then there exists a point $a' \in A'$ such that $A \setminus \{a\}$ is equivalent to $A' \setminus \{a'\}$. On the other hand, if A is equivalent (congruent) to A' and A is large enough, then usually its possible extensions are also determined: for each a, there exist only a small number of distinct elements a' such that $A \cup \{a\}$ is equivalent to $A' \cup \{a'\}$. Therefore, in order to bound the number of occurrences of a large pattern, it is usually sufficient to study small pattern fragments.

We have mentioned above that one can rephrase many extremal problems in combinatorial geometry as questions of type (1) (so-called *Turán-type* questions). Similarly, many *Ramsey-type* geometric coloring problems can also be formulated in this general setting.

(5) *Is it possible to color space with k colors such that there is no monochromatic occurrence of a given pattern?*

For point pairs in the plane under congruence, we obtain the famous Hadwiger–Nelson problem (Hadwiger, 1961): What is the smallest number of colors $\chi(\mathbb{R}^2)$ needed to color all points of the plane so that no two points at unit distance from each other get the same color?

THEOREM 2.3 $4 \leq \chi(\mathbb{R}^2) \leq 7$.

Another instance of question (5) is the following open problem from Erdős et al. (1973): Is it possible to color all points of the three-dimensional Euclidean space with three colors so that no color class contains two vertices at distance one and the midpoint of the segment determined by them? It is known that four colors suffice, but there exists no such coloring with two colors. In fact, Erdős et al. (1973) proved that for every d, the Euclidean d-space can be colored with four colors without creating a monochromatic triple of this kind.

2. A simple sample problem: Equivalence under translation

We illustrate our framework by analyzing the situation in the case in which two point sets are considered equivalent if and only if they are translates of each other. In this special case, we know the (almost) complete solution to problems (1)–(5) listed in the Introduction.

THEOREM 2.4 *Any set B of n points in d-dimensional space has at most $n+1-k$ subsets that are translates of a fixed set A of k points. This bound is attained if and only if $A = \{p, p+v, \ldots, p+(k-1)v\}$ and $B = \{q, q+v, \ldots, q+(n-1)v\}$ for some $p, q, v \in \mathbb{R}^d$.*

The proof is simple. Notice first that no linear mapping φ that keeps all points of B distinct decreases the maximum number of translates: if $A + t \subset B$, then $\varphi(A) + \varphi(t) \subset \varphi(B)$. Thus, we can use any projection into $\mathbb{R}$, and the question reduces to the following one-dimensional problem: Given real numbers $a_1 < \cdots < a_k$, $b_1 < \ldots, b_n$, what is the maximum number of values t such that $t + \{a_1, \ldots, a_k\} \subset \{b_1, \ldots b_n\}$. Clearly, $a_1 + t$ must be one of $b_1, \ldots, b_{n-k+1}$, so there are at most $n+1-k$ translates. If there are $n+1-k$ translates $t + \{a_1, \ldots, a_k\}$ that occur in $\{b_1, \ldots b_n\}$, for translation vectors $t_1 < \cdots < t_{n-k+1}$, then $t_i = b_i - a_1 = b_{i+1} - a_2 = b_{i+j} - a_{1+j}$, for $i = 1, \ldots, n-k+1$ and $j = 0, \ldots, k-1$. But then $a_2 - a_1 = b_{i+1} - b_i = a_{j+1} - a_j = b_{i+j} - b_{i+j-1}$, so all differences between consecutive a_j and b_i are the same. For higher-dimensional sets, this holds for every one-dimensional projection, which guarantees the claimed structure. In other words, the maximum is attained only for sets of a very special type, which answers question (1).

An asymptotically tight answer to (2), describing the dependence on the particular pattern, was obtained in Brass (2002).

THEOREM 2.5 *Let A be a set of points in d-dimensional space, such that the rational affine space spanned by A has dimension k. Then the maximum number of translates of A that can occur among n points in d-dimensional space is $n - \Theta(n^{(k-1)/k})$.*

Any set of the form $\{p, p+v, \ldots, p+(k-1)v\}$ spans a one-dimensional rational affine space. An example of a set spanning a two-dimensional rational affine space is $\{0, 1, \sqrt{2}\}$, so for this set there are at most $n - \Theta(n^{1/2})$ possible translates. This bound is attained, e.g., for the set $\{i + j\sqrt{2} \mid 1 \leq i, j \leq \sqrt{n}\}$.

In this case, it is also easy to answer question (3), i.e., to determine the minimum number of distinct patterns (translation-inequivalent subsets) determined by an n-element set.

THEOREM 2.6 *Any set of n points in d-dimensional space has at least $\binom{n-1}{k-1}$ distinct k-element subsets, no two of which are translates of each other. This bound is attained only for sets of the form $\{p, p+v, \ldots, p+(n-1)v\}$ for some $p, v \in \mathbb{R}^d$.*

By projection, it is again sufficient to prove the result on the line. Let $f(n,k)$ denote the minimum number of translation inequivalent k-element subsets of a set of n real numbers. Considering the set $\{1, \ldots, n\}$, we obtain that $f(n,k) \leq \binom{n-1}{k-1}$, since every equivalence class has a unique member that contains 1. To establish the lower bound, observe that, for any set of n real numbers, there are $\binom{n-2}{k-2}$ distinct subsets that contain both the smallest and the largest numbers, and none of them is translation equivalent to any other. On the other hand, there are at least $f(n-1,k)$ translation inequivalent subsets that do not contain the last element. So we have $f(n,k) \geq f(n-1,k) + \binom{n-2}{k-2}$, which, together with $f(n,1) = 1$, proves the claimed formula. To verify the structure of the extremal set, observe that, in the one-dimensional case, an extremal set minus its first element, as well as the same set minus its last element, must again be extremal sets, and for $n = k+1$ it follows from Theorem 2.4 that all extremal sets must form arithmetic progressions. Thus, the whole set must be an arithmetic progression, which holds, in higher-dimensional cases, for each one-dimensional projection.

The corresponding algorithmic problem (4) has a natural solution: Given two sets, $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_n\}$, we can fix any element of A, say, a_1, and try all possible image points b_i. Each of them specifies a unique translation $t = b_i - a_1$, so we simply have to test for each set $A + (b_i - a_1)$ whether it is a subset of B. This takes $\Theta(kn \log n)$ time. The running time of this algorithm is not known to be optimal.

PROBLEM 1 *Does there exist an $o(kn)$-time algorithm for finding all real numbers t such that $t + A \subset B$, for every pair of input sets A and B consisting of k and n reals, respectively?*

The Ramsey-type problem (5) is trivial for translates. Given any set A of at least two points $a_1, a_2 \in A$, we can two-color $\mathbb{R}^d$ without generating

any monochromatic translate of A. Indeed, the space can be partitioned into arithmetic progressions with difference $a_2 - a_1$, and each of them can be colored separately with alternating colors.

3. Equivalence under congruence in the plane

Problems (1) – (5) are much more interesting and difficult under congruence as the equivalence relation. In the plane, considering two-element subsets, the congruence class of a pair of points is determined by their distance. Questions (1) and (3) become the Erdős's famous problems, mentioned in the Introduction.

PROBLEM 2 *What is the maximum number of times the same distance can occur among n points in the plane?*

PROBLEM 3 *What is the minimum number of distinct distances determined by n points in the plane?*

The best known results concerning these questions were summarized in Theorems 2.1 and 2.2, respectively. There are several different proofs known for the currently best upper bound in Theorem 2.1 (see Spencer et al., 1984; Clarkson et al., 1990; Pach and Agarwal, 1995; Székely, 1997), which obviously does not depend on the particular distance (congruence class). This answers question (2). As for the lower bound of Katz and Tardos (2004) in Theorem 2.2, it represents the latest improvement over a series of previous results (Solymosi and Tóth, 2001; Székely, 1997; Chung et al., 1992; Chung, 1984; Beck, 1983; Moser1952).

The algorithmic problem (4) can now be stated as follows.

PROBLEM 4 *How fast can we find all unit distance pairs among n points in the plane?*

Some of the methods developed to establish the $O(n^{4/3})$ bound for the number of unit distances can also be used to design an algorithm for finding all unit distance pairs in time $O(n^{4/3} \log n)$ (similar to the algorithms for detecting point-line incidences; Matoušek, 1993).

The corresponding Ramsey-type problem (5) for patterns of size two is the famous Hadwiger – Nelson problem; see Theorem 2.3 above.

PROBLEM 5 *What is the minimum number of colors necessary to color all points of the plane so that no pair of points at unit distance receive the same color?*

If we ask the same questions for patterns of size k rather than point pairs, but still in the plane, the answer to (1) does not change. Given

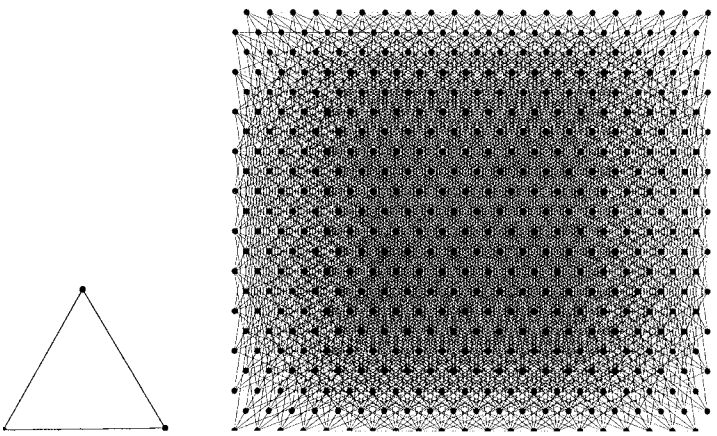


Figure 2.2. A unit equilateral triangle and a lattice section containing many congruent copies of the triangle

a pattern $A = \{a_1, \ldots, a_k\}$, any congruent image of A is already determined, up to reflection, by the images of a_1 and a_2. Thus, the maximum number of congruent copies of a set is at most twice the maximum number of (ordered) unit distance pairs. Depending on the given set, this maximum number may be smaller, but no results of this kind are known. As n tends to infinity, the square and triangular lattice constructions that realize $ne^{c \log n / \log \log n}$ unit distances among n points also contain roughly the same number of congruent copies of *any* fixed set that is a subset of a square or triangular lattice. However, it is likely that this asymptotics cannot be attained for most other patterns.

PROBLEM 6 *Does there exist, for every finite set A, a positive constant $c(A)$ with the following property: For every n, there is a set of n points in the plane containing at least $ne^{c(A) \log n / \log \log n}$ congruent copies of A?*

The answer is yes if $|A| = 3$.

Problem (3) on the minimum number of distinct congruence classes of k-element subsets of a point set is strongly related to the Problem of Distinct Distances, just like the maximum number of pairwise congruent subsets was related to the Unit Distance Problem. For if we consider ordered k-tuples instead of k-subsets (counting each subset $k!$ times), then two such k-tuples are certainly incongruent if their first two points determine distinct distances. For each distance s, fix a point pair that determines s. Clearly, any two different extensions of a point pair by filling the remaining $k-2$ positions result in incongruent k-tuples. This leads to a lower bound of $\Omega(n^{k-2+0.8641})$ for the minimum number of

distinct congruence classes of k-element subsets. Since a regular n-gon has $O(n^{k-1})$ pairwise incongruent k-element sets, this problem becomes less interesting for large k.

The algorithmic question (4) can also be reduced to the corresponding problem on unit distances. Given the sets A and B, we first fix $a_1, a_2 \in A$ and use our algorithm developed for detecting unit distance pairs to find all possible image pairs $b_1, b_2 \in B$ whose distance is the same as that of a_1 and a_2. Then we check for each of these pairs whether the rigid motion that takes a_i to b_i $(i = 1, 2)$ maps the whole set A into a subset of B. This takes $O^*(n^{4/3}k)$ time, and we cannot expect any substantial improvement in the dependence on n, unless we apply a faster algorithm for finding unit distance pairs. (In what follows, we write O^* to indicate that we ignore some lower order factors, i.e., $O^*(n^\alpha) = O(n^{\alpha+\varepsilon})$ for every $\varepsilon > 0$).

Many problems of Euclidean Ramsey theory can be interpreted as special cases of question (5) in our model. We particularly like the following problem raised in Erdős et al. (1975).

PROBLEM 7 *Is it true that, for any triple $A = \{a_1, a_2, a_3\} \subset \mathbb{R}^2$ that does not span an equilateral triangle, and for any coloring of the plane with two colors, one can always find a monochromatic congruent copy of A?*

It was conjectured in Erdős et al. (1975) that the answer to this question is yes. It is easy to see that the statement is not true for equilateral triangles A. Indeed, decompose the plane into half-open parallel strips whose widths are equal to the height of A, and color them red and blue, alternately. On the other hand, the seven-coloring of the plane, with no two points at unit distance whose colors are the same, shows that any given pattern can be avoided with seven colors. Nothing is known about coloring with three colors.

PROBLEM 8 *Does there exist a triple $A = \{a_1, a_2, a_3\} \subset \mathbb{R}^2$ such that any three-coloring of the plane contains a monochromatic congruent copy of A?*

4. Equivalence under congruence in higher dimensions

All questions discussed in the previous section can also be asked in higher dimensions. There are two notable differences. In the plane, the image of a fixed pair of points was sufficient to specify a congruence. Therefore, the number of congruent copies of any larger set was bounded from above by the number of congruent pairs. In d-space, however, one

has to specify d image points to determine a congruence, up to reflection. Hence, estimating the maximum number of congruent copies of a k-point set is a different problem for each $k = 2, \ldots, d$.

The second difference from the planar case is that starting from four dimensions, there exists another type of construction, discovered by Lenz, that provides asymptotically best answers to some of the above questions. For $k = \lfloor d/2 \rfloor$, choose k concentric circles of radius $1/\sqrt{2}$ in pairwise orthogonal planes in $\mathbb{R}^d$ and distribute n points on them as equally as possible. Then any two points from distinct circles are at distance one, so the number of unit distance pairs is $\left(\frac{1}{2} - 1/(2k) + o(1)\right)n^2$, which is a positive fraction of all point pairs. It is known (Erdős, 1960) that this constant of proportionality cannot be improved. Similarly, in this construction, any three points chosen from distinct circles span a unit equilateral triangle, so if $d \geq 6$, a positive fraction of all triples can be congruent. In general, for each $k \leq \lfloor d/2 \rfloor$, Lenz's construction shows that a positive fraction of all k-element subsets can be congruent. Obviously, this gives the correct order of magnitude for question (1). With some extra work, perhaps even the exact maxima can be determined, as has been shown for $k = 2$, $d = 4$ in Brass (1997) and van Wamelen (1999).

Even for $k > d/2$, we do not know any construction better than Lenz's, but for these parameters the problem is not trivial. Now one is forced to pick several points from the same circle, and only one of them can be selected freely. So, for $d = 3$, in the interesting versions of (1), we have $k = 2$ or 3 (now there is no Lenz construction). For $d \geq 4$, the cases $\lfloor d/2 \rfloor < k \leq d$ are nontrivial.

PROBLEM 9 *What is the maximum number of unit distances among n points in three-dimensional space?*

Here, the currently best bounds are $\Omega(n^{4/3} \log \log n)$ (Erdős, 1960) and $O^*(n^{3/2})$ (Clarkson et al., 1990).

PROBLEM 10 *What is the maximum number of pairwise congruent triangles spanned by a set of n points in three-dimensional space?*

Here the currently best lower and upper bounds are $\Omega(n^{4/3})$ (Erdős et al., 1989; Ábrego and Fernández-Merchant, 2002) and $O^*(n^{5/3})$ (Agarwal and Sharir, 2002), respectively. They improve previous results in Akutsu et al. (1998) and Brass (2000). For higher dimensions, Lenz's construction or, in the odd-dimensional cases, a combination of Lenz's construction with the best known three-dimensional point set (Erdős et al., 1989; Ábrego and Fernández-Merchant, 2002), are most likely to be

optimal. The only results in this direction, given in Agarwal and Sharir (2002), are for $d \leq 7$ and do not quite attain this bound.

PROBLEM 11 *Is it true that, for any $\lfloor d/2 \rfloor \leq k \leq d$, the maximum number of congruent k-dimensional simplices among n points in d-dimensional space is $O(n^{d/2})$ if d is even, and $O(n^{d/2-1/6})$ if d is odd?*

Very little is known about problem (2) in this setting. For point pairs, scaling again shows that all two-element patterns can occur the same number of times. For three-element patterns (triangles), the aforementioned $\Omega(n^{4/3})$ lower bound in Erdős et al. (1989) was originally established only for right-angle isosceles triangles. It was later extended in Ábrego and Fernández-Merchant (2002) to any fixed triangle. However, the problem is already open for full-dimensional simplices in 3-space. An especially interesting special case is the following.

PROBLEM 12 *What is the maximum number of orthonormal bases that can be selected from n distinct unit vectors?*

The upper bound $O(n^{4/3})$ is simple, but the construction of Erdős et al. (1989) that gives $O(n^{4/3})$ orthogonal pairs does not extend to orthogonal triples.

Question (3) on the minimum number of distinct patterns is largely open. For two-element patterns, we obtain higher-dimensional versions of the Problem of Distinct Distances. Here the upper bound $O(n^{2/d})$ is realized, e.g., by a cubic section of the d-dimensional integer lattice. The general lower bound of $\Omega(n^{1/d})$ was observed already in Erdős (1946). For $d = 3$, this was subsequently improved to $\Omega^*(n^{77/141})$ (Aronov et al., 2003) and to $\Omega(n^{0.564})$ (Solymosi and Vu, 2005). For large values of d, Solymosi and Vu (2005) got very close to finding the best exponent by establishing the lower bound $\Omega(n^{2/d-2/(d(d+2))})$. This extends, in the same way as in the planar case, to a bound of $\Omega(n^{k-2+2/d-2/(d(d+2))})$ for the minimum number of distinct k-point patterns of an n-element set, but even for triangles, nothing better is known. Lenz-type constructions are not useful in this context, because they span $\Omega(n^{k-1})$ distinct k-point patterns, as do regular n-gons.

As for the algorithmic problem (4), it is easy to find all congruent copies of a given k-point pattern A in an n-point set. For any $k \geq d$, this can be achieved in $O(n^d k \log n)$ time: fix a d-tuple C in A, and test all d-tuples of the n-point set B, whether they could be an image of C. If yes, test whether the congruence specified by them maps all the remaining $k - d$ points to elements of B. It is very likely that there are much faster algorithms, but, for general d, the only published improvement is by a factor of $\log n$ (de Rezende and Lee, 1995).

The Ramsey-type question (5) includes a number of problems of Euclidean Ramsey theory, as special cases.

PROBLEM 13 *Is it true that for every two-coloring of the three-dimensional space, there are four vertices of the same color that span a unit square?*

It is easy to see that if we divide the plane into half-open strips of width one and color them alternately by two colors, then no four vertices that span a unit square will receive the same color. On the other hand, it is known that any two-coloring of four-dimensional space will contain a monochromatic unit square (Erdős et al., 1975). Actually, the (vertex set of a) square is one of the simplest examples of a *Ramsey set*, i.e., a set B with the property that, for every positive integer c, there is a constant $d = d(c)$ such that under any c-coloring of the points of $\mathbb{R}^d$ there exists a monochromatic congruent copy of B. All boxes, all triangles (Frankl and Rödl, 1986), and all trapezoids (Kříž, 1992) are known to be Ramsey. It is a long-standing open problem to decide whether all finite subsets of finite dimensional spheres are Ramsey. If the answer is in the affirmative, this would provide a perfect characterization of Ramsey sets, for all Ramsey sets are known to be subsets of a sphere (Erdős et al., 1973).

The simplest nonspherical example, consisting of an equidistant sequence of three points along the same line, was mentioned at the end of the Introduction.

5. Equivalence under similarity

If we consider problems (1) – (5) with similarity (congruence and scaling) as the equivalence relation, again we find that many of the resulting questions have been extensively studied. Since any two point pairs are similar to each other, we can restrict our attention to patterns of size at least three. The first interesting instance of problem (1) is to determine or to estimate the maximum number of pairwise similar triangles spanned by n points in the plane. This problem was almost completely solved in Elekes and Erdős (1994). For any given triangle, the maximum number of similar triples in a set of n point in the plane is $\Theta(n^2)$. If the triangle is equilateral, we even have fairly good bounds on the multiplicative constants hidden in the Θ-notation (Ábrego and Fernández-Merchant, 2000). In this case, most likely, suitable sections of the triangular lattice are close to being extremal for (1). In general, the following construction from Elekes and Erdős (1994) always gives a quadratic number of similar copies of a given triangle $\{a, b, c\}$. Interpreting a, b, c as complex numbers 0, 1, z, consider the points $(i_1/n)z$,

$i_2/n+(1-i_2/n)z$, and $(i_3/n)z+(1-i_3/n)z^2$, where $0 < i_1, i_2, i_3 \leq n/3$. Then any triangle $(\beta-\alpha)z$, $\alpha+(1-\alpha)z$, $\beta z+(1-\beta)z^2$ is similar to 0, 1, z, which can be checked by computing the ratios of the sides. Thus, choosing $\alpha = i_2/n$, $\beta = i_3/n$, we obtain a quadratic number of similar copies of the triangle 0, 1, z.

The answer to question (1) for k-point patterns, $k > 3$, is more or less the same as for $k = 3$. Certain patterns, including all k-element subsets of a regular triangular lattice, permit $\Theta(n^2)$ similar copies, and in this case a suitable section of the triangular lattice is probably close to being extremal. For some other patterns, the order $\Theta(n^2)$ cannot be attained. All patterns of the former type were completely characterized in Laczkovich and Ruzsa (1997): for any pattern A of $k \geq 4$ points, one can find n points containing $\Theta(n^2)$ similar copies of A if and only if the cross ratio of every quadruple of points in A, interpreted as complex numbers, is algebraic. Otherwise, the maximum is slightly subquadratic. This result also answers question (2).

In higher dimensions, the situation is entirely different: we do not have good bounds for question (1) in any nontrivial case. The first open question is to determine the maximum number of triples in a set of n points in 3-space that induce pairwise similar triangles. The trivial upper bound, $O(n^3)$, was reduced to $O(n^{2.2})$ in Akutsu et al. (1998). On the other hand, we do not have any better lower bound than $\Omega(n^2)$, which is already valid in the plane. These estimates extend to similar copies of k-point patterns, $k > 3$, provided that they are planar.

PROBLEM 14 *What is the maximum number of pairwise similar triangles induced by n points in three-dimensional space?*

For full-dimensional patterns, no useful constructions are known. The only lower bound we are aware of follows from the lattice L which, in three dimensions, spans $\Omega(n^{4/3})$ similar copies of the full-dimensional simplex formed by its basis vectors or, in fact, of any k-element subset of lattice points. However, to attain this bound, we do not need to allow rotations: L spans $\Omega(n^{4/3})$ *homothetic* copies.

PROBLEM 15 *In three-dimensional space, what is the maximum number of quadruples in an n-point set that span pairwise similar tetrahedra?*

For higher dimensions and for larger pattern sizes, the best known lower bound follows from Lenz's construction for congruent copies, which again does not use the additional freedom of scaling. Since, for $d \geq 3$, we do not know the answer to question (1) on the maximum number occurrences, there is little hope that we would be able to answer question (2) on the dependence of this maximum number on the pattern.

Problem (3) on the minimum number of pairwise inequivalent patterns under similarity is an interesting problem even in the plane.

PROBLEM 16 *What is the minimum number of similarity classes of triangles spanned by a set of n points in the plane?*

There is a trivial lower bound of $\Omega(n)$: if we choose two arbitrary points, and consider all of their $n-2$ possible extensions to a triangle, then among these triangles each (oriented) similarity class will be represented only at most three times. Alternatively, we obtain asymptotically the same lower bound $\Omega(n)$ by just using the pigeonhole principle and the fact that the maximum size of a similarity class of triangles is $O(n^2)$. On the other hand, as shown by the example of a regular n-gon, the number of similarity classes of triangles can be $O(n^2)$. This leaves a huge gap between the lower and upper bounds.

For higher dimensions and for larger sets, our knowledge is even more limited. In three-dimensional space, for instance, we do not even have an $\Omega(n)$ lower bound for the number of similarity classes of triangles, while the best known upper bound, $O(n^2)$, remains the same. For four-element patterns, we have a linear lower bound (fix any triangle, and consider its extensions), but we have no upper bound better than $O(n^3)$ (consider again a regular n-gon). Here we have to be careful with the precise statement of the problem. We have to decide whether we count similarity classes of *full-dimensional* simplices only, or all similarity classes of possibly degenerate four-tuples. A regular $(n-1)$-gon with an additional point on its axis has only $\Theta(n^2)$ similarity classes of full-dimensional simplices, but $\Theta(n^3)$ similarity classes of four-tuples. In dimensions larger than three, nothing nontrivial is known.

In the plane, the algorithmic question (4) of finding all similar copies of a fixed k-point pattern is not hard: trivially, it can be achieved in time $O(n^2 k \log n)$, which is tight up to the $\log n$-factor, because the output complexity can be as large as $\Omega(n^2 k)$ in the worst case. For dimensions three and higher, we have no nontrivial algorithmic results. Obviously, the problem can always be solved in $O(n^d k \log n)$ time, by testing all possible d-tuples of the underlying set, but this is probably far from optimal.

The Ramsey-type question (5) has a negative answer, for any finite number of colors, even for homothetic copies. Indeed, for any finite set A and for any coloring of space with a finite number of colors, one can always find a monochromatic set similar (even homothetic) to A. This follows from the Hales–Jewett theorem (Hales and Jewett, 1963), which implies that every coloring of the integer lattice $\mathbb{Z}^d$ with a finite number

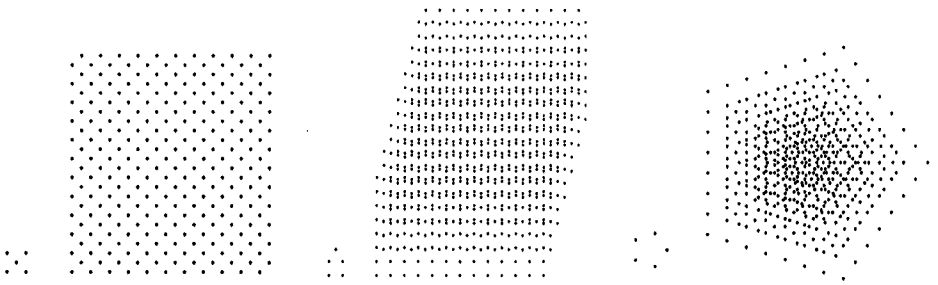


Figure 2.3. Three five-point patterns of different rational dimensions and three sets containing many of their translates

of colors contains a monochromatic homothetic copy of the lattice cube $\{1,\ldots,m\}^d$ (Gallai–Witt theorem; Rado, 1943; Witt, 1952).

6. Equivalence under homothety or affine transformations

For homothety-equivalence, questions (1) and (2) have been completely answered in all dimensions (van Kreveld and de Berg, 1989; Elekes and Erdős, 1994; Brass, 2002). The maximum number of homothetic copies of a set that can occur among n points is $\Theta(n^2)$; the upper bound $O(n^2)$ is always trivial, since the image of a set under a homothety is specified by the images of two points; and a lower bound of $\Omega(n^2)$ is attained by the homothetic copies of $\{1,\ldots,k\}$ in $\{1,\ldots,n\}$. The maximum order is attained only for this one-dimensional example. If the dimension of the affine space induced by a given pattern A over the rationals is k, then the maximum number of homothetic copies of A that can occur among n points is $\Theta(n^{1+1/k})$, which answers question (2).

Question (3) on the minimum number of distinct homothety classes of k-point subsets among n points, seems to be still open. As in the case of translations, by projection, we can restrict our attention to the one-dimensional case, where a sequence of equidistant points $\{0,\ldots,n-1\}$ should be extremal. This gives $\Theta(n^{k-1})$ distinct homothety classes. To see this, notice that as the size of the sequence increases from $n-1$ to n, the number of additional homothety classes that were not already present in $\{0,\ldots n-2\}$, is $\Theta(n^{k-2})$. (The increment certainly includes the classes of all k-tuples that contain 0, $n-1$, and a third number coprime to $n-1$.) Unfortunately, the pigeonhole principle gives only an $\Omega(n^{k-2})$ lower bound for the number of pairwise dissimilar k-point patterns spanned by a set of n numbers.

PROBLEM 17 *What is the minimum number of distinct homothety classes among all k-element subsets of a set of n numbers?*

The algorithmic problem (4) was settled in van Kreveld and de Berg (1989) and Brass (2002). In $O(n^{1+1/d}k\log n)$ time, in any n-element set of d-space one can find all homothetic copies of a given full-dimensional k-point pattern. This is asymptotically tight up to the $\log n$-factor. As mentioned in the previous section, the answer to the corresponding Ramsey-type question (5), is negative: one cannot avoid monochromatic homothetic copies of any finite pattern with any finite number of colors.

The situation is very similar for affine images. The maximum number of affine copies of a set among n points in d-dimensional space is $\Theta(n^{d+1})$. The upper bound is trivial, since an affine image is specified by the images of $d+1$ points. On the other hand, the d-dimensional "lattice cube," $\{1,\dots,n^{1/d}\}^d$, contains $\Omega(n^{d+1})$ affine images of $\{0,1\}^d$ or of any other small lattice-cube of fixed size.

The answer to question (2) is not so clear.

PROBLEM 18 *Do there exist, for every full-dimensional pattern A in d-space, n-element sets containing $\Omega(n^{d+1})$ affine copies of A?*

PROBLEM 19 *What is the minimum number of affine equivalence classes among all k-element subsets of a set of n points in d-dimensional space?*

For the algorithmic problem (4), the brute force method of trying all possible $(d+1)$-tuples of image points is already optimal. The Ramsey-type question (5) has again a negative answer, since every homothetic copy is also an affine copy.

7. Other equivalence relations for triangles in the plane

For triples in the plane, several other equivalence relations have been studied. An especially interesting example is the following. Two ordered triples are considered equivalent if they determine the same angle. It was proved in Pach and Sharir (1992) that the maximum number of triples in a set of n points in the plane that determine the same angle α is $\Theta(n^2\log n)$. This order of magnitude is attained for a dense set of angles α. For every other angle α, distribute as evenly as possible $n-1$ points on two rays that emanate from the origin and enclose angle α, and place the last point at the origin. Clearly, the number of triples determining angle α is $\Omega(n^2)$, which "almost" answers question (2). As for the minimum number of distinct angles determined by n points in the plane, Erdős conjectured that the answer to the following question is in the affirmative.

PROBLEM 20 *Is it true that every set of n points in the plane, not all on a line, determine at least $n-2$ distinct angles?*

This number is attained for a regular n-gon and for several other configurations.

The corresponding algorithmic question (4) is easy: list, for each point p of the set, all lines ℓ through p, together with the points on ℓ. Then we can find all occurrences of a given angle in time $O(n^2 \log n + a)$, where a is the number of occurrences of that angle. Thus, by the above bound from Pach and Sharir (1992), the problem can be solved in $O(n^2 \log n)$ time, which is optimal. The negative answer to the Ramsey-type question (5) again follows from the analogous result for homothetic copies: no coloring with a finite number of colors can avoid a given angle.

Another natural equivalence relation classifies triangles according to their areas.

PROBLEM 21 *What is the maximum number of unit-area triangles that can be determined by n points in the plane?*

An upper bound of $O(n^{7/3})$ was established in Pach and Sharir (1992), while it was pointed out in (Erdős and Purdy, 1971) that a section of the integer lattice gives the lower bound $\Omega(n^2 \log \log n)$. By scaling, we see that all areas allow the same multiplicities, which answers (2). However, problem (3) is open in this case.

PROBLEM 22 *Is it true that every set of n points in the plane, not all on a line, spans at least $\lfloor (n-1)/2 \rfloor$ triangles of pairwise different areas?*

This bound is attained by placing on two parallel lines two equidistant point sets whose sizes differ by at most one. This construction is conjectured to be extremal (Erdős and Purdy, 1977; Straus, 1978). The best known lower bound, $0.4142n - O(1)$, follows from Burton and Purdy (1979), using Ungar (1982).

The corresponding algorithmic problem (4) is to find all unit-area triangles. Again, this can be done in $O(n^2 \log n + a)$ time, where a denotes the number of unit area triangles. First, dualize the points to lines, and construct their arrangement, together with a point location structure. Next, for each pair (p, q) of original points, consider the two parallel lines that contain all points r such that pqr is a triangle of unit area. These lines correspond to points in the dual arrangement, for which we can perform a point location query to determine all dual lines containing them. They correspond to points in the original set that together with p and q span a triangle of area one. Each such query takes $\log n$ time plus the number of answers returned.

Concerning the Ramsey-type problem (4), it is easy to see that, for any 2-coloring of the plane, there is a monochromatic triple that spans a triangle of unit area. The same statement may hold for any coloring with a finite number of colors.

PROBLEM 23 *Is it true that for any coloring of the plane with a finite number of colors, there is a monochromatic triple that spans a triangle of unit area?*

The *perimeter* of triangles was also discussed in the same paper (Pach and Sharir, 1992), and later in Pach and Sharir (2004), where an upper bound of $O(n^{16/7})$ was established, but there is no nontrivial lower bound. The lattice section has $\Omega(ne^{c\log n/\log\log n})$ pairwise *congruent* triangles, which, of course, also have equal perimeters, but this bound is probably far from being sharp.

PROBLEM 24 *What is the maximum number of unit perimeter triangles spanned by n points in the plane?*

By scaling, all perimeters are equivalent, answering (2). By the pigeonhole principle, we obtain an $\Omega(n^{5/7})$ lower bound for the number of distinct perimeters, but again this is probably far from the truth.

PROBLEM 25 *What is the minimum number of distinct perimeters assumed by all* $\binom{n}{3}$ *triangles spanned by a set of n points in the plane?*

Here neither the algorithmic problem (4) nor the Ramsey-type problem (5) has an obvious solution. Concerning the latter question, it is clear that with a sufficiently large number of colors, one can avoid unit perimeter triangles: color the plane "cellwise," where each cell is too small to contain a unit perimeter triangle, and two cells of the same color are far apart. The problem of determining the minimum number of colors required seems to be similar to the question addressed by Theorem 2.3.

Acknowledgements Research supported by NSF CCR-00 98246, NSA H-98230, by grants from OTKA and PSC-CUNY.

References

Ábrego, B.M. and Fernández-Merchant, S. (2000). On the maximum number of equilateral triangles. I. *Discrete and Computational Geometry*, 23:129 – 135.

Ábrego, B.M. and Fernández-Merchant S. (2002). Convex polyhedra in $\mathbb{R}^3$ spanning $\Omega(n^{4/3})$ congruent triangles, *Journal of Combinatorial Theory. Series A*, 98:406 – 409.

Agarwal, P.K. and Sharir, M. (2002). On the number of congruent simplices in a point set. *Discrete and Computational Geometry*, 28:123–150.

Akutsu, T., Tamaki, H., and Tokuyama, T. (1998). Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. *Discrete and Computational Geometry*, 20:307–331.

Aronov, B., Pach, J., Sharir, M., and Tardos, G. (2003). Distinct distances in three and higher dimensions. In: *35th ACM Symposium on Theory of Computing*, pp. 541–546. Also in: *Combinatorics, Probability and Computing*, 13:283–293.

Beck, J. (1983). On the lattice property of the plane and some problems of Dirac, Motzkin and Erdős in combinatorial geometry. *Combinatorica*, 3:281–297.

Beck, J. and Spencer, J. (1984). Unit distances. *Journal of Combinatorial Theory. Series A*, 3:231–238.

Brass, P. (1997). On the maximum number of unit distances among n points in dimension four. In: I. Bárány et al. (eds.), *Intuitive Geometry*, pp. 277–290 Bolyai Society Mathematical Studies, vol. 4. Note also the correction of one case by K. Swanepoel in the review MR 98j:52030.

Brass, P. (2000). Exact point pattern matching and the number of congruent triangles in a three-dimensional pointset, In: M. Paterson (ed.), *Algorithms–ESA 2000*, pp. 112–119. Lecture Notes in Computer Science, vol. 1879, Springer-Verlag.

Brass , P. (2002). Combinatorial geometry problems in pattern recognition. *Discrete and Computational Geometry*, 28:495–510.

Burton, G.R. and Purdy, G.B. (1979). The directions determined by n points in the plane. *Journal of the London Mathematical Society*, 20:109–114.

Cantwell, K. (1996). Finite Euclidean Ramsey theory. *Journal of Combinatorial Theory. Series A*, 73:273–285.

Chung, F.R.K. (1984). The number of different distances determined by n points in the plane. *Journal of Combinatorial Theory. Series A*, 36:342–354.

Chung, F.R.K., Szemerédi, E., and Trotter, W.T. (1992) The number of different distances determined by a set of points in the Euclidean plane. *Discrete and Computational Geometry*, 7:1–11.

Clarkson, K.L., Edelsbrunner, H., Guibas, L., Sharir, M., and Welzl, E. (1990). Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete and Computational Geometry*, 5:99–160.

Elekes, G. and Erdős, P. (1994). Similar configurations and pseudo grids. In: K. Böröczky et. al. (eds.), *Intuitive Geometry*, pp. 85 – 104. Colloquia Mathematica Societatis János Bolyai, vol. 63.

Erdős, P. (1946). On sets of distances of n points. *American Mathematical Monthly*, 53:248 – 250.

Erdős, P. (1960). On sets of distances of n points in Euclidean space. *Magyar Tudományos Akadémia Matematikai Kutató Intézet Közleményei* 5:165 – 169.

Erdős, P., Graham, R.L., Montgomery, P., Rothschild, B.L., Spencer, J., and Straus, E.G. (1973). Euclidean Ramsey theorems. I. *Journal of Combinatorial Theory, Series A*, 14:341 – 363.

Erdős, P., Graham, R.L., Montgomery, P. Rothschild, B.L., Spencer, J., and Straus, E.G. (1975). Euclidean Ramsey theorems. III. In: A. Hajnal, R. Rado, and V.T. Sós (eds.), *Infinite and Finite Sets*, pp. 559 – 584. North-Holland, Amsterdam.

Erdős, P., Hickerson, D., and Pach, J. (1989). A problem of Leo Moser about repeated distances on the sphere, *American Mathematical Monthly*, 96:569 – 575.

Erdős, P. and Purdy, G. (1971). Some extremal problems in geometry, *Journal of Combinatorial Theory, Series A*, 10:246 – 252.

Erdős, P.and Purdy, G. (1977). Some extremal problems in geometry. V, In; *Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory and Computing*, pp. 569 – 578. Congressus Numerantium, vol. 19.

Frankl, P. and Rödl, V. (1986). All triangles are Ramsey. *Transactions of the American Mathematical Society*, 297:777 – 779.

Hadwiger, H. (1961). Ungelöste Probleme No. 40. *Elemente der Mathematik*, 16:103 – 104.

Hales, A.W. and Jewett, R.I. (1963). Regularity and positional games. *Transactions of the American Mathematical Society*, 106:222 – 229.

Józsa, S. and Szemerédi, E. (1975). The number of unit distances in the plane. In: A. Hajnal et al. (eds.), *Infinite and Finite Sets, Vol. 2*, pp. 939 – 950. Colloquia Mathematica Societatis János Bolyai vol. 10, North Holland.

Katz, N.H. and Tardos, G. (2004). Note on distinct sums and distinct distances. In: J. Pach (ed.), *Towards a Theory of Geometric Graphs*, pp. 119 – 126. Contemporary Mathematics, vol.342, American Mathematical Society, Providence, RI.

van Kreveld, M.J. and de Berg, M.T. (1989). Finding squares and rectangles in sets of points. In: M. Nagl (ed.), *Graph-Theoretic Concepts in Computer Science*, pp. 341 – 355. Lecture Notes in Computer Science, vol. 411, Springer-Verlag.

Kříž, I. (1992). All trapezoids are Ramsey. *Discrete Mathematics*, 108:59 – 62.

Laczkovich, M. and Ruzsa, I.Z. (1997). The number of homothetic subsets, In: R.L. Graham et al. (eds.), *The Mathematics of Paul Erdős. Vol. II*, pp. 294 – 302. Algorithms and Combinatorics, vol. 14, Springer-Verlag.

Matoušek, J. (1993). Range searching with efficient hierarchical cuttings. *Discrete and Computational Geometry*, 10:157 – 182.

Moser, L. (1952). On different distances determined by n points. *American Mathematical Monthly* , 59:85 – 91.

Pach, J. and Agarwal, P.K. (1995). *Combinatorial Geometry.* Wiley, New York.

Pach, J. and Sharir, M. (1992). Repeated angles in the plane and related problems. *Journal of Combinatorial Theory. Series A*, 59:12 – 22.

Pach, J. and Sharir, M. (2004). Incidences. In: J. Pach (ed.), *Towards a Theory of Geometric Graphs*, pp. 283 – 293. Contemporary Mathematics, vol. 342, American Mathematical Society, Providence, RI.

Rado, R. (1943). Note on combinatorial analysis. *Proceedings of the London Mathematical Society*, 48:122 – 160.

de Rezende, P.J. and Lee, D.T. (1995). Point set pattern matching in d-dimensions. *Algorithmica*, 13:387 – 404.

Solymosi, J. and Tóth, C.D. (2001). Distinct distances in the plane. *Discrete and Computational Geometry*, 25:629 – 634.

Solymosi, J. and Vu, V. (2005). Near optimal bounds for the number of distinct distances in high dimensions. Forthcoming in *Combinatorica.*

Spencer, J., Szemerédi, E., and Trotter, W.T. (1984). Unit distances in the Euclidean plane. In: B. Bollobás (ed.), *Graph Theory and Combinatorics*, pp. 293 – 304. Academic Press, London, 1984,

Straus, E.G. (1978). Some extremal problems in combinatorial geometry. In: Combinatorial Mathematics, pp. 308 – 312. Lecture Notes in Mathematics, vol. 686.

Székely, L.A. (1997). Crossing numbers and hard Erdős problems in discrete geometry. *Combinatorics, Probability and Computing*, 6:353 – 358.

Tardos, G. (2003). On distinct sums and distinct distances. *Advances in Mathematics*, 180:275–289.

Ungar, P. (1982). $2N$ noncollinear points determine at least $2N$ directions. *Journal of Combinatorial Theory. Series A*, 33:343–347.

van Wamelen, P. (1999). The maximum number of unit distances among n points in dimension four. *Beiträge Algebra Geometrie*, 40:475–477.

Witt, E. (1952). Ein kombinatorischer Satz der Elementargeometrie. *Mathematische Nachrichten*, 6:261–262.

Chapter 3

DATA DEPTH AND MAXIMUM FEASIBLE SUBSYSTEMS

Komei Fukuda
Vera Rosta

Abstract Various data depth measures were introduced in nonparametric statistics as multidimensional generalizations of ranks and of the median. A related problem in optimization is to find a maximum feasible subsystem, that is a solution satisfying as many constrainsts as possible, in a given system of linear inequalities. In this paper we give a unified framework for the main data depth measures such as the halfspace depth, the regression depth and the simplicial depth, and we survey the related results from nonparametric statistics, computational geometry, discrete geometry and linear optimization.

1. Introduction

The subject of this survey is a discrete geometric problem which was raised independently in statistics, in discrete and computational geometry, in political science and in optimization. The motivation in statistics to generalize the median and ranks to higher dimensions is very natural, as the mean is not considered to be a robust measure of central location. It is enough to strategically place one outlier to change the mean. By contrast, the median in one dimension is very robust, or has high breakdown point, as half of the observations need to be bad to corrupt the value of the median.

As a consequence, in nonparametric statistics, several data depth measures were introduced as multivariate generalizations of ranks to complement classical multivariate analysis, first by Tukey (1975), then followed by Oja (1983), Liu (1990), Donoho and Gasko (1992), Singh (1993), Rousseeuw and Hubert (1999a,b) among others. These measures, though seemingly different, have strong connections. In this survey we present ideas for unifying some of the different measures.

The *halfspace depth*, also known as *location depth* or *Tukey depth* introduced by Tukey (1974a,b) is perhaps the best known among the data depth measures in nonparametric statistics, and in discrete and computational geometry. It also has a strong connection to the maximum feasible subsystem problem, Max FS, in optimization. The halfspace depth of a point p relative to a data set S of n points in Euclidean space $\mathbb{R}^d$, is the smallest number of points of S in any closed halfspace with boundary through p. It is easy to see that the halfspace depth of p is the smallest number of points of S in any open halfspace containing p. A point of deepest location is called a *Tukey median*. Exact and heuristic algorithms are presented in Fukuda and Rosta (2004) to compute the halfspace depth of a point in any dimension using the hyperplane arrangement construction.

For a given data set S, the set D_k of all points in $\mathbb{R}^d$ with depth at least k is called the *contour* of depth k in statistics (Donoho and Gasko, 1992), though this expression is sometimes used just for the boundary of D_k. In discrete geometry, D_k is known as the *k-core* (Avis, 1993; Onn, 2001). This double terminology is a consequence of parallel research in several fields. It turns out that the halfspace depth computation is equivalent to the Max FS computation though the only common reference was the early complexity result of Johnson and Preparata (1978). The depth regions are convex and nested, that are critical in statistical estimation. Donoho and Gasko (1992) show that the halfspace depth measure has other statistically good properties, namely it leads to affine equivariant and robust estimators as the Tukey median has high breakdown point in any dimension (Donoho, 1982) (see (3.3) for the formal definition). It is not completely evident how to construct high breakdown estimators in high dimension, as shown by Donoho (1982). Many suggested location estimators do not have high breakdown points, namely, the iterative ellipsoidal trimming (Gnanadesikan and Kettenring, 1972), the sequential deletion of apparent outliers (Dempster and Gasko, 1981), the convex hull peeling (Bebbington, 1978) and the ellipsoidal peeling (Titterington, 1978).

For any dimension $d \geq 0$, as a consequence of Helly's theorem (Danzer et al., 1963), the maximum location depth is at least $\lceil \frac{n}{d+1} \rceil$. The set of points with at least this depth is called the *center* in computational geometry, and its computation in the plane drew considerable attention (Matoušek, 1992; Langerman and Steiger, 2000; Rousseeuw and Ruts, 1996; Naor and Sharir, 1990). While the problem is solved to optimality in the plane, in higher dimensions the computation of the center is much more challenging. Using Radon partitions Clarkson et al. (1993) compute approximate center points in any dimension, finding a point of

depth at least n/d^2 with high probability. This terminology of center might be misleading as for symmetric data the maximum depth is $n/2$, as observed by Donoho and Gasko (1992). They also mentioned that if the data set consists of the vertices of nested "aligned" simplices, then the lower bound is attained. Miller et al. (2001) proposed a method to compute the depth regions and their boundaries in the plane using a topological sweep. Unfortunately their method may not be easily generalized to higher dimensions. An exact and heuristic algorithm is presented (Fukuda and Rosta, 2004) in any dimension to compute the depth regions and their boundaries. Not surprisingly these are very high complexity computations and there is necessity for improvement.

Some of the main additional data depth measures have strong connection to the halfspace depth. The regression depth defined by Rousseeuw and Hubert (1999a,b) is a measure of how well a regression hyperplane fits a given data set (see (3.5) for the formal definition). It has been pointed out (Eppstein, 2003; Fukuda and Rosta, 2004; van Kreveld et al., 1999) that the computation of the regression depth of a hyperplane H can be done using enumeration of the dual hyperplane arrangement. The algorithm (Fukuda and Rosta, 2004) for the computation of halfspace depth of a point is based on a memory efficient hyperplane arrangement enumeration algorithm and can be immediately applied to compute the regression depth of a hyperplane. Amenta et al. (2000) showed that for any data set S there is a hyperplane whose regression depth is at least the Helly bound and thus the lower bound for the maximum regression depth is the same as for the maximum halfspace depth. Discrete geometry gives the theoretical background for many algorithms. Helly's theorem and its relatives (Danzer et al., 1963), the theorems of Tverberg, Charathéodory, and extensions by Bárány (1982); Bárány; Bárány and Onn (1997a); Matoušek (2002); Onn (2001); Tverberg (1966); Wagner (2003); Wagner and Welzl (2001) are essential for the unification of different data depth measures, and to describe their properties. For example there are bounds relating halfspace depth and simplicial depth based on these discrete geometry theorems, see Bárány (1982); Wagner (2003); Wagner and Welzl (2001). The simplicial depth of a point p relative to a given data set S in $\mathbb{R}^d$ was defined by Liu (1990) as the number (or proportion) of simplices containing p.

For $d = 2$, the problem of computing halfspace depth has been solved to optimal efficiency by computational geometers, Cole et al. (1987), Naor and Sharir (1990), Matoušek (1992), Langerman and Steiger (2000), Miller et al. (2001) and statisticians Rousseeuw and Ruts (1996).

In higher dimensions, the situation is very different. Let S be a set of n points on the d-dimensional sphere centered at the origin. The *closed*

(open) hemisphere problem is to find the closed (open) hemisphere containing the largest number of points of S. Johnson and Preparata (1978) proved that the closed (open) hemisphere problem is NP-complete, which clearly implies that the computation of the halfspace depth of a given point is also NP-complete. As a consequence very little effort was given to design exact and deterministic algorithm or even approximate one to compute the halfspace depth of a point relative to a given data set in higher dimensions. NP-hardness results concentrate on the worst case, that might hinder the possibility of computing partial information quickly in practice. For statistical applications, *primal-dual* algorithms might be more valuable, i.e., those algorithms that update both upper and lower bounds of the depth, and terminate as soon as the bounds coincide. Such an algorithm can provide useful information on the target depth even when the user cannot afford to wait for the algorithm to terminate. In addition, unlike "enumeration-based" algorithms whose termination depends on the completion of enumeration (of all halfspace partions), primal-dual algorithms might terminate much earlier than the worst-case time bound. The exact, memory efficient and highly parallelizable algorithm (Fukuda and Rosta, 2004) for computing the halfspace depth is primal-dual type.

Already Johnson and Preparata reformulated the closed (open) hemisphere problem to the equivalent form of finding the maximal feasible subsystem of a system of strict (weak) homogeneous linear inequalities, Max FS. There is an extensive literature for the complexity and computation of Max FS, using results of integer programming, independent from the computational geometry or statistics literature for data depth. The exact and heuristic algorithms (Fukuda and Rosta, 2004) for the computation of halfspace depth and related data depth measures, based on hyperplane arrangement construction algorithms can easily be adapted to solve Max FS.

In this survey we analyze the data depth literature in statistics, the related results in computational geometry, in discrete geometry and in the optimization literature. The importance of this area is evident for nonparametric statisticians for whom the computational efficiency is vital. Considering the high complexity of the problem this is a real challenge. Seeing the limits of existing results perhaps leads to new modified notions that hopefully will be computationally tractable. It make sence to go to this direction if all the computational difficulties of the existing definitions are examined. On the other hand the Max FS research can also profit from the results of others. Max FS computation has been applied extensively in many areas, for example in telecommunications (Rossi et al., 2001), neural networks (Amaldi, 1991), machine learning

(Bennett and Bredensteiner, 1997), image processing (Amaldi and Mattavelli, 2002) and in computational biology (Wagner et al., 2002).

2. Generalization of the median and multivariate ranks

Different notions of data depth were defined by Mahalanobis (1936); Tukey (1974a,b); Oja (1983); Liu (1990); Donoho and Gasko (1992); Singh (1993); Rousseeuw and Hubert (1999a,b) among others and were proposed as location parameters in the statistical literature, see Liu (2003). Depth ordering was used to define descriptive statistics and various statistical inference methods. Liu et al. (1999) introduce multivariate scale, skewness and kurtosis and present them graphically in the plane by simple curves. Liu and Singh (1993) proposed a quality index and Liu (1995) applied it with data depth based ordering to construct nonparametric control charts for monitoring multivariate processes. Other applications include multivariate rank tests by Liu (1992); Liu and Singh (1993), construction of confidence regions by Yeh and Singh (1997) and testing general multivariate hypotheses by Liu and Singh (1997).

An often referred paper in nonparametric statistics on the multivariate generalizations of the median is by Donoho and Gasko (1992) which continues earlier works of both and of many other statisticians. We follow their notations. In the statistical literature a good deal of effort is spent to demonstrate favorable properties of the various location estimates or data depth measures from the statistical point of view.

Tukey (1974a,b, 1977) introduced the notion of the halfspace depth of a point in a multivariate data set as follows: Let $X = \{X_1, X_2, \ldots, X_n\}$ be a data set in $\mathbb{R}^d$. If $d = 1$, the depth of a point or of the value x is defined as

$$\text{depth}_1(x, X) = \min(|\{i : X_i \leq x\}|, |\{i : X_i \geq x\}|). \tag{3.1}$$

If $d > 1$ the depth of a point $x \in \mathbb{R}^d$ is the least depth of x in any one-dimensional projection of the data set:

$$\begin{aligned} \text{depth}_d(x, X) &= \min_{|u|=1} \text{depth}_1(u^T x, \{u^T X_i\}) \\ &= \min_{|u|=1} |\{i : u^T X_i \geq u^T x\}|. \end{aligned} \tag{3.2}$$

In dimension one, the minimum and the maximum are points of depth one, the upper and lower quartiles are of depth $n/4$ and the median is of depth $n/2$. Tukey considered the use of contours of depth, the boundary of the regions determined by points of same depth, to indicate the shape of a two-dimensional data set and suggested to define multivariate rank

statistics. Since the median has the maximum depth value in dimension one, in higher dimensions a point with maximum depth could be considered a multivariate median. The data depth regions are convex whose shapes indicate the scale and correlation of the data.

The resulting measures have good statistical properties, for example they are affine equivariant, i.e. they remain the same after linear transformations of the data, and they are robust in any dimension. Donoho and Gasko (1992) proved that the generalization of the median has a breakdown point of at least $1/(d+1)$ in dimension d and it can be as high as $1/3$ for symmetric data set. A formal definition of a finite-sample breakdown point is in Donoho (1982). Let $X^{(n)}$ denote a given data set of size n and let T be the estimator of interest. Consider adding to $X^{(n)}$ another data set $Y^{(m)}$ of size m. If it is possible to make $T(X^{(n)} \cup Y^{(m)}) - T(X^{(n)})$ arbitrarily large, we say that the estimator breaks down under contamination fraction $m/(n+m)$. The breakdown point $\epsilon^*(T, X)$ is the smallest contamination fraction under which the estimator breaks down:

$$\epsilon^* = \min\left\{ \frac{m}{n+m} : \sup_{Y^{(m)}} \|T(X^{(n)} \cup Y^{(m)}) - T(X^{(n)})\| = \infty \right\} \qquad (3.3)$$

Thus the mean has breakdown point $1/(n+1)$ and the one-dimensional median has breakdown point $\frac{1}{2}$. We can say that the median is a robust measure, but not the mean.

Among location estimates the median has the best achievable breakdown point, as for translation equivariant estimators $\epsilon^* \leq \frac{1}{2}$. Maronna (1976) and Huber (1977) found that affine equivariant M estimates of location have breakdown points bounded above by $1/d$ in dimension d, i.e., such estimators can be upset by a relatively small fraction of strategically placed outliers. Donoho (1982) gives several other examples of affine equivariant estimators which do not have high breakdown points.

(a) *Convex hull peeling*: Iteratively the points lying on the boundary of a sample's convex hull are discarded, peeled away and finally the mean of the remaining observations is taken as the peeled mean. If the data set is in general position the breakdown point of any peeled mean is at most $\big(1/(d+1)\big)\big((n+d+1)/(n+2)\big)$ (Donoho, 1982).

(b) *Data cleaning* or sequential deletion of outlyers uses the Mahalanobis distance

$$D^2(X_i, X) = \big(X_i - \mathrm{Ave}(X)\big)^T \mathrm{Cov}^{-1}(X)\big(X_i - \mathrm{Ave}(X)\big) \qquad (3.4)$$

to identify the most discrepant observation relative to the data set X. At each stage the most discrepant data point relative to the

remaining data is removed. At some point a decision is made that all the outliers have been removed and the average of the remaining point is the "cleaned mean". If X is in general position, the breakdown point of any "cleaned mean" is at most $1/(d+1)$.

Both convex hull peeling and cleaned mean are affine equivariant. If the affine equivariance condition is relaxed to rigid-motion equivariance or to location equivariance, then it is easy to find high breakdown point estimators, for example the coordinatewise median is location equivariant and has breakdown point 1/2 in any dimension. The difficulty is being both coordinate free and robust (Donoho and Gasko, 1992). Rousseeuw (1985) showed that the center of the minimum volume ellipsoid containing at least half the data provided a method with breakdown point of nearly 1/2 in high dimensions, see Lopuhaä and Rousseeuw (1991). Oja (1983) introduced an affine equivariant multivariate median with interesting breakdown properties based on simplicial volumes, see Niinimaa et al. (1990). From the point of view of robustness, halfspace depth is considered interesting by Donoho and Gasko (1992) since the estimator $T_{(k)} = \text{Ave}\{X_i : \text{depth}(X_i; X) \geq k\}$ has breakdown point $\epsilon^* = k/(n+k)$, which means that the maximum depth controls what robustness is possible. Since the maximum depth is between $\lceil n/(d+1) \rceil$ and $n/2$ the breakdown point is close to $\frac{1}{3}$ for centrosymmetric distribution and it is at least $1/(d+1)$ for X in general position.

Another important statistical property for good data depth measure requires that the data depth regions be nested. This property follows easily for the halfspace depth. The simplicial depth of a point $x \in \mathbb{R}^d$ with respect to a given data set X was defined by Liu (1992), as the number (proportion) of simplices formed with points in X, that contain the point x. The simplicial depth is affine equivariant, but surprisingly the corresponding depth regions are not necessarily nested, as shown by examples already in the plane (Burr et al., 2003; Zuo and Serfling, 2000). This data depth measure is conceptually attractive by its simplicity but some modification to this definition might be necessary. Some modified definitions are suggested in Burr et al. (2003); Rafalin and Souvaine (2004). It turns out, see Fukuda and Rosta (2004), that the halfspace depth of a point x is actually the cardinality of the minimum transversal of the simplices containing x.

Regression is one of the most commonly used statistical tools for modeling related variables. Linear regression is optimal if the errors are normally distributed, but when outliers are present or the error distributions are not normal, linear regression is not considered robust. Rousseeuw and Hubert (1999a) introduced the notion of regression depth by generalizing the halfspace depth to the regression setting. In linear regres-

sion the aim is to fit to a data set $Z_n = \{\mathbf{z}_i = (x_{i1}, \ldots, x_{i,d-1}, y_i); i = 1, \ldots, n\} \subset \mathbb{R}^d$ a hyperplane of the form $y = \Theta_1 x_1 + \Theta_2 x_2 + \cdots + \Theta_{d-1} x_{d-1} + \Theta_d$ with $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_d)^T \in \mathbb{R}^d$. The x-part of each data point $\mathbf{z}_i$ is denoted by $\mathbf{x}_i = (x_{i1}, \ldots, x_{i,d-1})^T \in \mathbb{R}^{d-1}$ and the residuals of Z_n relative to the fit $\boldsymbol{\Theta}$ by $r_i(\boldsymbol{\Theta}) = y_i - \Theta_1 x_{i1} - \cdots - \Theta_{d-1} x_{i,d-1} - \Theta_d$. The *regression depth* of a fit $\boldsymbol{\Theta} \in \mathbb{R}^d$ relative to a data set $Z_n \subset \mathbb{R}^d$ is given by

$$\text{rdepth}(\Theta, Z_n) = \min_{\mathbf{u},v}(|\{i : r_i(\boldsymbol{\Theta}) \geq 0 \text{ and } \mathbf{x}_i^T\mathbf{u} < v\}| + |\{i : r_i(\boldsymbol{\Theta}) \leq 0 \text{ and } \mathbf{x}_i^T\mathbf{u} > v\}|) \quad (3.5)$$

where the minimum is over all unit vectors $\mathbf{u} = (u_1, \ldots, u_{d-1})^T \in \mathbb{R}^{d-1}$ and all $v \in \mathbb{R}$ with $\mathbf{x}_i^T\mathbf{u} \neq v$ for all $(\mathbf{x}_i^T, y_i) \in Z_n$.

Let h be a hyperplane in d-dimension and let p_h be the point dual to h in the dual hyperplane arrangement. The *regression depth* of a hyperplane h in d-dimension is the minimum number of hyperplanes crossed by any ray starting at the point p_h in the dual hyperplane arrangement. This geometric definition is the dual of the original definition given by Rousseeuw and Hubert (1999a). van Kreveld et al. (1999) mentioned that the regression depth can be computed by the enumeration of all unbounded cells of the dual hyperplane arrangement. The incremental hyperplane arrangement enumeration algorithm of Edelsbrunner et al. (1986) has $O(n^{d-1})$ time complexity and $O(n^{d-1})$ space requirement, the time complexity is optimal but the space complexity might be prohibitive for computation. Using in Fukuda and Rosta (2004) a memory efficient hyperplane arrangement enumeration code from Ferrez et al. (2005) makes possible the computation of the regression depth, though for high dimension it can also be too time consuming. An alternative definition of halfspace depth of a point p was given (Eppstein, 2003) as the minimum number of hyperplanes crossed by any ray starting at the point p, in the hyperplane arrangement determined by the given data set X. This looks identical to the regression depth definition given here showing that the two data depth measures are essentially the same.

Hyperplanes with high regression depth fit the data better than hyperplanes with low depth. The regression depth thus measures the quality of a fit, which motivates the interest in computing deepest regression depth. In Rousseeuw and Hubert (1999a) it is shown that the deepest fit is robust with breakdown value that converges to 1/3 for a large semiparametric model in any dimension $d \geq 2$. For general linear models they derive a monotone equivariance property. It seems that the data depth measures having the best statistical properties are the halfspace depth and the regression depth. It is not a surprise then, that these

measures attracted the most attention in all the relevant fields and thus justifies our concentration on these.

3. Helly's theorem and its relatives

In discrete geometry the following theorem is essential.

THEOREM 3.1 (HELLY (DANZER ET AL., 1963)) *Suppose* K *is a family of at least* $d+1$ *convex sets in* $\mathbb{R}^d$, *and* K *is finite or each member of* K *is compact. If each* $d+1$ *member of* K *have a common point, then there is a point common to all members of* K.

The characterization of the halfspace depth regions D_k and their boundaries is a direct consequence of Helly's theorem.

PROPOSITION 3.1 *Let* S *be a set of* n *points in* $\mathbb{R}^d$ *and* $k > 0$. *Then*

(a) *the halfspace depth region* D_k *is the intersection of all closed halfspaces containing at least* $n-k+1$ *points of* S,
(b) $D_{k+1} \subset D_k$,
(c) D_k *is not empty for all* $k \leq \lceil \frac{n}{d+1} \rceil$,
(d) *every full dimensional* D_k *is bounded by hyperplanes containing at least* d *points of* S *that span a* $(d-1)$*-dimensional subspace.*

Helly's theorem has many connections to other well-known discrete geometry theorems relevant for our study.

THEOREM 3.2 (TVERBERG, 1966) *Let* d *and* k *be given natural numbers. For any set* $S \subset \mathbb{R}^d$ *of at least* $(d+1)(k-1)+1$ *points there exist* k *pairwise disjoint subsets* $S_1, S_2, \ldots, S_k \subset S$ *such that* $\cap_{i=1}^{k} \operatorname{conv}(S_i) \neq \varnothing$.

Here $\operatorname{conv}(S_i)$ denotes the convex hull of the point set S_i. The sets $S_1, S_2, \ldots, S_k$, as in the theorem, are called a *Tverberg partition* and a point in the intersection of their convex hulls is called a *Tverberg point*, or a *k-divisible point*. The special case $k = 2$ is the Radon theorem, and accordingly we have a *Radon partition* and *Radon points*. Radon points are iteratively computed in an approximate center point algorithm by Clarkson et al. (1993). The algorithm finds a point of depth n/d^2 with high probability.

In order to design algorithms for the computation of the halfspace depth regions D_k, or to be able to decide whether D_k is empty for given k, or to find a point in D_k if not empty, it would be useful to know as much as possible about these regions. Unfortunately the existing discrete geometry results do not give too much hope for good characterization of D_k and their boundaries. In the discrete geometry literature D_k

is sometimes called the *k-core* or *Helly-core*, and the set of k-divisible points is the *k-split*. It is easy to see that if the k-split is not empty, then any point in the k-split has halfspace depth at least k. In particular if $k = \lceil n/(d+1) \rceil$, then the k-split is not empty as a consequence of the Tverberg's theorem, and a Tverberg point is also a center point.

It was conjectured by Reay (1982), Sierksma (1982) and others that the k-core equals the k-split in any dimension. In the plane, for $d = 2$, the k-split equals the k-core (Reay, 1982), but in higher dimensions the conjecture is not true. In fact the k-split can be a proper subset of the k-core, as shown by Avis (1993), Bárány, and Onn (2001) independently. Their respective examples are interesting. Avis' counterexample consists of nine points on the moment curve in $\mathbb{R}^3$. He showed that there are extreme points of the 3-core that are not 3-divisible, i.e., not in the 3-split. Onn generates many examples in dimensions higher than nine and interestingly reduces the NP-complete problem of edge 3-colourability of a 3-regular graph to the decision problem of checking whether the 3-split is empty or not. It follows that this is also an NP-complete problem. Bárány and Onn (1997a) also proved that to decide whether the k-split is empty or not, or to find a k-divisible point are strongly NP-complete, by showing that these are polynomial time reducible to the decision and search variants of the colourful linear programming problem. They prove that, if sets ("coloured points") $S_1, S_2, \ldots, S_k \subset \mathbb{Q}^d$ and a point $b \in \mathbb{Q}^d$ are given, it is strongly NP-complete to decide whether there is a colourful $T = \{s_1, s_2, \ldots, s_k\}$, where $s_i \in S_i$ for $1 \leq i \leq k$, such that $b \in \text{conv}(T)$, or if there is one to find it. (When all S_i-s are equal this is standard linear programming.)

Although the boundary of the halfspace depth regions D_k have no satisfactory characterization, it is still possible to design algorithm to compute D_k, for all k. The algorithm in Fukuda and Rosta (2004) for computing D_k is based on the enumeration of those cells in the dual hyperplane arrangement, that correspond to k-sets. A subset X of an n-point set S is called a *k-set* if it has cardinality k and there is an open halfspace H such that $H \cap S = X$. Therefore there is a hyperplane h, the boundary of H, that "strictly" separates X from the remaining points of S. A well-known problem in discrete and computational geometry is the *k-set problem*, to determine the maximum number of k-sets of an n-point set in $\mathbb{R}^d$, as a function of n and k. This problem turned out to be extremely challenging even in the plane, only partial results exist and mostly for $d \leq 3$. Some of these results were obtained using a coloured version of Tverberg's theorem (Alon et al., 1992).

Let S be a set of n points in general position in $\mathbb{R}^d$. An *S-simplex* is a simplex whose vertices belong to S. Boros and Füredi (1984) showed

that in the plane any centerpoint of S is covered by at least $\frac{2}{9}\binom{n}{3}$ S-triangles, giving a connection between halfspace depth and simplicial depth in $\mathbb{R}^2$. Bárány (1982) and Bárány and Onn (1997a) studied similar problems in arbitrary dimension. The Charathéodory theorem (Bárány, 1982) says, that if $S \subset \mathbb{R}^d$ and $p \in \text{conv}(S)$, then there is an S-simplex containing the point p. The maximal simplicial depth is therefore at least 1. The colourful Charathéodory theorem due to Bárány (1982) states that if there is a point p common to the convex hull of the sets, $M_1, M_2, \ldots, M_{d+1} \subset S$, then there is a *colourful S-simplex*, $T = \{m_1, m_2, \ldots, m_{d+1}\}$ where $m_i \in M_i$ for all i, containing p in its convex hull. Bárány (1982) combines the colourful Charathéodory theorem and Tverberg's theorem to show the following positive fraction theorem, also known as first selection lemma, that gives a non-trivial lower bound for the maximal simplicial depth of points in $\mathbb{R}^d$ with respect to S.

THEOREM 3.3 (BÁRÁNY, 1982) *Let S be an n-point set in $\mathbb{R}^d$. Then there exists a point p in $\mathbb{R}^d$ contained in at least $c_d\binom{n}{d+1}$ S-simplices, where c_d is a constant depending only on the dimension d.*

The proof gives $c_d = (d+1)^{-d}$ and $c_d = (d+1)^{-(d+1)}$ is given in Bárány and Onn (1997b). The value $c_2 = \frac{2}{9}$ obtained in Boros and Füredi (1984) is optimal. A recent result of Wagner (2003); Wagner and Welzl (2001) shows that in $d \geq 3$ dimensions every centerpoint is also covered by a positive fraction (depending only on d) of all S-simplices. In the proof for a more general first selection lemma they are using known discrete geometry results on face numbers of convex polytopes and the Gale transform.

Moreover, if the k-split is not empty, then any Tverberg point in the k-split is covered by at least $\binom{k}{d+1}$ S-simplices (Bárány, 1982). The k-split, when not empty, is the subset of the k-core, in this case there is a point with halfspace depth at least k contained in at least $\binom{k}{d+1}$ simplices. This gives a lower bound on the maximum simplicial depth of a point with halfspace depth at least k, and Bárány (1982) gives an upper bound for the maximum simplicial depth, showing that no point can be covered by more than $(1/2^d)\binom{n}{d+1}$ S-simplices, if the points of S are in general position. General bounds on the simplicial depth as a function of the halfspace depth are given in the special case of $d = 2$ in Burr et al. (2003).

The regression depth defined by Rousseeuw and Hubert (1999a,b) is a measure of how well a regression hyperplane fits a given data set. Amenta et al. (2000) showed that for any data set S there is a hyperplane whose regression depth is at least the Helly bound. It has been pointed out several times (Eppstein, 2003; Fukuda and Rosta, 2004; van

Kreveld et al., 1999) that the computation of the regression depth of a hyperplane H can be done using enumeration of the dual hyperplane arrangement. The algorithm in Fukuda and Rosta (2004) for the computation of the halfspace depth of a point is also based on the construction of the dual hyperplane arrangement and thus it can be immediately applied to compute the regression depth of a hyperplane. The fact that the same algorithm can be used to compute the halfspace depth of a point and the regression depth of a hyperplane indicates the equivalence of these two measures. Therefore we hope that it is possible to show that for any data set S there is a hyperplane whose regression depth is at least the Helly bound, using only Helly's theorem and its relatives instead of using Brouwer's fixpoint theorem as Amenta et al. do.

From the above basic discrete geometry results it is clear that Helly's theorem and its relatives give a theoretical framework for a unified treatment of the three basic data depth measures, namely the halfspace depth, the simplicial depth and the regression depth.

4. Complexity

Johnson and Preparata's 1978 paper, entitled "The densest hemisphere problem," contains the main complexity result related to this survey. It was motivated by a geometric problem that originated as a formalization of a political science situation. Let K be a set of n points on the unit sphere S^d. Find a hemisphere of S^d containing the largest number of points from K. The coordinates of the points in K correspond to preferences of n voters on d relevant political issues; the axis of the maximizing hemisphere corresponds to a position on these issues which is likely to be supported by a majority of the voters.

In Johnson and Preparata (1978) the problem is reformulated in terms of vectors and inner products, namely let $K = \{p_1, p_2, \ldots, p_n\}$ be a finite subset of $\mathbb{Q}^d$ and consider the following two parallel problems:

(a) *Closed hemisphere*:
Find $x \in \mathbb{R}^d$ such that $\|x\| = 1$ and $|\{i : p_i \in K \text{ and } p_i \cdot x \geq 0\}|$ is maximized.

(b) *Open hemisphere*:
Find $x \in \mathbb{R}^d$ such that $\|x\| = 1$ and $|\{i : p_i \in K \text{ and } p_i \cdot x > 0\}|$ is maximized.

This formulation is more general than the original geometric problem since it allows more than one point along a ray. The restriction to use only rational coordinates is useful as it places the problem in discrete form, to which computational complexity arguments can be applied. The closed hemisphere problem is the same as the computation of the half-

space depth of the origin, therefore the complexity result of the closed hemisphere problem can be applied immediately. Moreover the closed hemisphere problem is identical to the Max FS problem for homogeneous system of linear inequalities, and the open hemisphere problem is identical to the Max FS problem for strict homogeneous system of inequalities. A variant discussed by Reiss and Dobkin (1976) is to determine if there is a hemisphere which contains the entire set K. This is equivalent to linear programming, the question whether there is a feasible solution to the given system of linear inequalities.

In the same paper Johnson and Preparata showed that both the Closed and Open hemisphere problems are NP-complete and thus there is no hope to find polynomial time algorithm in general. They obtained this complexity result by reducing the previously known NP-complete MAX 2-SAT (maximum 2-satisfiability) problem to the hemisphere problems. The complexity results of Johnson and Preparata were extended by Teng (1991). Testing whether the halfspace depth of a point is at least some fixed bound is coNP-complete, even in the special case of testing whether a point is a center point.

The above mentioned NP-hardness results make very unlikely the existence of polynomial time methods for solving the halfspace depth problem. Approximate algorithms that provide solution that are guaranteed to be a fixed percentage away from the actual depth could be sometimes sufficient. There are studies comparing the approximability of optimization problems and various approximability classes have been defined (Kann, 1992). Strong bounds exist about the approximability of famous problems like maximum independent set, minimum set cover or minimum graph colouring and these results had strong influence on the study of approximability for other optimization problems.

It follows from Johnson and Preparata's result that Max FS with $\geq$ or $>$ relations is NP-hard even when restricted to homogeneous systems. Amaldi and Kann (1995, 1998) studied the approximability of maximum feasible subsystems of linear relations. Depending on the type of relations, they show that Max FS can belong to different approximability classes. These ranges from APX-complete problems which can be approximated within a constant but not within every constant unless P = NP, to NPO PB-complete problems that are as hard to approximate as all NP optimization problem with polynomially bounded objective function. Max FS with strict and nonstrict inequalities can be approximated within two but not within every constant factor.

Struyf and Rousseeuw (2000) propose a heuristic approximation algorithm for high-dimensional computation of the deepest location. The algorithm calculates univariate location depths in finite directions and con-

tinues as long as it monotonically increases univariate location depths. Only an approximation of location depth is computed at each step, using projections to a randomly chosen finite number of directions. There is no measure of how good this approximation is and there is no proof that the algorithm converges to a Tukey median.

Fixing the dimension as a constant is a common practice in computational geometry when a problem is hard. It is possible though that a very large function of d is hidden in the constants and these algorithms are rarely implemented in higher than 3 dimensions. Johnson and Preparata (1978) presented an algorithm to compute the closed hemisphere problem, if the dimension is fixed, based on iterative projections to lower dimensions. No implementation of this algorithm is known for arbitrary dimension. This algorithm has $\Theta(n^{d-1} \log n)$ time complexity. They considered this algorithm as an attractive method for cases in which d is a small integer, four or less. They also presented an algorithm for the open hemisphere problem when the dimension is fixed, that require $O(d2^{d-2}n^{d-1} \log n)$ time. Rousseeuw and Ruts (1996) and Struyf and Rousseeuw (2000) rediscovered the same deterministic algorithm when the dimension is fixed for the computation of the location depth of a point, requiring $\Theta(n^{d-1} \log n)$ time, corresponding to the closed hemisphere problem, and implemented it for $d \leq 3$.

Matoušek (1992) briefly describes approximation algorithms for the computation of a center point, of the center and of a Tukey median for point sets in fixed dimension which could theoretically be called efficient. A point is called an ϵ-approximate centerpoint for the data set S, if it has depth at least $(1 - \epsilon)n/(d + 1)$. For any fixed ϵ, an ϵ-approximate centerpoint can be found in $O(n)$ time with the constant depending exponentially on d and ϵ, and then a $O(n^d)$ algorithm is given to find a centerpoint. As he points it out, a large constant of proportionality can be hidden in the big-Oh notation. This algorithm has no suggested implementations and considered impractical if the dimension is not small. Clarkson et al. (1993) proposed approximation of a center point, finding n/d^2-depth points with proven high probability, using Radon points computation. It has a small constant factor, it is subexponential and can be optimally parallelized to require $O(\log^2 d \log \log n)$ time. Since the center is nonempty, this approximation of a center point can be far from the deepest location.

Given an n-point set S in the plane, Matoušek (1992) finds a Tukey median of S in time $O(n \log^5 n)$. A $\Theta(n \log n)$ lower bound was established for computing a Tukey median, and the upper bound was improved to $O(n \log^4 n)$ by Langerman and Steiger (2000). For $d = 2$, Cole et al. (1987) described an $O(n \log^5 n)$ algorithm to construct a cen-

terpoint, ideas in Cole (1987) could be used to improve the complexity to $O(n \log^3 n)$. Finally Jadhav and Mukhopadhyay (1994) gave a linear time algorithm to construct a center point in the plane. Naor and Sharir (1990) gave an algorithm to compute a center point in dimension three in time $O(n^2 \log^6 n)$.

Given an n-point set S in the plane, D_k, the set of points with half-space depth at least k can be computed in time $O(n \log^4 n)$ (Matoušek, 1992). In the statistics community the program HALFMED (Rousseeuw and Ruts, 1996) and BAGPLOT (Rousseeuw et al., 1999) have been used for this purpose. Miller et al. (2001) proposed an optimal algorithm that computes all the depth contours for a set of points in the plane in time $O(n^2)$ and allows the depth of a point to be queried in time $O(\log^2 n)$. This algorithm uses a topological sweep technique. Compared to HALFMED their algorithm seem to perform much better in practice. Krishnan et al. (2002), based on extensive use of modern graphics architectures, present depth contours computation that performs significantly better than the currently known implementations, outperforming them by at least one order of magnitude and having a strictly better asymptotic growth rate. Their method can only be used in the plane.

Rousseeuw, Hubert, Ruts and Struyf described algorithms for testing the regression depth of a given hyperplane, requiring time $O(n^3)$ in the plane (Rousseeuw and Hubert, 1999a) or time $O(n^{2d-1} \log n)$ in fixed dimensions $d \geq 3$ (Rousseeuw and Hubert, 1999b; Rousseeuw and Struyf, 1998). In the plane van Kreveld et al. (1999) found an algorithm for finding the optimum regression line in time $O(n \log^2 n)$ and it was improved by Langerman and Steiger (2003) to $O(n \log n)$. In any fixed dimension standard ϵ-cutting methods (Mulmuley and Schwarzkopf, 1997) can be used to find a linear time approximation algorithm that finds a hyperplane with regression depth within a factor $(1 - \epsilon)$ of the optimum. By a breadth-first search of the dual hyperplane arrangement one can find theoretically the hyperplane of maximum depth for a given point set in time $\Theta(n^d)$, (van Kreveld et al., 1999). There is no known implementation of this algorithm and the memory requirement is also $\Theta(n^d)$, prohibitive in higher dimensions.

5. Bounding techniques and primal-dual algorithms

Known complexity results in Section 4 suggest that even an ideal algorithm for any of the main problems might require too much time. In such circumstances, it is important to have an algorithm that gives useful information even if the computation is stopped before its comple-

tion. Typically, upper bounds and lower bounds of the optimal value are helpful for the user. Surprisingly, these informations are also useful for designing exact algorithms that might terminate much earlier than the worst-case complexity bounds.

We shall use the term *primal-dual* for algorithms that update both upper and lower bounds of the target measure and terminate as soon as the two bounds coincide.

5.1 Primal-dual algorithms for the halfspace depth

Let $S = \{p_1, \ldots, p_n\}$ be an n-point set and p be a given point in $\mathbb{R}^d$. Here we explain how can one design a primal-dual algorithm for the computation of the halfspace depth of p, based on the ideas used in Fukuda and Rosta (2004).

First of all, one can easily see that any hyperplane through p gives an upper bound for the halfspace depth of p. Any heuristic algorithm, such as the "LP-walk" method that starts from a random hyperplane to a better one in the neighborhood, can find a good candidate hyperplane quickly.

In our algorithm such a random walk is used. To compute exactly the halfspace depth of a point p with respect to the n-point set $S = \{p_1, \ldots, p_n\}$, an oriented hyperplane h is represented by the signvector $X \in \{+, -, 0\}^n$ of a cell in the dual hyperplane arrangement, so that an index j is in the positive support X^+ of the cell X, iff the corresponding point p_j is in the positive halfspace h^+ bounded by h. It is possible to restrict the dual hyperplane arrangement so that the point p is in the positive halfspace. Then the halfspace depth of the point p is the minimum $|X^+|$ over all restricted cells X. The greedy heuristic random walk starts at an arbitrary cell and uses LP successively to move to a neighboring cell with smaller cardinality of positive support if it exists. Successive application define a path starting at the first randomly selected cell, through cells whose signvectors have monotone decreasing number of positive signs, until no more local improvement can be made, arriving to a local minimum. Any local minimum gives an upper bound of the halfspace depth of the point p.

Lower bounds of the halfspace depth can be obtained using integer programming techniques and LP relaxation. In fact the halfspace depth problem, as pointed out by Johnson and Preparata (1978) is an optimization problem and therefore it is natural to use optimization techniques besides geometric or computational ones. A subset R of S is called *minimal dominating set* (MDS) for the point p, if the convex hull

of R contains p, and it is minimal with this property. It follows from Charathéodory's theorem, that an MDS must be a simplex. It is easy to see that the cardinality of a minimum transversal of all MDS's is the halfspace depth of the point p. Assume that the heuristics stops at a cell X. Then, for each $j \in X^+$, there exists at least one MDS R_j such that $R_j \cap X^+ = \{j\}$. Let I be any collection of MDS's containing at least one such R_j for each $j \in X^+$. Then X^+ is a minimal transversal of I. To compute a minimum transversal for I, first the sets $X^- \cup j$ are generated for all $j \in X^+$. Using successive LP's these are reduced to MDS's, forming the set I. Let $\mathbf{c}$ be the characteristic matrix of I, where each row corresponds to an MDS. Let $\mathbf{y}^T = (y_1, \ldots, y_n)$ be a 0/1 vector representing a transversal. The minimum transversal of I is a solution to the integer program: $\min \sum_{i=1}^{n} y_i$ subject to $\mathbf{cy} \geq \mathbf{1}$, $\mathbf{y} \in \{0,1\}^n$. Let us denote by c_I, the cardinality of the minimum transversal of the set I, and by c, the cardinality of the minimum transversal for all MDS's. The optimum value c_L obtained through the LP relaxation (with $0 \leq y_i \leq 1$), satisfies $c_L \leq c_I \leq c = \mathit{depth}$ (the halfspace depth of p). If this lower bound equals the upper bound obtained heuristically, then the global minimum is reached. A lower bound of the halfspace depth can be computed each time an upper bound is obtained by heuristics.

In order to guarantee the termination, one has to incorporate an enumeration scheme in a primal-dual type algorithm. Our primal-dual algorithm incorporates a reverse search algorithm of time complexity $O(n\,\mathrm{LP}(n,d)|\mathcal{C}_S^p|)$ and space complexity $O(n\,d)$, that constructs all $|\mathcal{C}_S^p|$ cells of the dual hyperplane arrangement, for any given S and p, where $\mathrm{LP}(n,d)$ denotes the time complexity of solving a linear program of d variables and n inequality constraints. While the algorithm terminates as soon as the current upper bound and lower bound coincide, the (worst-case) complexity of the primal-dual algorithm is dominated by the complexity of this enumeration algorithm.

To accelerate the termination, this algorithm incorporates a branch-and-bound technique that cuts off a branch in the cell enumeration tree when no improvement can be expected. The bound is based on the elementary fact that no straight line can cross a hyperplane twice. In the preprocessing the repetition of the above mentioned random walk heuristics gives reasonably good local minimum cell C which can become the root cell of the search enumeration tree. The algorithm enumerates the dual hyperplane arrangement's cells keeping in memory the current minimum positive support cardinality. The enumeration is done in reverse using an LP based oracle that can tell the neighbors of any given cell and a function $f(X)$, that computes for any given cell X its neighbor that is on the opposite side of the hyperplane that is first hit by a ray directed

from X to the root cell C. The branch-and-bound technique makes use of the simple observation that if $j \in C^- \cap X^+$ then $j \in Y^+$ for any cell Y below X on the search tree. If the number $g(X)$ of such indices j is larger than the current minimum *depth*, then no improvement can be made below the cell X, thus the corresponding branch is cut.

As we have seen previously, known and recent discrete geometry theorems (Bárány, 1982; Wagner and Welzl, 2001) give upper and lower bounds on the simplicial depth, some of them as a function of the halfspace depth when the data set S satisfies certain additional conditions. In this section we pointed out a different connection between the simplicial depth and the halfspace depth, namely that the halfspace depth of the point p is the minimum cardinality transversal of the simplices containing p.

5.2 Regression depth

The regression depth is a measure of how well a hyperplane fits the data, by checking how many data points must be crossed to obtain a nonfit, a vertical hyperplane, parallel to the dependent variable y's coordinate axis. Let $Z = \{z_1, z_2, \ldots, z_n\}$ be a set of n points in $\mathbb{R}^d$, h be a hyperplane in $\mathbb{R}^d$ and let the point p_h be the dual of h in the oriented dual hyperplane arrangement. The regression depth of the hyperplane h, with respect to Z is the minimum number of hyperplanes in the dual arrangement crossed by any ray starting at the point p_h. The same reverse search hyperplane arrangement enumeration algorithm can be used to compute the regression depth of a hyperplane as the one we used to compute the halfspace depth of a point. The regression depth becomes computable, as the algorithm is memory efficient, requiring $O(nd)$ space.

Let α be a direction and r_α be the ray starting at p_h in the direction α, in the dual hyperplane arrangement. There is an unbounded cell U in the direction α, for any direction α. The point p_h has a signvector corresponding to the face it is located in and each hyperplane crossed by the ray r_α will change the corresponding index to the one in U. To compute the regression depth of a hyperplane h relative to a point set $Z = \{z_1, z_2, \ldots, z_n\}$ is equivalent to finding an unbounded cell U of the dual oriented hyperplane arrangement that has the mimimum difference between its signvector, $\sigma(U)$ and the signvector of the point p_h, $\sigma(p_h)$. The regression depth of the hyperplane h is

$$\min_{U \in \mathcal{U}} |\sigma(U) - \sigma(p_h)|. \tag{3.6}$$

The algorithm enumerates the unbounded cells using the same reverse search algorithm in one lower dimension. It keeps in memory the lowest

current difference and outputs the one with minimal difference obtaining the regression depth.

5.3 Heuristics for the halfspace depth regions and deeper points

The bounding techniques and our primal-dual algorithms can be used for the computation of the halfspace depth regions. Let us choose an appropriate starting point s_1. Using the random walk greedy heuristics one can obtain the upper bound of the halfspace depth of the starting point s_1, that corresponds to a signvector with locally minimum number k of positive signs. The aim is to find a point with halfspace depth at least $k+1$.

The cell with k positive signs corresponds to a k-cell, or a hyperplane h_1, such that h_1 strictly separates k data points in h_1^+ from the remaining ones in h_1^-. Since it is a local minimum, the vertices of the cell are also vertices of the neighboring cells, all of them with $k+1$-positive signs. The hyperplanes in the primal corresponding to those vertices with at least d zeros in their signvectors are candidates to be boundary hyperplanes of D_{k+1}. The attention is restricted to those vertices of the local minimum k-cell in the dual hyperplane arrangement, which have d zeros replacing the negative signs of the k-cell. Any point with halfspace depth at least $(k+1)$ has to be in the feasible region determined by the hyperplanes corresponding to these vertices in the primal. If these hyperplanes do not determine a feasible region, then D_{k+1} must be empty and the deepest point has halfspace depth k.

Therefore one strategy to get deeper and deeper points can be the following: compute heuristically the halfspace depth of s_1, say it is k, by finding a local minimum cell with k positive signs. List all the vertices of this cell. All vertices have at most k positive signs and thus the corresponding closed halfspaces in the primal contain at least $n-k$ data points. Check whether the intersection of the halfspaces is empty. If it is empty, we have a certificate that the maximum depth is at most k. Otherwise we choose a point s_2 in this region away from s_1 that is in h_1^-. Compute heuristically its halfspace depth. If it is k, redo the same with h_2, getting a new smaller region. Then choose a point s_3 in $h_1^- \cap h_2^-$, deeper in the data than s_1, s_2. If the halfspace depth of s_2 is less than k, choose the midpoint of the line segment connecting s_1 and s_2 and redo the same by computing first the halfspace depth of this midpoint. Continue this procedure until the halfspaces corresponding to the vertices of a local minimum cell have empty intersection.

Rousseeuw and Hubert (1999b) designed a heuristic algorithm to find maximum regression depth in the plane, called the "catline."

6. The maximum feasible subsystem problem

The doctoral theses of Parker (1995) and Pfetsch (2002) consider the maximum feasible subsystem problem as their main subject of study. We follow the definitions and notations in Pfetsch (2002). Let $\Sigma\colon A\mathbf{x} \leq \mathbf{b}$ be an infeasible linear inequality system, with $A \in \mathbb{R}^{n\times d}$ and $\mathbf{b} \in \mathbb{R}^n$. To fit into the complexity theory the coefficients are finitely represented, rational or real algebraic numbers. The maximum feasible subsystem problem **Max FS** is to find a feasible subsystem of a given infeasible system $\Sigma\colon A\mathbf{x} \leq \mathbf{b}$, containing as many inequalities as possible.

A subsystem Σ' of an infeasible system $\Sigma\colon A\mathbf{x} \leq \mathbf{b}$ is an *irreducible inconsistent subsystem* IIS, if Σ' is infeasible and all of its proper subsystems are feasible.

The minimum irreducible inconsistent subsystem problem **Min IIS** is to find a minimum cardinality IIS of a given infeasible system $\Sigma\colon A\mathbf{x} \leq \mathbf{b}$. The **Min IIS Transversal** problem is to find an IIS-transversal T of minimum cardinality, where T is an *IIS-transversal*,[1] if $T \cap C \neq \varnothing$ for all IIS C of Σ. The Min IIS-transversal problem has the following integer programming formulation:

$$\min \sum_{i=1}^{n} y_i \tag{3.7}$$

$$\text{subject to } \sum_{i\in C} y_i \geq 1 \text{ for all IIS } C \text{ of } \Sigma, \text{ and} \tag{3.8}$$

$$y_i \in \{0,1\} \text{ for all } 1 \leq i \leq n. \tag{3.9}$$

The complexity of Max FS is strongly NP-hard, see Chakravarti (1994) and Johnson and Preparata (1978), even when the matrix A has only $-1, 1$ coefficients, or A is totally unimodular and $\mathbf{b}$ is an integer. In the special case when $[A \quad \mathbf{b}]$ is totally unimodular Max FS is solvable in polynomial time, see Sankaran (1993). The complexity of Max FS approximation was studied by Amaldi and Kann (1995, 1998). They showed that Max FS can be approximated within a factor of two, but it does not admit a polynomial-time approximation scheme unless P = NP. Max FS and Min IIS Transversal are polynomially equivalent, but their

[1] IIS-transversal is sometimes called *IIS cover*. We chose not to use this terminology in this paper, as it must be dualized to casted as a special case of the set cover problem.

approximation complexities are very different, namely Min IIS Transversal cannot be approximated within any constant.

Parker (1995) and Parker and Ryan (1996) developed an exact algorithm to solve Min IIS Transversal which solves the above mentioned integer program for a partial list of IISs, using an integer programming solver. Then either the optimal transversal is obtained, or at least one uncovered IIS is found and the process is iterated. The algorithm uses standard integer programming techniques, in particular branch-and-cut method, cutting planes combined with branch-and-bound techniques and linear programming relaxations.

Parker (1995) studied the associated 0/1 polytope, i.e. the convex hull of all incidence vectors of feasible subsystems of a given infeasible linear inequality system. Amaldi et al. (2003) continued the polyhedral study. They found a new geometric characterization of IISs and proved various NP-hardness results, in particular they proved that the problem of finding an IIS of smallest cardinality is NP-hard and hard to approximate. The IISs are the supports of the vertices of the alternative (Farkas-dual) polyhedron $P = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^T A = 0, \mathbf{y^T b} = -1, \mathbf{y} \geq 0\}$, where the number of IISs in the worst case is $\Theta(n^{\lfloor d'/2 \rfloor})$ and $d' = n - (d+1)$. Our experiments indicate that already for $d = 5$ and $n = 40$ this number is too large for the type of computation these algorithms suggest and the memory requirement is also prohibitive. Compared to this the exact algorithm developed in Fukuda and Rosta (2004) is memory efficient. Additionally our worst case time complexity is also much smaller. One of the main differences is that in our algorithm the time complexity is $O(n^d)$, the number of cells in the hyperplane arrangement given by Σ, while the IIS enumerating algorithms of Parker, Ryan and Pfetsch has time complexity $O(n^{(n-d-1)/2})$ that corresponds to the number of vertices of the dual polytope given by Farkas's lemma, which is much larger if n is large compared to d. It might be useful to combine all the existing ideas and proceed in parallel with the original and the Farkas dual, using all information obtained in both to improve the primal-dual aspects of these algorithms.

Parker, Ryan and Pfetsch also studied the IIS-hypergraphs, where each node corresponds to an inequality and each hyperedge corresponds to an IIS. An IIS-transversal hypergraph corresponds to an IIS-transversal. It is unknown whether there is an output sensitive algorithm, i.e., polynomial in the input and output sizes, that enumerates all IIS-transversals.

Pfetsch (2002) remarks that if an infeasible linear inequality system Σ: $\{A\mathbf{x} \leq \mathbf{b}\}$ is given, the IIS-transversal hypergraph corresponding to Σ can be computed using the affine oriented hyperplane arrangement

corresponding to Σ. For each point $z \in \mathbb{R}^d$ there are inequalities violated and the corresponding indices form the negative support $S^-(z)$. If we remove the inequalities in $S^-(z)$ from Σ a feasible system remains therefore $S^-(z)$ is an IIS-transversal for each $z \in \mathbb{R}^d$. Moreover if Σ does not contain implicit equations then every minimal IIS-transversal corresponds to a cell in the affine oriented hyperplane arrangement. Thus the IIS-transversal hypergraph can be generated by enumerating all cells of the arrangement. In Pfetsch (2002) it is suggested that all cells could be enumerated with the reverse search algorithm of Avis and Fukuda (1996), and then the ones that correspond to minimal IIS-transversals could be found. No details or implementation are reported in Pfetsch (2002).

Several greedy type heuristics were proposed to solve Min IIS Transversal in order to deal with infeasibility of large linear inequality systems. First the problem of identifying IISs with a small and possibly minimum number of inequalities was considered (Greenberg and Murphy, 1991) which is the same as solving Min IIS. Chinneck (1997) and Chinneck and Dravnieks (1991) proposed several greedy type heuristics, now available in commercial LP-solvers, such as CPLEX and MINOS, see Chinneck (1996a). Improved versions by Chinneck (1996b, 2001) give greedy type heuristic algorithms for Min IIS Transversal to avoid overlapping IISs and the resulting lack of information.

One main application consists of finding a linear classifier that distinguishes between two classes of data. A set of points is given in $\mathbb{R}^d$ each belonging to one of two classes. The aim is to find a hyperplane that separates the two classes with minimum possible number of misclassification. This hyperplane has good chance to classify a new data point correctly. The linear classification problem can be easily formulated as a Min IIS Transversal in $\mathbb{R}^{d+1}$. For this Min IIS Transversal application in machine learning several heuristics use methods from nonlinear programming, see Bennett and Bredensteiner (1997), Bennett and Mangasarian (1992) and Mangasarian (1994). Mangasarian (1999) introduced heuristics for Min IIS.

Agmon (1954) and Motzkin and Schoenberg (1954) developed the *relaxation method* to find a feasible solution of a system of linear inequalities. If the system is feasible and full-dimensional the fixed stepsize iteration process terminates in finite steps. If applied to an infeasible system, the procedure neither terminates nor converges, but decreasing step length after each iteration can result in convergence. Randomized decisions can also be incorporated together with other variants as used by Amaldi (1994) and Amaldi and Hauser (2001) successfully in some applications. This method can only be applied if no implicit equations

are present in the system, but many applications can be formulated with strict inequalities only, for example in machine learning, protein folding and digital broadcasting.

7. Conclusion

It became evident by looking at the respective literature in statistics, discrete and computational geometry and optimization that closely related problems have been studied in these fields. Though there has been strong interaction between researchers of statistics and computational geometers, the optimization community appears to be rather isolated. We reviewed many available tools from geometric and optimization computations, the combination of exact, primal-dual, heuristic and random algorithms for multivariate ranks, generalization of median, classification or infeasibility related questions. The purpose of this survey is to demonstrate that pooling together all relevant areas can help to get ahead in this very difficult subject. Since we have to deal with NP-hard problems which are also hard to approximate, there is almost no hope for finding a polynomial algorithm. However, this does not mean that we should give up hope for finding practical algorithms. The Travelling Salesman Problem (TSP) is a famous hard problem that has been solved successfully by the team Applegate et al. (1998). We strongly hope that similar efforts will be made to develop practical algorithms and implementations, perhaps exploiting parallel computation, for the data depth and maximum feasibility subsystem problems.

Acknowledgments. We would like to thank the referees for many helpful comments. The research of K. Fukuda is partially supported by an NSERC grant (RGPIN 249612), Canada. The research of V. Rosta is partially supported by an NSERC grant (RGPIN 249756), Canada.

References

Agmon, S. (1954). The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:382–392.

Alon, N., Bárány, I., Füredi Z., and Kleitman, D. (1992). Point selections and weak ϵ-nets for convex hulls. *Combinatorics, Probability and Computing*, 1(3):189–200.

Aloupis, G., Cortes, C., Gomez, F., Soss, M., and Toussaint, G. (2002). Lower bounds for computing statistical depth. *Computational Statistics and Data Analysis*, 40(2):223–229.

Aloupis, G., Langerman, S., Soss, M., and Toussaint, G. (2001). Algorithms for bivariate medians and a Fermat–Toricelli problem for

lines. *Proceedings of the 13th Canadian Conference on Computational Geometry*, pp. 21–24.

Amaldi, E. (1991). On the complexity of training perceptrons. In: T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (eds.), *Artificial Neural Networks*, pp. 55–60. Elsevier, Amsterdam.

Amaldi, E. (1994). *From Finding Maximum Feasible Subsystems of Linear Systems to Feedforward Neural Network Design.* Ph.D. thesis, Dept. of Mathematics, EPF-Lausanne, 1994.

Amaldi, E. and Hauser, R. (2001). Randomized relaxation methods for the maximum feasible subsystem problem. Technical Report 2001-90, DEI, Politecnico di Milano.

Amaldi, E. and Kann, V. (1995). The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147:181–210.

Amaldi, E. and Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260.

Amaldi E. and Mattavelli, M. (2002). The MIN PCS problem and piecewise linear model estimation. *Discrete Applied Mathematics*, 118:115–143.

Amaldi, E., Pfetsch M.E., and Trotter, L.E., Jr. (2003). On the maximum feasible subsystem problem, IISs, and IIS-hypergraphs. *Mathematical Programming*, 95(3):533–554.

Amenta, N., Bern, M., Eppstein D., and Teng, S.H. (2000). Regression depth and center points. *Discrete and Computational Geometry*, 23:305–323.

Andrzejak A. and Fukuda, K. (1999). Optimization over k-set polytopes and efficient k-set enumeration. In: *Proceedings of the 6th International Workshop on Algorithms and Data Structures* (WADS'99), pp. 1–12. Lecture Notes in Computer Science, vol. 1663, Springer, Berlin.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (1998). On the solution of traveling salesman problems. *Proceedings of the International Congress of Mathematicians*, Vol. III. *Documenta Mathematica*, 1998:645–656.

Avis, D. (1993). The m-core properly contains the m-divisible points in space. *Pattern Recognition Letters*, 14(9):703–705.

Avis, D. and Fukuda, K. (1992). A pivoting algorithm for convex hulls and vertex enumeration of arrangements of polyhedra. *Discrete and Computational Geometry*, 8:295–313.

Avis, D. and Fukuda, K. (1996). Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3):21–46.

Bárány, I. (1982). A generalization of Carathéodory's theorem. *Discrete Mathematics*, 40:141–152.

Bárány, I. Personal communication.

Bárány, I. and Onn, S. (1997a). Colourful linear programming and its relatives. *Mathematics of Operations Research*, 22:550–567.

Bárány, I. and Onn, S. (1997). Carathéodory's theorem, colourful and applicable. In: *Intuitive Geometry (Budapest, 1995)*, pp. 11–21. Bolyai Society Mathematical Studies, vol. 6. János Bolyai Math. Soc., Budapest.

Barnett, V. (1976). The ordering of multivariate data. *Journal of the Royal Statistical Society. Series A*, 139(3):318–354.

Bebbington, A.C. (1978). A method of bivariate trimming for robust estimation of the correlation coefficient. *Journal of the Royal Statistical Society. Series C*, 27:221–226.

Bennett, K.P. and Bredensteiner, E.J. (1997). A parametric optimization method for machine learning. *INFORMS Journal of Computing*, 9(3):311–318.

Bennett, K.P. and Mangasarian, O.L. (1992). Neural network training via linear programming. In: P.M. Pardalos (ed.), *Advances in Optimization and Parallel Computing*, pp. 56–67. North-Holland, Amsterdam.

Boros, E. and Z. Füredi, Z. (1984). The number of triangles covering the center of an n-set. *Geometriae Dedicata*, 17:69–77.

Burr, M.A., Rafalin E., and Souvaine, D.L. (2003). Simplicial depth: An improved definition, analysis, and efficiency for the finite sample case. DIMACS, Technical Reports no. 2003–28.

Chakravarti, N. (1994). Some results concerning post-infeasibility analysis. *European Journal of Operational Research*, 73:139–143.

Cheng, A.Y. and Ouyang, M. (2001). On algorithms for simplicial depth. In:*Proceedings of the 13th Canadian Conference on Computational Geometry*, pp. 53–56.

Chinneck, J.W. (1996a). Computer codes for the analysis of infeasible linear programs. *Operational Research Society Journal*, 47(1):61–72.

Chinneck, J.W. (1996b). An effective polynomial-time heuristic for the minimum-cardinality IIS set-covering problem. *Annals of Mathematics and Artificial Intelligence*, 17(1-2):127–144.

Chinneck, J.W. (1997). Finding a useful subset of constraints for analysis in an infeasible linear program. *INFORMS Journal of Computing*, 9(2):164–174.

Chinneck, J.W. (2001). Fast heuristics for the maximum feasible subsystem problem. *INFORMS Journal of Computing*, 13(3):210–223.

Chinneck, J.W. and Dravnieks, E.W. (1991). Locating minimal infeasible constraint sets in linear programs. *ORSA Journal on Computing*, 3(2):157–168.

Clarkson, K.L., Eppstein, D., Miller, G.L., Sturtivant, C., and Teng, S.H. (1996). Approximating center points with iterative Radon points. *International Journal of Computational Geometry and Applications*, 6:357–377.

Cole, R. (1987). Slowing down sorting networks to obtain faster sorting algorithms. *Journal of the ACM*, 34:200–208.

Cole, R., Sharir, M., and Yap, C. (1987). On k-hulls and related problems. *SIAM Journal on Computing*, 16(1):61–67.

Danzer, L., Grünbaum, B., and Klee, V. (1963). Helly's theorem and its relatives. In: *Proceedings of Symposia in Pure Mathematics*, vol. 7, pp. 101–180. Amer. Math. Soc., Providence, RI.

Dempster, A.P. and Gasko, M.G. (1981). New tools for residual analysis. *The Annals of Statistics*, 9:945–959.

Donoho, D.L. (1982). Breakdown properties of multivariate location estimators. Ph.D. qualifying paper, Dept. Statistics, Harvard Univ.

Donoho, D.L. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *The Annals of Statistics*, 20(4):1803–1827.

Donoho, D.L. and Huber, P.J. (1982). The notion of breakdown point. In: P.J. Bickel. K.A. Doksum and J.I. Hodges, Jr. (eds.), *Festschrift for Erich L. Lehmann in honor of his sixty-fifth birthday*, pp. 157–184. Wadsworth, Belmont, CA.

Eddy, W. (1982). Convex hull peeling. In: H. Caussinus (ed.), *COMPSTAT*, pp. 42–47. Physica-Verlag, Wien.

Edelsbrunner, H., O'Rourke, J., and Seidel, R. (1986). Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing*, 15:341–363.

Eppstein, D. (2003). Computational geometry and statistics. Mathematical Sciences Research Institute, Discrete and Computational Geometry Workshop.

Ferrez, J.A., Fukuda, K., and Liebling, T. (2005). Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm, *European Journal of Operations Research*, to appear.

Fukuda, K. (2002). *cddlib reference manual, cddlib Version 092b*. Swiss Federal Institute of Technology, Zürich.

Fukuda, K. and Rosta, V. (2004). Exact parallel algorithms for the location depth and the maximum feasible subsystem problems. In: C.A. Floudas and P.M. Pardalos (eds.), *Frontiers in Global Optimization*,

pp. 123–134. Kluwer Academic Publishers.

Gil, J., Steiger, W., and Wigderson, A. (1992). Geometric medians. *Discrete Mathematics*, 108(1-3):37–51.

Gleeson, J. and Ryan, J. (1990). Identifying minimally infeasible subsystems of inequalities. *ORSA Journal on Computing*, 2(1):61–63.

Gnanadesikan, R. and Kettenring, J.R. (1972). Robust estimates, residuals and outlier detection with multiresponse data. *Biometrics*, 28:81–124.

Greenberg, H.J. and Murphy, F.H. (1991). Approaches to diagnosing infeasible linear programs. *ORSA Journal on Computing*, 3(3):253–261.

Hettmansperger, T. and McKean, J. (1977). A robust alternative based on ranks to least squares in analyzing linear models. *Technometrics*, 19:275–284.

Hodges, J. (1955). A bivariate sign test. *The Annals of Mathematical Statistics*, 26:523–527.

Huber, P.J. (1977). Robust covariances. In: S.S. Gupta and D.S. Moore (eds.), *Statistical Decision Theory and Related Topics. II*, pp. 165–191. Academic, New York.

Huber, P.J. (1985). Projection pursuit (with discussion). *The Annals of Statistics*, 13:435–525.

Jadhav, S. and Mukhopadhyay, A. (1994). Computing a centerpoint of a finite planar set of points in linear time. *Discrete and Computational Geometry*, 12:291–312.

Johnson, D.S. and Preparata, F.P. (1978). The densest hemisphere problem. *Theoretical Computer Science*, 6:93–107.

Kann, V. (1992). *On the Approximability of NP-Complete Optimization Problems*. Ph.D. Thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm.

van Kreveld, M., Mitchell, J.S.B., Rousseeuw, P.J., Sharir, M., Snoeyink, J., and Speckmann, B. (1999). Efficient algorithms for maximum regression depth. In: *Proceedings of the 15th Symposium on Computational Geometry*, pp. 31–40. ACM.

Krishnan, S., Mustafa, N.H., and Venkatasubramanian, S. (2002). Hardware-assisted computation of depth contours. In: *13th ACM-SIAM Symposium on Discrete Algorithms*, pp. 558–567.

Langerman, S. and Steiger, W. (2000). Computing a maximal depth point in the plane. In: *Proceedings of the Japan Conference on Discrete and Computational Geometry* (JCDCG 2000).

Langerman, S. and Steiger, W. (2003). The complexity of hyperplane depth in the plane. *Discrete and Computational Geometry*, 30(2):299–309.

Liu, R. (1990). On a notion of data depth based on random simplices. *The Annals of Statistics*, 18(1):405–414.

Liu, R. (1992). Data depth and multivariate rank tests, L_1. In: Y. Dodge (ed.), *Statistical Analysis and Related Methods*, pp. 279–294. Elsevier, Amsterdam.

Liu, R. (1995). Control charts for multivariate processes. *Journal of the American Statistical Association*, 90:1380–1388.

Liu, R. (2003). Data depth: Center-outward ordering of multivariate data and nonparametric multivariate statistics. In: M.G. Akritas and D.N. Politis (eds.), *Recent advances and Trends in Nonparametric Statistics*, pp. 155–167. Elsevier.

Liu, R., Parelius, J., and Singh, K. (1999). Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion). *The Annals of Statistics*, 27:783–858.

Liu, R. and Singh, K. (1992). Ordering directional data: concepts of data depth on circles and spheres. *The Annals of Statistics*, 20:1468–1484.

Liu, R. and Singh, K. (1993). A quality index based on data depth and multivariate rank tests. *Journal of the American Statistical Association*, 88:257–260.

Liu, R. and Singh, K. (1997). Notions of limiting P-values based on data depth and bootstrap. *Journal of the American Statistical Association*, 91:266–277.

Lopuhaä, H.P. (1988). Highly efficient estimates of multivariate location with high breakdown point. Technical report 88-184, Delft Univ. of Technology.

Lopuhaä, H.P. and Rousseeuw, P.J. (1991). Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, 19:229–248.

Mahalanobis, P.C. (1936). On the generalized distance in statistics. *Proceedings of the National Academy India*, 12:49–55.

Mangasarian, O.L. (1994). Misclassification minimization. *Journal of Global Optimization*, 5(4):309–323.

Mangasarian, O.L. (1999). Minimum-support solutions of polyhedral concave programs. *Optimization*, 45(1-4):149–162.

Maronna, R.A. (1976). Robust M-estimates of multivariate location and scatter. *The Annals of Statistics*, 4:51–67.

Matoušek, J. (1992). Computing the center of a planar point set. In: J.E. Goodman, R. Pollack, and W. Steiger (eds.), *Discrete and Computational Geometry*, pp. 221–230. Amer. Math. Soc.

Matoušek, J. (2002). *Lectures on Discrete Geometry*, Graduate Texts in Mathematics, Springer-Verlag, New York.

Miller, K., Ramaswami, S., Rousseeuw, P., Sellares, T., Souvaine, D., Streinu I., and Struyf, A. (2001). Fast implementation of depth contours using topological sweep. *Proceedings of the Twelfth ACM-SIAM Symposium on Discrete Algorithms, Washington, DC*, pp. 690–699.

Motzkin, T.S. and Schoenberg, I.J. (1954). The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:393–404.

Mulmuley K. and Schwarzkopf, O. (1997). Randomized algorithms. In: *Handbook of Discrete and Computational Geometry*, Chapter 34, pp. 633–652. CRC Press.

Naor N. and Sharir, M. (1990). Computing a point in the center of a point set in three dimensions. *Proceedings of the 2nd Canadian Conference on Computational Geometry*, pp. 10–13.

Niinimaa, A., Oja, H., and Tableman, M. (1990). On the finite-sample breakdown point of the bivariate median. *Statistics and Probability Letters*, 10:325–328.

Oja, H. (1983). Descriptive statistics for multivariate distributions. *Statistics and Probability Letters*, 1:327–332.

Onn, S. (2001). The Radon-split and the Helly-core of a point configuration. *Journal of Geometry*, 72:157–162.

Orlik, P. and Terao, H. (1992). *Arrangements of Hyperplanes*. Springer.

Parker, M. (1995). *A Set Covering Approach to Infeasibility Analysis of Linear Programming Problems and Related Issues*. Ph.D. thesis, Department of Mathematics, University of Colorado at Denver.

Parker M. and Ryan, J. (1996). Finding the minimum weight IIS cover of an infeasible system of linear inequalities. *Annals of Mathematics and Artificial Intelligence*, 17(1-2):107–126.

Pfetsch, M. (2002). *The Maximum Feasible Subsystem Problem and Vertex-Facet Incidences of Polyhedra*. Ph.D. thesis, Technischen Universitat Berlin.

Rafalin, E. and Souvaine, D.L. (2004). Computational geometry and statistical depth measures, theory and applications of recent robust methods. In: M. Hubert, G. Pison, A. Struyf and S. Van Aelst (eds.). *Theory and applications of recent robust methods*, pp. 283–295. Statistics for Industry and Technology, Birkhäuser, Basel.

Reay, J.R. (1982). Open problems around Radon's theorem. In: D.C. Kay and M. Breen (eds.), *Convexity and Related Combinatorial Geometry*, pp. 151–172. Marcel Dekker, Basel.

Reiss, S. and Dobkin, D. (1976). The complexity of linear programming. Technical Report 69, Department of Computer Science, Yale University, New Haven, CT.

Rossi, F., Sassano, A., and Smriglio, S. (2001). Models and algorithms for terrestrial digital broadcasting. *Annals of Operations Research*,

107(3):267–283.

Rousseeuw, P.J. (1985). Multivariate estimation with high breakdown point. In: W. Grossman, G. Pflug, I. Vincze and W. Wertz (eds.), *Mathematical Statistics and Applications, Vol. B*, pp. 283–297. Reidel, Dordrecht.

Rousseeuw, P.J. and Hubert, M. (1999a). Regression depth. *Journal of American Statistical Association*, 94:388–402.

Rousseeuw, P.J. and Hubert, M. (1999b). Depth in an arrangement of hyperplanes. *Discrete and Computational Geometry*, 22:167–176.

Rousseeuw, P.J. and Leroy, A.M. (1987). *Robust Regression and Outlier Detection*. Wiley, New York.

Rousseeuw, P.J. and Ruts, I. (1996). Computing depth contours of bivariate clouds. *Computational Statistics and Data Analysis*, 23:153–168.

Rousseeuw, P.J., Ruts, I., and Tukey, J.W. (1999). The bagplot: A bivariate boxplot. *The American Statistician*, 53(4):382–387.

Rousseeuw, P.J. and Struyf, A. (1998). Computing location depth and regression depth in higher dimensions. *Statistics and Computing*, 8:193–203.

Ryan, J. (1991). Transversals of IIS-hypergraphs. *Congressus Numerantium*, 81:17–22.

Ryan, J. (1996). IIS-hypergraphs. *SIAM Journal on Discrete Mathematics*, 9(4):643–653.

Sankaran, J.K. (1993). A note on resolving infeasibility in linear programs by constraint relaxation. *Operations Research Letters*, 13:19–20.

Sierksma, G. (1982). Generalizations of Helly's theorem: Open problems. In: D.C. Kay and M. Breen (eds.), *Convexity and related Combinatorial Geometry*, pp. 173–192. Marcel Dekker, Basel.

Singh, K. (1993). On the majority data depth. Technical Report, Rutgers University.

Struyf, A. and Rousseeuw, P.J. (2000). High-dimensional computation of the deepest location. *Computational Statistics and Data Analysis*, 34:415–426.

Teng, S.H. (1991). *Points, Spheres and Separators: A Unified Geometric Approach to Graph Partitioning*. Ph.D. Thesis, Carnegie-Mellon Univ. School of Computer Science.

Titterington, D.M. (1978). Estimation of correlation coefficients by ellipsoidal trimming. *Journal of the Royal Statistical Society. Series C*, 27:227–234.

Tukey, J.W. (1974a). Order statistics. In mimeographed notes for Statistics 411, Princeton Univ.

Tukey, J.W. (1974b). Address to International Congress of Mathematics, Vancouver.

Tukey, J.W. (1975). Mathematics and the picturing of data. In: *Proceedings of the International Congress of Mathematicians, Vol. 2*, pp. 523–531.

Tukey, J.W. (1977). *Exploratory Data Analysis.* Addison-Wesley, Reading, MA.

Tverberg, H. (1966). A generalization of Radon's theorem, *Journal of the London Mathematical Society*, 41:123–128.

Wagner, M., Meller, J., and Elber, R. (2002). Large-scale linear programming techniques for the design of protein folding potentials. Technical Report TR-2002-02, Old Dominion University.

Wagner, U. (2003). *On* k*-Sets and Applications.* Ph.D. Thesis, Theoretical Computer Science, ETH Zurich.

Wagner, U. and Welzl, E. (2001). A continuous analogue of the upper bound theorem. *Discrete and Computational Geometry*, 26(3):205–219.

Yeh, A. and Singh, K. (1997). Balanced confidence sets based on the Tukey depth. *Journal of the Royal Statistical Society. Series B*, 3:639–652.

Zuo, Y. and Serfling, R. (2000). General notions of statistical depth functions. *Annals of Statististics*, 28(2):461–482.

Chapter 4

THE MAXIMUM INDEPENDENT SET PROBLEM AND AUGMENTING GRAPHS

Alain Hertz
Vadim V. Lozin

Abstract In the present paper we review the method of augmenting graphs, which is a general approach to solve the maximum independent set problem. Our objective is the employment of this approach to develop polynomial-time algorithms for the problem on special classes of graphs. We report principal results in this area and propose several new contributions to the topic.

1. Introduction

The maximum independent set problem is one of the central problems of combinatorial optimization, and the method of augmenting graphs is one of the general approaches to solve the problem. It is in the heart of the famous solution of the maximum matching problem, which is equivalent to finding maximum independent sets in line graphs. Recently, the approach has been successfully applied to develop polynomial-time algorithms to solve the maximum independent set problem in many other special classes of graphs. The present paper summarizes classical results and recent advances on this topic, and proposes some new contributions to it.

The organization of the paper is as follows. In the rest of this section we introduce basic notations. Section 2 presents general information on the maximum independent set problem, describes its relationship with other problems of combinatorial optimization, shows some applications, etc. In Section 3 we outline the idea of augmenting graphs and prove several auxiliary results related to this notion. Section 4 is devoted to the characterization of augmenting graphs in some special classes, and

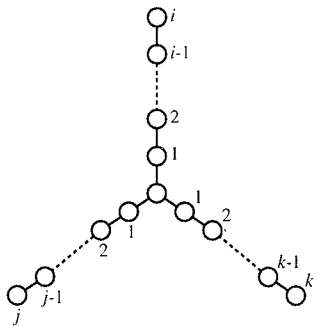


Figure 4.1. The graph $S_{i,j,k}$.

Section 5 describes algorithms to identify augmenting graphs of various types.

All graphs in this paper are undirected, without loops and multiple edges. For a graph G, we denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively, and by $\overline{G}$ the complement of G. Given a vertex x in G, we let $N(x) := \{y \in V(G) \mid xy \in E(G)\}$ denote the neighborhood of x, and $\deg(x) := |N(x)|$ the degree of x. The degree of G is $\Delta(G) := \max_{x \in V(G)} \deg(x)$. If W is a subset of $V(G)$, we denote by $N_W(x) := N(x) \cap W$ the neighborhood of x in the subset W, and by $N(W) := \bigcup_{x \in W} N_{V(G)-W}(x)$ the neighborhood of W. Also, $N_U(W) := N(W) \cap U$ is the neighborhood of W in a subset $U \subseteq V(G)$. As usual, P_n is the chordless path (chain), C_n is the chordless cycle and K_n is the complete graph on n vertices. By $K_{n,m}$ we denote the complete bipartite graph with parts of size n and m, and by $S_{i,j,k}$ the graph represented in Figure 4.1. In particular, $S_{1,1,1} = K_{1,3}$ is a claw, $S_{1,1,2}$ is a fork (called also a chair), and $S_{0,j,k} = P_{j+k+1}$. The graph obtained from $S_{1,1,k}$ by adding a new vertex adjacent to the two vertices of degree 1 of distance 1 from the vertex of degree 3 will be called Banner_k. This is a generalization of Banner_1 known in the literature simply as a *banner*.

2. The maximum independent set problem

An *independent set* in a graph (called also a *stable set*) is a subset of vertices no two of which are adjacent. There are different problems associated with the notion of independent set, among which the most important one is the MAXIMUM INDEPENDENT SET problem. In the *decision version* of this problem, we are given a graph G and an integer K, and the problem is to determine whether G contains an independent set of cardinality at least K. The *optimization version* deals with finding

in G an independent set of maximum cardinality. The number of vertices in a maximum cardinality independent set in G is called the *independence* (*stability*) *number* of G and is denoted $\alpha(G)$. One more version of the same problem consists in computing the independence number of G. All three versions of this problem are polynomially equivalent and we shall refer to any of them as the MAXIMUM INDEPENDENT SET (MIS) problem.

The MAXIMUM INDEPENDENT SET problem is NP-hard in general graphs and remains difficult even under substantial restrictions, for instance, for cubic planar graphs (Gareyet al., 1976)). Alekseev (1983) has proved that if a graph H has a connected component which is not of the form $S_{i,j,k}$, then the MIS is NP-hard in the class of H-free graphs. On the other hand, it admits polynomial-time solutions for graphs in special classes such as bipartite or, more generally, perfect graphs (Grötschel et al., 1984).

An independent set S is called *maximal* if no other independent set properly contains S. Much attention has been devoted in the literature to the problem of generating *all* maximal independent sets in a graph (see for example Johnson and Yannakakis, 1988; Lawler et al., 1980; Tsukiyama et al., 1977). Again, there are different versions of this problem depending on definitions of notions of "performance" or "complexity" (see for example Johnson and Yannakakis, 1988, for definitions): polynomial total time, incremental polynomial time, polynomial delay, specified order, polynomial space.

One more problem associated with the notion of independent set is that of finding in a graph a maximal independent set of minimum cardinality, also known in the literature as the INDEPENDENT DOMINATING SET problem. This problem is more difficult than the MAXIMUM INDEPENDENT SET in the sense that it is NP-hard even for bipartite graphs, where MIS can be solved in polynomial time.

In the present paper, we focus on the MAXIMUM INDEPENDENT SET problem. This is one of the central problems in graph theory that is closely related to many other problems of combinatorial optimization. For instance, if S is an independent set in a graph $G = (V, E)$, then S is a clique in the complement $\overline{G}$ of G and $V - S$ is a vertex cover of G. Therefore, the MAXIMUM INDEPENDENT SET problem in a graph G is equivalent to the MAXIMUM CLIQUE problem in $\overline{G}$, and the MINIMUM VERTEX COVER in G. A *matching* in a graph $G = (V, E)$, i.e., an independent set of edges, corresponds to an independent set of vertices in the line graph of G, denoted $L(G)$ and defined as follows: the vertices of $L(G)$ are the edges of G, and two vertices of $L(G)$ are adjacent if and only if their corresponding edges in G are adjacent. Thus, the MAXIMUM MATCHING problem coincides with the MAXIMUM INDEPENDENT

SET problem restricted to the class of line graphs. Unlike the general case, the MIS can be solved in polynomial time in the class of line graphs, which is due to the celebrated matching algorithm proposed in Edmonds (1965).

The weighted version of the MAXIMUM INDEPENDENT SET problem, also known as the VERTEX PACKING problem, deals with graphs whose vertices are weighted with positive integers, the problem being to find an independent set of maximum total weight. In Ebenegger et al. (1984), this problem has been shown to be equivalent to maximizing a pseudo-Boolean function, i.e., a real-valued function with Boolean variables. Notice that pseudo-Boolean optimization is a general framework for a variety of problems of combinatorial optimization such as MAX-SAT or MAX-CUT (Boros and Hammer, 2002).

There are numerous generalizations and variations around notions of independent sets and matchings (independent sets of edges). Consider, for instance, a subset of vertices inducing a subgraph with vertex degree at most k. For $k = 0$, this coincides with the notion of independent set. For $k = 1$, this notion is usually referred in the literature as a *dissociation set*. As shown in Yannakakis (1981), the problem of finding a dissociation set of maximum cardinality is NP-hard in the class of bipartite graphs. Both the MAXIMUM INDEPENDENT SET and the MAXIMUM DISSOCIATION SET problems belong to a more general class of hereditary subset problems (Halldórsson, 2000). Another generalization of the notion of independent set has been recently introduced under the name *k-insulated set* (Jagota et al., 2001).

Consider now a subset of vertices of a graph G inducing a subgraph H with vertex degree exactly 1. The set of edges of H is called an *induced matching* of G (Cameron, 1989). Similarly to the ordinary MAXIMUM MATCHING problem, the MAXIMUM INDUCED MATCHING can be reduced to the MAXIMUM INDEPENDENT SET problem by associating with G an auxiliary graph, which is the square of $L(G)$, where the square of a graph $H = (V, E)$ is the graph with vertex set V in which two vertices x and y are adjacent if and only if the distance between x and y in H is at most 2. However, unlike the MAXIMUM MATCHING problem, the MAXIMUM INDUCED MATCHING problem is NP-hard even for bipartite graphs with maximum degree 3 (Lozin, 2002a).

One more variation around matchings is the problem of finding in a graph a maximal matching of minimum cardinality, known also as the MINIMUM INDEPENDENT EDGE DOMINATING SET problem (Yannakakis and Gavril, 1980). This problem reduces to MIS by associating with the input graph G the total graph of G consisting of a copy of G, a copy of $L(G)$ and the edges connecting a vertex v of G to a vertex e of $L(G)$ if and

only if v is incident to e in G. By exploiting this association, Yannakakis and Gavril (1980) proved NP-hardness of the MAXIMUM INDEPENDENT SET problem for total graphs of bipartite graphs. Some more problems related to the notion of matching can be found in Müller (1990), Plaisted and Zaks (1980), and Stockmeyer and Vazirani (1982).

Among various applications of the MAXIMUM INDEPENDENT SET (MAXIMUM CLIQUE) problem let us distinguish two examples. The origin of the first one is the area of computer vision and pattern recognition, where one of the central problems is the matching of relational structures. In graph theoretical terminology, this is the GRAPH ISOMORPHISM, or more generally, MAXIMUM COMMON SUBGRAPH problem. It reduces to the MAXIMUM CLIQUE problem by associating with a pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ a special graph $G = (V, E)$ (known as the *association graph* Barrow and Burstal, 1976; Pelillo et al., 1999) with vertex set $V = V_1 \times V_2$ so that two vertices $(i, j) \in V$ and $(k, l) \in V$ are adjacent in G if and only if $i \neq k$, $j \neq l$ and $ik \in E_1 \iff jl \in E_2$. Then a maximum common subgraph of the graphs G_1 and G_2 corresponds to a maximum clique in G.

Another example comes from information theory. The graph theoretical model arising here can be roughly described as follows. An information source sends messages in the alphabet $X = \{x_1, x_2, \ldots, x_n\}$. Along the transmission some symbols of X can be changed to others because of random noise. Let G be a graph with $V(G) = X$ and $x_i x_j \in E(G)$ if and only if x_i and x_j can be interchanged during transmission. Then a noise-resistant code should consist of the symbols of X that constitute an independent set in G. Therefore, a largest noise-resistant code corresponds to a largest independent set in G.

For more information about the MAXIMUM CLIQUE (MAXIMUM INDEPENDENT SET) problem, including application, complexity issues, etc., we refer to Bomze et al. (1999).

In view of the NP-hardness of the MAXIMUM INDEPENDENT SET problem, one can distinguish three main groups of algorithms to solve this problem:

(1) non-polynomial-time algorithms,
(2) polynomial-time algorithms providing approximate solutions,
(3) polynomial-time algorithms that solve the problem exactly for graphs belonging to special classes.

Non-polynomial-time algorithms are generally impractical even for graphs of moderate size. It has been recently shown in Håstad (1999) that non-exact polynomial-time algorithms cannot approximate the size of a maximum independent set within a factor of $n^{1-\epsilon}$, which is viewed

as a negative result. The objective of the present paper is the algorithms of the third group.

As we mentioned already, the MAXIMUM INDEPENDENT SET problem has a polynomial-time solution in the classes of bipartite graphs and line graphs. In both cases, MIS reduces to the MAXIMUM MATCHING problem. For line graphs, this reduction has been described above. For bipartite graphs, the reduction is based on the fundamental theorem of König stating that the independence number of a bipartite graph G added to the number of edges in a maximum matching of G amounts to the number of vertices of G.

A polynomial-time solution to the MAXIMUM MATCHING problem is based on Berge's idea (Berge, 1957) that a matching in a graph is maximum if and only if there are no augmenting chains with respect to the matching. The first polynomial-time algorithm to find augmenting chains has been proposed by Edmonds in 1965. The idea of augmenting chains is a special case of a general approach to solve the MAXIMUM INDEPENDENT SET problem by means of augmenting graphs. The next section presents this approach in its general form.

3. Method of augmenting graphs

Let S be an independent set in a graph G. We shall call the vertices of S *black* and the remaining vertices of the graph *white*. A bipartite graph $H = (W, B, E)$ with the vertex set $W \cup B$ and the edge set E is called *augmenting* for S (and we say that S *admits* the augmenting graph) if

(1) $B \subseteq S$, $W \subseteq V(G) - S$,
(2) $N(W) \cap (S - B) = \emptyset$,
(3) $|W| > |B|$.

Clearly if $H = (W, B, E)$ is an augmenting graph for S, then S is not a maximum independent set in G, since the set $S' = (S - B) \cup W$ is independent and $|S'| > |S|$. We shall say that the set S' is obtained from S by H-augmentation and call the number $|W| - |B| = |S'| - |S|$ the *increment* of H.

Conversely, if S is not a maximum independent set, and S' is an independent set such that $|S'| > |S|$, then the subgraph of G induced by the set $(S - S') \cup (S' - S)$ is augmenting for S. Therefore, we have proved the following key result.

THEOREM OF AUGMENTING GRAPHS *An independent set S in a graph G is maximum if and only if there are no augmenting graphs for S.*

This theorem suggests the following general approach to find a maximum independent set in a graph G: begin with any independent set S in G and as long as S admits an augmenting graph H, apply H-augmentation to S. Clearly the problem of finding augmenting graphs is generally NP-hard, as the maximum independent set problem is NP-hard. However, this approach has proven to be a useful tool to develop approximate solutions to the problem (Halldórsson, 1995), to compute bounds on the independence number (Denley, 1994), and to solve the problem in polynomial time for graphs in special classes. For a polynomial-time solution, one has to

(a) find a complete list of augmenting graphs in the class under consideration,
(b) develop polynomial-time algorithms for detecting all augmenting graphs in the class.

Section 4 of the present paper analyzes problem (a) and Section 5 problem (b) for various graph classes. Analysis of problem (a) is based on characterization of bipartite graphs in classes under consideration. Clearly not every bipartite graph can be augmenting. For instance, a bipartite cycle is never augmenting, since the condition (3) fails for it. Moreover, without loss of generality we may exclude from our consideration those augmenting graphs, which are not minimal. An augmenting graph H for a set S is called *minimal* if no proper induced subgraph of H is augmenting for S. Some bipartite graphs that may be augmenting are never minimal augmenting. To give an example, consider a claw $K_{1,3}$. If it is augmenting for some independent set S, then its subgraph obtained by deleting a vertex of degree 1 also is an augmenting graph for S. The following lemma describes several necessary conditions for an augmenting graph to be minimal.

LEMMA 4.1 *If $H = (B, W, E)$ is a minimal augmenting graph for an independent set S, then*

(i) *H is connected;*
(ii) $|B| = |W| - 1$;
(iii) *for every subset $A \subseteq B$, $|A| < |N_W(A)|$.*

Proof. Conditions (i) and (ii) are obvious. To show (iii), assume $|A| \geq |N_W(A)|$ for some subset A of B. Then the vertices in $(B - A) \cup \big(W - N_W(A)\big)$ induce a proper subgraph of H which is augmenting too. □

Another notion, which is helpful in some cases, is the notion of maximum augmenting graph. An augmenting graph H for an independent set S is called *maximum* if the increment of any other augmenting graph

for S does not exceed the increment of H. The importance of this notion is due to the following lemma.

LEMMA 4.2 *Let S be an independent set in a graph G, and H an augmenting graph for S. Then the independent set obtained by H-augmentation is maximum in G if and only if H is a maximum augmenting graph for S.*

To conclude this section, let us mention that an idea similar to augmenting graphs can be applied to the INDEPENDENT DOMINATING SET problem. In this case, given a maximal independent set S in a graph G, we want to find a smaller maximal independent set. So, we define a bipartite graph $H = (B, W, E)$ to be a *decreasing* graph for S if

(1') $B \subseteq S$, $W \subseteq V(G) - S$,
(2') $N(W) \cap (S - B) = \emptyset$,
(3') $|W| < |B|$,
(4') $(S - B) \cup W$ is a maximal independent set in G.

The additional condition (4') makes the problem of finding decreasing graphs harder than that of finding augmenting graphs, though some results exploiting the idea of decreasing graphs are available in the literature (Boliac and Lozin, 2003a).

4. Characterization of augmenting graphs

The basis for characterization of augmenting graphs in a certain class is the description of bipartite graphs in that class. For a bipartite graph $G = (V_1, V_2, E)$, we shall denote by $\widetilde{G}$ the bipartite complement of G, i.e. $\widetilde{G} = (V_1, V_2, (V_1 \times V_2) - E)$. We call a bipartite graph G *prime* if any two distinct vertices of G have different neighborhoods.

Claw-free ($S_{1,1,1}$-free) graphs. In the class of *claw*-free graphs, no bipartite graph has a vertex of degree more than 2, since otherwise a claw arises. Therefore, every connected *claw*-free bipartite graph is either an even cycle or a chain. Cycles of even length and chains of odd length cannot be augmenting graphs, since they have equal number of black and white vertices. Thus, every minimal *claw*-free augmenting graph is a chain of even length.

P_4-free ($S_{0,1,2}$-free) graphs. It is a simple exercise to show that every connected P_4-free bipartite graph is complete bipartite. Therefore, every minimal augmenting graph in this class is of the from $K_{n,n+1}$ for some value of n.

Fork-free ($S_{1,1,2}$-free) graphs. Connected bipartite fork-free graphs G have been characterized in Alekseev (1999) as follows: either $\Delta(G) \leq 2$ or $\Delta(\widetilde{G}) \leq 1$. A bipartite graph G with $\Delta(\widetilde{G}) \leq 1$ has been called a *complex*. Thus, every minimal fork-free augmenting graph is either a chain of even length or a complex.

P_5-free ($S_{0,2,2}$-free) graphs. It has been shown independently by many researchers (and can be easily verified) that every connected P_5-free bipartite graph is $2K_2$-free, where a $2K_2$ is the disjoint union of two copies of K_2. The class of $2K_2$-free bipartite graphs was introduced in the literature under various names such as *chain graphs* (Yannakakis, 1981) or *difference graphs* (Hammeret al., 1990). The fundamental property of a chain graph is that the vertices in each part can be ordered under inclusion of their neighborhoods. Unfortunately, this nice property does not help in finding maximum independent sets in P_5-free graphs in polynomial time (the complexity status of the problem in this class is still an open question). So, augmenting graphs in many subclasses of P_5-free bipartite graphs have been characterized, among which we distinguish (P_5, banner)-free and $(P_5, K_{3,3} - e)$-free graphs.

It has been shown in Lozin (2000a) that in the class of (P_5, banner)-free graphs every minimal augmenting graph is complete bipartite. Here we prove a more general proposition, which is based on the following two lemmas.

LEMMA 4.3 *Let H be a connected bipartite banner-free graph. If H contains a C_4, then it is complete bipartite.*

Proof. Denote by H' a maximal induced complete bipartite subgraph of H containing the C_4. Let x be a vertex outside H' adjacent to a vertex in the subgraph. Then x must be adjacent to all the vertices in the opposite part of the subgraph, since otherwise H contains an induced banner. But then H' is not maximal. This contradiction proves that $H = H'$ is complete bipartite. □

LEMMA 4.4 *No minimal $(S_{1,2,2}, C_4)$-free augmenting graph H contains a $K_{1,3}$ as an induced subgraph.*

Proof. Let vertices a, b, c, d induce a $K_{1,3}$ in H with a being the center. Assume first that a is the only neighbor of b and c in the graph H. Then H is not minimal. Indeed, in case that a is a white vertex, this follows from Lemma 4.1(iii). If a is a black vertex, then $H[a, b, c]$ is a smaller augmenting graph. Now suppose without loss of generality that b has a neighbor $e \neq a$, and c has a neighbor $f \neq a$ in the graph H. Since

H is C_4-free, $e \neq f$ and $ec, ed, fb, fd \notin E(H)$. But now the vertices a, b, c, d, e, f induce an $S_{1,2,2}$. □

Combining these two lemmas with the characterization of *claw*-free minimal augmenting graphs, we obtain the following conclusion.

THEOREM 4.1 *Every minimal* $(S_{1,2,2}, \text{banner})$*-free augmenting graph is either complete bipartite or a chain of odd length. In particular, every minimal* (P_5, banner)*-free augmenting graph is complete bipartite.*

The class of $(P_5, K_{3,3} - e)$-free graphs generalizes (P_5, banner)-free graphs. The minimal augmenting $(P_5, K_{3,3} - e)$-free graphs have been characterized in Gerber et al. (2004b) as follows.

THEOREM 4.2 *Every minimal augmenting* $(P_5, K_{3,3} - e)$*-free graph is either complete bipartite or a graph obtained from a complete bipartite graph* $K_{n,n}$ *by adding a single vertex with exactly one neighbor in the opposite part.*

$S_{1,2,2}$-free graphs. The class of $S_{1,2,2}$-free bipartite graphs has been provided in Lozin (2000b) with the following characterization.

THEOREM 4.3 *Every prime* $S_{1,2,2}$*-free bipartite graph is either* $K_{1,3}$*-free or* $\widetilde{P}_5$*-free.*

The class of $S_{1,2,2}$-free graphs is clearly an extension of P_5-free graphs. Since the complexity of the problem is still open even for P_5-free graphs, it is worth characterizing subclasses of $S_{1,2,2}$-free graphs that do not contain the entire class of P_5-free graphs. One of such characterizations is given in Theorem 4.1. Now we extend this theorem to $(S_{1,2,2}, A)$-free bipartite graphs, where A is the graph obtained from a P_6 by adding an edge between two vertices of degree 2 at distance 3.

THEOREM 4.4 *A prime connected* $(S_{1,2,2}, A)$*-free bipartite graph* G *is* $S_{1,1,2}$*-free.*

Proof. By contradiction, assume G contains an $S_{1,1,2}$ with vertices a, b, c, d, e and edges ab, bc, cd, ce. Since G is prime, there must be a vertex f adjacent to e but not to d. Since G is bipartite, f is not adjacent to a and c. But then the vertices a, b, c, d, e, f induce either an $S_{1,2,2}$, if f is not adjacent to b, or an A, otherwise. □

$S_{2,2,2}$-free graphs. This is a rich class containing all the previously mentioned classes. Moreover, the bipartite graphs in this class include

also all bipartite permutation graphs (Spinrad et al., 1987) and all biconvex graphs (Abbas and Stewart, 2000). So, again we restrict ourselves to a special subclass of $S_{2,2,2}$-free bipartite graphs that does not entirely contain the class of P_5-free graphs. To this end, we recall that a *complex* is a bipartite graph every vertex of which has at most one non-neighbor in the opposite part. A *caterpillar* is a tree that becomes a path by removing the pendant vertices. A *circular caterpillar* G is a graph that becomes a cycle C_k by removing the pendant vertices. We call G a long circular caterpillar if $k > 4$. The following theorem has been proven in Boliac and Lozin (2001).

THEOREM 4.5 *A prime connected $(S_{2,2,2}, A)$-free bipartite graph is either a caterpillar or a long circular caterpillar or a complex.*

P_k-free graphs with $k \geq 6$ and $S_{1,2,j}$-free graphs with $j \geq 3$. It has been shown in Mosca (1999) that (P_6, C_4)-free augmenting graphs are simple augmenting trees (i.e., graphs $M_{r,0}$ with $r \geq 0$ in Figure 4.2). This characterization as well as the characterization of (P_5, banner)-free augmenting graphs has been extended in Alekseev and Lozin (2004) in two different ways.

THEOREM 4.6 *In the class of (P_7, banner)-free graphs every minimal augmenting graph is either complete bipartite or a simple augmenting tree or an augmenting plant (i.e., a graph $L^2_{r,0}$ with $r \geq 2$ in Figure 4.2).*

In the class of $(S_{1,2,3}, \text{banner})$-free graphs every minimal augmenting graph is either a chain or a complete bipartite graph or a simple augmenting tree or an augmenting plant.

Finally, in Gerber et al. (2004a), these results have been generalized in the following way.

THEOREM 4.7 *A minimal augmenting (P_8, banner)-free graph is one of the following graphs (see Figure 4.2 for definitions of the graphs):*

- *a complete bipartite graph $K_{r,r+1}$ with $r \geq 0$,*
- *a $L^1_{r,s}$ or a $L^2_{r,s}$ with $r \geq 2$ and $s \geq 0$,*
- *a $M_{r,s}$ with $r \geq 1$ and $r \geq s \geq 0$,*
- *a N_s with $s \geq 0$.*
- *one of the graphs $F_2, \ldots, F_5$.*

A minimal augmenting $(S_{1,2,4}, \text{banner})$-free graph is one of the following graphs:

- *a complete bipartite graph $K_{r,r+1}$ with $r \geq 0$,*
- *a path P_k with k odd ≥ 7,*
- *a $L^2_{r,0}$ with $r \geq 2$,*

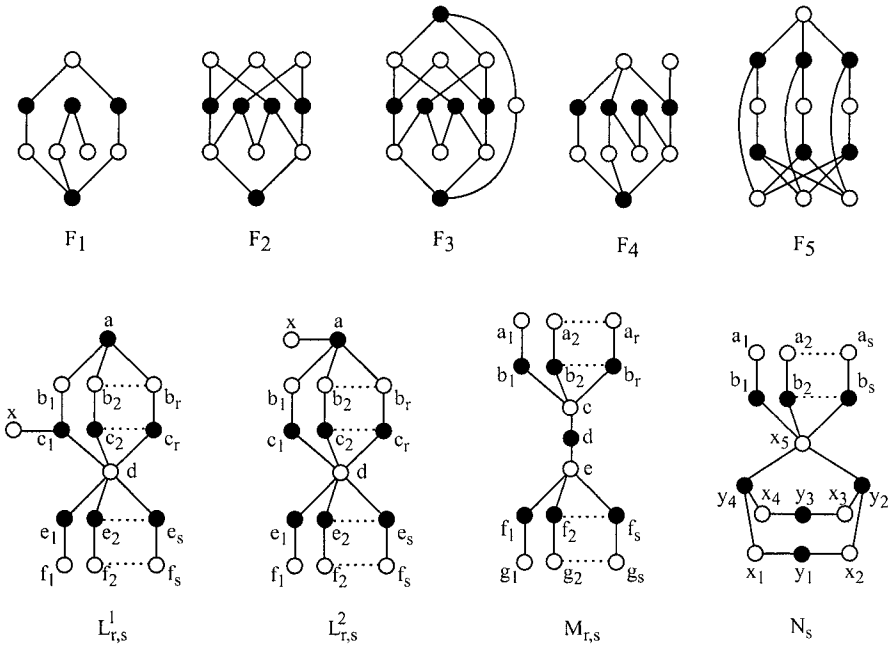


Figure 4.2. Minimal augmenting graphs

- *a* $M_{r,0}$ *with* $r \geq 1$,
- *one of the graphs* $F_1, \ldots, F_5, L^1_{3,0}, L^1_{2,1}, N_0, N_1$.

Notice that the set of (P_8, banner)-free augmenting graphs can be partitioned into two general groups. The first one contains infinitely many graphs of high vertex degree and "regular" structure, while the second group consists of finitely many graphs of bounded vertex degree. It is not a surprise. With simple arguments it has been shown in Lozin and Rautenbach (2003) that for any k and n, there are finitely many connected bipartite $(P_k, K_{1,n})$-free graphs. This observation has been generalized in Gerber et al. (2003) by showing that in the class of $(S_{1,1,j}, K_{1,n})$-free graphs there are finitely many connected bipartite graphs of maximum vertex degree more than 2. Now we extend this result as follows.

THEOREM 4.8 *For any three integers* j, k *and* n, *the class of* $(S_{1,2,j}, \text{Banner}_k, K_{1,n})$*-free graphs contains finitely many minimal augmenting graphs different from chains.*

Proof. To prove the theorem, consider a minimal augmenting graph H in this class that contains a $K_{1,3}$ with the center a_0. Denote by A_i the subset of vertices of H of distance i from a_0. In particular, $A_0 = \{a_0\}$.

Let m be an integer greater than $\max(k+2, j+2)$. Consider a vertex $a_m \in A_m$, and let $P = (a_0, a_1, \ldots, a_m)$ with $a_i \in A_i$ $(i = 1, \ldots, m)$ denote a shortest path connecting a_0 to a_m. Then a_2 has no other neighbor in A_1, except for a_1, since otherwise the vertices of P together with this neighbor would induce a Banner_{m-2}.

Since a_0 is the center of a $K_{1,3}$, we may consider two vertices in A_1 different from a_1, say b and c. Assume b has a neighbor d in A_2. Then d is not adjacent to a_3, since otherwise a_3 is the vertex of degree 3 in an induced $S_{1,2,m-3}$ (if $da_1 \notin E(H)$) or in an induced Banner_{m-3} (if $da_1 \in E(H)$). Consequently, d is not adjacent to a_1, since otherwise a_1 is the vertex of degree 3 in an induced Banner_{m-1}. But now a_0 is the vertex of degree 3 either in an induced $S_{1,2,m}$ (if $cd \notin E(H)$) or in an induced Banner_m (if $cd \in E(H)$). Therefore, vertices b and c have degree 1 in H, but then H is not minimal. This contradiction shows that $A_i = \emptyset$ for each $i > \max(k+2, j+2)$. Since H is $K_{1,n}$-free, there is a constant bounding the number of vertices in each A_i for $i \leq \max(k+2, j+2)$. Therefore, only finitely many minimal augmenting graphs in the class under consideration contain a $K_{1,3}$. □

We conclude this section with the characterization of prime $S_{1,2,3}$-free bipartite graphs found in Lozin (2002b), which may become a source for many other results on the maximum independent set problem in subclasses of $S_{1,2,3}$-free graphs.

THEOREM 4.9 *A prime $S_{1,2,3}$-free bipartite graph G is either disconnected or $\widetilde{G}$ is disconnected or G can be partitioned into an independent set and a complete bipartite graph or G is $K_{1,3}$-free or $\widetilde{G}$ is $K_{1,3}$-free.*

5. Finding augmenting graphs

5.1 Augmenting chains

Let S be an independent set in the line graph $L(G)$ of a graph G, and let M be the corresponding matching in G. The problem of finding an augmenting chain for S in $L(G)$ is equivalent to the problem of finding an augmenting chain with respect to M in G, and this problem can be solved by means of Edmonds' polynomial-time algorithm (Edmonds, 1965). In 1980, Minty and Sbihi have independently shown how to extend this result to the class of *claw*-free graphs that strictly contains the class of line graphs. More precisely, both authors have shown that the problem of finding augmenting chains in *claw*-free graphs is polynomially reducible to the problem of finding an augmenting chain with respect to a matching. This result has recently been generalized in two different ways:

(1) Gerber et al. (2003) have proved that by slightly modifying Minty's algorithm, one can determine augmenting chains in the class of $S_{1,2,3}$-free graphs;
(2) it is proved in Hertz et al. (2003) that the problem of finding augmenting chains in $(S_{1,2,i}, \text{banner})$-free graphs, for a fixed $i \geq 1$, is polynomially reducible to the problem of finding augmenting chains in *claw*-free graphs.

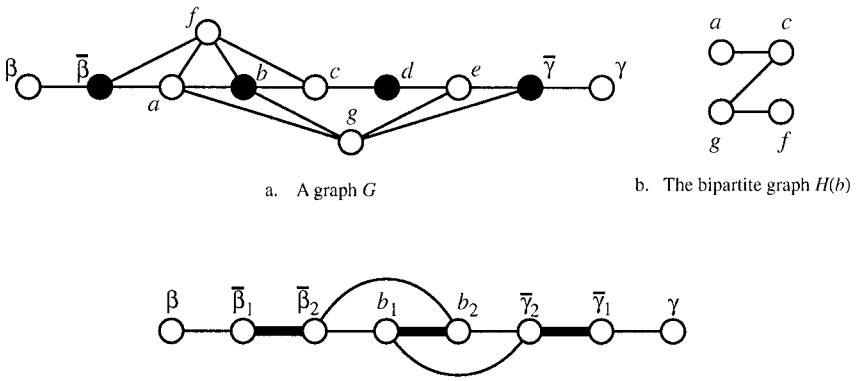
Both classes of $(S_{1,2,i}, \text{banner})$-free graphs ($i \geq 1$) and $S_{1,2,3}$-free graphs strictly contain the class of *claw*-free graphs. In this section, we first describe Minty's algorithm for finding augmenting chains in *claw*-free graphs. We then describe its extensions to the classes of $S_{1,2,3}$-free and $(S_{1,2,i}, \text{banner})$-free graphs.

5.1.1 Augmenting chains in *claw*-free graphs. Notice first that an augmenting chain for a maximal independent set S necessarily connects two non-adjacent white vertices β and γ, each of which has exactly one black neighbor, respectively, $\overline{\beta}$ and $\overline{\gamma}$. If $\overline{\beta} = \overline{\gamma}$, then the chain $(\beta, \overline{\beta}, \gamma)$ is augmenting for S. We can therefore assume that $\overline{\beta} \neq \overline{\gamma}$. We may also assume that any white vertex different from β and γ is not adjacent to β and γ, and has exactly two black neighbors (the vertices not satisfying the assumption are out of interest, since they cannot occur in any augmenting chain connecting β to γ).

Two white vertices having the same black neighbors are called *similar*. The similarity is an equivalence relation, and an augmenting chain clearly contains at most one vertex in each class of similarity. The similarity classes in the neighborhood of a black vertex b are called the *wings* of b. Let b be a black vertex different from $\overline{\beta}$ and $\overline{\gamma}$: if b has more than two wings, then b is defined as *regular*, otherwise it is *irregular*. In what follows, R denotes the set of black vertices that are either regular or equal to $\overline{\beta}$ or $\overline{\gamma}$. For illustration, the graph G depicted in Figure 4.3.a has one regular black vertex (vertex b), one irregular black vertex (vertex d), and R is equal to $\{b, \overline{\beta}, \overline{\gamma}\}$.

DEFINITIONS An *alternating chain* is a sequence $(x_0, x_1, \ldots, x_k)$ of distinct vertices in which the vertices are alternately white and black. Vertices x_0 and x_k are called the *termini* of the chain. If x_0 and x_k are both black (respectively white) vertices, then the sequence is called a black (respectively white) alternating chain.

Let b_1 and b_2 be two distinct black vertices in R. A black alternating chain with termini b_1 and b_2 is called an IBAP (for *irregular black alternating path*) if it is chordless and if all black vertices of the chain, except b_1 and b_2, are irregular. An IWAP (for *irregular white alternating*



a. A graph G

b. The bipartite graph $H(b)$

c. The corresponding Edmond's graph

Figure 4.3. Illustration of Minty's algorithm.

path) is a white alternating chain obtained by removing the termini of an IBAP.

For illustration, the graph G depicted in Figure 4.3.a has four IWAPs: (a), (f), (g) and (c, d, e). With this terminology, one can now represent an augmenting chain as a sequence $\big(I_0 = (\beta), b_0 = \overline{\beta}, I_1, b_1, I_2, \ldots, b_{k-1}, I_{k-1}, b_k = \overline{\gamma}, I_k = (\gamma)\big)$ such that

(a) the b_i $(0 < i < k)$ are distinct black regular vertices,
(b) the I_i $(0 < i < k)$ are pairwise mutually disjoint IWAPs,
(c) each b_i $(0 \leq i \leq k)$ is adjacent to the final terminus of I_i and to the initial one of I_{i+1},
(d) the white vertices in $I_1 \cup \cdots \cup I_{k-1}$ are pairwise non-adjacent.

Minty has proved that the neighborhood of each black vertex b can be decomposed into at most two subsets $N_1(b)$ and $N_2(b)$, called *node classes*, in such a way that no two vertices in the same node class can occur in the same augmenting chain for S. For vertices $\overline{\beta}$ and $\overline{\gamma}$, such a decomposition is obvious: one of the node classes contains the vertex β (respectively γ) and the other class includes all the remaining vertices in the neighborhood of $\overline{\beta}$ (respectively $\overline{\gamma}$). We assume that $N_1(\overline{\beta}) = \{\beta\}$ and $N_1(\overline{\gamma}) = \{\gamma\}$. For an irregular black vertex b, the decomposition also is trivial: the node classes correspond to the wings of b.

Now let b be a regular black vertex. Two white neighbors of b can occur in the same augmenting chain for S only if they are non-similar and non-adjacent. Define an auxiliary graph $H(b)$ as follows:

– the vertex set of $H(b)$ is $N(b)$

- two vertices u and v of $H(b)$ are linked by an edge if and only if u and v are non-similar non-adjacent vertices in G.

Minty has proved that $H(b)$ is bipartite. The two node classes $N_1(b)$ and $N_2(b)$ of a regular black vertex b therefore correspond to the two parts of the bipartite graph $H(b)$. For illustration, the bipartite graph $H(b)$ associated with b in the graph of Figure 4.3.b defines the partition of $N(b)$ into two node classes $N_1(b) = \{a, g\}$ and $N_2(b) = \{c, f\}$.

We now show how to determine the pairs (u, v) of vertices such that there exists an IWAP with termini u and v. Notice first that u and v must have a black neighbor in R. So let b_0 be a black vertex in R, and let W_1 be one of its wings ($W_1 = N_2(\overline{\beta})$ if $b_0 = \overline{\beta}$, and $W_1 = N_2(\overline{\gamma})$ if $b_0 = \overline{\gamma}$). The following algorithm determines the set P of pairs (u, v) such that u belongs to W_1 and is a terminus of an IWAP:

(1) Set $k := 1$;
(2) Let b_k denote the second black neighbor of the vertices in W_k; If b_k has two wings then go to Step 3. If b_k is regular and different from b_0 then go to Step 4. Otherwise STOP: P is empty;
(3) Let W_{k+1} denote the second wing of b_k. Set $k := k + 1$ and go to Step 2;
(4) Construct an auxiliary graph with vertex set $W_1 \cup \cdots \cup W_k$ and link two vertices by an edge if and only if they are non-adjacent in G and belong to two consecutive sets W_i and W_{i+1}. Orient all edges from W_i to W_{i+1};
(5) Determine the set P of pairs (u, v) such that $u \in W_1, v \in W_k$ and there exists a path from u to v in the auxiliary graph.

The last important concept proposed by Minty is the *Edmonds' Graph* which is constructed as follows:

- For each black vertex $b \in R$ do the following: create two vertices b_1 and b_2, link them by a black edge, and identify b_1 and b_2 with the two node classes $N_1(b)$ and $N_2(b)$ of b. In particular, $\overline{\beta}_1$ represents $N_1(\overline{\beta}) = \{\beta\}$ and $\overline{\gamma}_1$ represents $N_1(\overline{\gamma}) = \{\gamma\}$;
- Create two vertices β and γ, and link β to $\overline{\beta}_1$ and γ to $\overline{\gamma}_1$ by a white edge.
- Link b_i ($i = 1$ or 2) to b'_j ($j = 1$ or 2) with a white edge if there are two white vertices u and v in G such that $u \in N_i(b)$, $v \in N_j(b')$, and there exists an IWAP with termini u and v. Identify each such white edge with a corresponding IWAP.

The black edges define a matching M in the Edmonds' graph. If M is not maximum, then there exists an augmenting chain of edges $(e_0, \ldots, e_{2k})$ such that the even indexed edges are white, the odd-indexed edges are black, e_0 is the edge linking β to $\overline{\beta}_1$, and e_{2k} is the edge

linking γ to $\overline{\gamma}_1$. Such an augmenting chain of edges in the Edmonds' graph corresponds to an alternating chain C in G. Indeed, notice first that each white edge e_i with $2 \leq i \leq 2k-2$ corresponds to an IWAP whose termini will be denoted w_{i-1} and w_i. Also, each black edge e_i with $1 \leq i \leq 2k-1$ corresponds to a black vertex b_i. The alternating chain C is obtained as follows:

- replace e_0 by β, e_{2k} by γ, and each white edge $e_i (2 \leq i \leq 2k-2)$ by an IWAP with termini w_{i-1} and w_i;
- replace each black edge $e_i (1 \leq i \leq 2k-1)$ by the vertex b_i.

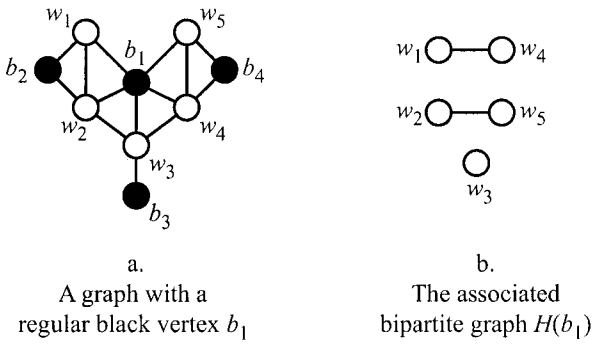
Minty has proved that C is chordless, and is therefore an augmenting chain for S in G. He has also proved that an augmenting chain for S in G corresponds to an augmenting chain with respect to M in the Edmonds' graph. Hence, determining whether there exists an augmenting chain for S in G, with termini β and γ, is equivalent to determining whether there exists an augmenting chain with respect to M in the Edmonds' graph.

For illustration, the Edmonds' graph associated with the graph in Figure 4.3.a is represented in Figure 4.3.c with bold lines for the black edges and regular lines for the white edges. The four IWAPs (a), (f), (g) and (c, d, e) correspond to the four white edges $\overline{\beta}_2 b_1$, $\overline{\beta}_2 b_2$, $b_1 \overline{\gamma}_2$ and $b_2 \overline{\gamma}_2$, respectively. The Edmonds' graph contains two augmenting chains: $(\beta, \overline{\beta}_1, \overline{\beta}_2, b_1, b_2, \overline{\gamma}_2, \overline{\gamma}_1, \gamma)$ and $(\beta, \overline{\beta}_1, \overline{\beta}_2, b_2, b_1, \overline{\gamma}_2, \overline{\gamma}_1, \gamma)$ which correspond to the augmenting chains $(\beta, \overline{\beta}, a, b, c, d, e, \overline{\gamma}, \gamma)$ and $(\beta, \overline{\beta}, f, b, g, \overline{\gamma}, \gamma)$ for S in G.

In summary, given two non-adjacent white vertices β and γ, each of which has exactly one black neighbor, the following algorithm either builds an augmenting chain for S with termini β and γ, or concludes that no such chain exists.

Minty's algorithm for finding an augmenting chain for S with termini β and γ in a *claw*-free graph.

(1) Partition the neighborhood of each regular black vertex b into two node classes $N_1(b)$ and $N_2(b)$ by constructing the bipartite graph $H(b)$ in which two white neighbors of b are linked by an edge if and only if they are non-adjacent and non-similar;
(2) Determine the set of pairs (u, v) of (not necessarily distinct) white vertices such that there exists an IWAP with termini u and v;
(3) Construct the Edmonds' graph and let M denote the set of black edges;
(4) If the Edmonds' graph contains an augmenting chain of edges with respect to M, then it corresponds to an augmenting chain for S in



a. A graph with a regular black vertex b_1

b. The associated bipartite graph $H(b_1)$

Figure 4.4. Bipartite graph associated with a regular black vertex.

G with termini β and γ; otherwise, there are no augmenting chains for S with termini β and γ.

5.1.2 Augmenting chains in $S_{1,2,3}$-free graphs. Gerber et al. (2003) have shown that Minty's algorithm can be adapted in order to detect augmenting chains in the class of $S_{1,2,3}$-free graphs that strictly contains the class of *claw*-free graphs. The algorithm described in Gerber et al. (2003) differs from Minty's algorithm in only two points. The first difference occurs in the definition of $H(b)$ where an additional condition is imposed for creating an edge in $H(b)$. More precisely, let us first define *special* pairs of vertices.

DEFINITION A pair (u, v) of vertices is *special* if u and v have a common black regular neighbor b, and if there is a vertex $w \in N(b)$ which is similar neither to u nor to v and such that either both of uw and vw or none of them is an edge in G.

It is shown in Gerber et al. (2003) that if (u, v) is a special pair of non-adjacent non-similar vertices in a $S_{1,2,3}$-free graph, then u and v cannot occur in a same augmenting chain. For a regular black vertex b, the graph $H(b)$ is defined as follows: the vertex set of $H(b)$ is $N(b)$, and two vertices u and v in $H(b)$ are linked by an edge if and only if (u, v) is a pair of non-special non-similar non-adjacent vertices in G. It is proved in Gerber et al. (2003) that $H(b)$ is bipartite. For illustration, the bipartite graph $H(b_1)$ associated with the regular black vertex b_1 in Figure 4.4.a is represented in Figure 4.4.b.

An isolated vertex in $H(b)$ cannot belong to an augmenting chain. Hence, an IWAP in an augmenting chain necessarily connects two white

vertices that are not isolated in the respective bipartite graphs associated with their black neighbors in R. This motivates the following definition.

DEFINITION Let $(b_1, w_1, \ldots, w_{k-1}, b_k)$ be an IBAP. The IWAP obtained by removing b_1 and b_k is *interesting* if w_1 and w_{k-1} are non-isolated vertices in $H(b_1)$ and $H(b_k)$, respectively.

Let W denote the set of white vertices w which have a black neighbor $b \in R$ such that w is an isolated vertex in $H(b)$. The set of pairs (u, v) such that there is an interesting IWAP with termini u and v can be determined in polynomial time by using the algorithm of the previous section that generates all IWAPs, and by removing a pair (u, v) if u or/and v belongs to W.

The Edmonds' graph is then constructed as in Minty's algorithm, except that white edges in the Edmonds' graph correspond to interesting IWAPs.

Now let S be an independent set in a $S_{1,2,3}$-free graph G, let β and γ be two non-adjacent white vertices, each of which has exactly one black neighbor, and let M denote the set of black edges in the corresponding Edmond's graph. It is proved in Gerber et al. (2003) that determining whether there exists an augmenting chain for S with termini β and γ is equivalent to determining whether there exists an augmenting chain with respect to M in the Edmonds' graph. In summary, the algorithm for finding augmenting chains in $S_{1,2,3}$-free graphs works as follows.

Algorithm for finding an augmenting chain for S with termini β and γ in a $S_{1,2,3}$- free graph.

(1) Partition the neighborhood of each regular black vertex b into two node classes $N_1(b)$ and $N_2(b)$ by constructing the bipartite graph $H(b)$ in which two white neighbors u and v of b are linked by an edge if and only if (u, v) is a pair of non-special non-adjacent non-similar vertices;
(2) Determine the set of pairs (u, v) of (not necessarily distinct) white vertices such that there exists an interesting IWAP with termini u and v;
(3) Construct the Edmond's graph and let M denote the set of black edges;
(4) If the Edmond's graph contains an augmenting chain of edges with respect to M, then it corresponds to an augmenting chain in G with termini β and γ; otherwise, there are no augmenting chains with termini β and γ.

The above algorithm is very similar to Minty's algorithm. It only differs in step 1 where an additional condition is imposed for introduc-

ing an edge in $H(b)$, and in step 2 where only interesting IWAPs are considered.

5.1.3 Augmenting chains in ($S_{1,2,i}$, banner)-free graphs.

Let $G = (V, E)$ be a ($S_{1,2,i}$, banner)-free graph, where i is any fixed strictly positive integer, and let S be a maximal independent set in G. An augmenting chain for S is of the form $P = (x_0, x_1, x_2, \ldots, x_{k-1}, x_k)$ (k is even) where the even-indexed vertices of P are white, and the odd-indexed vertices are black.

DEFINITION Let (x_0, x_k) be a pair of white non-adjacent vertices, each of which has exactly one black neighbor. A pair (L, R) of disjoint chordless alternating chains $L = (x_0, x_1, x_2)$ and $R = (x_{k-m}, x_{k-m+1}, \ldots, x_{k-1}, x_k)$ is said *candidate* for (x_0, x_k) if

- no vertex of L is adjacent to a vertex of R,
- each vertex x_j is white if and only if j is even, and
- $m = 2\lfloor \frac{i}{2} \rfloor$.

Augmenting chains with at most $i + 3$ vertices can be detected in polynomial time by inspecting all subsets of black vertices of cardinality at most $\frac{i+4}{2}$. It is proved in Hertz et al. (2003) that larger augmenting chains can be detected by applying the following algorithm for each pair (x_0, x_k) of white non-adjacent vertices, each of which has exactly one black neighbor.

(a) Remove from G all white vertices adjacent to x_0 or x_k as well as all white vertices different from x_0 and x_k which have 0, 1 or more than 3 black neighbors.

(b) Find all candidate pairs (L, R) of alternating chains for (x_0, x_k), and for each such pair, do steps (b.1) through (b.4):

  (b.1) remove all white vertices that have a neighbor in L or in R,

  (b.2) remove the vertices of L and R except for x_2 and x_{k-m},

  (b.3) remove all the vertices that are the center of a *claw* in the remaining graph,

  (b.4) in the resulting *claw*-free graph, determine whether there exists an augmenting chain for S with termini x_2 and x_{k-m}.

Step (b.4) can be performed by using the algorithm described in Section 5.1.1.

5.2 Complete bipartite augmenting graphs

In the present section we describe an approach to finding augmenting graphs every connected component of which is complete bipartite, i.e.,

P_4-free augmenting graphs. This approach has been applied first to fork-free graphs (Alekseev, 1999) and (P_5, Banner)-free graphs (Lozin, 2000a). Then it has been extended to the entire class of Banner-free graphs (Alekseev and Lozin, 2004) and to the entire class of P_5-free graphs (Boliac and Lozin, 2003b). We now generalize this approach to the class of Banner$_2$-free graphs that contains all the above mentioned classes.

Throughout the section G stands for a Banner$_2$-free graph and S for a maximal independent set in G. Let us call two white vertices x and y with $N_S(x) = N_S(y)$ *similar*. First, we partition the set of white vertices into similarity classes. Next, each class of similarity C is partitioned into co-components, i.e., subsets each of which forms a connected component in the complement to $G[C]$. Every co-component of a similarity class will be called a *node class*. Two node classes are *non-similar* if their vertices belong to different similarity classes.

Without loss of generality we shall assume that for any node class Q_i the following conditions hold:

$$|N_S(Q_i)| \geq 3, \tag{4.1}$$

each vertex in Q_i has a non-neighbor in the same node class. (4.2)

To meet condition (4.1), we first find augmenting graphs of the form $K_{1,2}$ or $K_{2,3}$. If S does not admit such augmenting graphs, we may delete node classes non-satisfying (4.1). Under condition (4.1), any vertex that has no non-neighbor in its own node class is of no interest to us. So, vertices non-satisfying condition (4.2) can be deleted.

Assuming (4.1) and (4.2), we prove the following lemma.

LEMMA 4.5 *Let Q_1 and Q_2 be two non-similar node classes. If there is a pair of non-adjacent vertices $x \in Q_1$ and $y \in Q_2$, then one of the following statements holds:*

(a) $\max(|N_S(Q_1) - N_S(Q_2)|, |N_S(Q_2) - N_S(Q_1)|) \leq 1$,
(b) $N_S(Q_1) \subseteq N_S(Q_2)$ *or* $N_S(Q_2) \subseteq N_S(Q_1)$,
(c) $N_S(Q_1) \cap N_S(Q_2) = \emptyset$ *and no vertex in* Q_1 *is adjacent to a vertex in* Q_2.

Proof. Assume first that the intersection $N_S(Q_1) \cap N_S(Q_2)$ contains a vertex a, and suppose (b) does not hold. Then we denote by b a vertex in $N_S(Q_1) - N_S(Q_2)$ and by c a vertex in $N_S(Q_2) - N_S(Q_1)$. We also assume by contradiction that $N_S(Q_2) - N_S(Q_1)$ contains a vertex $d \neq c$, and finally, according to the assumption (4.2), we let z be a vertex in Q_2 non-adjacent to y. If x is not adjacent to z, then the vertices a, b, c, x, y, z induce a Banner$_2$ in G. If x is adjacent to z, then a Banner$_2$ is induced by b, c, d, x, y, z.

Now we assume that $N_S(Q_1) \cap N_S(Q_2) = \emptyset$, and we denote by a and b two vertices in $N_S(Q_1)$ and by c and d two vertices in $N_S(Q_2)$. Suppose by contradiction that a vertex z in Q_1 is adjacent to a vertex w in Q_2. If x is not adjacent to w then one can find two vertices v_1 and v_2 on the path connecting z to x in $\overline{G[Q_1]}$ such that w is adjacent to v_1 but not to v_2. But then vertices a, b, c, v_1, v_2, w induce a Banner$_2$ in G, a contradiction. So we can assume that x is adjacent to w. But one can now find two vertices v_1 and v_2 on the path connecting w to y in $\overline{G[Q_2]}$ such that x is adjacent to v_1 but not to v_2. Hence, vertices b, c, d, x, v_1, v_2 induce a Banner$_2$ in G, a contradiction. □

Let us associate with G and S an auxiliary graph Γ as follows. The vertices of Γ are the node classes of G, and two vertices Q_i and Q_j are defined to be adjacent in Γ if and only if one of the following conditions holds:

- $\max\{|N_S(Q_i) - N_S(Q_j)|, |N_S(Q_j) - N_S(Q_i|\} \leq 1$,
- $N_S(Q_i) \subseteq N_S(Q_j)$ or $N_S(Q_j) \subseteq N_S(Q_i)$,
- each vertex of Q_i is adjacent to each vertex of Q_j in graph G.

In other words, due to Lemma 4.5, Q_i and Q_j are non-adjacent in Γ if and only if $N_S(Q_i) \cap N_S(Q_j) = \emptyset$ and no vertex in Q_i is adjacent to a vertex in Q_j. To each vertex Q_j of Γ we assign an integer number, the weight of the vertex, equal to $\alpha(G[Q_j]) - |N_S(Q_j)|$.

Consider now an independent set $Q = \{Q_1, \ldots, Q_p\}$ in the graph Γ. Let us associate with each vertex $Q_j \in Q$ a complete bipartite graph $H_j = (B_j, W_j, E_j)$ with $B_j = N_S(Q_j)$ and W_j being an independent set of maximum cardinality in $G[Q_j]$. By definition of the graph Γ, subsets $B_1, \ldots, B_p$ are pairwise disjoint and the union $\bigcup_{j=1}^{p} W_j$ is an independent set in G. Hence the union of graphs $H_1, \ldots, H_p$, denoted H_Q, is a P_4-free bipartite graph.

The increment of H_Q, equal to $\sum_{j=1}^{p}(|W_j| - |B_j|)$, coincides with the weight of Q, equal to $\sum_{j=1}^{p}(\alpha(G[Q_j]) - |N_S(Q_j)|)$. If the weight of Q is positive, then H_Q is an augmenting graph for S. Moreover, if Q is an independent set of maximum total weight in Γ, then the increment of H_Q is maximum over all P_4-free augmenting graphs for S. Indeed, if H is a P_4-free augmenting graph for S with larger increment, then the node classes corresponding to the components of H form an independent set in Γ the weight of which is obviously at least as large as the increment of H and hence is greater than that of Q, contradicting the assumption. We thus have proved the following lemma

LEMMA 4.6 *If Q is an independent set of maximum weight in the graph Γ, then the increment of the corresponding graph H_Q is maximum over all possible P_4-free augmenting graphs for S.*

Assume now that S admits no augmenting graphs containing a P_4, and let H be a P_4-free augmenting graph for S with maximum increment. Then, obviously, the independent set obtained from S by H-augmentation is of maximum cardinality. This observation together with Lemma 4.6 provide a way to reduce the independent set problem in Banner$_2$-free graphs to the following two subproblems:

(P_1) finding augmenting graphs containing a P_4;
(P_2) finding an independent set of maximum weight in the auxiliary graph Γ.

We formally fix the above proposition in the following recursive procedure.

ALPHA(G).
Input: A Banner$_2$-free graph G.
Output: An independent set S of maximum size in G.

(1) Find an arbitrary maximal under inclusion independent set S in G. If $S = V(G)$ go to 7.
(2) As long as possible apply H-augmentations to S with H containing a P_4.
(3) Partition the vertices of $V(G) - S$ into node classes $Q_1, \ldots, Q_k$.
(4) For every $j = 1, \ldots, k$, find a maximum independent set $W_j = \text{ALPHA}(G[Q_j])$.
(5) Construct the auxiliary graph Γ and find an independent set $Q = \{Q_1, \ldots, Q_p\}$ of maximum weight in it.
(6) If the weight of Q is positive, augment S by exchanging $N_S(Q_i)$ by W_i for each $i = 1, \ldots, p$.
(7) Return S and STOP.

In the rest of this section we show that the problem (P_2), i.e., finding an independent set of maximum weight in the auxiliary graph Γ, has a polynomial-time solution whenever G is a Banner$_2$-free graph.

Let us say that an edge Q_iQ_j in the graph Γ is of type A if $N_S(Q_i) \subseteq N_S(Q_j)$ or $N_S(Q_j) \subseteq N_S(Q_i)$ or $\max(|N_S(Q_i) - N_S(Q_j)|, |N_S(Q_j) - N_S(Q_i|) \leq 1$, and of type B otherwise. Particularly, for every edge Q_iQ_j of type B, we have $N_S(Q_i) - N_S(Q_j) \neq \emptyset$, $N_S(Q_j) - N_S(Q_i) \neq \emptyset$ and each vertex of Q_i is adjacent to each vertex of Q_j in the graph G.

CLAIM 4.1 *If vertices Q_1, Q_2, Q_3 induce a P_3 in Γ with edges Q_1Q_2 and Q_2Q_3, then at least one of these edges is of type A.*

Proof. Assume to the contrary that both edges are of type B. Denote by a a vertex of G in $N_S(Q_1) - N_S(Q_2)$ and by b a vertex of G in $N_S(Q_3) - N_S(Q_2)$. Let $q_j \in Q_j$ for $j = 1, 2, 3$. By the assumption (4.2),

q_1 must have a non-neighbor c in Q_1. Since Q_1 is not adjacent to Q_3 in Γ, b has no neighbor in Q_1 and a has no neighbor in Q_3 in G. But now the vertices a, b, c, q_1, q_2, q_3 induce a Banner$_2$ in G. □

CLAIM 4.2 *If vertices Q_1, Q_2, Q_3, Q_4 induce a P_4 in Γ with edges Q_1Q_2, Q_2Q_3, Q_3Q_4, then the mid-edge Q_2Q_3 is of type B and the other two edges are of type A.*

Proof. Assume by contradiction that the edge Q_1Q_2 is of type B. Then from Claim 4.1 it follows that Q_2Q_3 is of type A. Let $q_j \in Q_j$ for $j = 1, 2, 3, 4$. Denote by a a vertex of G in $N_S(Q_1) - N_S(Q_2)$ and by b a vertex in Q_1 non-adjacent to q_1. If q_2 is not adjacent to q_3, then we consider a vertex $c \in N_S(Q_2) \cap N_S(Q_3)$ and conclude that a, b, c, q_1, q_2, q_3 induce in G a Banner$_2$. Now let q_2 be adjacent to q_3. If q_3 is adjacent to q_4, then G contains a Banner$_2$ induced by vertices a, b, q_1, q_2, q_3, q_4. If q_3 is not adjacent to q_4, then the edge Q_3Q_4 is of type A and hence there is a vertex c in $N_S(q_3) \cap N_S(q_4)$. But then G contains a Banner$_2$ induced by vertices a, b, q_1, q_2, q_3, c. This contradiction proves that Q_1Q_2 is of type A. Symmetrically, Q_3Q_4 is of type A.

To complete the proof, assume that the mid-edge Q_2Q_3 is of type A too. Remember that $|N_S(Q_i)| \geq 3$ and hence $|N_S(Q_i) \cap N_S(Q_j)| \geq 2$ for any edge Q_iQ_j of type A. Since $N_S(Q_1)$ and $N_S(Q_3)$ are disjoint, we conclude that $N_S(Q_1) \cup N_S(Q_3) \subseteq N_S(Q_2)$. Similarly, $N_S(Q_2) \cup N_S(Q_4) \subseteq N_S(Q_3)$. This is possible only if $N_S(Q_1) = N_S(Q_4) = \emptyset$, which contradicts the maximality of S. □

REMARK In the concluding part of the proof of Claim 4.2 we did not use the fact that vertices Q_1 and Q_4 are non-adjacent in Γ, which means that no induced C_4 in Γ has three edges of type A. In conjunction with Claim 4.1 this implies

CLAIM 4.3 *In any induced C_4 in the graph Γ, adjacent edges have different types.*

Combining Claims 4.1, 4.2 and 4.3, we obtain

CLAIM 4.4 *Graph Γ contains no induced $K_{2,3}$, P_5, C_5 and* Banner.

Finally, we prove

CLAIM 4.5 *Graph Γ contains no induced $\overline{C_k}$ with odd $k > 5$.*

Proof. By contradiction, let $C_k = (x_1, x_2, \ldots, x_k)$ be an induced cycle of odd length $k > 5$ in the complement to Γ. Consider two consecutive vertices of the cycle, say x_1 and x_2. It is not hard to see that pairs $x_{k-2}x_1$

and x_2x_5 form mid-edges of induced P_4's in Γ. Hence by Claim 4.2 both edges are of type B. Now let us consider the set of edges $F = \{x_1x_5, x_1x_6, \ldots, x_1x_{k-3}, x_1x_{k-2}\}$ in Γ. Any two consecutive edges x_1x_j and x_1x_{j+1} in F belong to a C_4 induced by vertices x_1, x_j, x_2, x_{j+1} in Γ. Hence, by Claim 4.3, edges of type A in F strictly alternate with edges of type B. Since x_1x_{k-2} is of type B and k is odd, we conclude that x_1x_5 is an edge of type B in Γ. But then vertices x_1, x_5, x_2 induce a P_3 in Γ with both edges of type B, contradicting Claim 4.1. □

From Claims 4.4 and 4.5 we deduce that Γ is a Berge graph. It is known (Barré and Fouquet, 1999; Olariu, 1989) that the Strong Perfect Graph Conjecture is true in (P_5, banner)-free graphs. We hence conclude that

LEMMA 4.7 *Graph Γ is perfect.*

Lemma 4.7 together with the result in Grötschel et al. (1984) show that an independent set of maximum weight in the graph Γ can be found in polynomial time. The weights $\alpha(G[Q_j])$ to the vertices of Γ are computed recursively. Obviously, if every step of a recursive procedure can be implemented in polynomial time, and the number of recursive calls is bounded by a polynomial, then the total time of the procedure is polynomial as well. In algorithm ALPHA the recursion applies to vertex-disjoint subgraphs. Therefore, the number of recursive calls is polynomial. Every step of algorithm ALPHA, other than Step 2, has a polynomial time complexity. Thus, polynomial-time solvability of Step 2 would imply polynomiallity of the entire algorithm. The converse statement is trivial. As a result we obtain

THEOREM 4.10 *The maximum independent set problem in the class of* Banner$_2$*-free graphs is polynomially equivalent to the problem of finding augmenting graphs containing a P_4.*

5.3 Other types of augmenting graphs

In this section we give additional examples of algorithms that detect augmenting graphs in particular classes of graphs. Throughout this section we assume that G is a (P_8, banner)-free graph. As mentioned in Section 4, a minimal augmenting (P_8, banner)-free graph is one of the following graphs (see Figure 4.2 for definitions of the graphs):

- a complete bipartite graph $K_{r,r+1}$ with $r \geq 0$,
- a $L^1_{r,s}$ or a $L^2_{r,s}$ with $r \geq 2$ and $s \geq 0$,
- a $M_{r,s}$ with $r \geq 1$ and $r \geq s \geq 0$,
- a N_s with $s \geq 0$,

– one of the graphs $F_2, \ldots, F_5$.

An augmenting F_i $(2 \leq i \leq 5)$ can be found in polynomial time since these graphs have a number of vertices which does not depend on the size of G. Moreover, we know from the preceding section that complete bipartite augmenting graphs can be detected in polynomial time in banner-free graphs. In the present section we show how the remaining graphs listed above can be detected in polynomial time, assuming that G is (P_8, banner)-free.

We denote by W^i the set of white vertices having exactly i black neighbors. Given a white vertex w, we denote by $B(w) = N(w) \cap S$ the set of black neighbors of w. We first show how to find an augmenting $M_{r,s}$ with $r \geq 1$ and $r \geq s \geq 0$. We assume there is no augmenting $K_{1,2}$ (such augmenting graphs can easily be detected).

Consider three black mutually non-adjacent vertices a_1, c, e such that $a_1 \in W^1$, $|B(c)| \geq |B(e)|$, $B(a_1) \cap B(c) = \{b_1\}$, $B(c) \cap B(e) = \{d\}$ and $B(a_1) \cap B(e) = \emptyset$. Notice that we have chosen, on purpose, the same labeling as in Figure 4.2. The following algorithm determines whether this initial structure can be extended to an augmenting $M_{r,s}$ in G (with $r = |B(c)| - 1$ and $s = |B(e)| - 1$) (Gerber et al., 2004a).

(a) Determine $A = (B(c) \cup B(e)) - \{b_1, d\}$.
(b) For each vertex $u \in A$, determine the set $N_1(u)$ of white neighbors of u which are in W^1, and which are not adjacent to a_1, c or e.
(c) Let G' be the subgraph of G induced by $\bigcup_{u \in A} N_1(u)$:
   - if $\alpha(G') = |A|$ then $A \cup \{a_1, b_1, c, d, e\}$ together with any maximum independent set in G' induce the desired $M_{r,s}$;
   - otherwise, $M_{r,s}$ does not exist in G.

As shown in Gerber et al. (2004a), G' is $(\text{banner}, P_5, C_5, \text{fork})$-free when G is (P_8, banner)-free. Since polynomial algorithms are available for the computation of a maximum independent set in this class of graphs (Alekseev, 1999; Lozin, 2000a), the above Step (c) can be performed in polynomial time.

Finding an augmenting N_s with $s \geq 0$ is even simpler. Indeed, consider five white non-adjacent vertices $x_1, \ldots, x_5$ such that $x_i \in W^2$ $(i = 1, \ldots, 4)$, $\bigcup_{i=1}^{4} (\{x_i\} \cup B(x_i))$ induces a $C_8 = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ in G, and $B(x_5) \cap \{y_1, \ldots, y_4\} = \{y_2, y_4\}$. The following algorithm determines whether this initial structure can be extended to an augmenting N_s in G (with $s = |B(x_5)| - 2$) (Gerber et al., 2004a).

(a) Determine $A = B(x_5) - \{y_2, y_4\}$.
(b) For each vertex $u \in A$, determine the set $N_1(u)$ of white neighbors of u which are in W^1, and which are not adjacent to $x_1, \ldots, x_4$.
(c) Let G' be the subgraph of G induced by $\bigcup_{u \in A} N_1(u)$:

- if $\alpha(G') = |A|$ then $A \cup \{x_1, \ldots, x_5, y_1, \ldots, y_4\}$ together with any maximum independent set in G' induce the desired N_s.
- otherwise, N_s does not exist in G.

If G is (P_8, banner)-free then G' is the union of disjoint cliques (Gerber et al., 2004a), which means that a maximum independent set in G' can easily be obtained by choosing a vertex in each connected component of G'.

We finally show how augmenting $L^1_{r,s}$ and $L^2_{r,s}$ with $r \geq 2$ and $s \geq 0$ can be found in polynomial time, assuming there is no augmenting $P_3 = K_{1,2}$, $P_5 = M_{1,0}$ and $P_7 = M_{1,1}$ (these can easily be detected in polynomial time). Consider four white non-adjacent vertices b_1, b_2, d and x such that x belongs to W^1, b_1 and b_2 belong to W^2, $\{b_1, b_2, d\} \cup B(b_1) \cup B(b_2)$ induces a $C_6 = (c_1, b_1, a, b_2, c_2, d)$ in G, and x is adjacent to a or (exclusive) c_1. Notice that we have chosen, on purpose, the same labeling as in Figure 4.2. The following algorithm determines whether this initial structure can be extended to an augmenting $L^1_{r,s}$ or $L^2_{r,s}$ in G (with $r + s = |B(d)|$) (Gerber et al., 2004a).

(a) Determine $A = B(d) - \{c_1, c_2\}$ as well as the set $\overline{W}$ of white vertices which are not adjacent to x, b_1, b_2 or d.

(b) For each vertex $u \in A$, determine the set $N_1(u)$ of white neighbors of u which are in $W^1 \cap \overline{W}$ as well as the set $N_2(u)$ of white vertices in $W^2 \cap \overline{W}$ which are adjacent to both a and u.

(c) Let G' be the subgraph of G induced by the vertices in the set $\bigcup_{u \in A}(N_1(u) \cup N_2(u))$:
- if $\alpha(G') = |A|$ then $A \cup \{a, b_1, b_2, c_1, c_2, d, x\}$ together with any maximum independent set in G' induce the desired $L^1_{r,s}$ (if x is adjacent to c_1) or $L^2_{r,s}$ (if x is adjacent to a).
- otherwise, $L^1_{r,s}$ and $L^2_{r,s}$ do not exist in G.

Once again, it is proved in Gerber et al. (2004a) that the subgraph G' is (banner, P_5, C_5, fork)-free when G is (P_8, banner)-free, and this implies that the above Step (c) can be performed in polynomial time.

6. Conclusion

In this paper we reviewed the method of augmenting graphs, which is a general approach to solve the maximum independent set problem. As the problem is generally NP-hard, no polynomial-time algorithms are available to implement the approach. However, for graphs in some special classes, this method leads to polynomial-time solutions. In particular, the idea of augmenting graphs has been used to solve the problem for line graphs, which is equivalent to finding maximum matchings

in general graphs. The first polynomial-time algorithm for the maximum matching problem has been proposed by Edmonds in 1965. Fifteen years later Minty (1980) and Sbihi (1980) extended independently of each other the solution of Edmonds from line graphs to claw-free graphs. The idea of augmenting graphs did not see any further developments for nearly two decades. Recently Alekseev (1999) and Mosca (1999) revived the interest in this approach, which has led to many new results on the topic. This paper summarizes most of those results and proposes several new contributions. In particular, we show that in the class of $(S_{1,2,j}, \text{Banner}_k)$-free graphs for any fixed j and k, there are finitely many minimal augmenting graphs of bounded vertex degree different from augmenting chains. Together with polynomial-time algorithms to find augmenting chains in $(S_{1,2,j}, \text{Banner}_1)$-free graphs (Hertz et al., 2003) and $S_{1,2,3}$-free graphs (Gerber et al., 2003) this immediately implies polynomial-time solutions to the maximum independent set problem in classes of $(S_{1,2,j}, \text{Banner}_1, K_{1,n})$-free graphs and $(S_{1,2,3}, \text{Banner}_k, K_{1,n})$-free graphs, both generalizing claw-free graphs. The second class extends also $(P_5, K_{1,n})$-free graphs and $(P_2 + P_3, K_{1,n})$-free graphs for which polynomial-time solutions have been proposed by Mosca (1997) and Alekseev (2004), respectively. We believe the idea of augmenting graphs may lead to many further results for the maximum independent set problem and hope the present paper will be of assistance in this respect.

References

Abbas, N. and Stewart, L.K. (2000). Biconvex graphs: ordering and algorithms. *Discrete Applied Mathematics*, 103:1 – 19.

Alekseev, V.E. (1983) On the local restrictions effect on the complexity of finding the graph independence number. *Combinatorial-Algebraic Methods in Applied Mathematics*, Gorkiy University Press, Gorky, pp. 3 – 13, in Russian.

Alekseev, V.E. (1999). A polynomial algorithm for finding largest independent sets in fork-free graphs, *Diskretnyj Analiz i Issledovanie Operatsii, Seriya 1* 6(4):3 – 19. (In Russian, translation in *Discrete Applied Mathematics*, 135:3 – 16, 2004).

Alekseev, V.E. (2004). On easy and hard hereditary classes of graphs with respect to the independent set problem, *Discrete Applied Mathematics*, 132:17 – 26.

Alekseev, V.E. and Lozin, V.V. (2004) Augmenting graphs for independent sets. *Discrete Applied Mathematics*, 145:3 – 10.

Barré, V. and Fouquet, J.-L. (1999). On minimal imperfect graphs without induced P_5. *Discrete Applied Mathematics*, 94:9 – 33.

Barrow, H.G. and Burstal, R.M. (1976). Subgraph isomorphism, matching relational structures, and maximal cliques. *Information Processing Letters*, 4:83 – 84.

Berge, C. (1957). Two theorems in graph theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43:842 – 844.

Boliac, R. and Lozin, V.V. (2001). An attractive class of bipartite graphs. *Discussiones Mathematicae Graph Theory*, 21:293 – 301.

Boliac, R. and Lozin, V.V. (2003a). Independent domination in finitely defined classes of graphs. *Theoretical Computer Science*, 301:271 – 284.

Boliac, R. and Lozin, V.V. (2003b). An augmenting graph approach to the independent set problem in P_5-free graphs. *Discrete Applied Mathematics*, 131:567 – 575.

Bomze, I.M., Budinich, M., Pardalos, P.M., and Pelillo, M. (1999). The maximum clique problem. In: D.-Z. Du and P.M. Pardalos (eds), *Handbook of Combinatorial Optimization - Suppl. Vol. A*, Kluwer Academic Publishers, pp. 1 – 74, Boston, MA.

Boros, E. and Hammer, P.L. (2002). Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123:155 – 225.

Cameron, K. (1989). Induced matchings. *Discrete Applied Mathematics*, 24:97 – 102.

Denley, T. (1994). The independence number of graphs with large odd girth. *Electronic Journal of Combinatorics*, 1:Research Paper 9, 12 pp.

De Simone, C. and Sassano, A. (1993). Stability number of bull- and chair-free graphs. *Discrete Applied Mathematics*, 41:121 – 129.

Ebenegger, Ch., Hammer, P.L., and de Werra, D. (1984). Pseudo-Boolean functions and stability of graphs. *Annals of Discrete Mathematics*, 19:83 – 98.

Edmonds, J. (1965). Path, trees, and flowers. *Canadian Journal of Mathematics*, 17:449 – 467.

Garey, M.R., Johnson, D.S., and Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237 – 267.

Gerber, M.U., Hertz, A., and Lozin, V.V. (2003). *Augmenting Chains in Graphs Without a Skew Star*. RUTCOR Research Report, Rutgers University, USA.

Gerber, M.U., Hertz, A., and Lozin, V.V. (2004a). Stable sets in two subclasses of banner-free graphs. *Discrete Applied Mathematics*, 132:121 – 136.

Gerber, M.U., Hertz, A., and Schindl, D. (2004b). P_5-free augmenting graphs and the maximum stable set problem. *Discrete Applied Mathematics*, 132:109 – 119.

Grötschel, M., Lovász, L., and Schrijver, A. (1984). Polynomial algorithms for perfect graphs. *Annals of Discrete Mathematics*, 21:325 – 356.

Halldórsson, M.M. (1995). Approximating discrete collections via local improvements. *Proceedings of the Sixth SAIM-ACM Symposium on Discrete Algorithms (San Francisco, CA, 1995)*, pp. 160 – 169.

Halldórsson, M.M. (2000). Approximation of weighted independent sets and hereditary subset problems. *Journal of Graph Algorithms and Applications*, 4:1 – 16.

Hammer, P.L., Peled, U.N., and Sun, X. (1990). Difference graphs. *Discrete Applied Mathematics*, 28:35 – 44.

Håstad, J. (1999). Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105 – 142.

Hertz, A., Lozin, V.V., and Schindl, D. (2003). On finding augmenting chains in extensions of claw-free graphs. *Information Processing Letters* 86:311 – 316.

Jagota, A., Narasimhan, G., and Soltes, L. (2001). A generalization of maximal independent sets. *Discrete Applied Mathematics*, 109:223 – 235.

Johnson, D.S., Yannakakis, M., and Papadimitriou, C.H. (1988). On generating all maximal independent sets. *Information Processing Letters*, 27:119 – 123.

Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G. (1980). Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9:558 – 565.

Lozin, V.V. (2000a). Stability in P_5- and Banner-free graphs. *European J. Operational Research*, 125:292 – 297, 2000a.

Lozin, V.V. (2000b). E-free bipartite graphs. *Discrete Analysis and Operations Research*, Ser.1, 7:49 – 66, in Russian.

Lozin, V.V. (2002a). On maximum induced matchings in bipartite graphs. *Information Processing Letters*, 81:7 – 11.

Lozin, V.V. (2002b). Bipartite graphs without a skew star. *Discrete Mathematics*, 257:83 – 100.

Lozin, V.V. and Rautenbach, D. (2003). Some results on graphs without long induced paths. *Information Processing Letters*, 86:167 – 171.

Minty, G.J. (1980). On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory*, Ser.B, 28:284 – 304.

Mosca, R. (1997). Polynomial algorithms for the maximum independent set problem on particular classes of P_5-free graphs. *Information Pro-*

cessing Letters, 61:137 – 144.

Mosca, R. (1999). Independent sets in certain P_6-free graphs. *Discrete Applied Mathematics*, 92:177 – 191.

Müller, H. (1990). Alternating cycle-free matchings. *Order*, 7:11 – 21.

Olariu, S. (1989). The strong perfect graph conjecture for pan-free graphs. *Journal of Combinatorial Theory*, Ser. B, 47:187 – 191.

Pelillo, M., Siddiqi, K., and Zucker, S.W. (1999). Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:1105 - 1120.

Plaisted, D.A. and Zaks, S. (1980). An NP-complete matching problem. *Discrete Applied Mathematics*, 2:65 – 72.

Sbihi, N. (1980). Algorithme de recherche d'un independent de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics*, 29:53 – 76.

Spinrad, J., Brandstädt, A., and Stewart, L. (1987). Bipartite permutation graphs. *Discrete Applied Mathematics*, 18:279 – 292.

Stockmeyer, L. and Vazirani, V.V. (1982). NP-completeness of some generalizations of the maximum matching problems. *Information Processing Letters*, 15:14 – 19.

Tsukiyama, S., Ide, M., Ariyoshi, H., and Shirakawa, I. (1977). A new algorithm for generating all maximal independent sets. *SIAM Journal on Computing*, 6:505 – 516.

Yannakakis, M. (1981). Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10:310 – 327.

Yannakakis, M. and Gavril, F. (1980). Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38:364 – 372, 1980.

Chapter 5

INTERIOR POINT AND SEMIDEFINITE APPROACHES IN COMBINATORIAL OPTIMIZATION

Kartik Krishnan
Tamás Terlaky

Abstract Conic programming, especially semidefinite programming (SDP), has been regarded as *linear programming for the 21st century.* This tremendous excitement was spurred in part by a variety of applications of SDP in integer programming (IP) and combinatorial optimization, and the development of efficient primal-dual interior-point methods (IPMs) and various first order approaches for the solution of large scale SDPs. This survey presents an up to date account of semidefinite and interior point approaches in solving NP-hard combinatorial optimization problems to optimality, and also in developing approximation algorithms for some of them. The interior point approaches discussed in the survey have been applied directly to non-convex formulations of IPs; they appear in a cutting plane framework to solving IPs, and finally as a subroutine to solving SDP relaxations of IPs. The surveyed approaches include non-convex potential reduction methods, interior point cutting plane methods, primal-dual IPMs and first-order algorithms for solving SDPs, branch and cut approaches based on SDP relaxations of IPs, approximation algorithms based on SDP formulations, and finally methods employing successive convex approximations of the underlying combinatorial optimization problem.

1. Introduction

Optimization problems seem to divide naturally into two categories: those with *continuous* variables, and those with *discrete* variables, which we shall hereafter call combinatorial problems. In continuous problems, we are generally looking for a set of real numbers or even a function; in combinatorial optimization, we are looking for certain objects from a finite, or possibly countably infinite set, typically an integer, graph etc.

These two kinds of problems have different flavors, and the methods for solving them are quite different too. In this survey paper on interior point methods (IPMs) in combinatorial optimization, we are in a sense at the boundary of these two categories, i.e., we are looking at IPMs, that represent continuous approaches towards solving combinatorial problems usually formulated using discrete variables.

To better understand why one would adopt a continuous approach to solving discrete problems, consider as an instance the linear programming (LP) problem. The LP problem amounts to minimizing a linear functional over a polyhedron, and arises in a variety of applications in combinatorial optimization. Although the LP is in one sense a continuous optimization problem, it can be viewed as a combinatorial problem. The set of candidate solutions are extreme points of the underlying polyhedron, and there are only a finite (in fact combinatorial) number of them. Before the advent of IPMs, the classical algorithm for solving LPs was the simplex algorithm. The simplex algorithm can be viewed as a combinatorial approach to solving an LP, and it deals exclusively with the extreme point solutions; at each step of the algorithm the next candidate extreme point solution is chosen in an attempt to improve some performance measure of the current solution, say the objective value. The improvement is entirely guided by local search, i.e., the procedure only examines a neighboring set of configurations, and greedily selects one that improves the current solution. As a result, the search is quite myopic, with no consideration given to evaluate whether the current move is actually useful globally. The simplex method simply lacks the ability for making such an evaluation. Thus, although, the simplex method is quite an efficient algorithm in practice, there are specially devised problems on which the method takes a disagreeably exponential number of steps. In contrast, all polynomial-time algorithms for solving the LP employ a continuous approach. These include the ellipsoid method (Grötschel et al., 1993), or IPMs that are subsequent variants of the original method of Karmarkar (1984). It must be emphasized here that IPMs have both better complexity bounds than the ellipsoid method (we will say more on this in the subsequent sections), and the further advantage of being very efficient in practice. For LP it has been established that for very large, sparse problems IPMs often outperform the simplex method. IPMs are also applicable to more general conic (convex) optimization problems with efficiently computable self-concordant barrier functions (see the monographs by Renegar, 2001, and Nesterov and Nemirovskii, 1994). This includes important classes of optimization problems such as second order cone programming (SOCP) and semidefinite programming (SDP). For such problems, IPMs are indeed the algorithm of choice.

We now present the underlying ideas behind primal-dual IPMs (see Roos et al., 1997; Wright, 1997; Ye, 1997; Andersen et al., 1996) the most successful class of IPMs in computational practice. For ease of exposition, we consider the LP problem. We will later consider extensions to convex programming problems, especially the SDP, in Section 4.2. Consider the standard linear programming problem (LP)

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{LP}$$

with dual

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A^T y + s = c, \\ & s \geq 0, \end{aligned} \tag{LD}$$

where m and n represent the number of constraints and variables in the primal problem (LP), with $m < n$. Also, c, x, and s are vectors in $\mathbb{R}^n$, b and y are vectors in $\mathbb{R}^m$, and A is an $m \times n$ matrix with full row rank. The constraints $x, s \geq 0$ imply that these vectors belong to $\mathbb{R}^n_+$, i.e., all their components are non-negative. Similarly, $x > 0$ implies that $x \in \mathbb{R}^n_{++}$ (the interior of $\mathbb{R}^n_+$), i.e., all components of x are strictly positive.

The optimality conditions for LP include primal and dual feasibility and the complementary slackness conditions, i.e.,

$$\begin{aligned} Ax = b, \quad & x \geq 0, \\ A^T y + s = c, \quad & s \geq 0, \\ x \circ s = 0, & \end{aligned} \tag{5.1}$$

where $x \circ s = (x_i s_i)$, $i = 1, \ldots, n$ is the Hadamard product of the vectors x and s.

Consider perturbing the complementarity slackness conditions in (5.1) to $x \circ s = \mu e$, where e is the all-ones vector and $\mu > 0$ is a given scalar. Neglecting the inequality constraints in (5.1) for the moment this gives the following system:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ x \circ s &= \mu e. \end{aligned} \tag{5.2}$$

A typical feasible primal-dual IPM for LP starts with a strictly feasible (x, y, s) solution in $\mathbb{R}^n_{++}$, i.e., $x, s > 0$. The perturbed system (5.2) has an unique solution (x_μ, y_μ, s_μ) for each $\mu > 0$. Moreover, the set

$\{(x_\mu, y_\mu, s_\mu), \mu > 0\}$, also called the *central path*, is a smooth, analytic curve converging to an optimal solution (x^*, y^*, s^*) as $\mu \to 0$. In fact, this limit point is in the relative interior of the optimal set, and is a strictly complementary solution, i.e., $x^* + s^* > 0$ and $x^* \circ s^* = 0$.

If we solve (5.2) by Newton's method, we get the following linearized system

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - x \circ s. \end{aligned}$$

This system has a unique solution, namely

$$\begin{aligned} \Delta y &= (AXS^{-1}A^T)^{-1}(b - \mu As^{-1}), \\ \Delta s &= -A^T\Delta y, \\ \Delta x &= \mu s^{-1} - x - x \circ s^{-1} \circ \Delta s, \end{aligned} \tag{5.3}$$

where $X = \mathrm{Diag}(x)$ and $S = \mathrm{Diag}(s)$ are diagonal matrices, whose entries are the components of x and s, respectively. Since the constraints $x, s > 0$ were neglected in (5.2), one needs to take *damped* Newton steps. Moreover, the central path equations (5.2) are nonlinear and so it is impossible to obtain the point (x_μ, y_μ, s_μ) on the central path via damped Newton iterations alone. One requires a proximity measure $\delta(x, s, \mu)$ (see Roos et al., 1997; Wright, 1997) that measures how *close* the given point (x, y, s) is to the corresponding point (x_μ, y_μ, s_μ) on the central path. Finally, IPMs ensure that the sequence of iterates $\{(x, y, s)\}$ remain in some neighborhood of the central path by requiring that $\delta(x, s, \mu) \leq \tau$ for some $\tau > 0$, where τ is either an absolute constant or may depend on n.

We are now ready to present a generic IPM algorithm for LP.

Generic Primal-Dual IPM for LP.

Input. A, b, c, a starting point (x^0, y^0, s^0) satisfying the interior point condition (see Roos et al., 1997; Wright, 1997), i.e., $x^0, s^0 > 0$, $Ax^0 = b$, $A^Ty^0 + s^0 = c$, and $x^0 \circ s^0 = e$, a barrier parameter $\mu = 1$, a proximity threshold $\tau > 0$ such that $\delta(x^0, s^0, \mu) \leq \tau$, and an accuracy parameter $\epsilon > 0$.

(1) Reduce the barrier parameter μ.
(2) **If** $\delta(x, s, \mu) > \tau$ compute $(\Delta x, \Delta y, \Delta s)$ using (5.3).
(3) Choose some $\alpha \in (0, 1]$ such that $x + \alpha\Delta x, s + \alpha\Delta s > 0$, and proximity $\delta(x, s, \mu)$ appropriately reduced.
(4) Set $(x, y, s) = (x + \alpha\Delta x, y + \alpha\Delta y, s + \alpha\Delta s)$.

(5) **If** the duality gap $x^T s < \epsilon$ **then** stop,
else if $\delta(x, s, \mu) \leq \tau$ **goto** step 1,
else goto step 2.

We can solve an LP problem with rational data, to within an accuracy $\epsilon > 0$, in $O(\sqrt{n}\log(1/\epsilon))$ iterations (see Roos et al., 1997, for more details). This is the best iteration complexity bound for a primal-dual interior point algorithm. Most combinatorial optimization problems, other than flow and matching problems are NP-complete, all of which are widely considered unsolvable in polynomial time (see Garey and Johnson, 1979; Papadimitriou and Steiglitz, 1982, for a discussion on intractability and the theory of NP completeness). We are especially interested in these problems. One way of solving such problems is to consider successively strengthened convex relaxations (SDP/SOCP) of these problems in a branch-cut framework, and employing IPMs to solving these relaxations. On the other hand, semidefinite programming (SDP) has been applied with a great deal of success in developing approximation algorithms for various combinatorial problems, the showcase being the Goemans and Williamson (1995) approximation algorithm for the maxcut problem. The algorithm employs an SDP relaxation of the maxcut problem which can be solved by IPMs, followed by an ingenious randomized rounding procedure. The approximation algorithm runs in polynomial time, and has a worst case performance guarantee. The technique has subsequently been extended to other combinatorial optimization problems.

We introduce two canonical combinatorial optimization problems, namely the maxcut and maximum stable set problems, that will appear in the approaches mentioned in the succeeding sections.

(1) **Maxcut Problem:** Let $G = (V, E)$ denote an edge weighted undirected graph without loops or multiple edges. Let $V = \{1, \ldots, n\}$, $E \subset \{\{i, j\} : 1 \leq i < j \leq n\}$, and $w \in \mathbb{R}^{|E|}$, with $\{i, j\}$ the edge with endpoints i and j, with weights w_{ij}. We assume that $n = |V|$, and $m = |E|$. For $S \subseteq V$, the set of edges $\{i, j\} \in E$ with one endpoint in S and the other in $V \backslash S$ form the cut denoted by $\delta(S)$. We define the weight of the cut as $w(\delta(S)) = \sum_{\{i,j\} \in \delta(S)} w_{ij}$. The maximum cut problem, denoted as (MC), is the problem of finding a cut whose total weight is maximum.

(2) **Maximum Stable Set Problem:** Given a graph $G = (V, E)$, a subset $V' \subset V$ is called a stable set, if the induced subgraph on V' contains no edges. The maximum stable set problem, denoted by (MSS), is to find the stable set of maximum cardinality.

It must be mentioned that although (MC) and (MSS) are NP-complete problems, the maxcut problem admits an approximation algorithm, while no such algorithms exist for the maximum stable set problem unless P = NP (see Arora and Lund, 1996, for a discussion on the hardness of approximating various NP-hard problems).

This paper is organized as follows: Section 2 deals with non-convex potential function minimization, among the first techniques employing IPMs in solving difficult combinatorial optimization problems. Section 3 deals with interior point cutting plane algorithms, especially the analytic center cutting plane method (ACCPM) and the volumetric center method. These techniques do not require a knowledge of the entire constraint set, and consequently can be employed to solve integer programs (IPs) with exponential or possibly infinite number of constraints. They can also be employed as a certificate to show certain IPs can be solved in polynomial time, together with providing the best complexity bounds. Section 4 discusses the complexity of SDP and provides a generic IPM for SDP. This algorithm is employed in solving the SDP formulations and relaxations of integer programming problems discussed in the succeeding sections. Although IPMs are the algorithms of choice for an SDP, they are fairly limited in the size of problems they can handle in computational practice. We discuss various first order methods that exploit problem structure, and have proven to be successful in solving large scale SDPs in Section 5. Section 6 discusses branch and cut SDP approaches to solving IPs to optimality, advantages and issues involved in employing IPMs in branching, restarting, and solving the SDP relaxations at every stage. Section 7 discusses the use of SDP in developing approximation algorithms for combinatorial optimization. Section 8 discusses approaches employing successive convex approximations to the underlying IP, including recent techniques based on polynomial and copositive programming. We wish to emphasize that the techniques in Section 8 are more of a theoretical nature, i.e., we have an estimate on the number of liftings needed to solve the underlying IP to optimality, however the resulting problems grow in size beyond the capacity of current state of the art computers and software; this is in sharp contrast to the practical branch and cut approaches in Section 6. We conclude with some observations in Section 9, and also highlight some of the open problems in each area.

The survey is by no means complete; it represents the authors biased view of this rapidly evolving research field. The interested reader is referred to the books by Chvátal (1983), Papadimitriou and Steiglitz (1982) on combinatorial optimization, and Schrijver (1986) on linear and integer programming. The books by Roos et al. (1997), Wright (1997),

and Ye (1997) contain a treatment of IPMs in linear optimization. A recent survey on SOCP appears in Alizadeh and Goldfarb (2003). Excellent references for SDP include the survey papers by Vandenberghe and Boyd (1996), Todd (2001), the SDP handbook edited by Wolkowicz et al. (2000), and the recent monograph by De Klerk (2002). A repository of recent papers dealing with interior point approaches to solving combinatorial optimization problems appear in the following websites: Optimization Online, IPM Online, and the SDP webpage maintained by Helmberg. Finally, recent surveys by Laurent and Rendl (2003) and Mitchell et al. (1998) also complement the material in this survey.

2. Non-convex potential function minimization

The non-convex potential function approach was introduced by Karmarkar (1990); Karmarkar et al. (1991) as a nonlinear approach for solving integer programming problems. Warners et al. (1997a,b) also utilized this approach in solving frequency assignment problems (FAP), and other structured optimization problems. We present a short overview of the approach in this section.

Consider the following binary $\{-1, 1\}$ feasibility problem:

$$\text{find } \bar{x} \in \{-1, 1\}^n \text{ such that } \bar{A}\bar{x} \leq \bar{b}. \tag{5.4}$$

Let $\mathcal{I}$ denote the feasible set of (5.4). Binary feasibility problems arise in a variety of applications. As an example, we can consider the stable set problem on the graph $G = (V, E)$ with $n = |V|$. The constraints $\bar{A}x \leq \bar{b}$ are given by $x_i + x_j \leq 0$, $\{i, j\} \in E$, where the set of $\{-1, 1\}$ vectors $x \in \mathbb{R}^n$ correspond to incidence vectors of stable sets in the graph G, with $x_i = 1$ if node i is in the stable set, and $x_i = -1$ otherwise.

The problem (5.4) is NP-complete, and there is no efficient algorithm that would solve it in polynomial time. Therefore, we consider the following polytope $\mathcal{P}$, which is a relaxation of $\mathcal{I}$.

$$\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leq b\},$$

where $A = \begin{pmatrix} \bar{A} & I & -I \end{pmatrix}^T$, and $b = \begin{pmatrix} \bar{b} & e & e \end{pmatrix}^T$. Here I is the $n \times n$ identity matrix, and e is the vector of all ones. Finding a vector $x \in \mathcal{P}$ amounts to solving an LP problem, and can be done efficiently in polynomial time. Let $\mathcal{P}^0$ denote the relative interior of $\mathcal{P}$, i.e., $\mathcal{P}^0 = \{x \in \mathbb{R}^n : Ax < b\}$. Since $-e \leq x \leq e$, $\forall x \in \mathcal{P}$, we have $x^T x \leq n$, with equality occurring if and only if $x \in \mathcal{I}$. Thus, (5.4) can also be formulated as the following concave quadratic optimization problem with

linear constraints.

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i^2 \\ \text{s.t.} \quad & x \in \mathcal{P}. \end{aligned} \tag{5.5}$$

Since the objective function of (5.5) is non-convex, this problem is NP-complete as well. However, the global optimum of (5.5), when (5.4) is feasible, corresponds to ± 1 binary solutions of this problem.

Consider now a non-convex potential function $\phi(x)$, where

$$\phi(x) = \log \sqrt{n - x^T x} - \frac{1}{m} \sum_{i=1}^{m} \log s_i$$

and

$$s_i = b_i - \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, \ldots, m$$

are the slacks in the constraints $Ax \leq b$. We replace (5.5) in turn by the following non-convex optimization problem

$$\begin{aligned} \min \quad & \phi(x) \\ \text{s.t.} \quad & x \in \mathcal{P}. \end{aligned} \tag{P_ϕ}$$

Assuming $\mathcal{I} \neq \phi$, a simple observation reveals that x^* is a global minimum of (P_ϕ) if and only if $x^* \in \mathcal{I}$. To see this, note that since $\phi(x) = \log\big((n - x^T x)^{1/2} / \prod_{i=1}^{m} (b_i - a_i^T x)^{1/m}\big)$, the denominator of the log term of $\phi(x)$ is the geometric mean of the slacks, and is maximized at the analytic center of the polytope $\mathcal{P}$, whereas the numerator is minimized when $x \in \mathcal{I}$, since $-e \leq x \leq e$, $\forall x \in \mathcal{P}$. Karmarkar (1990); Karmarkar et al. (1991) solve (P_ϕ) using an interior point method. To start with, we will assume a strictly interior point, i.e., $x^0 \in \mathcal{P}'$. The algorithm generates a sequence of points $\{x^k\}$ in $\mathcal{P}'$. In every iteration we perform the following steps:

(1) Minimize a quadratic approximation of the potential function over an inscribed ellipsoid in the feasible region $\mathcal{P}$ around the current feasible interior point, to get the next iterate.
(2) Round the new iterate to an integer solution.
If this solution is feasible the problem is solved,
else goto step 1.
(3) When a local minimum is found, modify the potential function to avoid running into this minimum again, and restart the process.

These steps will be elaborated in more detail in the subsequent sections.

2.1 Non-convex quadratic function minimization

We elaborate on Step 1 of the algorithm in this subsection. This step is an interior point algorithm to solve (P_ϕ). It mimics a trust region method, except that the trust region is based on making good global approximations to the polytope $\mathcal{P}$.

Given $x^k \in \mathcal{P}^0$, the next iterate x^{k+1} is obtained by moving in a descent direction Δx from x^k, i.e., a direction such that $\phi(x^k + \alpha\Delta x) < \phi(x^k)$, where α is an appropriate step length. The descent direction Δx is obtained by minimizing a quadratic approximation of the potential function about the current point x^k over the Dikin ellipsoid, which can be shown to be inscribed in the polytope $\mathcal{P}$. The resulting problem (P_r) solved in every iteration is the following:

$$\begin{array}{ll} \min & \frac{1}{2}(\Delta x)^T H(\Delta x) + h^T(\Delta x) \\ \text{s.t.} & (\Delta x)^T A^T S^{-2} A(\Delta x) \leq r^2, \end{array} \qquad (P_r)$$

for some $0 \leq r \leq 1$. Here $S = \text{Diag}(s)$ and H and h are the Hessian, and the gradient of the potential function $\phi(x)$, respectively.

The problem (P_r), a trust region subproblem for some $r_\ell \leq r \leq r_u$, is approximately solved by an iterative binary search algorithm (see Conn et al., 2000; Karmarkar, 1990; Vavasis, 1991), in which one solves a series of systems of linear equations of the form

$$(H + \mu A^T S^{-2} A)\Delta x = -h,$$

where $\mu > 0$ is a real scalar. This system arises from the first order KKT optimality condition for (P_r). Since there are two iterative schemes at work, we will refer to the iterations employed in solving (P_ϕ) as *outer* iterations, and the iterations employed in solving (P_r) as *inner* iterations. In this terminology, each outer iteration consists of a series of inner iterations. We concentrate on the outer iterations first. Assume for simplicity that (P_r) is solved exactly in every outer iteration for a solution Δx^*. Let us define the *S-norm* of Δx^* as

$$\|\Delta x^*\|_S = \sqrt{(\Delta x^*)^T A^T S^{-2} A(\Delta x^*)}.$$

Since H is indefinite, the solution to (P_r) is attained on the boundary of the Dikin ellipsoid, giving $r = \|\Delta x^*\|_S$. On the other hand, the computed direction Δx^* need not be a descent direction for $\phi(x)$, since the higher order terms are neglected in the quadratic approximation. Karmarkar et al. (1991) however show that a descent direction can always be computed provided the radius r of the Dikin ellipsoid is decreased sufficiently. In the actual algorithm, in each outer iteration we solve (P_r)

for a priori bound (r_ℓ, r_u) on r, and if the computed Δx^* is not a descent direction, we reduce r_u, and continue with the process. Moreover, we stop these outer iterations with the conclusion that a *local minimum* is attained for (P_ϕ) as soon as the upper bound r_u falls below a user specified tolerance $\epsilon > 0$.

We now discuss the inner iterations, where a descent direction Δx is computed for (P_r): assuming we are in the kth outer iteration we have as input the current iterate x^k, a multiplier μ, lower and upper bounds (r_ℓ, r_u) on r, a flag ID, which is false initially, and is set to true if during any inner iteration an indefinite matrix $(H + \mu A^T S^{-2} A)$ is encountered. The algorithm computes $\Delta x^*(\mu)$ by solving the following system of linear equations.

$$\Delta x^*(\mu) = -(H + \mu A^T S^{-2} A)h.$$

We are assuming that $\mu > 0$ is chosen so that $(H + \mu A^T S^{-2} A)$ is positive definite. If this is not true for the input μ, the value of μ is increased, and the flag ID is set to true. This process is repeated until we have a nonsingular coefficient matrix. Once $\Delta x^*(\mu)$ is computed, we compute $r^* = \|\Delta x^*(\mu)\|_S$. One of the following four cases can then occur:

(1) If $r^* \leq r_\ell$ and ID is false, an upper bound on μ has been found; set $\mu_{\text{upper}} = \mu$, and μ is decreased either by dividing it by a constant > 1, or if a lower bound μ_{lower} on μ already exists by taking the geometric mean of the current μ and μ_{lower}. The direction Δx is recomputed with this new value of μ.

(2) If $r^* \geq r_u$, a lower bound on μ has been found; set $\mu_{\text{lower}} = \mu$, and μ is increased, either by multiplying it with some constant > 1, or if μ_{upper} already exists, by taking the geometric mean of μ and μ_{upper}. The direction Δx is recomputed with this new value of μ.

(3) If $r^* \leq r_\ell$, and ID is true, decreasing μ will still lead to an indefinite matrix; in this case the lower bound r_ℓ is reduced, and the direction Δx is recomputed.

(4) Finally, if $r_\ell \leq r^* \leq r_u$, the direction Δx is accepted.

2.2 Rounding schemes and local minima

We discuss Steps 2 and 3 of the algorithm in this subsection. These include techniques to round the iterates to ± 1 vectors, and schemes to modify the potential function to avoid running into the same local minima more than once.

(1) **Rounding schemes:** In Step 2 of the algorithm, the current iterate x^k is rounded to a ± 1 solution $\bar{x}$. Generally these rounding techniques are specific to the combinatorial problem being solved, but two popular choices include:

(a) Round to the nearest ± 1 vertex, i.e.,

$$\bar{x}_i = 1 \qquad \text{if } x_i^k \geq 0;$$
$$\bar{x}_i = -1 \qquad \text{if } x_i^k < 0.$$

(b) We can obtain a starting point x^0 by solving a linear relaxation of the problem, using an IPM. The rounding can then be based on a coordinate-wise comparison of the current solution point with the starting point, i.e.,

$$\bar{x}_i = 1 \qquad \text{if } x_i^k \geq x_i^0;$$
$$\bar{x}_i = -1 \qquad \text{if } x_i^k < x_i^0.$$

(2) **Avoiding the same local minima:** After a number of iterations, the interior point algorithm may lead to a local minimum. One way to avoid running into the same local minimum twice is the following: Let $\bar{x}$ be the rounded ± 1 solution and suppose $\bar{x} \notin \mathcal{I}$. It can be easily seen that

$$\bar{x}^T y = n, \qquad \text{for } y = \bar{x}$$
$$\bar{x}^T y \leq n-2, \qquad \forall y \in \{y \in \mathbb{R}^n : y_i \in \{-1,1\}, y \neq \bar{x}\}.$$

Thus we can add the cut $\bar{x}^T y \leq n-2$ without cutting off any integer feasible solution. After adding the cut the process is restarted from the analytic center of the new polytope. Although there is no guarantee that we won't run into the same local minimum again, in practice, the addition of the new cut changes the potential function and alters the trajectory followed by the algorithm.

Warners et al. (1997a,b) consider the following improvement in the algorithm arising from the choice of a different potential function: For the potential function $\phi(x)$ discussed earlier in the section, the Hessian H at the point x^k is given by

$$\begin{aligned} H &= \nabla^2 \phi(x^k) \\ &= -\frac{1}{f_0} I - \frac{2}{f_0^2} x^k x^{k^T} + \frac{1}{n} A^T S^{-2} A. \end{aligned}$$

For a general x^k this results in a dense Hessian matrix, due to the outer product term $x^k x^{k^T}$. This increases the computational effort in obtaining Δx since we have now to deal with a dense coefficient matrix. The sparsity of A can be utilized by employing rank 1 updates. Instead, Warners et al. (1997a,b) introduce the potential function.

$$\phi_w(x) = (n - x^T x) - \sum_{i=1}^{m} w_i \log s_i,$$

where $w = (w_1, \ldots, w_m)^T$ is a nonnegative weight vector. In this case the Hessian $H_w = -2I + A^T S^{-1} W S^{-1} A$, where $W = \text{Diag}(w)$. Now H_w is a sparse matrix, whenever the product $A^T A$ is sparse, and this fact can be exploited to solve the resulting linear system more efficiently. The weights $w_i^k \to 0$ during the course of the algorithm. Thus, initially when $w^k > 0$, the iterates x^k avoid the boundary of the feasible region, but subsequently towards optimality, these iterates approach the boundary, as any ± 1 feasible vector is at the boundary of the feasible region.

The technique has been applied to a variety of problems including satisfiability (Kamath et al., 1990), set covering (Karmarkar et al., 1991), inductive inference (Kamath et al., 1992), and variants of the frequency assignment problem (Warners et al., 1997a,b)).

3. Interior point cutting plane methods

In this section we consider interior point cutting plane algorithms, especially the analytic center cutting plane method (ACCPM) (Goffin and Vial, 2002; Ye, 1997) and the volumetric center method (Vaidya, 1996; Anstreicher, 1997, 2000, 1999). These techniques are originally designed for convex feasibility or optimization problems. To see how this relates to combinatorial optimization, consider the maxcut problem discussed in Section 1. The maxcut problem can be expressed as the following $\{-1, 1\}$ integer programming problem.

$$\begin{aligned} &\max \ w^T x \\ &\text{s.t.} \quad x(C \backslash F) - x(F) \leq |C| - 2 \quad \forall \text{ circuits } C \subseteq E \text{ and} \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{all } F \subseteq C \text{ with } |F| \text{ odd}, \\ &\qquad x \in \{-1, 1\}^m. \end{aligned} \tag{5.6}$$

Here w_{ij} represents the weight of edge $\{i, j\} \in E$. Let CHULL(G) represent the convex hull of the feasible set of (5.6). We can equivalently minimize the linear functional $w^T x$ over CHULL(G), i.e., we have replaced the maxcut problem via an equivalent convex optimization problem. Unfortunately, an exact description of CHULL(G) is unknown, and besides this may entail an exponential set of linear constraints. However, we can solve such problems by using interior point cutting plane methods discussed in this section.

Although we are primarily interested in optimization, we motivate these cutting plane methods via the convex feasibility problem; we will later consider extensions to optimization. Let $\mathcal{C} \subseteq \mathbb{R}^m$ be a convex set. We want to find a point $y \in \mathcal{C}$. We will assume that if the set $\mathcal{C}$ is nonempty then it contains a ball of radius ϵ for some tolerance $\epsilon > 0$. Further, we assume that $\mathcal{C}$ is in turn contained in the m dimensional

unit hypercube given by $\{y \in \mathbb{R}^m : 0 \leq y \leq e\}$, where e is the all ones vector. We also define $L = \log(1/\epsilon)$.

Since each convex set is the intersection of a (possibly infinite) collection of halfspaces, the convex feasibility problem is equivalent to the following (possibly semi-infinite) linear programming problem.

$$\text{Find } y \text{ satisfying } A^T y \leq c,$$

where A is a $m \times n$ matrix with independent rows, and $c \in \mathbb{R}^n$. As discussed earlier, the value of n could be infinite. We assume we have access to a *separation oracle*. Given $\bar{y} \in \mathbb{R}^m$, the oracle either reports that $\bar{y} \in \mathcal{C}$, or it will return a *separating hyperplane* $a \in \mathbb{R}^m$ such that $a^T y \leq a^T \bar{y}$ for every $y \in \mathcal{C}$. Such a hyperplane which passes through the query point $\bar{y} \notin \mathcal{C}$ will henceforth be referred to as a *central* cut. A weakened version of this cutting plane, hereafter referred to as a *shallow* cut, is $a^T y \leq a^T \bar{y} + \beta$, for some $\beta > 0$. It is interesting to note that the convex sets that have polynomial separation oracles are also those that have self-concordant barrier functionals whose gradient and Hessian are easily computable; the latter fact enables one to alternatively apply IPMs for solving optimization problems over such convex sets.

Generic cutting plane algorithm.

Input. Let $\mathcal{P} \supseteq \mathcal{C}$ be a computable convex set.

(1) Choose $\bar{y} \in \mathcal{P} \subseteq \mathbb{R}^m$.
(2) Present $\bar{y}$ to the separation oracle.
(3) **If** $\bar{y} \in \mathcal{C}$ we have solved the convex feasibility problem.
(4) **Else** use the constraint returned by the separation oracle to update $P = P \cup \{y : a^T y \leq a^T \bar{y}\}$ and **goto** step 2.

We illustrate the concept of an oracle for the maxcut problem. The maxcut polytope CHULL(G) does not admit a polynomial time separation oracle, but this is true for polytopes obtained from some of its faces. One such family of faces are the odd cycle inequalities; these are the linear constraints in (5.6). These inequalities form a polytope called the *metric* polytope. Barahona and Mahjoub (1986) describe a polynomial time separation oracle for this polytope, that involves the solution of n shortest path problems on an auxiliary graph with twice the number of nodes, and four times the number of edges.

The cutting plane approach to the feasibility problem can be extended to convex optimization problems by cutting on a violated constraint when the trial point is infeasible, and cutting on the objective function when the trial point is feasible but not optimal.

Interior point cutting plane methods set up a series of convex relaxations of $\mathcal{C}$, and utilize the analytic and volumetric centers of these convex sets as test points $\bar{y}$, that are computed in polynomial time by using IPMs. The relaxations are refined at each iteration by the addition of cutting planes returned by the oracle; some cuts may even conceivably be dropped. We will assume that each call to the oracle takes unit time.

We discuss the analytic center cutting plane method in Section 3.1, and the volumetric center method in Section 3.2.

3.1 Analytic center cutting plane methods

A good overview on ACCPM appears in the survey paper by Goffin and Vial (2002), and the book by Ye (1997). The complexity analysis first appeared in Goffin et al. (1996). The algorithm was extended to handle multiple cuts in Goffin and Vial (2000), and nonlinear cuts in Mokhtarian and Goffin (1998), Luo and Sun (1998), Sun et al. (2002), Toh et al. (2002), and Oskoorouchi and Goffin (2003a,b). The method has been applied to a variety of practical problems including stochastic programming (Bahn et al., 1997), multicommodity network flow problems (Goffin et al., 1997). A version of the ACCPM software (Gondzio et al., 1996) is publicly available. Finally, ACCPM has also appeared recently within a branch-and-price algorithm in Elhedhli and Goffin (2004).

Our exposition in this section closely follows Goffin and Vial (2002) and Goffin et al. (1996). We confine our discussion to the convex feasibility problem discussed earlier.

For the ease of exposition, we will assume the method approximates $\mathcal{C}$ via a series of increasingly refined polytopes $\mathcal{F}_D = \{y : A^T y \leq c\}$. Here A is an $m \times n$ matrix, $c \in \mathbb{R}^n$, and $y \in \mathbb{R}^m$. We will assume that A has full row rank, and $\mathcal{F}_D$ is bounded with a nonempty interior. The vector of slack variables $s = c - A^T y \in \mathbb{R}^n$, $\forall y \in \mathcal{F}_D$.

The analytic center of $\mathcal{F}_D$ is the unique solution to the following minimization problem.

$$\begin{aligned} \min \quad & \phi_D(s) = -\sum_{i=1}^{n} \log s_i \\ \text{s.t.} \quad & A^T y + s = c, \\ & s > 0. \end{aligned}$$

If we introduce the notion that $F(y) = \phi_D(c - A^T y)$, then the analytic center y^* of $\mathcal{F}_D$ is the minimizer of $F(y)$.

Assuming that $\mathcal{C} \subseteq \{y : 0 \leq y \leq e\}$, the complete algorithm is the following:

Analytic center cutting plane method.

Input. Let $\mathcal{F}_D^0 = \{y : 0 \leq y \leq e\}$, and $F_0(y) = -\sum_{i=1}^m \log(y_i(1 - y_i))$. Set $y^0 = e/2$.

(1) Compute y^k an approximate minimizer of $F_k(y)$.

(2) Present y^k to the oracle.

If $y^k \in \mathcal{C}$ **then** stop,

else the oracle returns the separating hyperplane with normal a^k passing through y^k. Update

$$\mathcal{F}_D^{k+1} = \mathcal{F}_D^k \cap \{y : (a^k)^T y \leq (a^k)^T y^k\},$$
$$F_{k+1}(y) = F_k(y) - \log((a^k)^T(y^k - y)).$$

Set $k = k + 1$ and **goto** step 1.

The formal proof of convergence of the algorithm is carried out in three steps. We will assume that the algorithm works with exact analytic centers.

- One first shows that a new analytic center can be found quickly after the addition of cuts. This is done in an iterative fashion using damped Newton steps, that are the inner iterations in the algorithm. Goffin and Vial (2002) show that an analytic center can be found in $O(1)$ iterations when one central cut is added in each iteration. In Goffin and Vial (2000), they also show that it is possible to add p cuts simultaneously, and recover a new analytic center in $O(p \log(p))$ Newton iterations.
- One then proceeds to establish bounds on the logarithmic barrier function $F(y)$. Let $\bar{y}^k$ be the exact analytic center of the polytope $\mathcal{F}_D^k$, i.e., the minimizer of

$$F_k(y) = -\sum_{i=1}^{2m+k} \log(c^k - (A^k)^T \bar{y}^k)_i.$$

We now establish upper, and lower bounds on $F_k(\bar{y}^k)$. If we are not done in the kth iteration, the polytope $\mathcal{F}_D^k$ still contains a ball of radius ϵ. If $\bar{y}$ is the center of this ball, then we have $\bar{s} = c - \mathcal{A}^T \bar{y} \geq \epsilon e$, giving

$$\begin{aligned} F_k(\bar{y}^k) &\leq F_k(\bar{y}) \\ &\leq (2m + k) \log\left(\frac{1}{\epsilon}\right). \end{aligned} \tag{5.7}$$

This is an upper bound on $F_k(\bar{y}^k)$. We can also obtain a lower bound on $F_k(\bar{y}^k)$ in the following manner. We only outline the

main steps, more details can be found in Goffin et al. (1996). Let H_i denote the Hessian of $F(y)$ evaluated at $\bar{y}^i$. We first obtain the bound

$$F_k(\bar{y}^k) \geq -\frac{k}{2}\log\left(\frac{1}{k}\sum_{i=1}^{k}(a_i^T H_{i-1}^{-1} a_i)\right) + 2m\log\left(\frac{1}{2}\right) \tag{5.8}$$

by exploiting the following self-concordance property of $F_j(y)$

$$F_j(y) \geq F_j(\bar{y}^j) + \sqrt{(y-\bar{y}^j)^T H_j (y-\bar{y}^j)} - \log\left(1+\sqrt{(y-\bar{y}^j)^T H_j (y-\bar{y}^j)}\right),$$

and applying this property recursively on $F_k(y)$. The bound is simplified in turn by bounding the Hessian H_i from below by a certain matrix, which is simpler to analyze. This yields the following upper bound on $\sum_{i=1}^{k} a_i^T H_{i-1}^{-1} a_i$

$$\sum_{i=1}^{k} a_i^T H_{i-1}^{-1} a_i \leq 2m^2 \log\left(1+\frac{k}{m^2}\right),$$

that is employed in the complexity analysis. Substituting this relation in (5.8) and simplifying the resulting formulas we have

$$F_k(\bar{y}^k) \geq -k\log(\sqrt{2}) + k\log\left(\frac{k/m^2}{\log(1+k/m^2)}\right) - 2m\log\left(\frac{1}{2}\right). \tag{5.9}$$

- A comparison of the two bounds (5.7) and (5.9) on $F_k(\bar{y}^k)$ yields the following upper bound on the number of outer iterations

$$k\log\left(\frac{k/m^2}{\log(1+k/m^2)}\right) \leq (2m+k)\log\left(\frac{1}{\epsilon}\right) + k\log(\sqrt{2}) + 2m\log\left(\frac{1}{2}\right). \tag{5.10}$$

This provides the proof of global convergence of the algorithm. It is clear from (5.10) that the algorithm terminates in a finite number of iterations, since the ratio $(k/m^2)/\log(1+k/m^2)$ tends to infinity as k approaches infinity, i.e., the left hand side grows superlinearly in k. Neglecting the logarithmic terms, an upper bound on the number of outer iterations is given by $O^*(m^2/\epsilon^2)$ (the notation O^* means that logarithmic terms are ignored).

The analysis presented above can be extended to approximate analytic centers (see Goffin et al., 1996) to yield a *fully* polynomial time algorithm for the convex feasibility problem. The ACCPM algorithm is not polynomial, since the complexity is polynomial in $1/\epsilon$ not $\log(1/\epsilon)$. There is a variant of ACCPM due to Atkinson and Vaidya (1995) (see also Mitchell (2003) for an easier exposition) which is polynomial with a complexity bound of $O(mL^2)$ calls to the oracle, but the algorithm requires dropping constraints from time to time, and also weakening the cuts returned by the oracle making them shallow. In the next section, we will discuss the volumetric center method which is a polynomial interior point cutting plane method, with a better complexity bound than ACCPM for the convex feasibility problem.

3.2 Volumetric center method

The volumetric center method is originally due to Vaidya (1996), with enhancements and subsequent improvements in Anstreicher (1997, 1999, 2000) and Mitchell and Ramaswamy (2000).

The complexity of the volumetric center algorithm is $O(mL)$ calls to the oracle, and either $O(mL)$ or $O(m^{1.5}L)$ approximate Newton steps depending on whether the cuts are shallow or central. The complexity of $O(mL)$ calls to the separation oracle is optimal — see Nemirovskii and Yudin (1983).

As in Section 3, we approximate the convex set $\mathcal{C}$ by the polytope $\mathcal{F}_\mathcal{D}(y) = \{y \in \mathbb{R}^m : A^T y \leq c\} \supseteq \mathcal{C}$, where A is an $m \times n$ matrix, and c is an n dimensional vector. Let y be a strictly feasible point in $\mathcal{F}_\mathcal{D}$, and let $s = c - A^T y > 0$. The volumetric barrier function for $\mathcal{F}_\mathcal{D}$ at the point y is defined as

$$V(y) = \frac{1}{2}\log\big(\det(AS^{-2}A^T)\big). \tag{5.11}$$

The volumetric center $\hat{y}$ of $F_D(y)$ is the point that minimizes $V(y)$. The volumetric center can also be defined as the point y chosen to maximize the volume of the inscribed Dikin ellipsoid $\{z \in \mathbb{R}^m : (z - y)^T(AS^{-2}A^T)(z - y) \leq 1\}$ centered at y.

The volumetric center is closely related to the analytic center of the polytope discussed in Section 3.1. It is closer to the geometrical center of the polytope, than the analytic center.

We also define variational quantities (Atkinson and Vaidya, 1995) for the constraints $A^T y \leq c$ as follows:

$$\sigma_j = \frac{a_j^T(AS^{-2}A^T)^{-1}a_j}{s_j^2}, \quad j = 1, \ldots, n.$$

These quantities give an indication of the relative importance of the inequality $a_j^T y \leq c_j$. The larger the value of σ_j, the more important the inequality. A nice interpretation of these quantities appears in Mitchell (2003). The variational quantities are used in the algorithm to drop constraints that are not important.

We present the complete algorithm below.

Volumetric center IPM.

Input. Given $F_D^0(y) = \{y \in \mathbb{R}^m : 0 \leq y \leq e\}$ with $C \subseteq F_D^0(y)$ and $n = 2m$ be the total number of constraints. Set $y^0 = e/2$, and let $0 < \epsilon < 1$ be the desired tolerance.

(1) **If** $V(y^k)$ is sufficiently large then stop with the conclusion that $\mathcal{C}$ is empty.
Else goto step 2.

(2) Compute σ_i for each constraint.
If $\sigma_{\bar{i}} = \min_{i=2m+1,\ldots,n} \sigma_i > \epsilon$ **goto** step 4,
else goto step 3.

(3) Call the oracle at the current point y^k.
If $y^k \in \mathcal{C}$ then stop,
else the oracle returns a separating hyperplane with normal a^k passing through y^k.
Update $F_D^{k+1} = F_D^k \cap \{y : (a^k)^T y \leq (a^k)^T y^k\}$, $n = n + 1$, and **goto** step 5.

(4) Drop the $\bar{i}$th constraint from the current feasible set, i.e., $F_D^{k+1} = F_D^k \backslash \{y : a_{\bar{i}}^T y \leq c_{\bar{i}}\}$, update $n = n - 1$, and **goto** step 5.

(5) Take a series of damped Newton steps to find a new approximate volumetric center. Set $k = k + 1$ and **goto** step 1.

We note that the box constraints $0 \leq y \leq e$ defining the initial polyhedral approximation are never dropped, and hence the polyhedral approximations have at least $2m$ constraints. In every iteration we either add or drop a constraint. It follows that in the kth iteration, the algorithm must have previously visited Step 4 where we add a constraint at least $k/2$ times, and Step 5 where we drop a constraint on no more than $k/2$ occasions. Else, the number of constraints would fall below $2m$. The formal proof of convergence of the algorithm proceeds in the following way:

- First, one shows that the number of Newton iterations in one call to Step 6 of the algorithm to find an approximate volumetric center is bounded. These are the inner iterations in the algorithm. The condition for a point to be an approximate volumetric center can be expressed as a condition on the norm of the gradient of the

volumetric barrier function in the norm given by an approximation to the Hessian of the volumetric barrier function. Formally, a point y is an approximate volumetric center if

$$\beta \|g(y)\|_{P(y)^{-1}} \leq \gamma, \tag{5.12}$$

for some appropriate $\gamma \leq \frac{1}{6}$, where

$$\beta = \min\left\{(2\sqrt{\sigma_{\bar{i}}} - \sigma_{\bar{i}})^{-1/2}, \sqrt{\frac{1+\sqrt{m}}{2}}\right\},$$

$g(y)$, and $P(y)$ are the gradient and an approximation to the Hessian of the volumetric barrier function $V(y)$ at the point y, respectively. In Step 6 one take a series of damped Newton steps of the form $\bar{y} = y + \alpha d$, where $P(y)d = -g(y)$. Anstreicher (1999) shows that when a central cut is added in Step 4, then an approximate volumetric center satisfying (5.12) could be recovered in $O(\sqrt{m})$ Newton steps. In this case, the direction first proposed in Mitchell and Todd (1992) is used to move away from the added cut, and the damped Newton iterations described above are used to recover an analytic center. On the other hand, when a cut is dropped in Step 5, Vaidya (1996) showed that an approximate volumetric center could be obtained in just one Newton iteration. In the original volumetric barrier (Vaidya, 1996), Vaidya weakened the cuts returned by the oracle (shallow cuts), and showed that a new approximate volumetric center could be obtained in $O(1)$ Newton steps (these are the number of Newton steps taken to recover an approximate analytic center in ACCPM with central cuts).

- The global convergence of the algorithm is established by showing that eventually the volumetric barrier function becomes too large for the feasible region to contain a ball of radius ϵ. This establishes an upper bound on the number of iterations required. For ease of exposition we shall assume that we are dealing with the exact volumetric center of the polyhedral approximation in every iteration. In reality this is not possible, however the analysis can be extended to include approximate volumetric centers. For example, Anstreicher (1997, 1999) shows that if the current polyhedral approximation F_D of $\mathcal{C}$ has n constraints, then if the value of the barrier functional at the volumetric center y of F_D is greater than $V_{max} = mL + m \log n$, then the volume of $\mathcal{C}$ is smaller than that of an m dimensional sphere of radius ϵ. He then establishes that the increase in the barrier function, when a constraint is added, is at least ΔV^+, and also the decrease is no more than ΔV^-, for

constants ΔV^+ and ΔV^- satisfying $0 < \Delta V^- < \Delta V^+$, and where $\Delta V = \Delta V^+ - \Delta V^- > 0$ is $O(1)$. Thus, we can bound the increase in the value of the volumetric barrier functional in the kth iteration as follows:

$$
\begin{aligned}
& V(y^k) - V(y^0) \\
& \quad \geq (\text{no of constraints added and still in relaxation}) \times \Delta V^+ \\
& \qquad (\text{no of constraints added and subsequently dropped}) \times \Delta V \\
& \quad \geq \Delta V \times (\text{total no of constraints added}) \\
& \quad \geq \frac{k \times \Delta V}{2},
\end{aligned}
$$

where the last inequality follows from the fact that the algorithm must have visited the separation oracle in Step 4 previously at least on $k/2$ occasions. Combining this with the maximum value $V_{\max}$, gives the complexity estimate that the volumetric center cutting plane algorithm either finds a feasible point in $\mathcal{C}$, or proves that it is empty in $O(mL)$ calls to the oracle, and $O(m^{1.5}L)$ Newton steps. The actual results in Anstreicher (1997) deal with approximate volumetric centers. The number of Newton steps can be brought down to $O(mL)$ if shallow cuts are employed as in Vaidya (1996).

The overall complexity of the volumetric center method is $O(mLT + m^{4.5}L)$ arithmetic operations, where T is the complexity of the oracle, for central cuts, and $O(mLT + m^4L)$ for shallow cuts. The ellipsoid method (see Grötschel et al., 1993) on the other hand takes $O(m^2LT + m^4L)$ arithmetic operations to solve the convex feasibility problem. Although the original algorithm due to Vaidya (1996) had the best complexity, it was not practical since the constants involved in the complexity analysis were very large, of the order of 10^7. The algorithm was substantially refined in Anstreicher (1997, 1999) significantly bringing down the maximum number of constraints required in the polyhedral approximation to $25n$ in Anstreicher (1999). Also, since the algorithm employs central cuts the number of Newton steps required in Step 6 is $O(\sqrt{m})$, which is significantly more than the $O(1)$ steps employed in the ACCPM algorithm in Section 3.1; whether this can be achieved for the volumetric center method is still an open question. Finally, we must mention that the computational aspects of the volumetric center method have not yet been entirely tested.

4. Complexity and IPMs for SDP

We consider the complexity of SDP in Section 4.1, and a generic interior point method (IPM) for solving the SDP, together with issues involved in an efficient implementation is presented in Section 4.2. This algorithm is employed in solving the SDP relaxations of combinatorial problems as discussed in the subsequent sections. Our exposition in this section is sketchy, and for details we refer the interested reader to the excellent surveys by De Klerk (2002), Todd (2001), Monteiro (2003), the habilitation thesis of Helmberg (2000a), and the Ph.D. dissertation of Sturm (1997).

Consider the semidefinite programming problem

$$\begin{array}{ll} \min & C \bullet X \\ \text{s.t.} & \mathcal{A}(X) = b, \\ & X \succeq 0, \end{array} \tag{SDP}$$

with dual

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & \mathcal{A}^T y + S = C, \\ & S \succeq 0, \end{array} \tag{SDD}$$

where the variables $X, S \in \mathcal{S}^n$ the space of real symmetric $n \times n$ matrices, $b \in \mathbb{R}^m$. Also $C \bullet X = \sum_{i,j=1}^n C_{ij} X_{ij}$ is the Frobenius inner product of matrices in $\mathcal{S}^n$. The linear operator $\mathcal{A}\colon \mathcal{S}^n \to \mathbb{R}^m$, and its adjoint $\mathcal{A}^T\colon \mathbb{R}^m \to \mathcal{S}^n$ are:

$$\mathcal{A}(X) = \begin{bmatrix} A_1 \bullet X \\ \vdots \\ A_m \bullet X \end{bmatrix} \quad \text{and} \quad \mathcal{A}^T y = \sum_{i=1}^m y_i A_i,$$

where the matrices $A_i \in \mathcal{S}^n$, $i = 1, \ldots, m$, and $C \in \mathcal{S}^n$ are the given problem parameters. The constraints $X \succeq 0$, $S \succeq 0$ are the only nonlinear (actually convex) constraints in the problem requiring that these matrices X and S are symmetric positive semi-definite matrices. We will hereafter assume that the matrices A_i, $i = 1, \ldots, m$ are linearly independent, that implies $m \leq \binom{n+1}{2}$.

If both the primal (SDP) and the dual (SDD) problems have strictly feasible (Slater) points, then both problems attain their optimal solutions, and the duality gap $X \bullet S = 0$ is zero at optimality. Most SDPs arising in combinatorial optimization satisfy this assumption. For more on strong duality we refer the reader to Ramana et al. (1997), and De Klerk et al. (1998) who discuss how to detect all cases that occur in SDP.

4.1 The complexity of SDP

In this section, we briefly review the complexity of SDP. Most results mentioned here can be found in the book by Grötschel et al. (1993), the Ph.D. thesis of Ramana (1993), the review by Ramana & Pardalos in the IPM handbook edited by Terlaky (1996), Krishnan and Mitchell (2003a), and Porkoláb and Khachiyan (1997).

We will assume that the feasible region of the SDP is contained in a ball of radius $R > 0$. The ellipsoid algorithm (see Theorem 3.2.1 in Grötschel et al., 1993) can find a solution X^* to this problem such that $|C \bullet X^* - \mathrm{OPT}| \leq \epsilon$ (OPT is the optimal objective value), in a number of arithmetic operations that is polynomial in m, n, $\log R$, and $\log(1/\epsilon)$ in the bit model. In Krishnan and Mitchell (2003a), for the particular choice of $R = 1/\epsilon$, it is shown that the ellipsoid method, together with an oracle that computes the eigenvector corresponding to the most negative eigenvalue of S during the course of the algorithm, takes $O\big((m^2n^3 + m^3n^2 + m^4)\log(1/\epsilon)\big)$ arithmetic operations. We can employ the volumetric barrier algorithm, discussed in Section 3, to improve this complexity. In Krishnan and Mitchell (2003a) it is shown that such an algorithm, together with the oracle mentioned above, takes $O\big((mn^3 + m^2n^2 + m^4)\log(1/\epsilon)\big)$ arithmetic operations. This is also slightly better than the complexity of primal-dual interior point methods to be discussed in Section 4.2, when there is no structure in the underlying SDP.

On the other hand, no polynomial bound has been established for the bit lengths of the intermediate numbers occurring in interior point methods solving an SDP (see Ramana & Pardalos in Terlaky, 1996). Thus, strictly speaking, these methods for SDP are not polynomial in the bit model.

We now address the issue of computing an exact optimal solution of an arbitrary SDP, when the problem data is rational. Rigorously speaking, this is not a meaningful question since the following pathological cases can occur for a feasible rational semidefinite inequality, that cannot occur in the LP case.

(1) It only has irrational solutions.
(2) All the rational solutions have exponential bitlength.

As a result, the solution may not be representable in polynomial size in the bit length model. However we can still consider the following semidefinite feasibility problem (SDFP).

DEFINITION 5.1 Given rational symmetric matrices $A_0, \ldots, A_m$ determine if the semidefinite system

$$\sum_{i=1}^{m} x_i A_i \preceq A_0$$

is feasible for some real $x \in \mathbb{R}^m$.

Ramana (1997) established that SDFP cannot be an NP-complete problem, unless NP = co-NP. In fact, Porkoláb and Khachiyan (1997) have shown that SDFP can actually be solved in polynomial time, if either m or n is a fixed constant. The complexity of SDFP remains one of the unsolved problems in SDP.

4.2 Interior Point Methods for SDP

In this section we consider primal-dual IPMs for SDP. These are in fact extensions of the generic IPM for LP discussed in Section 1.

The optimality conditions for the SDP problem (compare with (5.1) for LP in Section 1) include the following:

$$\begin{aligned} \mathcal{A}(X) &= b, & X &\succeq 0, \\ \mathcal{A}^T y + S &= C, & S &\succeq 0, \\ XS &= 0. \end{aligned} \tag{5.13}$$

The first two conditions represent primal and dual feasibility while the third condition gives the complementary slackness condition. Consider perturbing the complementary slackness conditions to $XS = \mu I$ for some $\mu > 0$. Ignoring the inequality constraints $X, S \succeq 0$ for the moment this gives the following system:

$$\begin{aligned} \mathcal{A}(X) &= b, \\ \mathcal{A}^T y + S &= C, \\ XS &= \mu I. \end{aligned} \tag{5.14}$$

We denote the solution to (5.14) for some fixed $\mu > 0$ by (X_μ, y_μ, S_μ). The set $\{(X_\mu, y_\mu, S_\mu)\}$ forms the *central path* that is a smooth analytical curve converging to an optimal solution (X^*, y^*, S^*), as $\mu \to 0$.

If we solve (5.14) by Newton's method, we get the following linearized system

$$\begin{aligned} \mathcal{A}\Delta X &= 0, \\ \mathcal{A}^T \Delta y + \Delta S &= 0, \\ \Delta X S + X \Delta S &= \mu I - XS. \end{aligned} \tag{5.15}$$

Since X and S are matrices, they do not always commute i.e., $XS \neq SX$. In fact, we have $m + n^2 + n(n+1)/2$ equations, but only $m + n(n+1)$ unknowns in (5.15), which constitutes an overdetermined system of linear equations. This is different from the LP case in Section 1, where X and S are diagonal matrices and hence commute. As a result, the solution ΔX may not be symmetric, and $X + \Delta X$ is not in the cone of symmetric positive semidefinite matrices $\mathcal{S}_+^n$. To ensure the symmetry of ΔX, Zhang (1998) introduces the symmetrization operator

$$H_P(M) = \tfrac{1}{2}(PMP^{-1} + (PMP^{-1})^T), \tag{5.16}$$

where P is a given nonsingular matrix, and uses this to symmetrize the linearized complementary slackness conditions, i.e., we replace the last equation in (5.15) by

$$H_P(\Delta XS + X\Delta S + XS) = \mu I. \tag{5.17}$$

A family of directions arises for various choices of P, that vary with regard to their theoretical properties, and practical efficiency, and it is still unclear which is the best direction in the primal-dual class. The Nesterov and Todd (1998) (NT) direction has the most appealing theoretical properties, and is shown to arise for a particular choice of $P = (X^{-1/2}(X^{1/2}SX^{1/2})^{-1/2}X^{1/2}S)^{1/2}$ in Todd et al. (1998). On the other hand, the H..K..M direction (proposed independently in Helmberg et al., 1996, Kojima et al., 1997, and Monteiro, 1997) is very efficient in practice (see Tütüncü et al., 2003), and also requires the least number of arithmetic operations per iteration. It arises for $P = S^{1/2}$, and a nice justification for this choice appears in Zhang (1998). However, since the NT direction employs a primal-dual scaling in P as opposed to a dual scaling in H..K..M, it is more efficient in solving difficult SDP problems. The H..K..M direction is also obtained in Helmberg et al. (1996) by solving the Newton system (5.15) for ΔX, and then symmetrizing ΔX by replacing it with $\frac{1}{2}(\Delta X + \Delta X^T)$. Finally, a good survey of various search directions appears in Todd (1999). As in IPMs for LP in Section 1, we need to take damped Newton steps. Similarly we introduce a proximity measure $\delta(X, S, \mu)$ that measures the proximity of (X, y, S) to (X_μ, y_μ, S_μ) on the central path. We present the generic IPM for SDP. For simplicity, we shall consider the H..K..M direction using the original interpretation of Helmberg et al. (1996).

Generic primal-dual IPM for SDP.

Input. $\mathcal{A}, b, C$, a feasible starting point (X^0, y^0, S^0) also satisfying the interior point condition, i.e., $X^0 \succ 0$, $S^0 \succ 0$, $\mathcal{A}(X^0) = b$, and

$\mathcal{A}^T y^0 + S^0 = C$. Further, we may assume without loss of generality that $X^0 S^0 = I$. Other parameters include a barrier parameter $\mu = 1$, a proximity threshold $\tau > 0$ such that $\delta(X^0, S^0, \mu) \leq \tau$, and an accuracy parameter $\epsilon > 0$.

(1) Reduce the barrier parameter μ.
(2) **If** $\delta(X, S, \mu) > \tau$ compute $(\Delta X, \Delta y, \Delta S)$ from (5.15) and replacing ΔX by $\frac{1}{2}(\Delta X + \Delta X^T)$.
(3) Choose some $\alpha \in (0, 1]$ so that $(X + \alpha \Delta X), (S + \alpha \Delta S) \succ 0$, and proximity $\delta(X, S, \mu)$ is suitably reduced.
(4) Set $(X, y, S) = (X + \alpha \Delta X, y + \alpha \Delta y, S + \alpha \Delta S)$.
(5) If $X \bullet S \leq \epsilon$ then **stop**,
else if $\delta(X, y, \mu) \leq \tau$ **goto** step 1,
else goto step 2.

One can solve an SDP with rational data to within a tolerance ϵ in $O\big(\sqrt{n}\log(1/\epsilon)\big)$ feasible iterations (see Todd, 2001, for more details). This is the best iteration complexity bound for SDP. Interestingly, this is the same bound as in the LP case.

We now examine the work involved in each iteration. The main computational task in each iteration is in solving the following normal system of linear equations.

$$\mathcal{A}(X\mathcal{A}^T(\Delta y)S^{-1}) \;=\; b \tag{5.18}$$

This system results from eliminating ΔS, and ΔX from (5.15). Let $M\colon \mathbb{R}^m \to \mathbb{R}^m$ be the linear operator given by $My = \mathcal{A}(X\mathcal{A}^T(y)S^{-1})$. The ith row of $M\Delta y$ is given by

$$A_i \bullet X\mathcal{A}^T(\Delta y)S^{-1} = \sum_{j=1}^{m} \Delta y_j \,\mathrm{Trace}(XA_iS^{-1}A_j).$$

Each entry of the matrix M thus has the form $M_{ij} = \mathrm{Trace}(XA_iS^{-1}A_j)$. This matrix is symmetric and positive definite, if we assume matrices A_i, $i = 1, \ldots, m$ are linearly independent in $\mathcal{S}^n$.

Solving for Δy requires $m^3/3$ flops, when the Cholesky decomposition is used. Moreover, M has be to recomputed in each iteration. An efficient way to build one row of M is the following

(1) Compute XA_iS^{-1} once in $O(n^3)$ time;
(2) Determine the m single elements via $XA_iS^{-1} \bullet A_j$ in $O(mn^2)$ arithmetic operations.

In total the construction of M requires $O(mn^3 + m^2n^2)$ arithmetic operations, and this is the most expensive operation in each iteration. On the whole, an interior point method requires $O\big(m(n^3 + mn^2 + m^2)\sqrt{n}\log(1/\epsilon)\big)$

arithmetic operations. For most of the combinatorial problems such as maxcut, the constraint matrices A_i have a rank one structure, and this reduces the computation of M to $O(mn^2 + m^2n)$ operations.

Excellent software based on primal-dual IPMs for SDP include CSDP by Borchers (1999), SeDuMi by Sturm (1999), and SDPT3 by Tütüncü et al. (2003). An independent benchmarking of various SDP software appears in Mittleman (2003).

In many applications the constraint matrices A_i have a special structure. The dual slack matrix S inherits this sparsity structure, while the primal matrix X is usually dense regardless of the sparsity. Benson et al. (2000) proposed a dual scaling algorithm that exploits the sparsity in the dual slack matrix. Also, Fukuda et al. (2000) and Nakata et al. (2003) employ ideas from the completion of positive semidefinite matrices (Grone et al., 1984; Laurent, 1998) to deal with dense X in a primal-dual IPM for SDP. Burer (2003) on the other hand utilizes these ideas to develop a primal-dual IPM entirely within the space of partial positive semidefinite matrices.

However, in most approaches, the matrix M is dense, and the necessity to store and factorize this dense matrix M limits the applicability of IPMs to problems with around 3000 constraints on a well equipped work station.

One way to overcome the problem of having to store the matrix M via the use of an iterative scheme, which only accesses this matrix through matrix vector multiplications, is discussed in Toh and Kojima (2002). This approach is not entirely straightforward since the Schur matrix M becomes increasingly ill-conditioned as the iterates approach the boundary. Hence, there is a need for good pre-conditioners for the iterative method to converge quickly. Recently, Toh (2003) has reported excellent computational results with a choice of a good preconditioner in solving the normal system of linear equations.

5. First order techniques for SDP

Interior point methods discussed in Section 4.2 are fairly limited in the size of problems they can handle. We discuss various first order techniques with a view of solving large scale SDPs in this section. As opposed to primal-dual interior point methods, these methods are mostly dual-only, and in some cases primal methods. These methods exploit the structure prevalent in combinatorial optimization problems; they are applicable in solving only certain classes of SDPs. Unlike IPMs there is no proof of polynomial complexity, and moreover these methods are not recommended for those problems, where a high accuracy is desired. Never-

theless excellent computational results have been reported for problems that are inaccessible to IPMs due to demand for computer time and storage requirements. A nice overview of such methods appears in the recent survey by Monteiro (2003). In this section, we will focus on the first order techniques which are very efficient in practice.

The first method is the spectral bundle method due to Helmberg and Rendl (2000). The method is suitable for large m, and recent computational results are reported in Helmberg (2003). The method is first order, but a second order variant which converges globally and which enjoys asymptotically a quadratic rate of convergence was recently developed by Oustry (2000).

The spectral bundle method works with the dual problem (SDD). Under an additional assumption that $\text{Trace}(X) = \beta$, for some constant $\beta \geq 0$, for all X in the primal feasible set, the method rewrites (SDD) as the following eigenvalue optimization problem.

$$\max \ \beta\lambda_{\min}(C - \mathcal{A}^T\mathrm{y}) + b^T\mathrm{y}, \tag{5.19}$$

where $\lambda_{\min}(S)$ denotes the smallest eigenvalue of S. Problem (5.19) is a concave non-smooth optimization problem, that is conveniently tackled by bundle methods for non-differentiable optimization. In the spectral bundle scheme the maximum eigenvalue is approximated by means of vectors in the subspace spanned by the bundle P which contains the important subgradient information. For simplicity we mention (see Krishnan and Mitchell, 2003b, for a discussion) that this can be interpreted as solving the following problem in lieu of (5.19)

$$\max \ \beta\lambda_{\min}(P^T(C - \mathcal{A}^T y)P) + b^T y, \tag{5.20}$$

whose dual is the following SDP

$$\begin{aligned} \min \ & (P^T C P) \bullet W \\ \text{s.t.} \ & (P^T A_i P) \bullet W = b_i, \quad i = 1, \ldots, m \\ & I \bullet W = \beta \\ & W \succeq 0. \end{aligned} \tag{5.21}$$

In the actual bundle method, instead of (5.20), we solve an SDP with a quadratic objective term; the quadratic term arises from the regularization term employed in the bundle method. For more details we refer the reader to Helmberg (2000a); Helmberg and Rendl (2000); Helmberg and Oustry (2000). In (5.21), we are approximately solving (SDP), by considering only a subset of the feasible X matrices. By keeping the number of columns r in P small, the resulting SDP can be solved quickly. The

dimension of the subspace P is roughly bounded by the square root of number of constraints. This follows from a bound by Pataki (1998) on the rank of extreme matrices in SDP. The optimum solution of (5.20) typically produces an indefinite dual slack matrix $S = (C - \mathcal{A}^T y)$. The negative eigenvalues and corresponding eigenvectors of S are used to update the subspace, P and the process is iterated. A recent primal active set approach for SDP which also deals with (5.21) has been recently developed by Krishnan et al. (2004).

Another variation of the low rank factorization idea mentioned above has been pursued by Burer and Monteiro (2003a). They consider factorizations $X = RR^T$, where $R \in \mathbb{R}^{n \times r}$, and instead of (SDP) they solve the following formulation for R

$$\begin{aligned} \min \quad & C \bullet (RR^T) \\ \text{s.t.} \quad & \mathcal{A}(RR^T) = b. \end{aligned}$$

This is a non-convex optimization problem that is solved using a modified version of the augmented Lagrangian method. The authors claim via extensive computational experiments that the method converges to the exact optimum value of (SDP), while a recent proof of convergence for a variant of this approach appears in Burer and Monteiro (2003b). As a particular case of this approach, Burer & Monteiro have employed rank two relaxations of maximum cut Burer et al. (2002b), and maximum stable set Burer et al. (2002c) problems with considerable computational success. The rank two relaxation is in fact an exact formulation of the maximum stable set problem.

We now turn to the method due to Burer et al. (2002a). This method complements the bundle approach discussed previously; it recasts the dual SDP as a non-convex but smooth unconstrained problem. The method operates on the following pair of SDPs.

$$\begin{aligned} \max \quad & C \bullet X \\ \text{s.t.} \quad & \operatorname{diag}(X) = d, \\ & \mathcal{A}(X) = b, \\ & X \succeq 0, \end{aligned} \tag{5.22}$$

with dual

$$\begin{aligned} \min \quad & d^T z + b^T y \\ \text{s.t.} \quad & \mathcal{A}^T y + \operatorname{Diag}(z) - S = C, \\ & S \succeq 0. \end{aligned} \tag{5.23}$$

Burer et al. consider only strictly feasible solutions of (5.23), i.e., $S = (\mathcal{A}^T y + \mathrm{Diag}(z) - C) \succ 0$. Consider now a Cholesky factorization of

$$S = (\mathrm{Diag}(v) + L_0)(\mathrm{Diag}(v) + L_0)^T, \tag{5.24}$$

where $v \in \mathbb{R}^n_{++}$, and L_0 is a strictly lower triangular matrix. In (5.24), there are $n(n+1)/2$ equations, and $m+n+n(n+1)/2$ variables. So one can use the equations to write $n(n+1)/2$ variables, namely z and L_0, in terms of the other variables v and y. Thus one can transform (5.23) into the following equivalent nonlinear programming problem

$$\begin{aligned} &\inf \ d^T z(v,y) + b^T y \\ &\text{s.t.} \ \ v > 0, \end{aligned} \tag{5.25}$$

where $z(v,y)$ indicates that z has been written in terms of v and y using (5.24). We note that the nonlinearity in (5.23) has been shifted from the constraints to the objective function, i.e., in the term $z(v,y)$ in (5.25). The latter problem does not attain its optimal solution, however we can use its intermediate solutions to approach the solution of (5.23) for a given $\epsilon > 0$. Moreover, the function $z(v,y)$ is a smooth analytic function. The authors then use a log-barrier term introducing the $v > 0$ constraint into the objective function, and suggest a potential reduction algorithm to solve (5.25); thus their approach amounts to reducing SDP to a non-convex, but smooth unconstrained problem. The main computational task is the computation of the gradient, and Burer et al. (2003) develop formulas that exploit the sparsity of the problem data. Although the objective function is non-convex, the authors prove global convergence of their method, and have obtained excellent computational results on large scale problems.

Other approaches include Benson and Vanderbei (2003), a dual Lagrangian approach due to Fukuda et al. (2002), and PENNON by Kocvara and Stingl (2003) that can also handle nonlinear semidefinite programs. A variant of the bundle method has also been applied to the Quadratic Assignment Problem (QAP) by Rendl and Sotirov (2003); their bounds are the strongest currently available for the QAP and this is one of the largest SDPs solved to date.

6. Branch and cut SDP based approaches

We discuss an SDP based branch and cut approach in this section that is designed to solving combinatorial optimization problems to optimality via a series of SDP relaxations of the underlying problem. Our particular emphasis is on the maxcut problem.

A branch and cut approach combines the advantages of cutting plane, and branch and bound methods. In a pure branch and bound approach

the relaxation is improved by dividing the problem into two subproblems, where one of the variables is restricted to taking certain values. The subproblems form a tree known as the branch and bound tree, rooted at the initial relaxation.

In a branch and cut approach cutting planes are added to the subproblems in the branch and bound tree, improving these relaxations until it appears that no progress can be made. Once this is the case, we resort to branching again. We do not discuss branch and cut LP approaches in this survey, but rather refer the reader to the survey by Mitchell et al. (1998).

Consider now the maxcut problem. As discussed in Section 1, for $S \subseteq V$ with cut $\delta(S)$, the maxcut problem (MC) can be written as

$$\max_{S \subset V} \sum_{\{i,j\} \in \delta(S)} w_{ij}. \tag{MC}$$

Without loss of generality, we can assume that our graph is complete. In order to model an arbitrary graph in this manner, define $w_{ij} = 0$, $\{i,j\} \notin E$. Finally, let $A = (w_{ij})$ be the weighted adjacency matrix of the graph.

We consider an SDP relaxation of the maxcut problem in this section. The maxcut problem can be formulated as the following integer program (5.26) in the x variables, where $x_i = 1$ if vertex $i \in S$, and -1 if $i \in V \backslash S$

$$\max_{x \in \{-1,1\}^n} \sum_{i,j=1}^{n} w_{ij} \frac{1 - x_i x_j}{4}. \tag{5.26}$$

A factor of $\frac{1}{2}$ accounts the fact that each edge is considered twice. Moreover, the expression $(1 - x_i x_j)/2$ is 0 if $x_i = x_j$, i.e., if i and j are in the same set, and 1 if $x_i = -x_j$. Thus $(1 - x_i x_j)/2$ yields the *incidence vector* of a cut associated with a cut vector x, evaluating to 1 if and only if edge $\{i,j\}$ is in the cut. Exploiting the fact that $x_i^2 = 1$, we have

$$\begin{aligned} \frac{1}{4} \sum_{i,j=1}^{n} w_{ij}(1 - x_i x_j) &= \frac{1}{4} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} w_{ij} x_i^2 - \sum_{j=1}^{n} w_{ij} x_i x_j \right) \\ &= \frac{1}{4} x^T (\mathrm{Diag}(Ae) - A) x. \end{aligned} \tag{5.27}$$

The matrix $L = \mathrm{Diag}(Ae) - A$ is called the *Laplacian* matrix of the graph G. Letting $C = \frac{1}{4}L$, we find that the maxcut problem can be interpreted as a special case of the following more general $\{+1, -1\}$ integer programming problem

$$\max_{x \in \{-1,1\}^n} x^T C x. \tag{5.28}$$

We are now ready to derive a semidefinite programming relaxation for the maxcut problem. First note that $x^TCx = \text{Trace}(Cxx^T)$. Now consider $X = xx^T$, i.e., $X_{ij} = x_ix_j$. Since $x \in \{-1, 1\}^n$, the matrix X is positive semidefinite, and its diagonal entries are equal to one. Thus (5.28) is equivalent to the following problem

$$\begin{aligned} \max \quad & C \bullet X \\ \text{s.t.} \quad & \text{diag}(X) = e, \\ & X \succeq 0, \\ & \text{rank}(X) = 1. \end{aligned} \tag{5.29}$$

The rank restriction is a non-convex constraint. To get a convex problem one drops the rank one restriction, and arrives at the following semidefinite programming relaxation of the maxcut problem

$$\begin{aligned} \max \quad & C \bullet X \\ \text{s.t.} \quad & \text{diag}(X) = e, \\ & X \succeq 0, \end{aligned} \tag{5.30}$$

and its dual

$$\begin{aligned} \min \quad & e^Ty \\ \text{s.t.} \quad & S = \text{Diag}(y) - C, \\ & S \succeq 0. \end{aligned} \tag{5.31}$$

Lemaréchal and Oustry (1999) and Poljak et al. (1995) derive the SDP relaxation (5.30) by taking the dual of the Lagrangian dual of (5.26), which incidentally is (5.31). We will refer to the feasible region of (5.30) as the elliptope. A point that must be emphasized is that the elliptope is no longer a polytope. Thus (5.30) is actually a non-polyhedral relaxation of the maxcut problem.

These semidefinite programs satisfy strong duality, since $X = I$ is strictly feasible in the primal problem, and we can generate a strictly feasible dual solution by assigning y an arbitrary positive value. In fact, setting $y_i = 1 + \sum_{j=1}^{n} |C_{ij}|$ and $S = \text{Diag}(y) - C$ should suffice.

We can improve the relaxation (5.30) using the following linear inequalities.

(1) **The odd cycle inequalities**

$$X(\mathcal{C}\backslash\mathcal{F}) - X(\mathcal{F}) \leq |\mathcal{C}| - 2 \quad \text{for each cycle } \mathcal{C}, \mathcal{F} \subset \mathcal{C}, |\mathcal{F}| \text{ odd.} \tag{5.32}$$

These include among others the triangle inequalities. They provide a complete description of the cut polytope for graphs not contractible

to K_5 (see Barahona, 1983; Seymour, 1981). Although there are an exponential number of linear constraints in (5.32), Barahona and Mahjoub (1986) (see also Grötschel et al., 1993) describe a polynomial time separation oracle for these inequalities, that involves solving n shortest path problems on an auxiliary graph with twice the number of nodes, and four times the number of edges. Thus it is possible to find the most violated odd cycle inequality in polynomial time.

(2) **The hypermetric inequalities**
These are inequalities of the form (5.33)

$$aa^T \bullet X \geq 1, \quad \text{where } a \in \mathcal{Z}^n, \sum_{i=1}^{n} a_i \text{ odd}$$
$$\text{and } \min\{(a^T x)^2 : x \in \{-1,1\}^n\} = 1. \quad (5.33)$$

For instance, the triangle inequality $X_{ij}+X_{ik}+X_{jk} \geq -1$ can be written as a hypermetric inequality by letting a to be the incidence vector of the triangle (i,j,k). On the other hand the other inequality $X_{ij}-X_{ik}-X_{jk} \geq -1$ can be written in a similar way, except that $a_k = -1$. Although there are a countably infinite number of them, these inequalities also form a polytope known as the hypermetric polytope (Deza and Laurent, 1997). The problem of checking violated hypermetric inequalities is NP-hard (Avis, 2003; Avis and Grishukhin, 1993). However, Helmberg and Rendl (1998) describe simple heuristics to detect violated hypermetric inequalities.

We sketch a conceptual SDP cutting plane approach for the maxcut problem in this section.

An SDP cutting plane approach for maxcut.

(1) **Initialize.** Start with (5.30) as the initial SDP relaxation.
(2) **Solve the current SDP relaxation.** Use a primal-dual IPM as discussed in Section 4.2. This gives an upper bound on the optimal value of the maxcut problem.
(3) **Separation.** Check for violated odd cycle inequalities. Sort the resulting violated inequalities, and add a subset of the most violated constraints to the relaxation.
If no violated odd cycle inequalities are found **goto** step 5.
(4) **Primal heuristic.** Use the Goemans and Williamson (1995) randomized rounding procedure (discussed in Section 7) to find a good incidence cut vector. This is a lower bound on the optimal value.
(5) **Check for termination. If** the difference between the upper bound and the value of the best cut is small, **then** stop.

If no odd cycle inequalities were found in step 3 **then goto** step 4. **Else goto** step 2.

(6) **Branching.** Resort to branch and bound as discussed in Section 6.1.

The choice of a good SDP branch and cut approach hinges on the following:

- **Choice of a good initial relaxation:** The choice of a good initial relaxation is important, and provides a tight upper bound on the maxcut value. The SDP relaxation (5.30) is an excellent choice; it is provably tight in most cases. Although, better initial SDP relaxations (Anjos and Wolkowicz, 2002a,b; Lasserre, 2002; Laurent, 2004) do exist, they are more expensive to solve. In contrast the polyhedral cutting plane approaches rely on poor LP relaxations, the ratio of whose bounds to the maxcut optimal value can be as high as 2 (Poljak and Tuza, 1994). Recently, Krishnan and Mitchell (2004) have proposed an semidefinite based LP cut-and-price algorithm for solving the maxcut problem, where one uses an LP cutting plane subroutine for solving the dual SDP relaxation (5.31).
- **Generating good lower bounds:** The Goemans–Williamson rounding procedure in Step 4 is an algorithm for generating incidence cut vectors, that provide good lower bounds. We will see in Section 7 that this procedure is instrumental in developing a 0.878 approximation algorithm for the maxcut problem.
- **Choice of good cutting planes:** It is important to use good cutting planes that are facets of the maxcut polytope, and use heuristics for finding such constraints quickly. In the above cutting plane approach for instance we might first check for violated triangle inequalities by complete enumeration, and use the Barahona-Mahjoub separation oracle when we run out of triangle inequalities (Mitchell, 2000).
- **Choice of the branching rule:** Typically we may have to resort to branch and bound in Step 6. It is important to choose a good branching rule to keep the size of the branch and bound tree small. We present a short discussion on branch and bound in an SDP branch and cut framework in Section 6.1.
- **Warm start:** One of the major shortcomings of an SDP branch and cut approach, where a primal-dual IPM is employed in solving the SDP relaxations is the issue of restarting the SDP relaxations after the addition of cutting planes. Although some warm start strategies do exist for the maxcut problem (Mitchell, 2001), they are prohibitively expensive. We will discuss some of these strategies in Section 6.2. There do exist simplex-like analogues for SDP

(Pataki, 1996a,b; Krishnan et al., 2004), and dual simplex variants of these schemes could conceivably be used to re-optimize the SDP relaxations after the addition of cutting planes.

6.1 Branch and bound in the SDP context

We provide a short overview on branch and bound within the SDP context in this section. Some excellent references for branch and bound within the SDP context of the maxcut problem are Helmberg and Rendl (1998), and Mitchell (2001).

Consider $X = V^T V$, with $V = (v_1, \ldots, v_n)$. We want to branch based on the values of $X_{ij} = (v_i^T v_j)$. Typically this is the most fractional variable, i.e., the X_{ij} closest to zero. The branching scheme is based on whether vertices i and j should be on the same side of the cut or on opposite sides. With this branching rule X_{ki} and X_{kj} are also then constrained to be either the same or different, $\forall k = \{1, \ldots, n\} \backslash \{i, j\}$. This means that the problem can be replaced by an equivalent semidefinite program of dimension one less. Without loss of generality let us assume that we are branching on whether vertices $n-1$ and n are on the same or opposite sides. Let we write the Laplacian matrix L in (5.30) as

$$L = \begin{bmatrix} \bar{L} & p_1 & p_2 \\ p_1^T & \alpha & \beta \\ p_2^T & \beta & \gamma \end{bmatrix}.$$

Here $\bar{L} \in \mathcal{S}^{n-2}$, $p_1, p_2 \in \mathbb{R}^{n-2}$ and α, β, and $\gamma \in \mathbb{R}$. The SDP relaxation that corresponds to putting both $n-1$ and n on the same side is

$$\begin{array}{ll} \max & \frac{1}{4} \begin{bmatrix} \bar{L} & p_1 + p_2 \\ p_1^T + p_2^T & \alpha + 2\beta + \gamma \end{bmatrix} \bullet X \\ \text{s.t.} & \operatorname{diag}(X) = e, \\ & X \succeq 0, \end{array} \tag{5.34}$$

with dual

$$\begin{array}{ll} \min & e^T y \\ \text{s.t.} & S = \operatorname{Diag}(y) - \frac{1}{4} \begin{bmatrix} \bar{L} & p_1 + p_2 \\ p_1^T + p_2^T & \alpha + 2\beta + \gamma \end{bmatrix}, \\ & S \succeq 0. \end{array} \tag{5.35}$$

Note that $X, S \in \mathcal{S}^{n-1}$, and $y \in \mathbb{R}^{n-1}$, i.e., not only do we have a semidefinite program of dimension one less, but the number of constraints in (5.34) has dropped by one as well. This is because performing

the same transformation (as the Laplacian) on the nth coefficient matrix $e_n e_n^T$ leaves it as $e_{n-1}e_{n-1}^T$, which is in fact the $(n-1)$th coefficient matrix.

On the other hand, putting $n-1$ and n on opposite sides, we get a similar SDP relaxation, with the Laplacian matrix now being

$$\frac{1}{4}\begin{bmatrix} \bar{L} & p_1 - p_2 \\ p_1^T - p_2^T & \alpha - 2\beta + \gamma \end{bmatrix}.$$

It is desirable that we use the solution of the parent node, in this case the solution of (5.30), to speed up the solution of the child (5.34). As we mentioned previously, this is a major issue in the SDP, since there is no analogue to the dual simplex method, unlike the LP case for reoptimization. More details on this can be found in Mitchell (2001).

Another important issue is determining good bounds for each of the subproblems, so that some of these subproblems in the branch and bound tree could be *fathomed*, i.e., not explicitly solved. In the LP approach, we can use reduced costs to estimate these bounds, and hence fix some of the variables without having to solve both subproblems. In the SDP case things are not so easy, since the constraints $-1 \leq X_{ij} \leq 1$ are not explicitly present in the SDP relaxation (they are implied through the $\mathrm{diag}(X) = e$ and $X \succeq 0$ constraints). Thus, the dual variables corresponding to these constraints are not directly available. Helmberg (2000b) describes a number of approaches to fix variables in semidefinite relaxations.

6.2 Warm start strategies for the maxcut problem

In cutting plane algorithms it is of fundamental importance that reoptimization is carried out in reasonable time after the addition of cutting planes. Since the cutting planes cut off the optimal solution X^{prev} to the previous relaxation, we need to generate a new strictly feasible point X^{start} for restarting the method.

We first discuss two strategies of restarting the primal problem since this is the more difficult problem.

(1) **Backtracking along iterates:**
This idea is originally due to Mitchell and Borchers (1996) for the LP. The idea is to store all the previous iterates on the central path, during the course of solving the original SDP relaxation (5.30), and restart from the last iterate that is strictly feasible with respect to the new inequalities. Also, this point is hopefully close to the new

central path, and the interior point algorithm will work better if this is the case.

(2) **Backtracking towards the analytic center:**
This was employed in Helmberg and Rendl (1998). The idea is to backtrack towards I along a straight line between the last iterate X^{prev} and I. Thus we choose $X^{\text{start}} = (\lambda X^{\text{prev}} + (1-\lambda)I)$ for some $\lambda \in [0,1)$. Since the identity matrix I is the analytic center of the feasible region of (5.30), it is guaranteed that the procedure will terminate with a strictly feasible primal iterate.

Restarting the dual which has additional variables corresponding to the number of cutting planes in the primal is relatively straightforward, since we can get into the dual SDP cone $S \succeq 0$, by assigning arbitrarily large values to the first n components of y (that originally appear in $\text{Diag}(y)$).

7. Approximation algorithms for combinatorial optimization

One of the most important applications of SDP is in developing approximation algorithms for various combinatorial optimization problems. The euphoria began with an 0.878 GW approximation algorithm (Goemans and Williamson, 1995) for the maxcut problem, and the technique has since been applied to a variety of other problems. For some of these problems such as MAX 3SAT, the SDP relaxation (Karloff and Zwick, 1997) provides the tightest approximation algorithm possible unless P = NP.

We discuss the GW algorithm in detail below. The algorithm works with the SDP relaxation (5.30) for the maxcut problem we introduced in Section 6. We outline the main steps in the algorithm as follows:

The Goemans – Williamson (GW) approximation algorithm for maxcut.

(1) Solve the SDP relaxation (5.30) to get a primal matrix X.
(2) Compute $V = (v_1, \ldots, v_n)$ such that $X = V^T V$. This can be done either by computing the Cholesky factorization of X, or by computing its spectral decomposition $X = P\Lambda P^T$, with $V = \sqrt{\Lambda} P^T$.
(3) Randomly partition the unit sphere in $\mathbb{R}^n$ into two half spheres H_1 and H_2 (the boundary in between can be on either side), and form the bipartition consisting of $V_1 = \{i : v_i \in H_1\}$ and $V_2 = \{i : v_i \in H_2\}$. The partitioning is carried out in practice by generating a random vector r on the unit sphere, and assigning i to V_1 if $v_i^T r \geq 0$, and V_2 otherwise. In practice, one may repeat this procedure more than once, and pick the best cut obtained.

Hereafter, we refer to Step 3 as the *GW rounding procedure*. It is important to note that Step 3 gives a lower bound on the optimal maxcut solution, while the SDP relaxation in Step 1 gives an upper bound. The entire algorithm can be derandomized as described in Mahajan and Hariharan (1999).

A few notes on the GW rounding procedure: For any factorization of $X = V^T V$ in Step 2, the columns of V yield vectors v_i, $i = 1, \ldots, n$. Since we have $\operatorname{diag}(X) = e$, each vector v_i is of unit length, i.e., $\|v_i\| = 1$. Associating a vector v_i with node i, we may interpret v_i as the relaxation of $x_i \in \{-1, 1\}$ to the n dimensional unit sphere. Thus we are essentially solving

$$\begin{aligned} \max \quad & \sum_{i,j=1}^{n} \frac{L_{ij}}{4} v_i^T v_j \\ \text{s.t.} \quad & \|v_i\| = 1 \quad \forall i = 1, \ldots, n, \\ & v_i \in \mathbb{R}^n. \end{aligned} \tag{5.36}$$

This vector formulation provides a way to interpret the solution to the maxcut SDP. Since v_i and v_j are unit vectors, $v_i^T v_j$ is the cosine of the angle between these vectors. If all the edge weights w_{ij} are nonnegative, the off diagonal entries of the Laplacian matrix are negative. Thus, if the angle between the vectors is large, we should separate the corresponding vertices, if it is small we put them in the same set (since this would improve the objective function in the vector formulation). In order to avoid conflicts, Goemans and Williamson (1995) consider the random hyperplane technique mentioned in Step 3. This step is in accord with our earlier intuition, since vectors with a large angle between them are more likely to be separated, since the hyperplane can end up between them.

The hyperplane with normal r in Step 3 of the algorithm divides the unit circle into two halfspheres, and an edge $\{i, j\}$ belongs to the cut $\delta(S)$ if and only if the vectors v_i and v_j do not belong to the same halfsphere. The probability that an edge $\{i, j\}$ belongs to $\delta(S)$ is equal to $\arccos(v_i^T v_j)/\pi$, and the expected weight $E(w(S))$ of the cut $\delta(S)$ is

$$\begin{aligned} E(w(S)) &= \sum_{i,j=1}^{n} \frac{L_{ij}}{4} \frac{\arccos(v_i^T v_j)}{\pi} \\ &= \sum_{i,j=1}^{n} \frac{L_{ij}}{4} \frac{1 - v_i^T v_j}{2} \frac{2}{\pi} \frac{\arccos(v_i^T v_j)}{1 - v_i^T v_j} \\ &\geq 0.878 \times (\text{objective value of relaxation (5.36)}) \\ &\geq 0.878 \times (\text{optimal maxcut value}). \end{aligned}$$

The second to last inequality holds if we assume that all the edge weights are nonnegative, and from the observation that

$$\min_{-1\leq x\leq 1} \frac{2}{\pi}\frac{\arccos(x)}{1-x} \geq 0.878.$$

The last inequality from the fact that the objective value of relaxation (5.36) provides an upper bound on the maxcut solution. Hence, we have an 0.878 approximation algorithm for the maxcut problem, when all the edge weights are nonnegative. On the negative side Håstad (1997) showed that it is NP-hard to approximate the maxcut problem to within a factor of 0.9412.

For the general case where $L \succeq 0$, Nesterov (1998) showed that the GW rounding procedure gives an $\frac{2}{\pi}$ approximation algorithm for the maxcut problem.

Interestingly, although, the additional inequalities such as triangle inequalities (mentioned with regard to the metric polytope) improve the SDP relaxation, they do not necessarily give better approximation algorithms. On the negative side Karloff (1999) exhibited a set of graphs for which the optimal solution of relaxation (5.30) satisfies all the triangle inequalities as well, so after the GW rounding procedure we are still left with a 0.878 approximation algorithm.

Goemans and Williamson (1995) show that the randomized rounding procedure performs well if the ratio of the weight of the edges in the cut, to those in the graph is more than 85%. If this is not true, then it pays to introduce more randomness in the rounding procedure. Zwick (1999) considers the randomized rounding as applied to $(\gamma I + (1-\gamma)X)$ rather than X, for some appropriate $\gamma \in [0,1]$.

There have been several extensions of SDP and the randomized rounding technique to other combinatorial optimization problems. These include quadratic programming (Nesterov, 1998; Ye, 1999), maximum bisection (Frieze and Jerrum, 1997; Ye, 2001), max k-cut problem (Frieze and Jerrum, 1997; Goemans and Williamson, 2001) and more recently in De Klerk et al. (2004b), graph coloring (Karger et al., 1998), vertex cover (Kleinberg and Goemans, 1998), maximum satisfiability problem (Goemans and Williamson, 1995; De Klerk and Van Maaren, 2003; De Klerk et al. , 2000; Anjos, 2004), Max 2SAT (Feige and Goemans, 1995), Max 3SAT (Karloff and Zwick, 1997), and finally the maximum directed cut problem (Goemans and Williamson, 1995; Feige and Goemans, 1995). A nice survey on the techniques employed in designing approximation algorithms for these problems can be found in Laurent and Rendl (2003), while a good overview of the techniques for satisfiability, graph coloring, and max k-cut appears in the recent monograph by De Klerk (2002).

8. Convex approximations of integer programming

The results in this section are based on recent results by Nesterov (2000), Lasserre (2001, 2002), Parrilo (2003) and De Klerk and Pasechnik (2002); Bomze and De Klerk (2002). A nice survey of these methods also appears in Laurent and Rendl (2003).

8.1 Semidefinite approximations of polynomial programming

Consider the following polynomial programming problem

$$\begin{aligned} &\min \ g_0(x) \\ &\text{s.t.} \ \ g_k(x) \geq 0, \quad k = 1, \ldots, m, \end{aligned} \tag{5.37}$$

where $g_k(x)$, $k = 0, \ldots, m$ are polynomials in $x = (x_1, \ldots, x_n)$. This is a general problem which encompasses $\{0, 1\}$ integer programming problems, since the condition $x_i \in \{0, 1\}$ can be expressed as the polynomial equation $x_i^2 - x_i = 0$. The importance of (5.37) is that, under some technical assumptions, this problem can be approximated by a sequence of semidefinite programs. This result, due to Lasserre (2001), relies on the fact that certain nonnegative polynomials can be expressed as sums of squares (SOS)[1] of other polynomials. Also, see Nesterov (2000), Parrilo (2003), and Shor (1998) for using SOS representations of polynomials for approximating (5.37).

We give a brief overview of some of the main ideas underlying this approach. For ease of exposition we shall confine our attention to the unconstrained problem

$$g^* = \min\{g(x), x \in \mathbb{R}^n\} \tag{5.38}$$

where without loss of generality we assume $g(x)$ is a polynomial of even degree $2d$. Let

$$[1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, x_2x_3, \ldots, x_n^2, \ldots, x_1^{2d}, \ldots, x_n^{2d}]$$

be a basis for $g(x)$. Let

$$S_{2d} = \left\{ \alpha \in \mathcal{Z}_+^n : \sum_i \alpha_i \leq 2d \right\},$$

[1]This is not to be confused with *specially ordered sets* commonly used in integer programming.

and let $s(2d) = |S_{2d}|$. The above basis can then be conveniently represented as $\{x^\alpha\}$, $\alpha \in S_{sd}$. We write $g(x) = \sum_{\alpha \in S_{2d}} \gamma_\alpha x^\alpha$, with $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, where $\gamma = \{\gamma_\alpha\} \in \mathbb{R}^{s(2d)}$ is the coefficient vector of $g(x)$ in the basis. Then problem (5.38) can also be written as

$$g^* = \max\{\lambda \text{ s.t. } g(x) - \lambda \geq 0, \ \forall x \in \mathbb{R}^n\}. \tag{5.39}$$

This problem encompasses integer and non-convex optimization problems, and consequently is NP hard. However, lower bounds on g^* can be obtained by considering sufficient conditions for the polynomial $g(x) - \lambda \geq 0$ on $\mathbb{R}^n$. One such requirement is that $g(x) - \lambda$ be expressible as a sum of squares of polynomials, i.e., have an SOS representation. Thus,

$$g^* \geq \max\{\lambda \text{ s.t. } g(x) - \lambda \text{ has an SOS representation}\}. \tag{5.40}$$

Problem (5.40) can be expressed as a semidefinite program. To see this, let $z = \{x^\alpha\}$ with $\alpha \in S_d$ be the basis vector consisting of all monomials of degree $\leq d$. Then one can easily verify that $g(x)$ has an SOS representation if and only if $g(x) = z^T X z$ for some positive semidefinite matrix X. For $\gamma \in S_{2d}$, let

$$B_\gamma = \sum_{\substack{\alpha,\beta \in S_d \\ \alpha+\beta=\gamma}} E_{\alpha,\beta},$$

where $E_{\alpha,\beta}$ is the elementary matrix with all zero entries except entries 1 at positions (α,β) and (β,α). Using this we have:

$$\begin{aligned} z^T X z &= \sum_{\alpha,\beta \in S_d} X_{\alpha,\beta} x^{\alpha+\beta}, \\ &= \sum_{\gamma \in S_{2d}} x^\gamma \sum_{\substack{\alpha,\beta \in S_d, \\ \alpha+\beta=\gamma}} X_{\alpha,\beta}, \\ &= \sum_{\gamma \in S_{2d}} x^\gamma (B_\gamma \bullet X). \end{aligned}$$

Assuming the constant term g_0 in the polynomial $g(x)$ is zero, and comparing coefficients in $g(x) - \lambda = \sum_{\gamma \in S_{2d}} x^\gamma (B_\gamma \bullet X)$ for $\gamma = 0$, we have $\lambda = -B_0 \bullet X$. Hence, one can equivalently write (5.40) as the following SDP

$$\begin{aligned} \max \ & -B_0 \bullet X \\ \text{s.t. } & B_\gamma \bullet X = g_\gamma, \quad \gamma \in S_{2d} \backslash \{0\}, \\ & X \succeq 0, \end{aligned} \tag{5.41}$$

with dual

$$\begin{aligned} \min & \sum_{\alpha \in S_{2d}} g_\alpha y_\alpha \\ \text{s.t.} & \sum_{\alpha \in S_{2d}} B_\alpha y_\alpha \succeq 0. \end{aligned} \tag{5.42}$$

The dual (5.42) has an equivalent interpretation in the theory of moments, and forms the basis for the original approach of Lasserre (2001). Another advantage of this dual approach of Lasserre (2001), over the primal approach of Parrilo (2003), is that it also yields certificates ensuring that an optimal solution is attained in the series of relaxations, and also gives a mechanism for extracting these solutions (see Henrion and Lasserre, 2003b).

In general for a polynomial with even degree $2d$ in n variables, the SDP (5.41) has $\binom{n+2d}{2d}$ constraints, where X is a matrix in $\mathcal{S}^{\binom{n+d}{d}}$. The lower bound from (5.41) is equal to g^* if the polynomial $g(x) - \lambda$ has an SOS representation; this is true for $n = 1$, but not in general if $n \geq 2$. In such cases, one can estimate g^* asymptotically by a sequence of SDPs, if one assumes that an upper bound R is known a priori on the norm of a global minimizer x of $g(x)$ (Lasserre, 2001), by using a theorem of Putinar (1993) for SOS representations of the positive polynomial $g(x) - \lambda + \epsilon$ on the set $\{x : \|x\| \leq R\}$. This gives a sequence of SDP approximations, whose objective values asymptotically converge to g^*. A similar approach has been adopted by Lasserre (2001) for the constrained case (5.37).

In the $\{0, 1\}$ case, when the constraints $x_i^2 - x_i = 0$ are part of the polynomials in the constraint set, Lasserre (2002) shows there is finite convergence in n steps. Laurent (2003) shows that the Lassere approach is actually a strengthened version of the Sherali and Adams (1990) lift and project procedure, and since the latter scheme converges in at most n steps so does the above approach. Other lift and project methods include Lovász and Schrijver (1991), and Balas et al. (1993) in the context of estimating the convex hull of the feasible set of $\{0, 1\}$ programming problems, and the successive convex approximations to non-convex sets introduced in Kojima and Tuncel (2000). We also refer the reader to Laurent (2003), and the recent survey by Laurent and Rendl (2003) for a comparison of these various approaches. Finally, MATLAB code based on the above approach have been developed by Prajna et al. (2002) and Henrion and Lasserre (2003a).

8.2 Copositive formulations of IP and SDP approximations of copositive programs

As another instance of convex approximations to integer programming, we consider the problem of finding the stability number of a graph. This problem can be expressed as a copositive program (Quist et al., 1998; Bomze et al., 2000), that is a convex optimization problem. Recently, De Klerk and Pasechnik (2002) apply the technique of approximating the copositive cone through a series of semidefinite approximations introduced by Parrilo (2003), and use this to estimate the stability number of the graph to any degree of accuracy. We present a brief overview of their approach in this section.

The stability number of a graph $G = (V, E)$, denoted by $\alpha(G)$, can be expressed as the solution to a copositive programming problem (Quist et al., 1998); this is based on an earlier representation of $\alpha(G)$ due to Motzkin and Strauss (1965) that amounts to minimizing a particular quadratic function over the simplex. This copositive program (5.43) is given by:

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & S = \lambda I + yA - ee^T, \\ & S \in \mathcal{C}_n, \end{aligned} \tag{5.43}$$

with dual

$$\begin{aligned} \max \quad & ee^T \bullet X \\ \text{s.t.} \quad & I \bullet X = 1, \\ & A \bullet X = 0, \\ & X \succeq 0, \end{aligned} \tag{5.44}$$

where $\lambda, y \in \mathbb{R}$, e is the all-ones vector, A is the adjacency matrix of the graph $G = (V, E)$, and $\mathcal{C}_n = \{X \in \mathcal{S}^n : d^T X d \geq 0, \forall d \geq 0\}$ is the set of $n \times n$ symmetric copositive matrices. The problem (5.43) is not solvable in polynomial time since the decision problem whether a matrix is copositive or not is NP-hard (Murthy and Kabadi, 1987). In fact, De Klerk and Pasechnik (2002) show that the equality constraints in (5.44) can be combined together as $(A + I) \bullet X = 1$. Thus, we can drop the additional variable y in (5.43), and rewrite the slack matrix as $S = \lambda(I + A) - ee^T$.

A sufficient condition for a matrix M to be copositive is $M \succeq 0$. In fact, setting $S \succeq 0$ in (5.43) gives a constrained version of (5.45) which represents the Lovász theta function (see Lovász, 1979; Grötschel et al., 1993) and is given by

$$\begin{aligned} &\min \lambda \\ &\text{s.t. } S = \lambda I + \sum_{\{i,j\}\in E} y_{ij} E_{ij} - ee^T, \\ &\quad S \succeq 0. \end{aligned} \tag{5.45}$$

Here $E_{ij} \in \mathcal{S}^n$ is the elementary matrix with all zero entries, except entries 1 in positions (i,j) and (j,i), corresponding to edge $\{i,j\}$ in the graph. In the search for stronger sufficient conditions for copositivity, Parrilo (2003, 2000) proposes approximating the copositive cone using SOS representations of polynomials. To see this, note that a matrix $M \in \mathcal{C}_n$ if and only if the polynomial

$$g_M(x) = \sum_{i,j=1}^{n} M_{ij} x_i^2 x_j^2$$

is nonnegative on $\mathbb{R}^n$. Therefore, a sufficient condition for M to be copositive is that $g_M(x)$ has an SOS representation, or more generally the polynomial $g_M(x)(\sum_{i=1}^n x_i^2)^r$ has an SOS representation for some integer $r \geq 0$. In fact a theorem due to Polya suggests that M is copositive, then $g_M(x)(\sum_{i=1}^n x_i^2)^r$ has an SOS representation for some r. An upper bound on r is given by Powers and Reznick (2001).

Let $\mathcal{K}_n^r$ to be the set of symmetric matrices for which $g_M(x)(\sum_{i=1}^n x_i^2)^r$ has an SOS representation. We then have the following hierarchy of approximations to $\mathcal{C}_n$.

$$\begin{aligned} \mathcal{S}_+^n \subseteq \mathcal{K}_n^0 \subseteq \ldots \\ \subseteq \mathcal{K}_n^r = \mathcal{C}_n. \end{aligned} \tag{5.46}$$

For each $\mathcal{K}_n^r$, one can define the parameter

$$\gamma^r(G) = \min \ \lambda \ \text{ s.t. } \ \lambda I + yA - ee^T \in \mathcal{K}_n^r, \tag{5.47}$$

where $\gamma^r(G) = \alpha(G)$ for some r. It was remarked in Section 8.1 that the SOS requirement on a polynomial can be written as a semidefinite program, and so (5.47) represents a hierarchy of semidefinite programs, whose objective values eventually converge to the stability number of the graph. Parrilo (2003) gives explicit SDP representations for $\mathcal{K}_n^r$, $r = 0, 1$. For instance $S \in \mathcal{K}_n^0$, if and only if $S = P + N$, for $P \succeq 0$, and $N \geq 0$. For the stable set problem, this first lifting gives the Schrijver formulation (Schrijver, 1979) of the Lovász theta function. In particular, using the estimate in Powers and Reznick (2001), De Klerk and Pasechnik (2002) show that

$$\alpha(G) = \lfloor \gamma^r(G) \rfloor, \quad \text{if } r \geq \alpha^2(G).$$

One obtains the same result by applying the hierarchy of SDP approximations due to Lasserre (discussed in Section 8.1) on the original Motzkin and Strauss (1965) formulation for the maximum stable set problem. In fact, De Klerk et al. (2004a) have shown that the copositive programming approach mentioned in this section and polynomial programming approach of Section 8.1 are equivalent for the problem of minimizing a quadratic function over the simplex (standard quadratic programming problem).

Recently, Bomze and De Klerk (2002) developed the first polynomial time approximation scheme (PTAS) for the standard quadratic programming problem, by applying a similar technique of LP and SDP approximations to the copositive cone. A good account also appears in the recent survey by De Klerk (2002).

As of now, copositive programming has only been applied to the standard quadratic programming problem De Klerk (2003). It is therefore interesting to speculate on other classes of problems that can be modelled as copositive programs.

9. Conclusions

We have presented an overview of some of the most recent developments in IPMs for solving various combinatorial optimization problems. IPMs are adapted in a number of ways to solving the underlying discrete problem; directly via a potential reduction approach in Section 2, in conjunction with an oracle in a cutting plane approach in Section 3, or applied to SDP relaxations or other convex reformulations of these problems as discussed in Sections 6 and 8. SDP is a major tool in continuous approaches to combinatorial problems, and IPMs of Section 4 can also be used in conjunction with ingenious randomized rounding schemes to generate solutions for various combinatorial optimization problems with provable performance guarantees. This was the topic of Section 7.

We conclude with a summary of some of the important issues, and open problems in the topics discussed:

(1) The interior point cutting plane methods of Section 3, especially ACCPM, and its variants have been applied to solve a variety of convex optimization problems with some degree of practical success. It is interesting to speculate whether ACCPM is indeed a polynomial time solution procedure for the convex feasibility problem. The volumetric center IPM on the other hand has the best complexity among cutting plane methods which is provably optimal, and has rendered the classical ellipsoid algorithm obsolete. Recent work by Anstreicher (1999) has considerably improved the constants involved

in the analysis of the algorithm, and it would be interesting to consider practical implementations of this algorithm in the near future.

(2) The primal-dual IPMs described in Section 4.2 are indeed the algorithms of choice for SDP; however as of now they are fairly limited in the size of problems they can handle in computational practice. The ability of future IPMs to handle large SDPs will depend to a great extent on the design of good pre-conditioners (see Toh, 2003; Toh and Kojima, 2002), that are required in an iterative method to solve the normal system of equations. On the other hand, the first order approaches discussed in Section 5 exploit the structure in the underlying SDP problem, and are consequently able to solve larger problems; albeit to a limited accuracy.

(3) On the theoretical side, the complexity of the semidefinite feasibility problem (SDFP) discussed in Section 4.1 is still an open problem.

(4) There have been several applications of SDP to hard discrete optimization problems as discussed in Section 7 of this survey. However, to the best of our knowledge, there have been relatively few applications of second order cone programming (SOCP) in combinatorial optimization. In this regard we note the work of Kim and Kojima (2001) and Muramatsu and Suzuki (2002). An open question is whether one could develop good approximation algorithms for combinatorial optimization using SOCP relaxations of the underlying problem, since the SOCP can be solved more quickly than SDP using IPMs.

(5) An important issue in the branch and cut approaches discussed in Section 6 is that of restarting the new relaxation with a strictly interior point after branching, or the addition of cutting planes. In this regard, it is interesting to consider dual analogues of the primal active set approaches investigated in Krishnan et al. (2004), which conceivably (like the dual simplex method for LP) could be employed for re-optimization.

(6) One of the major applications of the SDP is its use in developing approximation algorithms for various combinatorial optimization problems as discussed in Section 7. In many cases, such as the MAX 3 SAT problem, the SDP in conjunction with rounding schemes provides the tightest possible approximation algorithms for these problems unless P = NP. Recently, there has been renewed interest in SDP approximations to polynomial and copositive programming, which are provably exact in the limit. We discussed some of these ideas in Section 8. Although, there are a variety of problems that can be modelled as polynomial programs, the situation with respect to copositive programming is far less clear. In this regard it

is interesting to speculate on the classes of problems, that can be written as copositive programs.

Acknowledgments The authors would like to thank an anonymous referee whose comments greatly improved the presentation of the paper.

References

Alizadeh, F. and Goldfarb, D. (2003). Second-order cone programming. *Mathematical Programming*, 95:3 – 51.

Alizadeh, F., Haeberly, J.P.A., and Overton, M.L. (1998). Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 8:746 – 751.

Andersen, E.D., Gondzio, J., Mészáros, Cs., and Xu, X. (1996). Implementation of interior point methods for large scale linear programming. In: T. Terlaky (ed.), *Interior Point Methods for Linear Programming*, pp. 189 – 252. Kluwer Academic Publishers, The Netherlands.

Anjos, M.F. (2004). An improved semidefinite programming relaxation for the satisfiability problem. *Mathematical Programming*, to appear.

Anjos, M.F. and Wolkowicz, H. (2002a). Geometry of semidefinite maxcut relaxations via matrix ranks. *Journal of Combinatorial Optimization*, 6:237 – 270.

Anjos, M.F. and Wolkowicz, H. (2002b). Strengthened semidefinite relaxations via a second lifting for the maxcut problem. *Discrete Applied Mathematics*, 119:79 – 106.

Anstreicher, K.M. (1997). On Vaidya's volumetric cutting plane method for convex programming. *Mathematics of Operations Research*, 22:63 – 89.

Anstreicher, K.M. (1999). Towards a practical volumetric cutting plane method for convex programming. *SIAM Journal on Optimization*, 9:190 – 206.

Anstreicher, K.M. (2000). The volumetric barrier for semidefinite programming. *Mathematics of Operations Research*, 25:365 – 380.

Arora, S. and Lund, C. (1996). Hardness of approximations. In: D. Hochbaum (ed.), *Approximation Algorithms for NP-hard Problems*, Chapter 10. PWS Publishing.

Atkinson, D.S. and Vaidya, P.M. (1995). A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69:1 – 43.

Avis, D. (2003). On the complexity of testing hypermetric, negative type, k-gonal and gap inequalities. In: J. Akiyama, M. Kano (eds.), *Discrete*

and Computational Geometry, pp. 51 – 59. Lecture Notes in Computer Science, vol. 2866, Springer.

Avis, D. and Grishukhin, V.P. (1993). A bound on the k-gonality of facets of the hypermetric cone and related complexity problems. *Computational Geometry: Theory & Applications*, 2:241 – 254.

Bahn, O., Du Merle, O., Goffin, J.L., and Vial, J.P. (1997). Solving nonlinear multicommodity network flow problems by the analytic center cutting plane method. *Mathematical Programming*, 76:45 – 73.

Balas, E., Ceria, S., and Cornuejols, G. (1993). A lift and project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295 – 324.

Barahona, F. (1983). The maxcut problem in graphs not contractible to K_5. *Operations Research Letters*, 2:107-111.

Barahona, F. and Mahjoub, A.R. (1986). On the cut polytope. *Mathematical Programming*, 44:157 – 173.

Benson, H.Y. and Vanderbei, R.J. (2003). Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming. *Mathematical Programming*, 95:279 – 302.

Benson, S.J., Ye, Y., and Zhang, X. (2000). Solving large-scale semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10:443 – 461.

Bomze, I., Dur, M., De Klerk, E., Roos, C., Quist, A.J., and Terlaky, T. (2000). On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, 18:301 – 320.

Bomze, I. and De Klerk, E. (2002). Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24:163 – 185.

Borchers, B. (1999). CSDP: A C library for semidefinite programming. *Optimization Methods and Software*, 11:613 – 623.

Burer, S. (2003). Semidefinite programming in the space of partial positive semidefinite matrices. *SIAM Journal on Optimization*, 14:139 – 172.

Burer, S. and Monteiro, R.D.C. (2003a). A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95:329 – 357.

Burer, S. and Monteiro, R.D.C. (2003b). Local minima and convergence in low-rank semidefinite programming. Technical Report, Department of Management Sciences, University of Iowa.

Burer, S., Monteiro, R.D.C., and Zhang, Y. (2002a). Solving a class of semidefinite programs via nonlinear programming. *Mathematical Programming*, 93:97 – 102.

Burer, S., Monteiro, R.D.C., and Zhang, Y. (2002b). Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12:503–521.

Burer, S., Monteiro, R.D.C., and Zhang, Y. (2002c). Maximum stable set formulations and heuristics based on continuous optimization. *Mathematical Programming*, 94:137–166.

Burer, S., Monteiro, R.D.C., and Zhang, Y. (2003). A computational study of a gradient based log-barrier algorithm for a class of large-scale SDPs. *Mathematical Programming*, 95:359–379.

Chvátal, V. (1983). *Linear Programming*. W.H. Freeman and Company.

Conn, A.R., Gould, N.I.M., and Toint, P.L. (2000). *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM, Philadelphia, PA.

De Klerk, E. (2002). *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*. Applied Optimization Series, vol. 65, Kluwer Academic Publishers.

De Klerk, E. (2003). Personal communication.

De Klerk, E., Laurent, M., and Parrilo, P. (2004a). On the equivalence of algebraic approaches to the minimization of forms on the simplex. In: D. Henrion and A. Garulli (eds.), *Positive Polynomials in Control*, LNCIS, Springer, to appear.

De Klerk, E. and Pasechnik, D. (2002). Approximating the stability number of graph via copositive programming. *SIAM Journal on Optimization*, 12:875–892.

De Klerk, E., Pasechnik, D.V., and Warners, J.P. (2004b). Approximate graph coloring and max-k-cut algorithms based on the theta function. *Journal of Combinatorial Optimization*, to appear.

De Klerk, E., Roos, C., and Terlaky, T. (1998). Infeasible-start semidefinite programming algorithms via self-dual embeddings. *Fields Institute Communications*, 18:215–236.

De Klerk, E. and Van Maaren, H. (2003). On semidefinite programming relaxations of 2+p-SAT. *Annals of Mathematics of Artificial Intelligence*, 37:285–305.

De Klerk, E., Warners, J., and Van Maaren, H. (2000). Relaxations of the satisfiability problem using semidefinite programming. *Journal of Automated Reasoning*, 24:37–65.

Deza, M.M. and Laurent, M. (1997). *Geometry of Cuts and Metrics*. Springer-Verlag, Berlin.

Elhedhli, S. and Goffin, J.L. (2004). The integration of an interior-point cutting plane method within a branch-and-price algorithm. *Mathematical Programming*, to appear.

Feige, U. and Goemans, M. (1995). Approximating the value of two prover proof systems withe applications to MAX-2SAT and MAX-

DICUT. *Proceedings of the 3rd Isreal Symposium on Theory of Computing and Systems*, pp. 182–189. Association for Computing Machinery, New York.

Frieze, A. and Jerrum, M.R. (1997). Improved approximation algorithms for max k-cut and max bisection. *Algorithmica*, 18:61–77.

Fukuda, M., Kojima, M., Murota, K., and Nakata, K. (2000). Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, 11:647–674.

Fukuda, M., Kojima, M., and Shida, M. (2002). Lagrangian dual interior-point methods for semidefinite programs. *SIAM Journal on Optimization*, 12:1007–1031.

Garey, M.R., and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Company, San Francisco, CA.

Goemans, M. and Williamson, D.P. (1995). Improved approximation algorithms for max cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145.

Goemans, M. and Williamson, D.P. (2001). Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. In: *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 443–452. Association for Computing Machinery, New York.

Goffin, J.L., Gondzio, J., Sarkissian, R., and Vial, J.P. (1997). Solving nonlinear multicommodity flow problems by the analytic center cutting plane method. *Mathematical Programming*, 76:131–154.

Goffin, J.L., Luo, Z.Q., and Ye, Y. (1996). Complexity analysis of an interior point cutting plane method for convex feasibility problems. *SIAM Journal on Optimization*, 6:638–652.

Goffin, J.L. and Vial, J.P. (2000). Multiple cuts in the analytic center cutting plane method. *SIAM Journal on Optimization*, 11:266–288.

Goffin, J.L. and Vial, J.P. (2002). Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. *Optimization Methods and Software*, 17:805–867.

Gondzio, J., du Merle, O., Sarkissian, R., and Vial, J.P. (1996). ACCPM — A library for convex optimization based on an analytic center cutting plane method. *European Journal of Operations Research*, 94:206–211.

Grone, B., Johnson, C.R., Marques de Sa, E., and Wolkowicz, H. (1984). Positive definite completions of partial Hermitian matrices. *Linear Algebra and its Applications*, 58:109–124.

Grötschel, M., Lovász, L., and Schrijver, A. (1993). *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag.

Håstad, J. (1997). Some optimal inapproximability results. *Proceedings of the 29th ACM Symposium on Theory and Computing*, pp. 1 – 10.

Helmberg, C. (2000a). *Semidefinite Programming for Combinatorial Optimization.* Habilitation Thesis, ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin.

Helmberg, C. (2000b). Fixing variables in semidefinite relaxations. *SIAM Journal on Matrix Analysis and Applications*, 21:952 – 969.

Helmberg, C. (2003). Numerical evaluation of SBmethod. *Mathematical Programming*, 95:381 – 406.

Helmberg, C. and Oustry, F. (2000). Bundle methods to minimize the maximum eigenvalue function. In: H. Wolkowicz, R. Saigal, and L. Vandenberghe (eds.), *Handbook of Semidefinite Programming*, pp. 307 – 337. Kluwer Academic Publishers.

Helmberg, C. and Rendl, F. (1998). Solving quadratic (0,1) problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82:291 – 315.

Helmberg, C. and Rendl, F. (2000). A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10:673 – 696.

Helmberg, C., Rendl, F., Vanderbei, R., and Wolkowicz, H. (1996). An interior point method for semidefinite programming. *SIAM Journal on Optimization*, 6:673 – 696.

Helmberg, C. Semidefinite programming webpage.
`http://www-user.tu-chemnitz.de/~helmberg/semidef.html`

Henrion, D. and Lasserre, J. (2003a). Gloptipoly: Global optimization over polynomials with MATLAB and SeDuMi. *Transactions on Mathematical Software*, 29:165 – 194.

Henrion, D. and Lasserre, J. (2003b). Detecting global optimality and extracting solutions in Globtipoly. Technical Report LAAS-CNRS.

Interior-Point Methods Online.
`http://www-unix.mcs.anl.gov/otc/InteriorPoint`

Kamath, A.P., Karmarkar, N., Ramakrishnan, K.G., and Resende M.G.C. (1990). Computational experience with an interior point algorithm on the satisfiability problem. *Annals of Operations Research*, 25:43 – 58.

Kamath, A.P., Karmarkar, N., Ramakrishnan, K.G., and Resende, M.G.C. (1992). A continuous approach to inductive inference. *Mathematical Programming*, 57:215 – 238.

Karger, D., Motwani, R., and Sudan, M. (1998). Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45:246 – 265.

Karloff, H. (1999). How good is the Goemans-Williamson MAX CUT algorithm? *SIAM Journal on Computing*, 29:336–350.

Karloff, H. and Zwick, U. (1997). A 7/8 approximation algorithm for MAX 3SAT? In: *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pp. 406–415. IEEE Computer Science Press, Los Alamitos, CA.

Karmarkar, N. (1984). A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395.

Karmarkar, N. (1990). An interior point approach to NP-complete problems. *Contemporary Mathematics*, 114:297–308.

Karmarkar, N., Resende, M.G.C., and Ramakrishnan, K.G. (1991). An interior point approach to solve computationally difficult set covering problems. *Mathematical Programming*, 52:597–618.

Kim, S. and Kojima, M. (2001). Second order cone programming relaxations of nonconvex quadratic optimization problems. *Optimization Methods & Software*, 15:201–224.

Kleinberg, J. and Goemans, M. (1998). The Lovász theta function and a semidefinite programming relaxation of vertex cover. *SIAM Journal on Discrete Mathematics*, 11:196–204.

Kocvara, M. and Stingl, M. (2003). PENNON: A code for convex nonlinear and semidefinite programming. *Optimization Methods & Software*, 18:317–333.

Kojima, M., Shindoh, S., and Hara, S. (1997). Interior-point methods for the monotone linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7:86–125.

Kojima, M. and Tuncel, L. (2000). Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM Journal on Optimization*, 10:750–778.

Krishnan, K. and Mitchell, J.E. (2003a). Properties of a cutting plane method for semidefinite programming. Technical Report, Department of Computational & Applied Mathematics, Rice University.

Krishnan, K. and Mitchell, J.E. (2003b). An unifying survey of existing cutting plane methods for semidefinite programming. *Optimization Methods & Software*, to appear; AdvOL-Report No. 2004/1, Advanced Optimization Laboratory, McMaster University.

Krishnan, K. and Mitchell, J.E. (2004). Semidefinite cut-and-price approaches for the maxcut problem. AdvOL-Report No. 2004/5, Advanced Optimization Laboratory, McMaster University.

Krishnan, K., Pataki, G., and Zhang, Y. (2004). A non-polyhedral primal active set approach to semidefinite programming. Technical Report, Dept. of Computational & Applied Mathematics, Rice University, forthcoming.

Lasserre, J.B. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11:796–817.

Lasserre, J.B. (2002). An explicit exact SDP relaxation for nonlinear 0-1 programs. *SIAM Journal on Optimization*, 12:756–769.

Laurent, M. (1998). A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems. In : *Topics in Semidefinite and Interior Point Methods*, The Fields Institute for Research in Mathematical Sciences, Communications Series, vol. 18, American Mathematical Society, Providence, RI, AMS.

Laurent, M. (2003). A comparison of the Sherali-Adams, Lovász–Schrijver and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28:470–496.

Laurent, M. (2004). Semidefinite relaxations for max-cut. In: M. Grötschel (ed.), *The Sharpest Cut, Festschrift in honor of M. Padberg's 60th birthday*, pp. 291–327. MPS-SIAM.

Laurent, M. and Rendl, F. (2003). Semidefinite programming and integer programming. Technical Report PNA-R0210, CWI, Amsterdam.

Lemaréchal, C. and Oustry, F. (1999). Semidefinite relaxations and Lagrangian duality with applications to combinatorial optimization. Technical Report RR-3710, INRIA Rhone-Alpes.

Lovász, L. (1979). On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7.

Lovász, L. and Schrijver, A. (1991). Cones of matrices and set functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190.

Luo, Z.Q. and Sun, J. (1998). An analytic center based column generation method for convex quadratic feasibility problems. *SIAM Journal on Optimization*, 9:217–235.

Mahajan, S. and Hariharan, R. (1999). Derandomizing semidefinite programming based approximation algorithms. *SIAM Journal on Computing*, 28:1641–1663.

Mitchell, J.E. (2000). Computational experience with an interior point cutting plane algorithm. *SIAM Journal on Optimization*, 10:1212–1227.

Mitchell, J.E. (2003). Polynomial interior point cutting plane methods. *Optimization Methods & Software*, 18:507–534.

Mitchell, J.E. (2001). Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems. *Journal of Combinatorial Optimization*, 5:151–166.

Mitchell, J.E. and Borchers, B. (1996). Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. *Annals of Operations Research*, 62:253–276.

Mitchell, J.E., Pardalos, P., and Resende, M.G.C. (1998). Interior point methods for combinatorial optimization. *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, 1:189 – 297.

Mitchell, J.E. and Ramaswamy, S. (2000). A long step cutting plane algorithm for linear and convex programming. *Annals of Operations Research*, 99:95 – 122.

Mitchell, J.E. and Todd, M.J. (1992). Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, 56:245 – 284.

Mittleman, H.D. (2003). An independent benchmarking of SDP and SOCP software. *Mathematical Programming*, 95:407 – 430.

Mokhtarian, F.S. and Goffin, J.L. (1998). A nonlinear analytic center cutting plane method for a class of convex programming problems. *SIAM Journal on Optimization*, 8:1108 – 1131.

Monteiro, R.D.C. (1997). Primal-dual path following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7:663 – 678.

Monteiro, R.D.C. (2003). First and second order methods for semidefinite programming. *Mathematical Programming*, 97:209 – 244.

Motzkin, T.S. and Strauss, E.G. (1965). Maxima for graphs and a new proof of a theorem of Turan. *Canadian Journal of Mathematics*, 17:533 – 540.

Muramatsu, M. and Suzuki, T. (2002). A new second-order cone programming relaxation for maxcut problems. *Journal of Operations Research of Japan*, to appear.

Murthy, K.G. and Kabadi, S.N. (1987). Some NP-complete problems in quadratic and linear programming. *Mathematical Programming*, 39:117 – 129.

Nakata, K. Fujisawa, K., Fukuda, M., Kojima, M., and Murota, K. (2003). Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results. *Mathematical Programming*, 95:303 – 327.

Nemirovskii, A.S. and Yudin, D.B. (1983). *Problem Complexity and Method Efficiency in Optimization*. John Wiley.

Nesterov, Y.E. (1998). Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9:141 – 160.

Nesterov, Y.E. (2000). Squared functional systems and optimization problems. In: J.B.G. Frenk, C. Roos, T. Terlaky, and S. Zhang (eds.). *High Performance Optimization*, pp. 405 – 440. Kluwer Academic Publishers.

Nesterov Y.E. and Nemirovskii, A. (1994). *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Math-

ematics, SIAM, Philadelphia, PA.

Nesterov, Y.E. and Todd, M.J. (1998). Primal dual interior point methods for self-scaled cones. *SIAM Journal on Optimization*, 8:324–364.

Optimization Online. `http://www.optimization-online.org`

Oskoorouchi, M. and Goffin, J.L. (2003a). The analytic center cutting plane method with semidefinite cuts. *SIAM Journal on Optimization*, 13:1029–1053.

Oskoorouchi, M. and Goffin, J.L. (2003b). An interior point cutting plane method for convex feasibility problems with second order cone inequalities. *Mathematics of Operations Research*, to appear; Technical Report, College of Business Administration, California State University, San Marcos.

Oustry, F. (2000). A second order bundle method to minimize the maximum eigenvalue function. *Mathematical Programming*, 89:1–33.

Papadimitriou, C.H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall.

Parrilo, P.A. (2000). *Structured Semidefinite Programs and Semialgebraic Methods in Robustness and Optimization.* Ph.D. Thesis, California Institute of Technology.

Parrilo, P.A. (2003). Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96:293–320.

Pataki, G. (1996a). Cone-LPs and semidefinite programs: geometry and simplex type method. *Proceedings of the 5th IPCO Conference*, pp. 162–174. Lecture Notes in Computer Science, vol. 1084, Springer.

Pataki, G. (1996b). *Cone Programming and Nonsmooth Optimization: Geometry and Algorithms.* Ph.D. Thesis, Graduate School of Industrial Administration, Carnegie Mellon University.

Pataki, G. (1998). On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23:339–358.

Poljak, S., Rendl, F., and Wolkowicz, H. (1995). A recipe for semidefinite relaxations for $\{0,1\}$-quadratic programming. *Journal of Global Optimization*, 7:51–73.

Poljak, S. and Tuza, Z. (1994). The expected relative error of the polyhedral approximation of the maxcut problem. *Operations Research Letters*, 16:191–198.

Porkoláb, L. and Khachiyan, L. (1997). On the complexity of semidefinite programs. *Journal of Global Optimization*, 10:351–365.

Powers, V. and Reznick, B. (2001). A new bound for Polya's theorem with applications to polynomials positive on polyhedra. *Journal of Pure and Applied Algebra*, 164:221–229.

Prajna, S., Papachristodoulou, A., and Parrilo, P.A. (2002). SOSTOOLS: Sum of squares optimization toolbox for MATLAB. `http://www.cds.caltech.edu/sostools`.

Putinar, M. (1993). Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42:969 – 984.

Quist, A.J., De Klerk, E., Roos, C., and Terlaky, T. (1998). Copositive relaxations for general quadratic programming. *Optimization Methods and Software*, 9: 185 – 209.

Ramana, M. (1993). *An Algorithmic Analysis of Multiquadratic and Semidefinite Programming Problems.* Ph.D. Thesis, The John Hopkins University.

Ramana, M. (1997). An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77:129 – 162.

Ramana, M., Tuncel, L., and Wolkowicz, H. (1997). Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7:641 – 662.

Rendl, F. and Sotirov, R. (2003). Bounds for the quadratic assignment problem using the bundle method. Technical Report, University of Klagenfurt, Universitaetsstrasse 65-67, Austria.

Renegar, J. (2001). *A Mathematical View of Interior-Point Methods in Convex Optimization.* MPS-SIAM Series on Optimization.

Roos, C., Terlaky, T., and Vial, J.P. (1997). *Theory and Algorithms for Linear Optimization: An Interior Point Approach.* John Wiley & Sons, Chichester, England.

Schrijver, A. (1979). A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, 25:425 – 429.

Schrijver, A. (1986). *Theory of Linear and Integer Programming.* Wiley-Interscience, New York.

Seymour, P.D. (1981). Matroids and multicommodity flows. *European Journal of Combinatorics*, 2:257 – 290.

Sherali, H.D. and Adams, W.P. (1990). A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411 – 430.

Shor, N. (1998). *Nondifferentiable Optimization and Polynomial Problems.* Kluwer Academic Publishers.

Sturm, J.F. (1997). *Primal-Dual Interior Point Approach to Semidefinite Programming.* Tinbergen Institute Research Series, vol. 156, Thesis Publishers, Amsterdam, The Netherlands; Also in: Frenk et al. (eds.), *High Performance Optimization*, Kluwer Academic Publishers.

Sturm, J.F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625 – 653.

Sun, J., Toh, K.C., and Zhao, G.Y. (2002). An analytic center cutting plane method for the semidefinite feasibility problem. *Mathematics of Operations Research*, 27:332 – 346.

Terlaky, T. (ed.) (1996). *Interior Point Methods of Mathematical Programming*. Kluwer Academic Publishers.

Todd, M.J. (1999). A study of search directions in interior point methods for semidefinite programming. *Optimization Methods and Software*, 12:1 – 46.

Todd, M.J. (2001). Semidefinite optimization. *Acta Numerica*, 10:515 – 560.

Todd, M.J., Toh, K.C., and Tütüncü, R.H. (1998). On the Nesterov – Todd direction in semidefinite programming. *SIAM Journal on Optimization*, 8:769 – 796.

Toh, K.C. (2003). Solving large scale semidefinite programs via an iterative solver on the augmented system. *SIAM Journal on Optimization*, 14:670 – 698.

Toh, K.C. and Kojima, M. (2002). Solving some large scale semidefinite programs via the conjugate residual method. *SIAM Journal on Optimization*, 12:669 – 691.

Toh, K.C., Zhao, G.Y., and Sun, J. (2002). A multiple-cut analytic center cutting plane method for semidefinite feasibility problems. *SIAM Journal on Optimization*, 12:1026 – 1046.

Tütüncü, R.H., Toh, K.C., and Todd, M.J. (2003). Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95:189 – 217.

Vaidya, P.M. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73:291 – 341, 1996.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38:49 – 95.

Vavasis, S.A. (1991). *Nonlinear Optimization*. Oxford Science Publications, New York.

Warners, J.P., Jansen, B., Roos, C., and Terlaky, T. (1997a). A potential reduction approach to the frequency assignment problem. *Discrete Applied Mathematics*, 78:252 – 282.

Warners, J.P., Jansen, B., Roos, C., and Terlaky, T. (1997b). Potential reduction approaches for structured combinatorial optimization problems. *Operations Research Letters*, 21:55 – 65.

Wright, S.J. (1997). *Primal-Dual Interior Point Methods*. SIAM, Philadelphia, PA.

Wolkowicz, H., Saigal, R., and Vandenberghe, L. (2000). *Handbook on Semidefinite Programming*, Kluwer Academic Publishers.

Ye, Y. (1997). *Interior Point Algorithms: Theory and Analysis.* John Wiley & Sons, New York.

Ye, Y. (1999). Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, 84:219 – 226.

Ye, Y. (2001). A .699 approximation algorithm for max bisection. *Mathematical Programming*, 90:101 – 111.

Zhang, Y. (1998). On extending some primal-dual interior point algorithms from linear programming to semidefinite programming. *SIAM Journal on Optimization*, 8:365 – 386.

Zwick, U. (1999). Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to maxcut and other problems. *Proceedings of 31st STOC*, pp. 496 – 505.

Chapter 6

BALANCING MIXED-MODEL SUPPLY CHAINS

Wieslaw Kubiak

Abstract This chapter studies balancing lean, mixed-model supply chains. These supply chains respond to customers' demand by setting rates for delivery of each model and pull supplies for model production from upstream suppliers whenever needed. The chapter discusses algorithms for obtaining balanced model delivery sequences as well as suppliers option delivery and productions sequences. It discusses various factors that shape these sequences. The chapter also explores some insights into the structure and complexity of the sequences gained through the concept of balanced words developed in word combinatorics. The chapter discusses open problems and further research.

1. Introduction

Benchmark supply chains offer their members a sustainable competitive advantage through difficult to replicate business processes. The growing awareness of this fact has made supply chains the main focus of successful strategies for an increasing number of business enterprises, see Shapiro (2001), Bowersox et al. (2002) and Simchi-Levi et al. (2003).

The main insight gained through preliminary research on supply chains is that information sharing between different nodes of a chain counteracts harmful effects of unbalanced and unsynchronized supply and demand in the chain (Lee et al., 1997). This shared information includes both demand and production patterns as well as, though less often, capacity constraints. Improved balance of supply and demand in the chain achieved by sharing information reduces inventories and shortages throughout the chain and consequently allows the chain members to benefit from lower costs.

A *mixed-model* supply chain is intended to deliver a large number of customized models of a product (for example a car or a PC computer)

to customers. Each model is differentiated from other models by its option and supplier *content*. The main objective of such chain is to keep the supply of each model as close to its *demand* as possible. For instance, if the chain is to supply three models a, b and c such that the demand for a is 50%, for b 30%, and for c the remaining 20% of the total demand for the product, then the chain should ideally produce and deliver each model at the rates 0.5, 0.3 and 0.2, respectively. This has reportedly been the main goal of many benchmark lean, mixed-model supply chains, see for example an excellent account of Toyota just-in-time supply chain by Monden (1998). Accordingly, the chain sets its model delivery sequence, that is the order in which it intends to deliver the models to its customers, to follow the rate of demand for each model as closely as possible at any moment during the sequence time horizon. By doing so the chain satisfies the customer demands for a variety of models without holding large inventories or incurring large shortages of the models.

Due to the "pull" synchronization of lean supply chains, once the model delivery sequence is fixed at the final (or model) level of the chain, the option delivery sequences at all other levels are also inherently fixed. Consequently, suppliers have to precisely follow the delivery sequence of each option they deliver to the next level of the chain. The model delivery sequence is thus a *pace-maker* for the whole chain. The supply chain pace is set by the *external* demand through the demand rates for various models and the model delivery sequences are designed so that the actual rates *deviate* from these rates only minimally. Since the model delivery sequence is discrete not continuous there always will be some deviation from demand rates. Furthermore, since this pace is set for the chain according to external demand rates, it is generally independent of the *internal* capacity constraints of supply chain. These capacity constraints, unfortunately, *distort* the delivery sequence. For instance, to address capacity constraints at a supplier node the model delivery sequence may be set so that models supplied by the supplier be paced at the rate 1:10, meaning at most one out of each 10 models in the sequence should be supplied by the supplier.

These two main factors, external demand rates and internal capacity constraints, shape the model delivery sequence so that it features different models evenly spread throughout the sequence. This form of the sequence, however, may remain at odds with the most desirable supplier *production* sequence. The latter's goal, being upstream the supply chain, is often to take advantage of the *economies of scale* by reducing setup costs incurred by frequent switching production from one option to another. The supplier prefers long runs or *batches* over short passed from

the model level. The model level being closer to customer can hardly afford the luxury of long production runs. To minimize his costs the supplier maintains some inventory of finished options that allows him to *batch* together few orders of the same option. Therefore, the supplier needs to decide which orders to batch and how to schedule the batches to meet all deadlines imposed by model delivery sequence and, at the same time, to minimize the number of setups.

The chapter is organized as follows. Section 2 formally defines lean, mixed-model supply chains. Section 3, reviews algorithms for the model variation problem which consists in generating model delivery sequences to minimize deviations between the model demand and supply levels. Section 5 shows how much this deviation increases for suppliers upstream the supply chain. Section 4 introduces and explores a link between model delivery sequences and *balanced* words. The latter have been shown to minimize expected workload of resources in computing and communication networks by Altman et al. (2000) and thus appear promising for balancing mixed-model supply chain as well. In balanced words the numbers of occurrences of each letter in any two of their factors of the same size differ by at most one. These words feature a number desirable properties, for instance there is only polynomial number of distinct factors of a given size in any balanced word. However, one of the main insights gained from the famous *Frankel's Conjecture* for balanced words is that they can only be built for very special sets of model demand rates. Therefore, model delivery sequences being balanced words are extremely rare in practice. Interestingly, it is always possible to obtain a 3-balanced sequence for any set of demand rates. Section 6 shows that the incorporation of supplier's temporary capacity constraints into the model delivery sequence renders the model variation problem NP-hard in the strong sense. The section also reviews algorithms for this extended problem. Section 7 discusses minimization of the number of setups in delivery feasible supplier production sequences. These production sequences can be converted into required delivery sequences with the use of an inventory buffer of limited size. We show that obtaining such sequences with minimum number of setups is NP-hard in the strong sense. However, we prove that for fixed buffer size this can be done in polynomial time. Finally, Section 8 gives concluding remarks and directions for further research.

2. Lean, mixed-model supply chains

A mixed-model supply chain has a set $\{0, 1, \ldots, \mathcal{S}\}$ of suppliers. The supplier s offers supplies from its list $\mathcal{S}_s = \{(s, 1), \ldots, (s, n_s)\}$ of n_s

supplies. The supplies of different suppliers are connected by directed arcs as follows. There is an arc from (s_i, p) to (s_j, q) if and only if s_i asks s_j to supply q for its p. The arc $((s_i, p), (s_j, q))$ is weighted by the number (or amount) of q needed for a *unit* of p. The set of supplies $\bigcup_{s=0}^{\mathcal{S}} \mathcal{S}_s$ and the set of arcs $\mathcal{A}$ between supplies make up a weighted, acyclic digraph. Without loss of generality we shall assume that $s_i < s_j$ for any arc $((s_i, p), (s_j, q))$ in this graph. The supplies $S_0 = \{(0, 1), \ldots, (0, n_0)\}$ at *Level 1* will be called *models*. For simplicity, we denote model $(0, j)$ by j and the number of models n_0 by n. To avoid duplicates in the supply chain, we assume that any two nodes of the digraph have different out-sets and no node has out-degree 1. In fact we assume that the digraphs are *multistage* digraphs, as virtually all supply chains appear to have this structure simplifying feature, see Shapiro (2001); Bowersox et al. (2002), and Simchi-Levi et al. (2003).

Each path p from model m to (s, i) represents a demand for (s, i) originating from m. The size of this demand is the *product* of all weights along the path. Therefore, the total demand for (s, i) originating from m is the sum of path demands over all paths from m to (s, i). For instance, in Figure 6.1, there are two paths from model 1 to $(4, 1)$ both with weight 1, therefore the total demand for $(4, 1)$ originating from 1 equals 2. Each supplier s aggregates its demand over all supplies on its list $\mathcal{S}_s$. For supplier 4 the demand originating from model 1 is (112), from model 2, (12233), and from model 3, (233). In our notation, supply i for a given model is listed the number of times equal to the unit demand for i originating from the model. Each of these lists will be referred to as a *kit* to emphasize the fact that suppliers do not deliver an individual part or a subassembly required by models but rather a complete collection required by the model, a common practice in manufacturing (Bowersox et al., 2002). Thus, model 1 needs the entire kit (112) from supplier 4 rather than two 1's and one 2 delivered separately. We shall also refer to kit as *option*. Notice that a model may require at most one kit from a supplier. The supplier *content* of models is defined by an n by $\mathcal{S} + 1$ matrix $\mathcal{C}$, where $\mathcal{C}_{is} = 1$ if model i requires a kit (option) from supplier s and $\mathcal{C}_{is} = 0$ otherwise.

We assume that the supply chain operates in a *pull* mode. That is any supply at a *higher* level is drawn as needed by a *lower* level. Therefore, it is a sequence of models at *Level 1* that determines the *delivery* sequence of each supplier at every level higher than 1 (upstream) and the supplier must exactly follow this delivery sequence. For instance a sequence of models 1231121321 at *Level 1* results in the option delivery sequence

$$(12)(1)(12)(12)(1)(12)(1)(12)$$

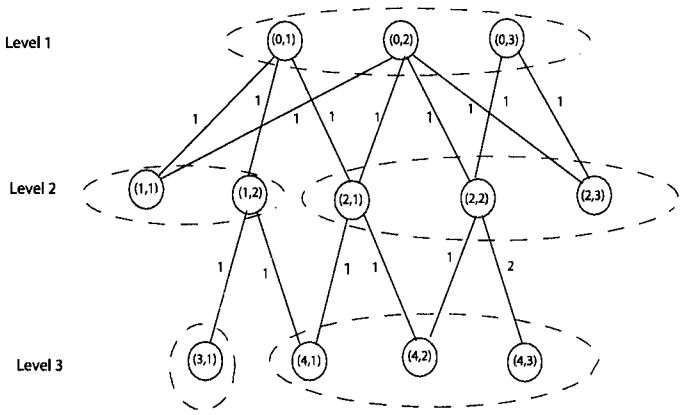


Figure 6.1. Mixed-model supply chain with three levels and five suppliers (or chain nodes): one at Level 1 supplying three models, two at Level 2, and two at Level 3.

for supplier 1 at *Level 2*, and the option delivery sequence

$$(112)(12233)(233)(112)(112)(12233)(112)(233)(12233)(112)$$

for supplier 4 at *Level 3*. The demand for model j is denoted by d_j and assumed given. The demand for any other supply can easily be *derived* from demand for models and the option content of each model.

3. The model rate variation problem

This section formulates the model variation problem and presents algorithms for its solution. For *models* $1, \ldots, n$ of a product with their positive integer *demands* $d_1, \ldots, d_n$ during a time horizon, for instance a daily, a weekly or a monthly demand, the demand rate for model i is defined as the ratio $r_i = d_i/D$, where $D = \sum_{i=1}^{n} d_i$. We require the *actual* delivery level of each model to remain as close as possible to the *ideal* level, $r_i k$, $k = 1, \ldots, D$, at any moment k during the time horizon. Conveniently, the rates sum up to 1 and consequently can be also looked at as the probabilities of a discrete *probability* distribution over models in a possible stochastic analysis of the chains, however, we shall not proceed with this analysis here leaving it for further research.

Figure 6.2 illustrates the problem for an instance with model a produced along with two other models b and c. In the example, the demands for models a, b and c are $d_a = 5$, $d_b = 3$, and $d_c = 2$, respectively. Consequently, the demand rates for the three models are $r_a = 0.5$, $r_b = 0.3$, and $r_c = 0.2$. The ideal delivery level for a is set by the straight line $0.5k$ in Figure 6.2. For convenience, we assume that k takes on real values in the interval $[0, D]$. The actual delivery levels, on the other hand,

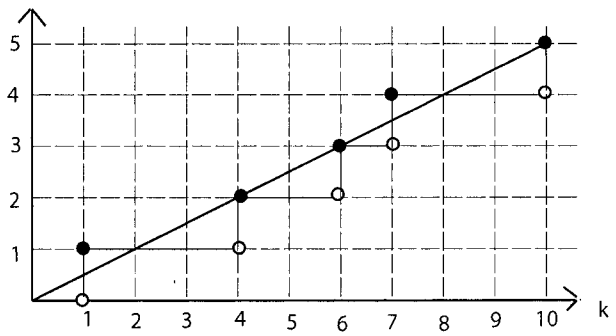


Figure 6.2. The target $0.5k$ line and the actual delivery level for model a with its copies in positions 1, 4, 6, 7 and 10 of the delivery sequence.

depend on the sequence in which models a, b and c are delivered. Here, for instance,we assume the following delivery sequence

$$abcabaacba.$$

This sequence keeps delivery levels for all models *simultaneously* within 1 unit of their respective target levels, as the reader can easily be convinced by Figure 6.2 for model a.

Following Monden (1998); Miltenburg (1989), and Kubiak (1993) we shall formulate the problem as the problem of minimizing the *total deviation* of the actual delivery levels from the target levels as follows.

Let $f_1, \ldots, f_n$ be n convex and symmetric functions of a single variable, the deviation, all assuming minimum 0 at 0. Find a sequence $S = s_1, \ldots, s_D$, of models $1, \ldots, n$, where model i occurs exactly d_i times that minimizes the following objective function,

$$F(S) = \sum_{i=1}^{n} \sum_{k=1}^{D} f_i(x_{ik} - r_i k), \tag{6.1}$$

where x_{ik} the number of model i occurrences (or the number of model i *copies*) in the prefix $s_1, \ldots, s_k$ of S.

An optimal solution to this problem can be found by reducing the problem to the assignment problem (Kubiak and Sethi, 1991, 1994). The main idea behind this reduction is as follows. We define $Z_j^i = \lceil (2j-1)/2r_i \rceil$ as the *ideal position* for the jth copy of model i. Though sequencing the copies in their ideal positions minimizes $F(S)$, it is likely infeasible since more than one copy may compete for the same position, which can only be occupied by one copy. Therefore, we need to resolve the competition in an optimal fashion so to minimize $F(S)$. Fortunately,

this can be done efficiently by solving an *assignment* problem, which we now define.

Let $X = \{(i,j,k) \mid i = 1,\ldots,n; j = 1,\ldots,d_i; k = 1,\ldots,D\}$. Define cost $C^i_{jk} \geq 0$ for $(i,j,k) \in X$ as follows:

$$C^i_{jk} = \begin{cases} \sum_{l=k}^{Z^i_j - 1} \psi^i_{jl} & \text{if } k < Z^i_j, \\ 0, & \text{if } k = Z^i_j, \\ \sum_{l=Z^i_j}^{k-1} \psi^i_{jl}, & \text{if } k > Z^i_j, \end{cases} \tag{6.2}$$

where for symmetric functions f_i, $Z^i_j = \lceil (2j-1)/2r_i \rceil$ is the ideal position for the jth copy of product i, and

$$\begin{aligned} \psi^i_{jl} &= |f_i(j - lr_i) - f_i(j - 1 - lr_i)| \\ &= \begin{cases} f_i(j - lr_i) - f_i(j - 1 - lr_i), & \text{if } l < Z^i_j, \\ f_i(j - 1 - lr_i) - f_i(j - lr_i), & \text{if } l \geq Z^i_j. \end{cases} \end{aligned} \tag{6.3}$$

Notice that the point $(2j-1)/2r_i$ is the crossing point of $f_i(j-1-kr_i)$ and $f_i(j-kr_i)$, $j = 1,\ldots,d_i$.

Let $S \subseteq X$, we define $V(S) = \sum_{(i,j,k)\in S} C^i_{jk}$, and call S *feasible* if it satisfies the following three constraints:

($\mathcal{A}$) For each k, $k = 1,\ldots,D$, there is exactly one pair (i,j), $i = 1,\ldots,n$; $j = 1,\ldots,d_i$ such that $(i,j,k) \in S$.

($\mathcal{B}$) For each pair (i,j), $i = 1,\ldots,n$; $j = 1,\ldots,d_i$, there is exactly one k, $k = 1,\ldots,D$, such that $(i,j,k) \in S$.

($\mathcal{C}$) If $(i,j,k),(i,j',k') \in S$ and $k < k'$, then $j < j'$.

Constraints ($\mathcal{A}$) and ($\mathcal{B}$) are the well known assignment problem constraints, constraints ($\mathcal{C}$) impose an order on copies of a product and will be elaborated upon later.

Consider any set S of D triples (i,j,k) satisfying ($\mathcal{A}$), ($\mathcal{B}$), and ($\mathcal{C}$). Let $\alpha(S) = \alpha(S)_1,\ldots,\alpha(S)_D$, where $\alpha(S)_k = i$ if $(i,j,k) \in S$ for some j, be a sequence corresponding to S. By ($\mathcal{A}$) and ($\mathcal{B}$) sequence $\alpha(S)$ is feasible for $d_1,\ldots,d_n$. The following theorem ties $F(\alpha(S))$ and $V(S)$ for any feasible S.

THEOREM 6.1 *We have*

$$F(\alpha(S)) = V(S) + \sum_{i=1}^{n} \sum_{k=1}^{D} \inf_j f_i(j - kr_i). \tag{6.4}$$

Proof. See Kubiak and Sethi (1994). □

Notice that $\sum_{i=1}^{n}\sum_{k=1}^{D}\inf_j f_i(j-kr_i)$ in (6.4) is constant, that is independent of S. An optimal set S can not be found by simply solving the assignment problem with constraints $(\mathcal{A})$ and $(\mathcal{B})$, and the costs as in (6.2), for which many efficient algorithms exist. The reason being constraint $(\mathcal{C})$, which is not of the assignment type. Informally, $(\mathcal{C})$ ties up copy j of a product with the j-th ideal position for the product and it is necessary for Theorem 6.1 to hold. In other words, for a set S satisfying $(\mathcal{A})$ and $(\mathcal{B})$ but not $(\mathcal{C})$ we may generally have inequality in (6.3). However, the following theorem remedies this problem.

THEOREM 6.2 *If S satisfies $(\mathcal{A})$ and $(\mathcal{B})$, then S' satisfying $(\mathcal{A})$, $(\mathcal{B})$ and $(\mathcal{C})$, and such that*

$$V(S) \geq V(S'),$$

can be constructed in $O(D)$ steps. Furthermore, each product occupies the same positions in $\alpha(S')$ as it does in $\alpha(S)$.

Proof. See Kubiak and Sethi (1994). □

We have the following two useful properties of optimal solutions. First, the set of optimal solutions $\mathcal{S}^*$ includes *cyclic* solutions whenever functions f_i are symmetric. That is, if the greatest common divisor $g = \gcd(d_1, \ldots, d_n)$ of demands $d_1, \ldots, d_n$ is greater than 1, then the optimal solution for demands $d_1/g, \ldots, d_n/g$ repeated g times gives an optimal solution for $d_1, \ldots, d_n$ (Kubiak, 2003b). Second, if $\alpha \in \mathcal{S}^*$, then $\alpha^R \in \mathcal{S}^*$ where α^R is a mirror reflection of α.

This approach to solving the model variation problem applies to any l_p-norm ($F = l_p$), in particular to l_∞-norm. In the latter case the approach minimizes maximum deviation where the objective function becomes

$$H(S) = \min \max_{i,k} f_i(x_{ik} - r_i k).$$

Steiner and Yeomans (1993) considered the same absolute deviation function, $f_i(x_{ik} - r_i k) = |x_{ik} - r_i k|$, for all models, and suggested an algorithm based on the following theorem of Steiner and Yeomans (1993); Brauner and Crama (2001), and Kubiak (2003c).

THEOREM 6.3 *If a sequence S with maximum absolute deviation not exceeding B exists, then copy j of model i, $i = 1, \ldots, n$ and $j = 1, \ldots, d_i$ occupies a position in the interval $[E(i,j), L(i,j)]$, where*

$$E(i,j) = \left\lceil \frac{j - B}{r_i} \right\rceil$$

and

$$L(i,j) = \left\lfloor \frac{j-1+B}{r_i} + 1 \right\rfloor.$$

The *feasibility* test for a given B is based on Glover (1967) Earliest Due Date algorithm for testing the existence of a perfect matching in a *convex* bipartite graph G. The graph $G = (V_1 \cup V_2, \mathcal{E})$ is made of the set $V_1 = \{1, \ldots, D\}$ of positions and the set $V_2 = \{(i,j) \mid i = 1, \ldots, n; j = 1, \ldots, d_i\}$ of copies. The edge $(k, (i,j)) \in \mathcal{E}$ if and only if $k \in [E(i,j), L(i,j)]$. The algorithm assigns position k to the copy (i,j) with the smallest value of $L(i,j)$ among all the available copies with $(k, (i,j)) \in \mathcal{E}$, if such exist. Otherwise, no sequence for B exists. The results of Brauner and Crama (2001); Meijer (1973), and Tijdeman (1980) show the following bounds on the optimal B^*.

THEOREM 6.4 *The optimal value B^* satisfies the following inequalities*

$$B^* \geq \frac{1}{\Delta_i} \left\lfloor \frac{\Delta_i}{2} \right\rfloor,$$

for $i = 1, \ldots, n$, where $\Delta_i = D/\gcd(d_i, D)$ and

$$B^* \leq 1 - \max\left\{ \frac{1}{D}, \frac{1}{2(n-1)} \right\}.$$

The quota methods of *apportionment* introduced by Balinski and Young (1982), see also Balinski and Shahidi (1998), and studied by Still (1979) proved the existence of solutions with $B^* < 1$ already in the seventies.

Theorem 6.4 along with the fact that the product DB^* is integer allow the binary search to find the optimum B^* and the corresponding matching by doing $O(\log D)$ tests for B.

Other efficient algorithms based on the reduction to the *bottleneck* assignment problem were suggested, by Kubiak (1993) and developed by Bautista et al. (1997).

Corominas and Moreno (2003) recently observed that optimal solutions for the total deviation problem may result in maximum deviation being greater than 1 for some instances, they give $n = 6$ models $d_1 = d_2 = 23$, and $d_3 = d_4 = d_5 = d_6 = 1$ as an example. However, it is worth noticing that a large computational study, Kovalyov et al. (2001), tested 100,000 randomly selected instances *always* finding that optimal solution to the total absolute deviation problem have maximum absolute deviation less or equal 1, which indicates that most solutions minimizing total deviation will have maximum deviation $B \leq 1$.

4. Balanced words and model delivery sequences

This section explores some insights into the solutions to the model rate variation problem gained from combinatorics on words. We use the terminology and the notation borrowed from this area which we now briefly review as they will be also used in the following sections.

The models $\{1,\ldots,n\}$ will be viewed as the letters of a finite alphabet $\mathcal{A} = \{1,\ldots,n\}$. We consider both finite and infinite words over $\mathcal{A}$. A solution to the model variation problem will then be viewed a finite word of length D on $\mathcal{A}$, where the letter i occurs exactly d_i times. This word can be concatenated *ad infinitum* to obtain a periodic, infinite word on $\mathcal{A}$. We write $S = s_1 s_2 \ldots$, where $s_i \in \mathcal{A}$ is the i-th letter of S. The index i will be called the position of the letter s_i in the word s. A *factor* of length (size) $b \geq 0$ of S is word x such that $x = s_i \ldots s_{i+b-1}$. The length of word x is denoted by $|x|$. The empty word is the word of length 0. If x is a factor of a word, then $|x|_i$ denotes the number of i's in x.

We recall from Section 3 that sequencing copy j of model i in its ideal position $\lceil (2j-1)/2r_i \rceil$ minimizes both the total deviation and the maximum deviation, however, leads to an infeasible solution whenever more than one copy *competes* for the same ideal position in the sequence. The algorithms discussed in Section 3 show how to efficiently resolve the conflicts so that the outcome is an optimal sequence, minimizing either total or maximum deviations. Let us now drop the ceiling in the definition of ideal positions and consider an infinite, periodic sequence of numbers $(2j-1)/2r_i = jD/d_i - D/2d_i = (j-1)D/d_i + D/2d_i$. We build an infinite word on $\mathcal{A}$ using these numbers as follows. Label the points $\{(j-1)D/d_i + D/2d_i, j \in \mathbb{N}\}$ by the letter i. Consider $\bigcup_{i=1}^{n} \{(j-1)D/d_i + D/2d_i, j \in \mathbb{N}\}$ and the corresponding sequence of labels. Each time there is a tie we chose i over j whenever $i < j$. Notice that here higher priority is always given to a lower index whenever a conflict needs to be settled. This way we obtain what Vuillon (2003) refers to as an *hypercubic billiard* word with angle vector $\alpha = (D/d_1, D/d_2, \ldots, D/d_n)$ and starting point $\beta = (D/2d_1, D/2d_2, \ldots, D/2d_n)$. Vuillon (2003) proves the following theorem.

THEOREM 6.5 *Let x be an infinite hypercubic billiard word in dimension n of angle α and starting point β. Then x is $(n-1)$-balanced.*

The c-balanced words, $c > 0$, are defined as follows.

DEFINITION 6.1 (C-BALANCED WORD) *A* c-balanced word *on alphabet $\{1,2,\ldots,n\}$ is an infinite sequence $S = s_1 s_2 \ldots$ such that*

(1) *$s_j \in \{1,2,\ldots,n\}$ for all $j \in \mathbb{N}$, and*

(2) *if x and y are two factors of S of the same size, then* $\big||x|_i - |y|_i\big| \leq c$, *for all* $i = 1, 2, \ldots, n$.

Theorem 6.5 shows that the *priority* based conflict resolution applied whenever there is a competition for an ideal position results in c being almost of the size of the alphabet, in fact 1 less than this size. However, Jost (2003) proves that the conflict resolution provided by any algorithm minimizing maximum deviation leads to c being constant. He proves the following theorem.

THEOREM 6.6 *For a word S obtained by infinitely repeating a sequence with maximum deviation B for n models with demands $d_1, \ldots, d_n$. We have:*

- *If* $B < \frac{1}{2}$, *then S is 1-balanced.*
- *If* $B < \frac{3}{4}$, *then S is 2-balanced.*
- *If* $B < 1$, *then S is 3- balanced.*

For instance, the infinite word generated by the word

$$abcabaacba$$

is 2-balanced as its maximum deviation equals $\frac{1}{2}$ but not 1-balanced, factors bc and aa differ by 2 on the latter a.

The opposite claim does not hold, for instance, any sequence for n models with their demands all equal 1 is a 1-balanced word though its maximum deviation equals $1 - 1/n$, and thus greater than half for $n \geq 3$. It remains an open question to show whether or not there always is a 2-balanced word for *any* given set of demands $d_1, \ldots, d_n$.

In the hierarchy of balanced words, the 1-balanced words, or just balanced words, have attracted most attention thus far, see Vuillon (2003); Altman et al. (2000) and Tijdeman (2000) for review of recent results on balanced words. Berthé and Tijdeman (2002) observe that the number of balanced words of length m is bounded by a *polynomial* of m, which makes the balanced words very rare. The polynomial complexity of balanced words would reduce a number of possible delivery sequences through the supply chain which could have obvious advantages for their management, as well balanced words would optimally balance suppliers workload according to the results of Altman et al. (2000). However, balanced words turn out to be out of reach in practice. Indeed, according to Frankel's conjecture, Altman et al. (2000) and Tijdeman (2000), there is only *one* such word on n letter alphabet with *distinct* densities.

CONJECTURE 6.1 (FRAENKEL'S CONJECTURE) *There exists a periodic, balanced word on $n \geq 3$ letters with rates $r_1 < r_2 < \cdots < r_n$ if and only if $r_i = 2^{i-1}/(2^n - 1)$.*

Though this conjecture remains open, a simpler one for *periodic*, symmetric and balanced words has recently been proven by Kubiak (2003a), see also Brauner et al. (2002), which indicates that the balanced words will indeed be very rare generally and as the solutions to the model rate variation problem in particular.

THEOREM 6.7 (FRAENKEL'S SYMMETRIC CASE) *There exists a periodic, symmetric and balanced word on $n \geq 3$ letters with rates $r_1 < r_2 < \cdots < r_n$, if and only if the rates verify $r_i = 2^{i-1}/(2^n - 1)$.*

Theorem 6.4 shows that there always is an optimal solution with $B < 1$, and the Theorem 6.6 shows that such solutions are 3-balanced. These two ensure that 3-balanced words can be obtained for any set of demands $d_1, \ldots, d_n$. However, Berthé and Tijdeman (2002) observe the number of c-balanced words of length m is *exponential* in m for any $c > 1$.

5. Option delivery sequences

A supplier s option delivery sequence can be readily obtained from the model delivery sequence S and the supplier content matrix C by *deleting* from S all models i not supplied by s, that is those with $C_{is} = 0$. This deletion increases deviation between the ideal and actual *option* delivery levels for suppliers as we show in this section. Let us first introduce some necessary notation.

- $A_s \subseteq \{1, \ldots, n\}$ — the subset of models supplied by s.
- $A_{sj} \subseteq A_s$ — the subset of models requiring option j of supplier s.
- $r_{sj} = \sum_{m \in A_{sj}} d_m / \sum_{m \in A_s} d_m$.
- $r_{A_{sj}} = (\sum_{m \in A_{sj}} d_m)/D = \sum_{m \in A_{sj}} r_m$.
- $r_{A_s} = (\sum_{m \in A_s} d_m)/D = \sum_{m \in A_s} r_m$.

We notice that

$$r_{sj} = \frac{r_{A_{sj}}}{r_{A_s}}.$$

First, we investigate the maximum deviation in the *option* delivery sequence of supplier s. Supplier s has total derived demand $D_s = \sum_{m \in A_s} d_m$ and the derived demand for its option j equals $d_{sj} = \sum_{m \in A_{sj}} d_m$. A model delivery sequence S with x_{mk} copies of model m out of first k copies delivered results in actual total derived demand $\sum_{m \in A_s} x_{mk}$ for supplier s out of which $\sum_{m \in A_{sj}} x_{mk}$ is demand for option j of s. Therefore, the maximum deviation for the option delivery sequence of supplier s equals

$$\max_{j,k} \left| \sum_{m \in A_{sj}} x_{mk} - r_{sj} \sum_{m \in A_s} x_{mk} \right|. \tag{6.5}$$

However, for S with maximum deviation B^* we have

$$kr_m - B^* \leq x_{mk} \leq kr_m + B^* \tag{6.6}$$

for any model m and k, and consequently

$$kr_{A_{sj}} - |A_{sj}|B^* \leq \sum_{m \in A_{sj}} x_{mk} \leq kr_{A_{sj}} + |A_{sj}|B^*$$

and

$$kr_{A_s} - |A_s|B^* \leq \sum_{m \in A_s} x_{mk} \leq kr_{A_s} + |A_s|B^*.$$

Thus,

$$\sum_{m \in A_{sj}} x_{mk} = kr_{A_{sj}} + \epsilon_{A_{sj}},$$

where $|\epsilon_{A_{sj}}| \leq |A_{sj}|B^*$ and

$$\sum_{m \in A_s} x_{mk} = kr_{A_s} + \epsilon_{A_s},$$

where $|\epsilon_{A_s}| \leq |A_s|B^*$.

Therefore, (6.5) becomes

$$\max_{j,k} \left| kr_{A_{sj}} - \frac{r_{A_{sj}}}{r_{A_s}}(kr_{A_s} + \epsilon_{A_s}) + \epsilon_{A_{sj}} \right| \tag{6.7}$$

or

$$\max_{j,k} |\frac{r_{A_{sj}}}{r_{A_s}}\epsilon_{A_s} - \epsilon_{A_{sj}}|. \tag{6.8}$$

Notice that in fact both ϵ_{A_s} and $\epsilon_{A_{sj}}$ depend on k. Obviously,

$$\epsilon_{A_s} = \epsilon_{A_s \setminus A_{sj}} + \epsilon_{A_{sj}}.$$

Thus,

$$\max_{j,k} \left| \frac{r_{A_{sj}}}{r_{A_s}}\epsilon_{A_s \setminus A_{sj}} - \left(1 - \frac{r_{A_{sj}}}{r_{A_s}}\right)\epsilon_{A_{sj}} \right| \leq \max_j \{r_{sj}|A_s|B^* + (1 - 2r_{sj})|A_{sj}|B^*\} \tag{6.9}$$

but, since $|A_{sj}| \leq |A_s|$ and $1 - 2r_{sj} \leq 1 - r_{sj}$, we have

$$r_{sj}|A_s| + (1 - 2r_{sj})|A_{sj}| \leq |A_s|.$$

Finally,

$$\max_{j,k} \left| \sum_{k \in A_{sj}} x_{mk} - r_{sj} \sum_{m \in A_s} x_{mk} \right| \leq |A_s|B^*. \tag{6.10}$$

We have just proved the following theorem.

THEOREM 6.8 *The maximum deviation of the option delivery sequence for supplier s who supplies $|A_s|$ different models out of n produced may increase $|A_s|$ times in comparison with the maximum deviation of the model delivery sequence.*

Theorems 6.4 and 6.6 show that the model delivery sequence minimizing maximum deviation are 3-balanced. However, Theorem 6.8 proves that the maximum deviation of the option delivery sequence of supplier s grows proportionally to the number of models s supplies. Therefore, the option delivery sequence becomes less balanced. We have the following result.

THEOREM 6.9 *The option delivery sequence for supplier s is $\lfloor 4|A_s|B^*\rfloor$-balanced.*

Proof. For supplier s consider k and k_Δ, $\Delta \geq 1$ such that between k and k_Δ there are exactly Δ copies of models requiring some option from s. That is

$$\sum_{m\in A_s} x_{mk_\Delta} - \sum_{m\in A_s} x_{mk} = \Delta.$$

We then have

$$-|A_s|B^* \leq \sum_{k\in A_{sj}} x_{mk} - r_{sj} \sum_{m\in A_s} x_{mk} \leq |A_s|B^*,$$

and

$$-|A_s|B^* \leq \sum_{k\in A_{sj}} x_{mk_\Delta} - r_{sj} \sum_{m\in A_s} x_{mk_\Delta} \leq |A_s|B^*,$$

which results in

$$-2|A_s|B^* \leq \sum_{k\in A_{sj}} x_{mk_\Delta} - \sum_{k\in A_{sj}} x_{mk} - r_{sj}\Delta \leq 2|A_s|B^*$$

for each k. Therefore, the numbers of option j occurrences in any two supplier s delivery subsequences of length Δ differ by at most $\lfloor 4|A_s|B^*\rfloor$. □

6. Temporal supplier capacity constraints

Thus far, we have required that the model deliver sequence S keeps up with the demand rates for models but ignored the capacity constraints of suppliers in a supply chain. This may render S difficult to implement in

the chain since S may *temporarily* impose too much strain on supplier's resources by setting too high a temporal delivery pace for their options. This section addresses this temporal suppliers capacity constraints. We assume that supplier s is a subject to a *capacity* constraint in the form $p_s : q_s$, which means that *at most* p_s models of S in each consecutive sequence of q_s models of S may need options supplied by s. The problem consists in finding a sequence S of length D over models $\{1, \ldots, n\}$ where i occurs exactly d_i times and which respects capacity constraints for each supplier s. Clearly, in order for a feasible model sequence S to exist the capacity constraints must satisfy the condition $D/q_s p_s \geq \sum_{i \in \{i: S_{sj}=1\}} d_i$ for all s, otherwise the demands d_i for models will not be met. For instance, in the example from Table 6.1 demand for supplier 4 equals 6 which is less than $11 \cdot 2/3$, with $2 : 3$ capacity constraint for supplier 2. Table 6.2 presents a feasible sequence for this example.

We now prove that the problem to decide whether or not there is a model delivery sequence that respects suppliers capacity constraints is NP-complete in the strong sense. This holds even if all suppliers have the same capacity constraints $1 : \alpha$ for some positive integer α, that is for each supplier s at most 1 in each consecutive α models of the model deliver sequence may require an option delivered by s. We refer to the problem as temporal supplier capacity problem. We have the following theorem.

Table 6.1. An instance of the temporary supplier capacity problem.

supplier	*capacity*	*models*					
		1	2	3	4	5	6
1	2:3	1	0	0	0	1	1
2	2:3	0	0	1	1	0	1
3	1:2	1	0	0	0	1	0
4	3:5	1	1	0	1	0	0
5	2:5	0	0	1	0	0	0
demands		2	3	1	1	2	2

Table 6.2. A feasible sequence of models.

supplier	*sequence*										
	2	2	1	3	5	2	2	1	4	5	6
1	0	0	1	0	1	0	1	1	0	1	1
2	0	0	0	1	0	0	0	1	1	0	1
3	0	0	1	0	1	0	1	0	0	1	0
4	1	1	1	0	0	1	1	0	1	0	0
5	0	0	0	1	0	0	0	0	0	0	0

THEOREM 6.10 *The temporal supplier capacity problem is strongly NP-complete.*

Proof. Our transformation is from the graph coloring problem, see Garey and Johnson (1979). Let graph $G = (V, E)$ and $k \geq 2$ make up an instance of the graph coloring problem. Let $|V| = n$ and $|E| = m$. Take k *disjoint* isomorphic copies of G, $G^1 = (V^1, E^1), \ldots, G^k = (V^k, E^k)$. Let $\mathcal{G} = (\mathcal{V} = \bigcup_{i=1}^{k} V^i, \mathcal{E} = \bigcup_{i=1}^{k} E^i)$ be the union of the k copies. Now, consider an independent set S on n nodes, that is the graph $S = (N = \{1, \ldots, n\}, \varnothing)$. Take $k+1$ *disjoint* copies isomorphic of S, $S^1 = (N^1, \varnothing), \ldots, S^k = (N^k, \varnothing)$. Add an edge between any two nodes of $\mathcal{N} = \bigcup_{i=1}^{k+1} N^i$ being in different copies of S to make a graph $\mathcal{S} = (\mathcal{N}, \mathcal{X} = \bigcup_{i \neq j} N_i \times N_j)$. Notice that $N^1, \ldots, N^{k+1}$ are independent sets of $\mathcal{N}$ each with cardinality n. Finally, consider a *disjoint* union of $\mathcal{G}$ and $\mathcal{N}$, that is $\mathcal{H} = \mathcal{G} \cup \mathcal{N} = (\mathcal{V} \cup \mathcal{N}, \mathcal{E} \cup \mathcal{X})$. Clearly, the union has $nk + n(k+1)$ nodes and $mk + k^2 n$ edges, and thus its size is polynomially bounded in n, m and k and consequently polynomial in the size of the input instance of the graph coloring problem.

Consider the *node-arc* incidence matrix I of graph $\mathcal{H}$. In fact, its transposition I^T. The columns of I^T correspond to the nodes of $\mathcal{H}$ and they, in turn, correspond to *models*. The rows of I^T correspond to the edges of $\mathcal{H}$ and they, in turn, correspond to *suppliers*. The demand for each model equals one. The capacity constraint for each supplier in $\mathcal{E}$ is $1 : (n+1)$, and the capacity constraint for each supplier in $\mathcal{X}$ is $1 : (n+1)$ as well. We shall refer to any supplier in $\mathcal{E}$ as the $\mathcal{E}$-supplier, and to any supplier in $\mathcal{X}$ as $\mathcal{X}$-supplier.

(if) Assume there is a coloring of G using no more than k colors. Then, obviously, there is a coloring of G using *exactly* k colors. The coloring defines a partition of V into k independent sets $W_1, \ldots, W_k$. Let $W_j^i \subseteq V^i$ be a copy of the independent set W_j inside of the copy G^i of G. Define the sets

$$\begin{aligned} A_1 &= W_1^1 \cup W_2^2 \cup \cdots \cap W_k^k, \\ A_2 &= W_2^1 \cup W_3^2 \cup \cdots \cap W_1^k, \\ &\vdots \\ A_k &= W_k^1 \cup W_1^2 \cup \cdots \cap W_{k-1}^k. \end{aligned}$$

These sets partition set $\mathcal{V}$, moreover, each of them is an independent set of $\mathcal{G}$ of cardinality n. Given the sets, let us sequence them as follows

$$N^1 A_1 N^2 A_2 \ldots A_k N^{k+1}. \tag{6.11}$$

To obtain a sequence of models we sequence models in each set arbitrarily. Next, we observe that each set N^j is independent thus no $\mathcal{X}$-supplier is used twice by models in N^j. Furthermore, there are n models with no $\mathcal{X}$-supplier between N^j and N^{j+1}, $j = 1, \ldots, k$. Consequently, any two models with an $\mathcal{X}$-supplier are separated by at least n models without this $\mathcal{X}$-supplier, and therefore the sequence (6.11) respects the $1 : (n+1)$ capacity constraint for each $\mathcal{X}$-supplier. Finally, we observe that each set A_j, $j = 1, \ldots, n$ is independent, thus no $\mathcal{E}$-supplier is used twice by models in A_j. Moreover, there are n models with no $\mathcal{X}$-supplier between A^j and A^{j+1}, $j = 1, \ldots, k-1$. Thus, any two models with an $\mathcal{E}$-supplier are separated by at least n models without this $\mathcal{E}$-supplier, and therefore the sequence (6.11) respects the $1 : (n+1)$ capacity constraint for each $\mathcal{E}$-supplier. Therefore, sequence (6.11) is a feasible model sequence in the supplier capacity problem.

(*only if*) Let s be a feasible sequence of models. Let us assume for the time being that s is of the following form

$$S = N^1 M_1 N^2 M_2 \ldots M_k N^{k+1} \tag{6.12}$$

where $\bigcup_{j=1}^{k} M_j = \mathcal{V}$ and $|M_j| = n$ for $j = 1, \ldots, k$. Consider models in V^1 and the sets

$$V_i = M_i \cap V^1, i = 1, \ldots, k.$$

Obviously, $\bigcup_{i=1}^{k} V_i = V^1$ and the sets V_i are independent. Otherwise, there would be an edge (a, b) between some models a and b of some V_i. Then, however, the $\mathcal{E}$-supplier (a, b) would be used by both a and b models in M_i of length n which would make s infeasible by violating the $1 : (n+1)$ capacity constraint for the $\mathcal{E}$-supplier (a, b). Consequently, coloring each V_i with a distinct color would provide a coloring of G^1 using k colors. Since G^1 is an isomorphic copy of G, then the coloring would be a required coloring of G itself.

It remains to show that a feasible sequence of the form (6.12) always exists. To this end, let us consider the following decomposition of s into $2k+1$ subsequences of equal length n,

$$S = \gamma_1 \gamma_2 \ldots \gamma_{2k+1},$$

where

$$\gamma_i = s_{(i-1)n+1} \ldots s_{in}, i = 1, \ldots, 2k+1, \tag{6.13}$$

For each γ_i there is at most one N^j whose models are in γ_i. Otherwise, the $1 : (n+1)$ constraint for some $\mathcal{X}$-supplier would be violated. Consequently, no N^j can share γ_i, $i = 1, \ldots, 2k+1$ with any other N^l, $j \neq l$. However, since there are only $2k+1$ subsequences γ_i, then there

must be N^{j^*} which models *completely* fill in one of the subsequences γ_i. Let us denote this sequence by γ. Neither the subsequence immediately to the left of γ, if any, nor to the right of γ, if any, may include models from $\bigcup_{j\neq j^*,j=1}^{k+1} N^j$. Otherwise, the $1:(n+1)$ constraint for some $\mathcal{X}$-supplier would be again violated. Consequently, there are at most $2k-1$ subsequences with models from $\bigcup_{j\neq j^*,j=1}^{k+1} N^j$ in s, but this again implies the existence of $N^{j^{**}}$, $j^* \neq j^{**}$, which models *completely* fill in one of the subsequences γ_i, say γ^*. Furthermore, neither the subsequence immediately to the left of γ^*, if any, nor to the right of γ^*, if any, may include models from $\bigcup_{j\neq j^{**},j=1}^{k+1} N^j$. By continuing this argument we reach a conclusion that for any feasible s there is a one-to-one mapping f of $\{N^1,\ldots,N^{k+1}\}$ into $\{\gamma_1,\ldots,\gamma_{2k+1}\}$ such that the sequence $f(N^i)$ is made up of models from N^i only, $i=1,\ldots,k+1$. Also, if γ_i and γ_j are mapped into then $|i-j|\geq 2$. This mapping f is only possibly if s is of the form (6.11), which we needed to prove. □

The temporary supplier capacity problem is closely related to the car sequencing problem. The latter was shown NP-complete in the strong sense by an elegant transformation from the Hamiltonian path problem by Gent (1998), though his transformation requires different capacity constraints for different car options. The car sequencing problem is often solved by constraint programming, ILOG (2001). Drexl and Kimms (2001) propose an integer programming model to minimize maximum deviation from optimal positions, which is different from though related to the model variation problem discussed in Section 3, over all sequences satisfying suppliers capacity constraints. The LP-relaxation of their model is then solved by *column generation* technique to provide lower bound which is reported tight in their computational experiments. See also Kubiak et al. (1997) for a dynamic programming approach the temporal supplier capacity problem.

7. Optimization of production sequence

Suppliers do not need to assume their option delivery sequence to become exactly their production sequence. In fact the two may be quite different, which leaves suppliers some room for minimization of number of setups in their production sequence. For instance, in car industry when it comes to supplying components of great diversity and expensive to handle, an order is sent to a supplier, for example electronically, when a car enters assembly line. The supplier then has to produce the component, and to deliver it within a narrow time window, following the order sequence, Guerre-Chaley et al. (1995) and Benyoucef et al. (2000). However, if production for a local buffer is allowed, then the buffer per-

mits *permutation* of production sequence to obtain the required option delivery sequence. The options may leave the buffer in different order than they enter it, the former being the option delivery order, the latter the production order. The size b of the buffer limits the permutations that can be thus obtained. The goal of the supplier is to achieve the delivery sequence at minimal costs, in particular to find the best trade-off between the buffer size and the number of setups in the production sequence, Benyoucef et al. (2000).

Let us consider, for instance, an option delivery sequence

$$S = ababacabaca.$$

This sequence, is 2-balanced (though $B = 10/11$) and has 11 batches thus, by definition, the same number of setups.

A *batch* is a factor of S made of the same letter, which can not be extended either to the right or to the left by the same letter. Thus, the decomposition of S into batches is unique. The number of letters in a batch will be referred to as the batch *size* and the position of the batch last letter will be referred to as the batch *deadline*.

On the other hand the following *production* sequence

$$P = aaabbbccaaa,$$

has 4 batches only. Table 6.3 shows how a buffer of size 3 allows to convert P into S. Therefore, a buffer of size 3 allows to reduce the number of setups more than twice.

Though the buffer allows for the reduction of the number of setups, it does not prevent an option from being produced *too early* and consequently waiting in the buffer for too long for its position in S. To remedy this undesirable effect we put a limit, e, on flow time, that is the time between entering and leaving the buffer by an option.

We call a production sequence P (b, e)-delivery feasible, or just delivery feasible, for S if it can be converted into S by using a buffer of size b so that the maximum flow time does not exceed e. The permutation defined by P will be denoted by π_P. We have the following lemma.

LEMMA 6.1 *The production sequence P is (b, e)-delivery feasible if and only if $\pi_P(i) - i < b$ and $i - \pi_P(i) < e$ for each $i = 1, \ldots, |P|$.*

Proof. Assume that $\pi_P(i) - i < b$ and $i - \pi_P(i) < e$ for each $i = 1, \ldots, |P|$. The position i in delivery sequence S becomes $\pi_P(i)$ in the production sequence P. Thus, $\pi_P(i)$ is among $1, \ldots, i + b - 1$, and at the same time among $i - e + 1, \ldots, |P|$. The former ensures that the i must be in the buffer and thus ready for delivery. The latter ensures that the i waits

Table 6.3. The build up of delivery sequence S from production sequence P using buffer of size 3.

time	*delivery*	*buffer*	*production*
1	-	{$\overleftarrow{a}$,a,a}	bbbccaaa
2	a	{a,a,$\overleftarrow{b}$}	bbccaaa
3	ab	{$\overleftarrow{a}$,a,b}	bccaaa
4	aba	{a,$\overleftarrow{b}$,b}	ccaaa
5	abab	{$\overleftarrow{a}$,b,c}	caaa
6	ababa	{b,$\overleftarrow{c}$,c}	aaa
7	ababac	{b,c,$\overleftarrow{a}$}	aa
8	ababaca	{$\overleftarrow{b}$,c,a}	a
9	ababacab	{$\overleftarrow{c}$,a,a}	-
10	ababacaba	{$\overleftarrow{c}$,-,a}	-
11	ababacabac	{-,-,$\overleftarrow{a}$}	-
11	ababacabaca	{-,-,-}	-

no longer than e for its position in S. Thus, P is delivery feasible. Now assume that $\pi_P(i) - i \geq b$ or $i - \pi_P(i) \geq e$ for some $i = 1, \ldots, |P|$. Consider the smallest such i. Thus, $\pi_P(i) - i \geq b$ or $i - \pi_P(i) \geq e$. Thus, i is not among $1, \ldots, i + b - 1$, thus not in the buffer and not ready for delivery or, it is among $1, \ldots, i - e$, thus waits in the buffer for at least $e + 1$. Thus P is not delivery feasible. □

We assume that the production sequence respects batches of S, that is if $S = s_1 \ldots s_m$ where $s_1, \ldots, s_m$ are batches in S, then the permutation π_P of options (letters) translates in a permutation σ of batches such that $P = s_{\sigma(1)} \ldots s_{\sigma(m)}$

7.1 The limits on setup reduction, buffer size and flow time

In this section, we develop some bounds on the buffer size b and flow time e, but first we investigate the limits on reduction of the number of setups in production sequence for given buffer size b and flow time e.

THEOREM 6.11 *The buffer of size $b \geq 2$ with the limit on maximum flow $e \geq 2$ can reduce the number of batches, and consequently the number of setups, at most $2\min\{b, e\} - 1$ times in comparison with S.*

Proof. Consider an option delivery sequence $S = s_1 \ldots s_m$, where s_i, $i = 1, \ldots, m$, are batches of S, and its delivery-feasible production sequence $P = s_{\sigma^{-1}(1)} \ldots s_{\sigma^{-1}(m)} = p_1 \ldots p_l$, where $l \leq m$ and p_i are batches of P, $i = 1, \ldots, l$. We have $\sigma(i) - i < b$ for all i since P is delivery feasible.

Next, consider a batch $p_j = s_{\sigma^{-1}(i^*)} \dots s_{\sigma^{-1}(i^*+k-1)}$ of type t in P. We shall prove that $k < 2b$. By contradiction, suppose $k \geq 2b$. Then, there are at least $2b-1$ non-t batches between $\sigma^{-1}(i^*)$ and $\sigma^{-1}(i^*+k-1)$ in S, for there must be at least one non-t batch between any two consecutive t batches $s_{\sigma^{-1}(i^*+j)}$ and $s_{\sigma^{-1}(i^*+j+1)}$ of S, $j = 0, \dots, k-2$. Then, $\alpha \geq 0$ of them would end up in $p_1 \dots p_{j-1}$, and $\beta \geq 0$ in $p_{j+1} \dots p_l$, where $\alpha + \beta = k \geq 2b - 1$. Furthermore, all batches $s_1, \dots, s_{\sigma^{-1}(i^*)-1}$ of S must be in $p_1 \dots p_{j-1}$. Otherwise, let $a < \sigma^{-1}(i^*)$ be the earliest of them to end up in $p_{j+1} \dots p_l$, that is $\sigma(a) \geq i^* + k \geq i^* + 2b - 1$. Then all batches s_1 to s_{a-1} would be in $p_1 \dots p_{j-1}$ and thus $i^* \geq a$. Therefore, $\sigma(a) - a \geq 2b - 1 + (i^* - a) \geq b$, which leads to a contradiction since P is delivery-feasible. Consequently,

$$i^* \geq \sigma^{-1}(i^*) + \alpha.$$

Moreover, $\alpha < b$ for otherwise, $\sigma(i^*) - i^* = \alpha \geq b$ and P would not be delivery feasible. Now, consider the earliest non-t batch between t batches $\sigma^{-1}(i^*)$ and $\sigma^{-1}(i^* + k - 1)$ that ends up in $p_{j+1} \dots p_l$ in P. Let it be s_c. Then,

$$\sigma(c) - c \geq i^* + k \geq \sigma^{-1}(i) + \alpha + k - c.$$

Since there are β t batches among $s_{\sigma^{-1}(i^*)} \dots s_{\sigma^{-1}(i^*+k-1)}$ that follow c in S, we have

$$\sigma^{-1}(i^* + k - \beta) = \sigma^{-1}(i^* + \alpha) \geq c,$$

and, thus, it remains to show that

$$\sigma^{-1}(i^*) + k + \alpha - \sigma^{-1}(i^* + \alpha) \geq b.$$

However,

$$\sigma^{-1}(i^* + \alpha) - \sigma^{-1}(i^*) = 2\alpha + 1,$$

and thus

$$-2\alpha - 1 + k + \alpha = k - \alpha - 1 > k - b - 1 \geq b - 1,$$

which again leads to a contradiction since P is delivery-feasible, and proves that $k < 2b$. That is the number of batches in P is no more than $2b - 1$ times higher than in S. To complete the proof we observe that by taking the mirror reflection of S and e instead of b we can repeat the argument that we just presented showing that $k \leq 2e$. Therefore, $k \leq \min\{2b, 2e\}$, which completes the proof. □

We now develop some bounds on the buffer size b and flow time e. The rate-based bound on b follows from the following theorem.

THEOREM 6.12 *Any option delivery sequence S which is a c-balanced word will keep copies of all i's with $r_i \geq \frac{c}{b}$ in buffer of size b at any time.*

Proof. For i with demand d_i there always is a factor of S of size b with at least $d_i/\lceil D/b \rceil$ i's. If d_i is sufficiently large so that $d_i/\lceil D/b \rceil \geq c$, then each of the factors of size b must include at least one i. Otherwise, there would be one such factor with at least $c+1$ i's and at least one such factor with none, which would lead to a contradiction for S is c-balanced. However, i with $d_i/\lceil D/b \rceil \geq c$ implies that

$$r_i \geq \frac{c}{b},$$

which proves the theorem. □

Consequently, only the i's with rates not less than $\frac{c}{b}$ can always be found in a factor of size b of the delivery sequence, for those i's with

$$r_i < \frac{c}{b}$$

this cannot be ensured. The Theorem 6.12 suggests choosing b based on a threshold rate r^* by requiring that b is large enough so that all i's with $r_i \geq r^*$ be always present in the buffer of size b.

Other bounds on b and e can be obtained from the well known result of Jackson (1955) on the optimality of earliest due date sequences (EDD) for the maximum lateness problem on a single machine. The minimum buffer size b^* required to ensure the number of batches equal the number of options $|\mathcal{A}|$ is determined by the maximum lateness, denoted by $L_{\max}$, of the EDD sequence of letters (options). The EDD sequence puts a single batch of letter i in position i according to the ascending order of due dates $d_1 \leq \cdots \leq d_n$, where $d_i = f_i + p_i - 1$ and f_i is the position of the first letter i in S and $p_i = |S|_i$ is the number of i's in S. It is well known, Jackson (1955), that the EDD order minimizes maximum lateness of a set of jobs with processing times p_i and due dates d_i on a single machine. Therefore, extending a deadline of each job (batch) by $L_{\max}$ will result in a sequence with no job being late. Equivalently, the buffer of size $b^* = L_{\max} + 1$ will produce a (b^*, ∞)-delivery feasible production sequence having $|\mathcal{A}|$ batches. By the optimality of $L_{\max}$ no smaller buffer is able to ensure this feasibility. The minimum flow time e^* required to ensure $|\mathcal{A}|$ batches can be calculated similarly. To this end, we define $r_i = l_i - p_i + 1$, where l_i is the position of the last letter i in S. The earliest release date first (ERD) sequence orders a single batch of letter i in position i according to the ascending order of release dates $r_1 \leq \cdots \leq r_n$. This sequence minimizes maximum earliness $E_{\max}$,

which follows again from Jackson (1955). Therefore, reducing a release date of each job by $E_{\max}$ will result in a sequence with no job being started before its release dates. Equivalently, the flow $e^* = E_{\max} + 1$ will produce a (∞, e^*)-delivery-feasible production sequence having $|\mathcal{A}|$ batches.

7.2 Complexity of the number of setups minimization

We prove computational complexity of the number of setups minimization problem subject to the (b, e)-constraint in this section.

THEOREM 6.13 *The problem of obtaining a (b, e)-delivery feasible production sequence with minimum number of setups is NP-hard in the strong sense.*

Proof. The transformation is from the 3-partition problem, Garey and Johnson (1979). We sketch the proof for an instance with the set of $3n$ elements $E = \{1, \ldots, 3n\}$ with positive integer sizes $a_1, \ldots, a_{3n}$ such that $\sum_{i=1}^{3n} a_i = nB$. Let us define $b = e = (n+1)B+n$ in the (b, e)-constraint, and the option delivery sequence S as follows

$$\mathbf{1}L_1\mathbf{2}L_2 \ldots (\mathbf{n+1})L_{n+1}\mathbf{1}M_1\mathbf{2}M_2 \ldots \mathbf{n}M_n(\mathbf{n+1})R_1\mathbf{1}R_2\mathbf{2} \ldots R_{n+1}(\mathbf{n+1}).$$

In S, all batches L_i and R_i are of the same letter (option) L and R, respectively, and all of them of the same length $B + 2$. Moreover, each M_i is of length B, and all M_i's hold $3n$ letters corresponding to the $3n$ elements of A in an arbitrary but fixed order. Therefore, there are $|\mathcal{A}| = (n + 1) + 2 + 3n$ letters in S which obviously is also the minimum possible number of batches. We show that there is a 3 partition of A if and only if there is a (b, e)-constrained production sequence for S with $|\mathcal{A}|$ batches.

(if) Let $A_1, \ldots, A_n$ be a 3 partition, then the following sequence

$$L\mathbf{111}A_1\mathbf{222}A_2\mathbf{333}\ldots\mathbf{nnn}A_n(\mathbf{n+1})(\mathbf{n+1})(\mathbf{n+1})R$$

has $|\mathcal{A}|$ batches and respects the (b, e)-constraints.

(only if) Consider the three batches of letter $\mathbf{i}$, $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{n+1}$ in S. If the earliest of them is in position j, then the next is in position $j + e$ and the last in position $j + 2e$ in S. Thus, if one wants to create one batch for this option, then one needs to find a permutation π such that

$$\pi(j) = \pi(j + e) - 1$$

and

$$\pi(j + 2e) = \pi(j + e) + 1.$$

Since the production sequence defined by π must be delivery feasible, then

$$|\pi(j) - j| = |\pi(j+e) - 1 - j| \leq e$$

and

$$|\pi(j+2e) - j - 2e| = |\pi(j+e) + 1 - j - 2e| \leq e.$$

Thus, from the first equation we get

$$\pi(j+e) \leq e + j + 1$$

and from the second

$$e + j + 1 \leq \pi(j+e).$$

Consequently

$$\pi(j+e) = e + j + 1.$$

Therefore, the production sequence has the letter **i** in positions $j+e$, $j+e+1$, and $j+e+2$. Obviously, $j = (i-1)(B+2)+i$. Consequently, letter **i** occupies positions $(i-1)(B+2)+i+e$, $(i-1)(B+2)+i+e+1$, and $(i-1)(B+2)+i+e+2$. This pattern leaves a gap of size $i(B+2)+i+1+e-((i-1)(B+2)+i+e+2)-1 = B$ in between batches of letter **i** and of letter $\mathbf{i+1}$ for other letters. These gaps, however, cannot be filled in be either L or R since the two require long batches of size nB in a solution with $|\mathcal{A}|$ batches. Thus, the gaps can only be filled by the short batches of letters $1, \ldots, 3n$. None of them, however, can be split as this would violate the optimality of the solution. Therefore, if letters l, j and k occur in between batches of letters **i** and $\mathbf{i+1}$, then

$$a_l + a_j + a_k = B.$$

Notice that by definition of the 3 partition problem the total size of any two elements in E is less than B and the total size of any four is greater than B. Therefore, the letters $\mathbf{1}, \ldots, \mathbf{n+1}$ partition the letters $1, \ldots, 3n$ into n sets with the total size of each equal B, which gives the required 3 partition. □

The problem of obtaining a (b, ∞)-feasible sequence that minimizes the number of setups is NP-hard provided that S is succinctly coded as a sequence of batches, where each batch is specified by its letter and size, we refer the reader to Kubiak et al. (2000) for this complexity proof. However, it remains open whether or not there is a pseudo-polynomial time algorithm for the (b, ∞)-constrained problem.

7.3 Algorithm for minimization of the number of setups

Consider an option delivery sequence $S = s_1 \dots s_m$, where s_i, $i = 1, \dots, m$, are batches of S, and $|S| = T$. Let option (letter) i has its m_i batches in positions $i_1, \dots, i_{m_i}$ of S. For batch i_j let its size be p_{i_j} and its deadline d_{i_j}. Recall form the beginning of Section 7 that the size of a batch equals the number of letters in the batch and the deadline of a batch is the position of its last letter.

For any letter i, an optimal production sequence P merges some batches of S into a single batch of P. The batch $[j,k]$ of i obtained by merging batches $i_j, \dots, i_k$ for $1 \leq j \leq k \leq m_i$ has size

$$p_{i[j,k]} = \sum_{l=j}^{k} p_{i_l},$$

deadline, that is the position of its last letter

$$d_{i[j,k]} = d_{i_j} + p_{i[j+1,k]} + b - 1,$$

and release date, that is the position of its first letter

$$r_{i[j,k]} = d_{i_k} - p_{i[j,k]} - e + 1.$$

Since any batch must meet the (b,e)-constraint in P, the deadline ensures that the batch $[j,k]$ is not too late for any of composing it batches of S, whereas the release date ensures that the batch $[j,k]$, is not too early for any of composing it batches of S. Meeting the two simultaneously can only be possible if $d_{i_j} - p_{i_j} \geq r_{i[j,k]}$. Otherwise, the batch $[j,k]$ can be discarded for it will never occur in a production sequence respecting the (e,b)-constraint. From now on, we consider only *feasible* candidates for batches in the production sequence.

Let $(i,[j,k],s)$, where $i = 1, \dots, n$, feasible batch $[j,k]$, and $s = 1, \dots, T$. We build a digraph $\mathcal{G}$ where a node is any triple $(i,([j,k],s)$ with a possible starting point s of $[j,k]$ in the interval $[r_{i[j,k]}, d_{i[j,k]}]$. In addition, we have two nodes, start B and finish F. There is an arc between S and any $(i,[j,k],s=1)$, and an arc between any $(i,[j,m_i],s = T - p_{i[j,m_i]} - 1)$ and the F. Finally, there is an arc between $(i,[j,k],s)$ and $(i',[j',k'],s')$ if and only if

$$i \neq i'$$

and

$$s' = s + p_{i[j,k]}.$$

The length of each arc starting with B is 1. The arc linking $(i,[j,k],s)$ with $(i',[j',k'],s')$, has length 1 as it represents a setup between a batch of i and a batch of i' which are by definition different. The length of any arc finishing with F is 0. There are $O(nm^2T)$ nodes and $O(nm^4T)$ arcs in $\mathcal{G}$. For any path

$$B(i_1,[j_1,k_1],s_1)\ldots(i_m,[j_m,k_m],s_m)F$$

from B to F, we have $s_1=1$, $s_l=s_{l-1}+p_{i_l[j_lk_l]}$, for $l=2,\ldots,m$ and $s_m=T-p_{i_m[j_mk_m]}$ in $\mathcal{G}$. Furthermore, the length of the shortest path in $\mathcal{G}$ is a *lower bound* on the number of setups. However, the path may not be feasible as it may pass two nodes $(i,[j,k],s)$ and $(i,[j',k'],s')$ with overlapping intervals $[j,k]$ and $[j',k']$. In order to avoid this overlap along a path we need to keep track of the batches used in reaching a given node v. We now describe how this can be done. For $v=(i,[j,k],s)$ define,

$$\mu_v=\{k_{ij},k_{ij+1},\ldots,k_{ik}\},$$

where $k_{ij},k_{ij+1},\ldots,k_{ik}$ are the positions of batches $j,\ldots,k$ of i in the delivery sequence S. We associate with each node v of $\mathcal{G}$ a set M_v calculated as follows.

Start at B and recursively calculate the set M_v for each node v of $\mathcal{G}$ finishing once the M_F is calculated. Proceed as follows, initially

$$M_S=\{(\varnothing,0)\}.$$

Next, let $1,\ldots,l$ be all *immediate* predecessors of v, $v\neq F$, with their sets $M_1,\ldots,M_l$, respectively. Define

$$D_v=\{\mu:\exists_{k,t}(\mu,t)\in M_k\},$$

and

$$C_v=\{\mu:\mu\in D_v,\mu\cap\mu_v=\varnothing\}.$$

If $C_v=\varnothing$, then delete v for it overlaps on some batch with any path leading to v. Otherwise, for each $\mu\in C_v$ let

$$t_\mu=\min\{t:\exists_k(\mu,t)\in M_k\}.$$

Then,

$$M_v=\{(\mu\cup\mu_v,t_\mu+1):\mu\in C_v\}.$$

We observe that if $(\mu,t)\in M_v$, then there is a path from B to v of length t that uses all, and only, batches in positions in μ, and there is no shorter path from B to v using all, and only, batches in positions in μ.

Finally, let $1, \ldots, l$ be all *immediate* predecessors of F with their sets $M_1, \ldots, M_l$, respectively. Define

$$t_F = \min\{t : \exists_k(\mu, t) \in M_k\},$$

the number of setups in the solution to the problem. We have the following lemma.

LEMMA 6.2 *The t_F is the minimum number of setups subject to the (e, b)- constraints. The solution can be found by backtracking the sequence $t_F, t_F - 1, \ldots, 0$ from F to B.*

Proof. For any t in the sequence t_F, $t_F - 1, \ldots, 0$ from F to B, there is μ and v such that $(\mu, t) \in M_v$. Therefore, there is a path from B to v of length t that uses all, and only batches, in positions in μ, and there is no shorter path from B to v using, all and only, batches in positions in μ. □

We now estimate the number of pairs (μ, t) that need to be generated by this algorithm in order to eliminate infeasible paths.

LEMMA 6.3 *The number of different pairs (μ, t) does not exceed $m^2/2 \times (b + e)2^{\min\{b,e\}}$.*

Proof. We begin by calculating the number of distinct sets $\mu \subseteq \{1, \ldots, m\}$ constructed by the algorithm. Consider any non-empty μ. Then, there is k such that $\{1, \ldots, k\} \subseteq \mu$. Let k^* be the largest such k. Then, $k^* + 1 \notin \mu$. Also, let l^* be the largest element of μ. Thus, $\{l^* + 1, \ldots, m\} \cap \mu = \varnothing$. We have $|\mu \setminus \{1, \ldots, k^*\}| < b$. Otherwise, the batch k^*+1 would end up in position $k^*+|\mu\setminus\{1, \ldots, k^*\}|+1 \geq k^*+b+1$, that is too late. Furthermore, $l^* - k^* - |\mu \setminus \{1, \ldots, k^*\}| < e$. Otherwise, the batch l^* would end up in the position not latter than $l^* - (l^* - k^* - |\mu \setminus \{1, \ldots, k^*\}|) \leq l^* - e$, that is too early. Consequently,

$$|\mu \setminus \{1, \ldots, k^*\}| < b$$

and

$$|\{1, \ldots, l^*\} \setminus \mu| < e.$$

Therefore, for given k^* and l^*, $1 \leq k^* < l^* \leq m$, $l^* - k^* \geq 2$, there are at most $2^{l^*-k^*-1}$ sets μ. Denote $z = l^* - k^*$. Then the total number of number of sets μ is

$$\sum_{z=2}^{\min\{b+e-1,m\}} (m - z) \sum_{i=0}^{\min\{b,e,z\}} \binom{z}{i} \leq (b + e)\frac{m}{2}2^{\min\{b,e\}}.$$

Finally, $t \leq m$. Thus the lemma holds. □

The following corollary follows immediately from Lemma 6.2.

COROLLARY 6.1 *If at least one of b and e is constant then the algorithm is polynomial.*

The reader is referred to Benyoucef et al. (2000) for review of the literature on a closely related problem of changeover minimization problem.

8. Concluding remarks and further research

This chapter studied balancing lean, mixed-model supply chains. These chains respond to customers' demand by setting demand rates for each model produced and pulling supplies required for production whenever they are needed. To balance and synchronize these supply chains, it is important to find a balanced model delivery sequence for a given set of demand. Two main goals shape this sequence. The external, meeting demand rates, and, the internal, satisfying the temporary chain capacity constraints. The chapter discussed algorithms for setting up the model delivery sequence as well as supplier option delivery and productions sequences. The chapter introduced and explored a link between model delivery sequences and balanced words, and showed that though balanced words result in optimal workload balancing, Altman et al. (2000), they are not sufficient for all possible sets of demand rates. The real-live model delivery sequences are either 2-balanced or 3-balanced at best, that is if they disregard temporary capacity constraints. It is, however, an open problem to show how well these sequences balance the chain workload in comparison with balanced words. As well, it would be interesting to further investigate the concept of complexity of model delivery sequences based on their numbers of factors. By reducing this complexity supply chain could reduce the number of different demand patterns in option delivery sequences and thus reduce variability present in the chain. Finally, the chapter discussed optimization of suppliers production sequences. In particular, it discussed the problem of minimizing the number of setups for a given buffer size and maximum flow time limit. It proved the problem complexity and proposed algorithms for the problem.

Acknowledgments This research has been supported by NSERC Grant OGP0105675.

References

Altman, E., Gaujal, B., and Hordijk, A. (2000). Balanced sequences and optimal routing. *Journal of the Association for Computing Machinery*, 47:754–775.

Balinski, M. and Shahidi, N. (1998). A simple approach to the product rate variation problem via axiomatics. *Operations Research Letters*, 22:129–135.

Balinski, M. and Young, H.P. (1982). *Fair Representation: Meeting the Ideal of One Man, One Vote*. Yale University Press, New Haven, CT.

Bautista, J., Companys, R., and Corominas, A. (1997). Modelling and solving the production rate variation problem. *Trabajos de Investigacion Operativa*, 5:221–239.

Benyoucef, L., Kubiak, W., and Penz, B. (2000). Minimizing the number of changeovers for a multi-product single machine scheduling problem with deadlines. *International Conference on Management and Control of Production and Logistics-MCPL'2000*, CD-Rom, Grenoble, France.

Berthé, V. and Tijdeman, R. (2002). Balance properties of multi-dimensional words. *Theoretical Computer Science* 273:197–224.

Bowersox, D.J., Closs, D.J., and Cooper, M.B. (2002). *Supply Chain Logistics Management*. McGraw-Hill Irwin.

Brauner, N. and Crama, Y. (2001). *Facts and Questions About the Maximum Deviation Just-In-Time Scheduling Problem*. Research Report G.E.M.M.E. No 0104, University of Liège, Liège.

Brauner, N., Jost, V., and Kubiak, W. (2002). *On Symmetric Fraenkel's and Small Deviations Conjectures*. Les cahiers du Laboratoire Leibniz-IMAG, no 54, Grenoble, France.

Corominas, A. and Moreno, N. (2003). About the relations between optimal solutions for different types of min-sum balanced JIT optimisation problems. Forthcoming in *Information Processing and Operational Research*.

Drexl, A. and Kimms, A. (2001). Sequencing JIT mixed-model assembly lines under station-load and part-usage constraints. *Management Science*, 47:480–491.

Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco.

Gent, I.P. (1998). *Two Results on Car-Sequencing Problems*. Report APES-02-1998, Dept. of Computer Science, University of Strathclyde.

Glover, F. (1967). Maximum matching in a convex bipartite graph. *Naval Research Logistics Quarterly*, 4:313–316.

Guerre-Chaley, F., Frien, Y., and Bouffard-Vercelli, R. (1995). An efficient procedure for solving a car sequencing problem. *1995 IN-*

RIA/IEEE Symposium on Emerging Technologies and Factory Automation, Volume 2, pp. 385 – 394. IEEE.

ILOG, Inc. (2001). *ILOG Concert Technology 1.1. User's Manual.*

Jackson, J.R. (1955). *Scheduling a Production Line to Minimize Maximum Tardiness.* Research Report 43, Management Science Research Project, University of California, Los Angeles.

Jost, V. (2003). *Deux problèmes d'approximation diophantine : Le partage proportionnel en nombres entiers et les pavages équilibres de z.* DEA ROCO, Laboratoire Leibniz-IMAG.

Kovalyov, M.Y., Kubiak, W., and Yeomans, J.S. (2001). A computational study of balanced JIT optimization algorithms. *Information Processing and Operational Research*, 39:299 – 316.

Kubiak, W. (1993). Minimizing variation of production rates in just-in-time systems: A survey. *European Journal of Operational Research*, 66:259 – 271.

Kubiak, W.(2003a). On small deviations conjecture. *Bulletin of the Polish Academy of Sciences*, 51:189 – 203.

Kubiak, W. (2003b). Cyclic just-in-time sequences are optimal. *Journal of Global Optimization*, 27:333 – 347.

Kubiak, W. (2003c). The Liu – Layland problem revisited. Forthcoming in *Journal of Scheduling.*

Kubiak, W. and Sethi, S.P. (1991). A note on level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, 37:121 – 122.

Kubiak, W. and Sethi, S.P. (1994). Optimal just-in-time schedules for flexible transfer lines. *The International Journal of Flexible Manufacturing Systems*, 6:137 – 154.

Kubiak, W., Benyoucef, L., and Penz, B. (2000). *Complexity of the Just-In-Time Changeover Cost Minimization Problem.* Gilco Report RR 2000-02.

Kubiak, W., Steiner, G., and Yeomans, S. (1997). Optimal level schedules in mixed-model, multi-level just-in-time assembly systems. *Annals of Operations Research*, 69:241 – 259.

Lee, H.L., Padmanabhan, P., and Whang, S. (1997). The paralyzing curse of the bullwhip effect in a supply chain. *Sloan Management Review*, 93 – 102.

Meijer, H.G. (1973). On a distribution problem in finite sets. *Koninklijke Nederlandse Akademie van Wetenschappen. Indagationes Mathematicae*, 35:9 – 17.

Miltenburg, J.G. (1989). Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, 35:192 – 207.

Monden, Y. (1998). *Toyota Production Systems*, 3rd edition. Institute of Industrial Engineers.

Shapiro, J.F. (2001). *Modeling the Supply Chain*. Duxbury.

Simchi-Levi, D., Kaminski, Ph., and Simchi-Levi, E. (2003). *Designing and Managing the Supply Chain*, 2nd edition. McGraw-Hill Irwin.

Steiner, G. and Yeomans, S. (1993). Level schedules for mixed-model, just-in-time production processes. *Management Science*, 39:728 – 735.

Still, J.W. (1979). A class of new methods for congressional apportionment. *SIAM Journal on Applied Mathematics* 37:401 – 418.

Tijdeman, R. (1980). The chairman assignment problem. *Discrete Mathematics*, 32:323 – 330.

Tijdeman, R. (2000). Exact covers of balanced sequences and Fraenkel's conjecture. In: F. Halter-Koch and R.F. Tichy (eds.), *Algebraic Number Theory and Diophantine Analysis*, pp. 467 – 483. Walter de Gruyter, Berlin – New York.

Vuillon, L. Balanced Words, *Rapports de Recherche 2003-006*, LIAFA CNRS, Université Paris 7.

Chapter 7

BILEVEL PROGRAMMING: A COMBINATORIAL PERSPECTIVE

Patrice Marcotte
Gilles Savard

Abstract Bilevel programming is a branch of optimization where a subset of variables is constrained to lie in the optimal set of an auxiliary mathematical program. This chapter presents an overview of two specific classes of bilevel programs, and in particular their relationship to well-known combinatorial problems.

1. Introduction

In optimization and game theory, it is frequent to encounter situations where conflicting agents are taking actions according to a predefined sequence of play. For instance, in the Stackelberg version of duopolistic equilibrium (Stackelberg, 1952), a *leader* firm incorporates within its decision process the reaction of the *follower* firm to its course of action. By extending this concept to a pair of arbitrary mathematical programs, one obtains the class of *bilevel programs*, which allow the modeling of many decision processes. The term "bilevel programming" appeared for the first time in a paper by Candler and Norton (1977), who considered a multi-level formulation in the context of agricultural economics. Since that time, hundreds of papers have been dedicated to this topic. The reader interested in the theory and applications of bilevel programming is referred to the recent books by Shimizu et al. (1997), Luo et al. (1996), Bard (1998), and Dempe (2002).

Generically, a bilevel program assumes the form

$$\begin{aligned} \min_{x,y} \; & f(x,y) \\ \text{s.t.} \; & (x,y) \in X \\ & y \in \mathcal{S}(x), \end{aligned}$$

where $\mathcal{S}(x)$ denotes the solution set of a mathematical program parameterized in the vector x, i.e.,

$$\begin{aligned} \mathcal{S}(x) = \arg\min_{y} \; & g(x,y) \\ \text{s.t. } & (x,y) \in Y. \end{aligned}$$

In this formulation, the leader is free, whenever the set $\mathcal{S}(x)$ does not shrink to a singleton, to select an element of $\mathcal{S}(x)$ that suits her best. This corresponds to the *optimistic* formulation. Alternatively, the *pessimistic* formulation refers to the case where the leader protects herself against the worst possible situation, and is formulated as

$$\begin{aligned} \min_{x} \max_{y} \; & f(x,y) \\ \text{s.t. } & (x,y) \in X \\ & y \in \mathcal{S}(x). \end{aligned}$$

The scope of this chapter is limited to the optimistic formulation. The reader interested in the pessimistic formulation is referred to Loridan and Morgan (1996).

In many applications, the lower level corresponds to an equilibrium problem that is best represented as a (parametric) variational inequality or, equivalently, a generalized equation. We then obtain an MPEC (*Mathematical Program with Equilibrium Constraints*), that is expressed as[1]

$$\begin{aligned} \text{MPEC}: \quad \min_{x,y} \; & f(x,y) \\ \text{s.t. } & (x,y) \in X \\ & y \in Y(x) \\ & -G(x,y) \in N_{Y(x)}(y), \end{aligned}$$

where $Y(x) = \{y : (x,y) \in Y\}$ and $N_C(z)$ denotes the normal cone to the set C at the point z. If the vector function G represents the gradient of a differentiable convex function g and the set Y is convex, then MPEC reduces to a bilevel program. Conversely, an MPEC can be reformulated as a standard bilevel program by noting that a vector y is solution of the lower level variational inequality if and only if it globally minimizes, with respect to the argument y, the strongly convex function $\text{gap}(x,y)$

[1]Throughout the paper, we assume that vectors on the left-hand side of an inner product are row vectors. Symmetrically, right-hand side vectors are understood to be column vectors. Thus primal (respectively dual) variables usually make up column (respectively row) vectors. Transpose are only used when absolutely necessary.

defined as (see Fukushima, 1992):

$$\text{gap}(x,y) = \max_{y' \in Y(x)} G(x,y)(y-y') - \frac{1}{2}\|y-y'\|^2.$$

Being generically non-convex and non-differentiable, bilevel programs are intrinsically hard to solve. For one, the linear bilevel program which corresponds to the simple situation where all functions involved are linear, is strongly $\mathcal{NP}$-hard (see Section 2.2). Further, determining whether a solution is locally optimal is also strongly $\mathcal{NP}$-hard (Vicente et al., 1994). In view of these results, most research has followed two main avenues, either *continuous* or *combinatorial*. The continuous approach is mainly concerned with the characterization of necessary optimality conditions and the development of algorithms that generate sequences converging toward a local solution. Along that line, let us mention works based on the implicit function approach (Kočvara and Outrata, 1994), on classical nonlinear programming techniques such as SQP (Sequential Quadratic Programming) applied to a single-level reformulation of the bilevel problem (Scholtes and Stöhr, 1999) or smoothing approaches (Fukushima and Pang, 1999; Marcotte et al., 2001)]). Most work done on MPECs adopts the latter point of view.

The combinatorial approach takes a global optimization point of view and looks for the development of algorithms with a guarantee of global optimality. Due to the intractability of the bilevel program, these algorithms are limited to specific subclasses possessing features such as linear, bilinear or quadratic objectives, which allow for the development of "efficient" algorithms. We consider two classes that are amenable to a global approach, namely bilevel programs involving linear or bilinear objectives. The first class is important as it encompasses a large number of combinatorial problems (e.g., 0-1 mixed integer programs) while the second allows for the modeling of a rich class of pricing applications. This chapter focuses on the combinatorial structure of these two classes.

2. Linear bilevel programming

The linear/linear bilevel problem (LLBP) takes the form

$$
\begin{aligned}
\mathsf{LLBP}: \quad & \max_{x,y} c_1 x + d_1 y \\
& \text{s.t. } A_1 x + B_1 y \leq b_1 \\
& \quad x \geq 0 \\
& \quad y \in \arg\max_{y} d_2 y \\
& \qquad \text{s.t. } A_2 x + B_2 y \leq b_2 \\
& \qquad\quad y \geq 0,
\end{aligned}
$$

where $c_1 \in \mathbb{R}^{n_x}$, d_1, $d_2 \in \mathbb{R}^{n_y}$, $A_1 \in \mathbb{R}^{n_u \times n_x}$, $A_2 \in \mathbb{R}^{n_l \times n_x}$, $b_1 \in \mathbb{R}^{n_u}$, $B_1 \in \mathbb{R}^{n_u \times n_y}$, $B_2 \in \mathbb{R}^{n_l \times n_y}$, $b_2 \in \mathbb{R}^{n_l}$. The constraints $A_1 x + B_1 y \leq b_1$ (respectively $A_2 x + B_2 y \leq b_2$) are the upper (respectively lower) level constraints. The linear term $c_1 x + d_1 y$ (respectively $d_2 y$) is the upper (respectively lower) level objective function, while x (respectively y) is the vector of upper (respectively lower) level variables.[2] To characterize the solution of LLBP, the following definitions are useful.

DEFINITION 7.1 (1) The **feasible set** of LLBP is defined as

$$\Omega = \{(x, y) : x \geq 0, y \geq 0, A_1 x + B_1 y \leq b_1,\ A_2 x + B_2 y \leq b_2\}.$$

(2) For every $x \geq 0$, the **feasible set of the lower level problem** is defined as

$$\Omega_y(x) = \{y : y \geq 0,\ B_2 y \leq b - A_2 x\}.$$

(3) The **trace** of the lower level problem with respect to the upper level variables is

$$\Omega_x^2 = \{x : x \geq 0,\ \Omega_y(x) \neq \emptyset\}.$$

(4) For a given vector $x \in \Omega_x^2$, the *set of optimal solutions of the lower problem* is

$$S(x) = \{y : y \in \arg\max\{d_2 y : y \in \Omega_y(x)\}.$$

A point (x, y) is said to be **rational** if $x \in \Omega_x^2$ and $y \in S(x)$.

(5) The **optimal value function** for $x \in \Omega_x^2$ is

$$v(x) = d_2 y, \quad y \in \mathcal{S}(x).$$

(6) The **admissible set** (also called **induced region**) is

$$\Upsilon = \{(x, y) : x \geq 0,\ A_1 x + B_1 y \leq b_1,\ y \in S(x)\}.$$

[2] We slightly abuse notation and use the letter y to denote both the optimal solution (left-hand side) and the argument (right-hand side) of the lower level program.

A point (x, y) is **admissible** if it is feasible and lies in $S(x)$.

Based on the above notations, we characterize optimal solutions for the LLBP.

DEFINITION 7.2 A point (x^*, y^*) is optimal for LLBP if it is admissible and, for all admissible (x, y), there holds $c_1 x^* + d_1 y^* \geq c_1 x + d_1 y$.

Note that, whenever the upper level constraints involve no lower level variables, then rational points are also admissible. The converse may fail to hold in the presence of joint upper level constraints.

To illustrate some geometric properties of bilevel programs (see Figure 7.1), let us consider the following two-dimensional example:

$$\begin{aligned}
&\max_{x,y} -x - 4y \\
&\text{s.t. } x \geq 0 \\
&\quad y \in \arg\max_{y} y \\
&\quad \text{s.t. } -2x - y \leq 8 \\
&\qquad\quad -3x + 2y \leq 6 \\
&\qquad\quad 5x + 6y \leq 60 \\
&\qquad\quad 2x + y \leq 16 \\
&\qquad\quad 2x - 5y \leq 0 \\
&\qquad\quad y \geq 0.
\end{aligned}$$

The left-hand side graphs (b) and (d) illustrate the example's geometry, while right-hand side graphs (c) and (e) correspond to the bilevel program obtained after moving the next-to-last constraint from the lower to the upper level, showing the impact of upper level constraints on the admissible set. We observe that the admissible set, represented by thick lines, is not convex. Indeed, its analytic expression is

$$\Upsilon = \{(x, y) : x \geq 0,\ A_1 x + B_1 y \leq b_1\} \cap \Gamma_S$$

where

$$\Gamma_S = \{(x, y) : x \in \Omega_x^2,\ d_2 y = v(x)\}$$

represents the union of a finite (possibly empty) set of polyhedra (Savard, 1989). Based on a result of Hogan (1973), one can show that the multivalued mapping $S(x)$ is closed, whenever the set Ω is compact. In the particular case where $S(x)$ shrinks to a singleton for every $x \in \Omega_x^2$, it follows that the reaction function $y(x) = S(x)$ is continuous.

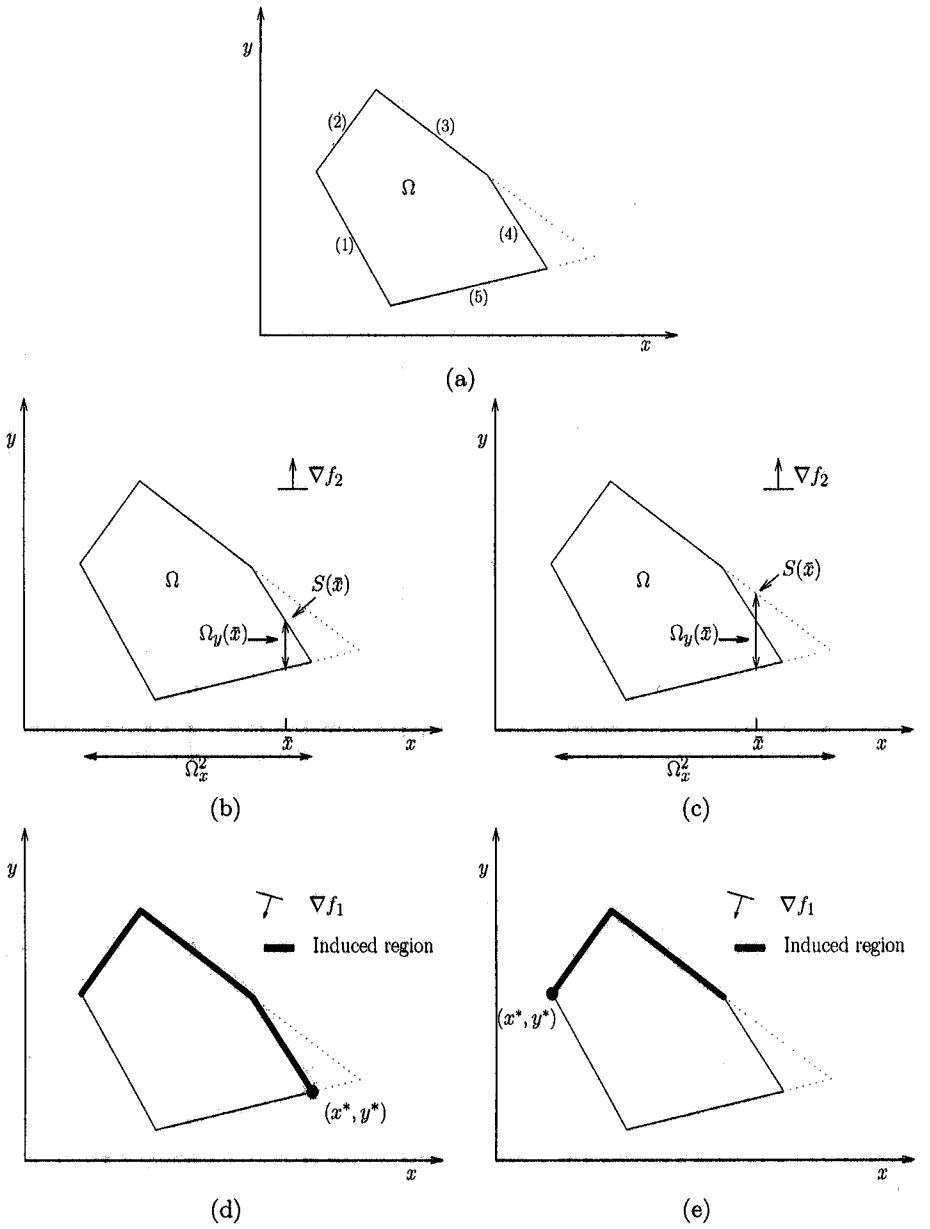


Figure 7.1. Two linear/linear bilevel programs

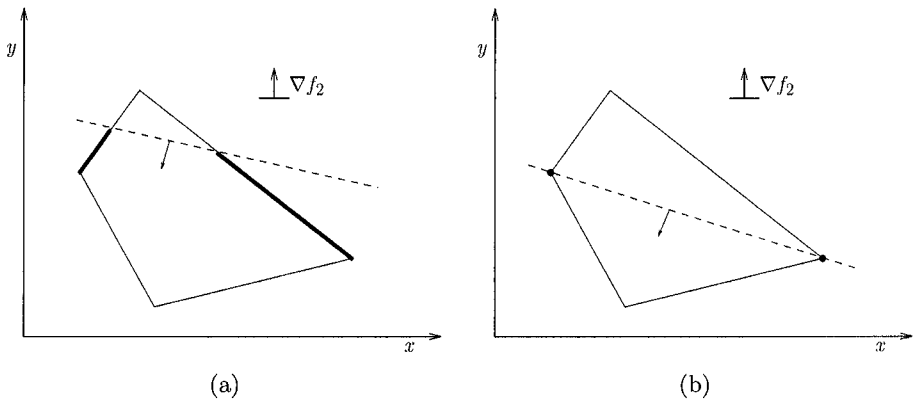


Figure 7.2. A disconnected admission set

As seen in Figure 7.1 the presence of joint upper level constraints may considerably modify the structure of the admissible set. It can make this set disconnected, finite or even empty. This is illustrated in Figure 7.2, where a single upper level constraint is slided. In the next section, we will construct a bilevel program with an admissible set corresponding solely of integer points.

The following theorem is a direct consequence of the polyhedral nature of the admissible set. It emphasizes the combinatorial nature of the LLBP.

THEOREM 7.1 *If LLBP has a solution, an optimal solution is attained at an extreme point of* Ω.

The combinatorial nature of bilevel programming can also be observed by studying the single-level reformulation obtained by replacing the lower level problem by its (necessary and sufficient) optimality conditions:

$$
\begin{aligned}
\mathsf{LLBP}_1: \quad & \max_{x,y,\lambda} c_1 x + d_1 y \\
& \text{s.t. } A_1 x + B_1 y \leq b_1 \\
& \quad A_2 x + B_2 y \leq b_2 \\
& \quad \lambda B_2 \geq d_2 \\
& \quad \lambda(b_2 - A_2 x - B_2 y) = 0 \\
& \quad (\lambda B_2 - d_2) y = 0 \\
& \quad x \geq 0,\ y \geq 0,\ \lambda \geq 0,
\end{aligned}
$$

where $\lambda \in \mathbb{R}^{n_l}$. The combinatorial nature is entirely captured by the two orthogonality constraints, which can actually be added to form a single

constraint. Their disjunctive nature relates the LLBP to linear mixed integer programming and allows for the development of algorithms based on enumeration and/or cutting plane approaches.

2.1 Equivalence between LLBP and classical problems

In this section, we show that simple polynomial transformations allow to formulate linear mixed 0-1 integer programs ($\mathsf{MIP}_{0\text{-}1}$) and bilinear disjoint programs (BDP) as linear bilevel programs, and vice versa. The interest in these reformulations goes beyond the complexity issue. Indeed, Audet (1997) and Audet et al. (1997) have uncovered equivalences between algorithms designed to solve mixed integer programs and LLBP. They have shown that the HJS algorithm of Hansen et al. (1992) designed for solving the LLBP can be mapped onto a standard branch-and-bound method (see for instance Beale and Small, 1965) for addressing an equivalent mixed 0-1 program, provided that mutually consistent branching rules are implemented. One may therefore claim that the mixed 0-1 algorithm is subsumed (the authors use the term *embedded*) by the bilevel algorithm. This result shows that the structure of both problems is virtually indistinguishable, and that any algorithmic improvement on one problem can readily be adapted to the other (Audet et al., 1997): solution techniques developed for solving mixed 0-1 programs may be tailored to the LLBP, and vice versa.

2.1.1 LLBP and $\mathsf{MIP}_{0\text{-}1}$. The linear mixed 0-1 programming problem ($\mathsf{MIP}_{0\text{-}1}$) is expressed as

$$\begin{aligned} \mathsf{MIP}_{0\text{-}1}: \quad & \max_{x,u} cx + eu \\ & \text{s.t. } Ax + Eu \leq b \\ & \quad x \geq 0,\ u \text{ binary valued,} \end{aligned}$$

where $c \in \mathbb{R}^{n_x}$, $e \in \mathbb{R}^{n_u}$, $A \in \mathbb{R}^{m \times n_x}$, $E \in \mathbb{R}^{m \times n_u}$, $b \in \mathbb{R}^m$.

We first note that the binary condition is equivalent to:

$$\begin{aligned} & 0 \leq u \leq \mathbf{1} \\ & 0 = \min\{u, \mathbf{1} - u\}, \end{aligned}$$

where $\mathbf{1}$ denotes the vector of "all ones". Next, by introducing an upper level variable y, and defining a second level problem such that the optimal solution corresponds to this minimum, we obtain the equivalent

bilevel programming reformulation:

$$\begin{aligned}
\mathsf{LLBP}_2: \quad & \max_{x,y,u} cx + eu \\
& \text{s.t. } Ax + Eu \leq b \\
& \quad 0 \leq u \leq \mathbf{1} \\
& \quad x \geq 0 \\
& \quad y = 0 \\
& \quad y \in \arg\max_{w} \sum_{i=1}^{n_u} w_i \\
& \qquad \text{s.t. } w \leq u \\
& \qquad\quad w \leq \mathbf{1} - u.
\end{aligned}$$

where $y, w \in \mathbb{R}^{n_u}$. In this formulation, the integrality constraints are no more required, as they are enforced by the upper level constraints $y = 0$, together with the lower level optimality conditions.

In general, upper level constraints make the problem more difficult to solve. Actually, some algorithms only address instances where such constraints are absent. However, as suggested by Vicente et al. (1996), the constraint $y = 0$ can be enforced by incorporating an *exact penalty* within the leader's objective, i.e., there exists a threshold value M^* such that, whenever M exceeds M^*, the solution of the following bilevel program satisfies the condition $y = 0$, i.e., the integrality condition:

$$\begin{aligned}
\mathsf{LLBP}_3: \quad & \max_{x,y,u} cx + eu - M\mathbf{1}y \\
& \text{s.t. } Ax + Eu \leq b \\
& \quad 0 \leq u \leq \mathbf{1} \\
& \quad x \geq 0 \\
& \quad y \in \arg\max_{w} \sum_{i=1}^{n_u} w_i \\
& \qquad \text{s.t. } w \leq u \\
& \qquad\quad w \leq \mathbf{1} - u.
\end{aligned}$$

Conversely, LLBP may be polynomially reduced to $\mathsf{MIP}_{0\text{-}1}$. First, one replaces the lower level problem by its optimality conditions, yielding a single-level program with the complementarity constraints

$$\begin{aligned}
& \lambda(b_2 - A_2x - B_2y) = 0 \\
& (\lambda B_2 - d_2)y = 0.
\end{aligned}$$

The second transformation consists in linearizing the complementarity constraints by introducing two binary vectors u and v and a sufficiently large finite constant $L > 0$, the existence of which is discussed in Vicente et al. (1996):

$$\begin{aligned} &b_2 - A_2x - B_2y \leq L(\mathbf{1} - u), && \lambda \leq Lu^{\mathsf{T}}, \\ &y \leq L(\mathbf{1} - v) && \lambda B_2 - d_2 \leq Lv^{\mathsf{T}}. \end{aligned}$$

This leads to the equivalent $\mathsf{MIP}_{0\text{-}1}$ reformulation of LLBP:

$$\begin{aligned} \mathsf{MIP}_{\mathsf{LLBP}}: \quad & \max_{x,y,\lambda,u,v} c_1x + d_1y \\ & \text{s.t. } A_1x + B_1y \leq b_1 \\ & \quad x \geq 0 \\ & \quad A_2x + B_2y \leq b_2 && -\lambda B_2 \leq -d_2 \\ & \quad y \geq 0 && \lambda \geq 0 \\ & \quad -A_2x - B_2y + Lu \leq L\mathbf{1} - b_2 && \lambda - Lu^{\mathsf{T}} \leq 0 \\ & \quad y + Lv \leq L\mathbf{1} && \lambda B_2 - Lv^{\mathsf{T}} \leq d_2 \\ & \quad u \text{ binary valued} && v \text{ binary valued.} \end{aligned}$$

2.1.2 LLBP and BILP. The disjoint bilinear programming problem BILP was introduced by Konno (1971) to generalize Mills' approach (Mills, 1960) for computing Nash equilibra (Nash, 1951) of bimatrix games. It can be expressed as follows:

$$\begin{aligned} \mathsf{BILP}: \quad & \max_{x,u} cx - uQx + ud \\ & \text{s.t. } Ax \leq b_1 \\ & \quad uB \leq b_2 \\ & \quad x \geq 0 \\ & \quad u \geq 0, \end{aligned}$$

where $c \in \mathbb{R}^{n_x}$, $d \in \mathbb{R}^{n_u}$, $Q \in \mathbb{R}^{n_u \times n_x}$, $A \in \mathbb{R}^{n_v \times n_x}$, $B \in \mathbb{R}^{n_u \times n_y}$, $b_1 \in \mathbb{R}^{n_v}$, $b_2 \in \mathbb{R}^{n_y}$, and the matrix Q assumes no specific structure.

By exploiting the connection between LLBP and BILP, Audet et al. (1999) and Alarie et al. (2001) have been able to construct improved branch-and-cut algorithms for the BILP. Their approach relies on the separability, with respect to the vectors x and u, of the feasible set of BILP. Let us introduce the sets $X = \{x \geq 0 : Ax \leq b_1\}$ and $U = \{u \geq 0 : uB \leq b_2\}$. If both sets are nonempty and the optimal solution of BILP is bounded, we can rewrite BILP as

$$\mathsf{BILP}_2: \quad \max_{x \in X} cx + \max_{u \in U} u(d - Qx).$$

For fixed $x \in X$, one can replace the inner optimization problem by its dual, to obtain

$$\begin{aligned} &\max_{x \in X} cx + \min_{y} b_2 y \\ &\text{s.t. } Qx + By \geq d \\ &\qquad y \geq 0. \end{aligned}$$

Under the boundedness assumption, the dual of the inner problem is feasible and bounded for each $x \in X$. In a symmetric way, one can reverse the roles of x and u to obtain the equivalent formulation

$$\begin{aligned} &\max_{u \in U} ud + \min_{v} vb_1 \\ &\text{s.t. } uQ + vA \geq c \\ &\qquad v \geq 0. \end{aligned}$$

Thus, the solution of BILP can be obtained by solving either one of the symmetric bilevel programs

$$\begin{aligned} &\text{LLBP}_4 \\ &\max_{x,y} cx + b_2 y \\ &\text{s.t. } Ax \leq b_1 \\ &\qquad x \geq 0 \\ &\qquad y \in \arg\min_{y} b_2 y \\ &\qquad\qquad \text{s.t. } Qx + By \geq d \\ &\qquad\qquad\qquad y \geq 0 \end{aligned} \qquad \begin{aligned} &\text{LLBP}_5 \\ &\max_{u,v} ud + vb_1 \\ &\text{s.t. } uB \leq b_2 \\ &\qquad u \geq 0 \\ &\qquad v \in \arg\min_{v} vb_1 \\ &\qquad\qquad \text{s.t. } uQ + vA \geq c \\ &\qquad\qquad\qquad v \geq 0, \end{aligned}$$

These two problems correspond to "max-min" programs, i.e., bilevel program involving opposite objective functions.

If BILP is unbounded, the above transformations are no longer valid as the inner problem may prove infeasible for some values of $x \in X$ (or $u \in U$). For instance, the existence of a ray (unbounded direction) in u-space implies that there exist $\bar{x} \in X$ and $\bar{u}$ with $\bar{u}B \leq 0$ such that $\bar{u}(d - Q\bar{x}) > 0$. Equivalently there exists a vector $\bar{x}$ such that the inner problem in BILP_2 is unbounded, which implies in turn that its dual is infeasible with respect to $\bar{x}$.

In order to be equivalent to BILP, LLBP_4 should therefore select an x-value for which the lower level problem is infeasible. However, this is inconsistent with the optimal solution of a bilevel program being admissible. Actually, Audet et al. (1999) have shown that determining whether there exists an x in X such that

$$Y(x) = \{y \geq 0 : By \geq d - Qx\}$$

is empty, is strongly $\mathcal{NP}$-complete. Equivalently, determining if BILP is bounded is strongly $\mathcal{NP}$-complete. This result was achieved by constructing an auxiliary bilinear program BILP$'$ (always bounded) such that BILP is unbounded whenever the optimal value of BILP$'$ is positive. Based on this technique, the bilevel reformulation can be used to "solve" separable bilinear programs, whether they are bounded or not.

2.2 Complexity of linear bilevel programming

While one may derive complexity results about bilevel programs via the bilinear programming connection, it is instructive to perform reductions directly from standard combinatorial problems. After Jeroslow (1985) initially showed that LLBP is $\mathcal{NP}$-hard, Hansen et al. (1992) proved $\mathcal{NP}$-hardness, using a reduction from KERNEL (see Garey and Johnson, 1979)). Vicente et al. (1994) strengthened these results and proved that checking strict or local optimality is also $\mathcal{NP}$-hard. In this section, we present different proofs, based on a reduction from 3-SAT.

Let $x_1, \ldots, x_n$ be n Boolean variables and

$$F = \bigwedge_{i=1}^{m} (l_{i1} \vee l_{i2} \vee l_{i3})$$

be a 3-CNF formula involving m clauses with literals l_{ij}.[3] To each clause $(l_{i1} \vee l_{i2} \vee l_{i3})$ we associate a linear Boolean inequality of the form

$$v_{i1} + v_{i2} + v_{i3} \geq 1$$

where

$$v_{ij} = \begin{cases} x_k & \text{if } l_{ij} = x_k, \\ 1 - x_k & \text{if } l_{ij} = \overline{x}_k. \end{cases}$$

According to this scheme, the inequality

$$x_1 + (1 - x_4) + x_6 \geq 1$$

corresponds to the clause $(x_1 \vee \overline{x}_4 \vee x_6)$. Using matrix notation, the inequalities take the form

$$A_S x \geq \mathbf{1} + c$$

where A_S is a matrix with entries in $\{0, 1, -1\}$, and the elements of the vector c lie between -3 and 0. By definition, S is satisfiable if and only

[3] A literal consists in a variable or its negation.

if a feasible binary solution of this linear system exists. We have seen that it is indeed easy to force variables to take binary values trough a bilevel program. The reduction makes use of this transformation.

THEOREM 7.2 *The linear bilevel program LLBP is strongly $\mathcal{NP}$-hard.*

Proof. Consider the following LLBP:

$$\begin{aligned}
&\min_{x,z} F(x,z) = \sum_{i=1}^{n} z_i \\
&\text{s.t. } A_S x \geq \mathbf{1} + c \\
&\quad 0 \leq x_i \leq 1 \quad i = 0, \ldots, n \\
&\quad z \in \arg\max \left\{ \sum_{i=1}^{n} z_i : \begin{array}{l} z_i \leq x_i \\ z_i \leq 1 - x_i \\ z \geq 0 \end{array} \right\}
\end{aligned}$$

We claim that S is satisfiable if and only if the optimal solution of the LLBP is 0 (note that 0 is a lower bound on the optimal value). First assume that S is satisfiable and let $x = (x_1, \ldots, x_n)$ be a truth assignment for S. Then the first level constraints are verified and the sole feasible lower level solution corresponds to setting $z_i = 0$ for all i. Since this rational solution (x, z) achieves a value of 0, it is optimal. Assume next that S is not satisfiable. Any feasible x-solution must be fractionary and, since every rational solution satisfies $z_i = \min\{x_i, 1 - x_i\}$, at least one z_i must assume a positive value, and the objective $F(x, z)$ cannot be driven to zero. This completes the proof. □

COROLLARY 7.1 *There is no fully polynomial approximation scheme for LLBP unless $\mathcal{P} = \mathcal{NP}$.*

To prove the local optimality results, Vicente, Savard and Júdice adapted techniques developed by Pardalos and Schnitger (1988) for non-convex quadratic programming, where the problem of checking (strict or not) local optimality was proved to be equivalent to solving a 3-SAT problem. The present proof differs slightly from the one developed in Vicente et al. (1994).

The main idea consists in constructing an equivalent but *degenerate* bilevel problem of 3-SAT. For that, we *augment* the Boolean constraints with an additional variable x_0, change the right hand-side to 3/2, and bound the x variables. For each instance S of 3-SAT, let us consider the

constraint set:

$$\begin{aligned} & A_S x + I x_0 \geq \tfrac{3}{2} + c \\ & \tfrac{1}{2} - x_0 \leq x_i \leq \tfrac{1}{2} + x_0 \qquad i = 1, \ldots, n \\ & x_i \geq 0 \qquad\qquad\qquad\qquad i = 0, \ldots, n. \end{aligned}$$

Obviously, the solution $x^* = (0, \frac{1}{2}, \ldots, \frac{1}{2})$ satisfies the above linear inequalities, but this does not guarantee that S is satisfiable. Hence, we will consider a bilevel program that will have, at this solution, the same objective value than we would obtain if S is satisfiable.

THEOREM 7.3 *Checking strict local optimality in linear bilevel programming is $\mathcal{NP}$-hard.*

Proof. Consider the following instance of a linear bilevel program:

$$\begin{aligned} & \min_{x,l,m,z} F(x,l,m,z) = \sum_{i=1}^{n} z_i \\ & \text{s.t. } A_S x + I x_0 \geq \tfrac{3}{2} + c \\ & \quad \tfrac{1}{2} - x_0 \leq x_i \leq \tfrac{1}{2} + x_0, \qquad i = 1, \ldots, n \\ & \quad x_i \geq 0, \qquad\qquad\qquad\quad i = 0, \ldots, n \\ & \quad l, m, z \in \arg\max \Big\{ \sum_{i=1}^{n} z_i : \; x_i - l_i = \tfrac{1}{2} - x_0 \\ & \qquad\qquad\qquad\qquad\qquad\quad x_i + m_i = \tfrac{1}{2} + x_0 \\ & \qquad\qquad\qquad\qquad\qquad\quad z_i \leq l_i, \; z_i \leq m_i \; i = 1, \ldots, n \\ & \qquad\qquad\qquad\qquad\qquad\quad z \geq 0 \qquad\qquad\qquad\qquad \Big\} \end{aligned}$$

Let $x^* = (0, \frac{1}{2}, \ldots, \frac{1}{2})$ and $l^* = m^* = z^* = 0$. We claim that S is satisfiable if and only if the point (x^*, l^*, m^*, z^*) is not a strict minimum. Since all variables z_i are forced to be nonnegative then:

$$F(x, l, m, z) \geq 0.$$

First, assume that S is satisfiable. Let $x_1, \ldots, x_n$ be a true assignment for S and set, for any $x_0 \in [0, \frac{1}{2}]$

$$\bar{x} = \begin{cases} \frac{1}{2} - x_0 & \text{if } x_i = 0, \\ \frac{1}{2} + x_0 & \text{if } x_i = 1, \end{cases}$$

i.e., $\bar{x}$ satisfies the upper level constraints. Furthermore $\bar{l} = 0$, $\bar{m} = 0$ and $\bar{z} = 0$ is the optimal solution of the lower level problem for $\bar{x}$ fixed.

Hence $(\bar{x}, \bar{l}, \bar{m}, \bar{z})$ belongs to the induced region associated with the linear bilevel program. Since $F(\bar{x}, \bar{l}, \bar{m}, \bar{z}) = 0$, we claim that $(\bar{x}, \bar{l}, \bar{m}, \bar{z})$ is a global minimum of the linear bilevel program.

Clearly, $F(x, l, m, z) = 0$ if and only if $x_i \in \{\frac{1}{2} - x_0, \frac{1}{2} + x_0\}$, for all $i = 1, \ldots, n$. If this last condition holds, then $l_i = 0$ or $m_i = 0$ and $z_i = 0$ for all $i = 1, \ldots, n$ and $F(x, l, m, z) = 0$. Since x_0 can be chosen arbitrarily close to 0, x^* cannot be a strict local minimum.

Assume next that (x^*, l^*, m^*, z^*) is not a strict local minimum. There exists a rational point (x^1, l^1, m^1, z^1) such that $F(x^1, l^1, m^1, z^1) = 0$, and this point satisfies $l^1 = m^1 = z^1 = 0$ and $x_i^1 = \frac{1}{2} - x_0$ or $x_i^1 = \frac{1}{2} + x_0$ for all i and some x_0. Then the assignment

$$\begin{cases} x_i = 0 & \text{if } x_i^1 = \frac{1}{2} - x_0, \\ x_i = 1 & \text{if } x_i^1 = \frac{1}{2} + x_0 \end{cases}$$

is a truth assignment for S. □

THEOREM 7.4 *Checking local optimality in linear bilevel programming is $\mathcal{NP}$-hard.*

The proof, which is based on complexity results developed in Pardalos and Schnitger (1988) and Vicente et al. (1994), will not be presented. Let us however mention that the underlying strategy consists in slightly discriminating against the rational points assuming value 0, through the addition of penalty factor with respect to x_0, yielding the LLBP

$$\begin{aligned} \min_{x,l,m,z,w} \; & F(x, l, m, z, w) = \sum_{i=1}^{n} z_i - \frac{1}{2n} \sum_{i=1}^{n} w_i \\ \text{s.t. } & A_S x \geq \tfrac{3}{2} + c \\ & \tfrac{1}{2} - x_0 \leq x_i \leq \tfrac{1}{2} + x_0 \qquad i = 1, \ldots, n \\ & x_i \geq 0 \qquad\qquad\qquad\qquad i = 0, \ldots, n \\ & l, m, z, w \in \arg\max \Big\{ \sum_{i=1}^{n} z_i - \sum_{i=1}^{n} w_i : \\ & \qquad\qquad x_i - l_i = \tfrac{1}{2} - x_0 \\ & \qquad\qquad x_i + m_i = \tfrac{1}{2} + x_0 \\ & \qquad\qquad z_i \leq l_i, \; z_i \leq m_i \; i = 1, \ldots, n \\ & \qquad\qquad w_i \geq x_i - \tfrac{1}{2} \; w_i \geq \tfrac{1}{2} - x_i, \; i = 1, \ldots, n \\ & \qquad\qquad z, w \geq 0. \Big\} \end{aligned}$$

3. Optimal pricing via bilevel programming

Although much attention has been devoted to linear bilevel programs, their mathematical structure does not fit many real life situations, where it is much more likely that interaction between conflicting agents occurs through the model's objectives rather than joint constraints. In this section, we consider such an instance that, despite its simple structure, forms the paradigm that lies behind large-scale applications in revenue management and pricing, such as considered by Côté et al. (2003).

3.1 A simple pricing model

Let us consider a firm that wants to price independently (bundling is not allowed) a set of products aimed at customers having specific requirements and alternative purchasing sources. If the requirements are related in a linear manner to the resources (products), one obtains the bilinear-bilinear bilevel program (BBBP):

$$\begin{aligned} \mathsf{BBBP}: \quad & \max_{t,x,y} tx \\ & \text{s.t. } (x,y) \in \arg\min_{x,y}(c+t)x + dy \\ & \qquad Ax + By = b \\ & \qquad x, y \geq 0, \end{aligned}$$

where t denotes the upper level decision vector, (c, d) the "before tax" price vector , (x, y) the consumption vector, (A, B) the "technology matrix" and b the demand vector. In the above, a trade-off must be achieved between high t-values that price the leader's products away from the customer(s), and low prices that induce a low revenue.

In a certain way, the structure of BBBP is dual to that of LLBP, in that the constraint set is separable and interaction occurs only through the objective functions. The relationship between LLBP and BBBP actually goes further. By replacing the lower level program by its primal-dual characterization, one obtains the equivalent bilinear and single-level program

$$\begin{aligned} & \max_{t,x,y} tx \\ & \text{s.t. } Ax + By = b \\ & \qquad x, y \geq 0 \\ & \qquad \lambda A \leq c + t \\ & \qquad \lambda B \leq d \\ & \qquad (c + t - \lambda A)x = 0 \\ & \qquad (d - \lambda B)y = 0. \end{aligned}$$

Without loss of generality, one can set $t = \lambda A - c$. Indeed, if $x_i > 0$, $t_i = (\lambda A)_i - c_i$ follows from the next-to-last orthogonality conditions whereas, if $x_i = 0$, the leader's objective is not affected by the value of t_i. Now, a little algebra yields:

$$tx = \lambda Ax - cx = \lambda(b - By) - cx = \lambda b - (cx + dy)$$

and one is left with a program involving a single nonlinear (actually bilinear and separable) constraint, that can be penalized to yield the bilinear program

$$\begin{aligned} \mathsf{PENAL}: \quad & \max_{x,y,\lambda} \lambda b - (cx + dy) - M(d - \lambda B)y \\ & \text{s.t. } Ax + By = b \\ & \quad x, y \geq 0 \\ & \quad \lambda B \leq d. \end{aligned}$$

Under mild feasibility and compactness assumptions, it has been shown by Labbé et al. (1998) that there exists a finite value M^* of the penalty parameter M such that, for every value of M larger than M^*, any optimal solution of the penalized problem satisfies the orthogonality constraint $(d - \lambda B)y = 0$, i.e., the penalty is exact.[4] Since the penalized problem is bilinear and separable, optimality must be achieved at some extreme point of the feasible polyhedron. Moreover, the program can, using the techniques of Section 2.1.2, be reformulated as a linear bilevel program of a special type.

The reverse transformation, from a generic LLBP to BBBP, is not straightforward and could not be achieved by the authors. However, since BBBP is strongly $\mathcal{NP}$-hard, such polynomial transformation *must* exist.

3.2 Complexity

In this section, we consider a subclass of BBBP initially considered by Labbé et al., where the feasible set $\{(x, y) : Ax + By = b, x, y \geq 0\}$ is that of a multicommodity flow problem, without upper bound constraints on the links of the network. For a given upper level vector t, a solution to the lower level problem corresponds to assigning demand to shortest

[4] Be careful though: the stationary points of the penalized and original problems need not be in one-to-one relationship!

paths linking origin and destination nodes. This yields:

$$\text{TOLL}: \quad \max_{t,x,y} t\sum_{k\in K} x^k$$
$$\left.\begin{array}{l} \text{s.t. } (x^k, y^k) \in \arg\min_{x^k,y^k} tx^k + dy^k \\ Ax^k + By^k = b^k \\ x^k, y^k \geq 0 \end{array}\right\} \quad \forall k \in K,$$

where (A, B) denotes the node-arc incidence matrix of the network, and b^k denotes the demand vector associated with the origin-destination pair, or "commodity" $k \in K$.

Note that since a common toll vector t applies to *all* commodities, TOLL does not quite fit the format of BBBP. However, by setting $x = \sum_{k\in K} x^k$ for both objectives[5] and incorporating the compatibility constraint $x = \sum_{k\in K} x^k$ (at either level), we obtain a *bona fide* BBBP.

THEOREM 7.5 *TOLL is strongly $\mathcal{NP}$-hard, even when $|K| = 1$.*

The proof relies on the reduction on the reformulation of 3-SAT as toll problem involving a single origin-destination pair, and is directly adapted form the paper by Roch et al. (2004). Let $x_1, \ldots, x_n$ be n Boolean variables and

$$F = \bigwedge_{i=1}^{m} (l_{i1} \vee l_{i2} \vee l_{i3}) \tag{7.1}$$

be a 3-CNF formula consisting of m clauses with literals (variables or their negations) l_{ij}. For each clause, we construct a "cell", i.e., a sub-network comprising one toll arc for each literal. Cells are connected by a pair of parallel arcs, one of which is toll-free, and by arcs linking literals that cannot be simultaneously satisfied (see Figure 7.3).

The idea is the following: if the optimal path goes through toll arc T_{ij}, then the corresponding literal l_{ij} is TRUE. The sub-networks are connected by two parallel arcs, a toll-free arc of cost 2 and a toll arc of cost 0, as shown in Figure 7.3.

If F is satisfiable, we want the optimal path to go through a single toll arc per sub-network (i.e., one TRUE literal per clause) and simultaneously want to make sure that the corresponding assignment of variables is consistent; i.e., paths that include a variable and its negation must

[5]This is allowed by the fact that the lower level constraints are separable by commodity.

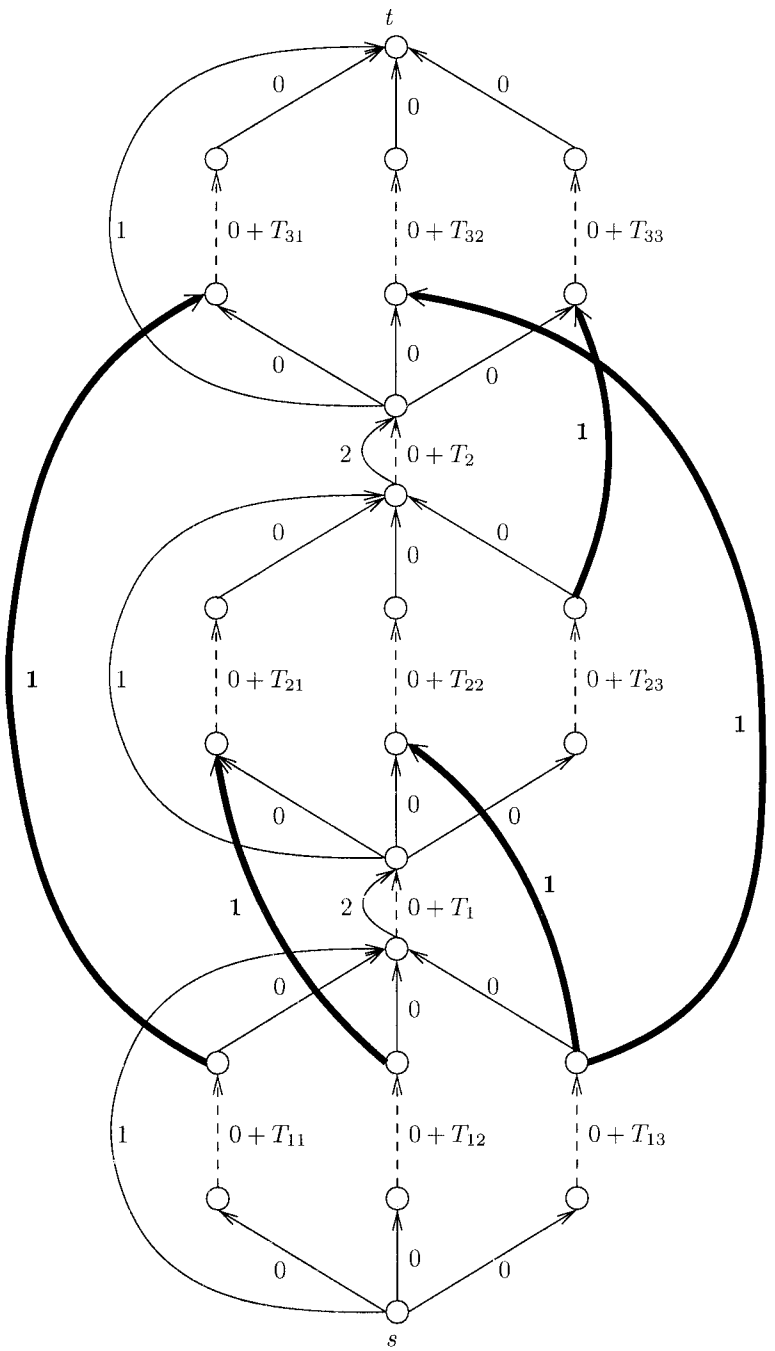


Figure 7.3. Network for the formula $(x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee x_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee x_3 \vee x_4)$. Inter-clause arcs are bold. Path through T_{12}, T_{22}, T_{32} is optimal ($x_2 = x_3 =$TRUE).

be ruled out. For that purpose, we assign to every pair of literals corresponding to a variable and its negation an inter-clause toll-free arc between the corresponding toll arcs (see Figure 7.3). As we will see, this implies that *inconsistent* paths, involving a variable and its negation, are suboptimal.

Since the length of a shortest toll-free path is $m + 2(m-1) = 3m - 2$ and that of a shortest path with zero tolls is 0, $3m-2$ is an upper bound on the revenue. We claim that F is satisfiable if and only if the optimal revenue is equal to that bound.

Assume that the optimal revenue is equal to $3m - 2$. Obviously, the length of the optimal path when tolls are set to 0 must be 0, otherwise the upper bound cannot be reached. To achieve this, the optimal path has to go through one toll arc per sub-network (it cannot use inter-clause arcs) and tolls have to be set to 1 on selected literals, $C + 1$ on other literals and 2 on tolls T_k, $\forall\ k$. We claim that the optimal path does not include a variable and its negation. Indeed, if that were the case, the inter-clause arc joining the corresponding toll arcs would impose a constraint on the tolls between its endpoints. In particular, the toll T_k immediately following the initial vertex of this inter-clause arc would have to be set at most to 1, instead of 2. This yields a contradiction. Therefore, the optimal path must correspond to a consistent assignment, and F is satisfiable (note: if a variable and its negation do not appear on the optimal path, this variable can be set to any value).

Conversely if F is satisfiable, at least one literal per clause is TRUE in a satisfying assignment. Consider the path going through the toll arcs corresponding to these literals. Since the assignment is consistent, the path does not simultaneously include a variable and its negation, and no inter-clause arc limits the revenue. Thus, the upper bound of $3m - 2$ is reached on this path.

Another instance, involving several commodities but restricting each path to use a single toll arc, also proved $\mathcal{NP}$-hard. Indeed, consider the "river tarification problem", where users cross a river by either using one of many toll bridges, or by flying directly to their destination on a toll-free arc. The proof of $\mathcal{NP}$-completeness also makes use of 3-SAT, but there is a twist: each cell now corresponds to a variable rather than a clause, and is thus "dual" to the previous transformation (see Grigoriev et al., 2004, for details). Apart of its elegance, the dual reduction has the advantage of being related to the corresponding optimization problem, i.e., one can maximize the number of satisfied clauses by solving the related TOLL problem. This is not true of the primal reduction, where the truth assignment is only valid when the Boolean formula can be satisfied. Indeed, the solution of the TOLL reduction may attain a near-

optimal value of $3m-3$ with only one clause being satisfied, thus making the truth assignment of the variables irrelevant. For instance, consider an instance where a variable and its negation appear as literals in the first and last clauses.[6] Then, a revenue of $3m-3$, one less than the optimal revenue, is achieved on the path that goes through the two literals and the toll-free link between them, by setting the tolls on the two toll arcs of that path to 0 and $3m-3$ respectively.

We conclude this section by mentioning that TOLL is polynomially solvable when the number of toll arcs is bounded by some constant. If the set of toll arcs reduces to a singleton, a simple ordering strategy can be applied (see Labbé et al., 1998). In the general case, path enumeration yields a polynomial algorithm that is unfortunately not applicable in practice (see Grigoriev et al., 2004). Other polynomial cases have been investigated by van Hoesel et al. (2003).

3.3 The traveling salesman problem

Although the relationship between the traveling salesman problem (TSP in short) and TOLL is not obvious, the first complexity result involved TSP or, to be more precise, the Hamiltonian path problem (HPP). The reduction considered in Labbé et al. (1998) goes as follows: Given a directed graph with n nodes, among them two distinguished nodes: an origin s and a destination t the destination, we consider the graph obtained by creating a toll-free arc from s to t, with length $d_{st} = n-1$. Next, we endow the remaining arcs, all toll arcs, with cost -1 and impose a lower bound of 2 on all of them. Then, it is not difficult to see that the maximal toll revenue, equal to $2n-2$, is obtained by setting $t_a = 2$ on the arcs of any Hamiltonian path, and $t_a = n+1$ elsewhere.

The weakness of that reduction is that it rests on two assumptions that are not required in the reductions presented in the previous sections, that is, negativity of arc lengths and lower bounds on toll values. Notwithstanding, the relationship between TOLL and TSP has proved fruitful. To see this, let us follow Marcotte et al. (2003) and consider a TSP involving a graph G and a length vector c. First, we transform the TSP into an HPP by duplicating the origin node s and replacing all arcs (i,s) by arcs from i to t. It is clear that the solutions to TSP and HPP are in one-to-one correspondence. Second, we incorporate a toll-free arc (s,t) with cost n, we set the fixed cost of the remaining arcs to $-1+c_a/L$ and the lower bounds on tolls to $2-c_a/L$, where L is some suitably large

[6]Remark: The clauses involving the two opposite literals can always be made the first and the last, through a straightforward permutation. This shows that the model is sensitive to the rearrangement of clauses.

constant, $L = n \times \max_a\{c_a\}$ for instance. Then, any solution to the toll problem on the modified network yields a shortest Hamiltonian path. This toll problem takes the form

$$\begin{aligned}
&\max_{t,x,y} \sum_a t_a x_a \\
&(x,y) \in \arg\min_{x,y} \sum_a (-1 + c_a/L + t_a)x_a + n y_{st} \\
&\qquad \text{s.t. flow conservation} \\
&\qquad\qquad x \geq 0.
\end{aligned}$$

Replacing the lower level linear program by its optimality conditions, one obtains a linear program including additional complementarity constraints. The latter, upon the introduction of binary variables, can be linearized to yield a MIP formulation of the TSP that, after some transformations, yields:

$$\begin{aligned}
&\min_{x,u} \sum_a c_{ij} x_{ij} \\
&\text{s.t.} \quad \sum_j x_{ij} = 1 && \forall i \\
&\qquad \sum_i x_{ij} = 1 && \forall j \\
&\qquad u_i - u_j \leq (n-2) + (1-n)x_{ij} + (3-n)x_{ji} && \forall (i,j) \\
&\qquad u_j \leq (n-2) + (3-n)x_{1j} + x_{j(n+1)} && \forall j \neq 1 \\
&\qquad u_j \geq (n-3)x_{j(n+1)} - x_{1j} + 2 && \forall j \neq 1 \\
&\qquad x \text{ binary valued,}
\end{aligned}$$

where u corresponds to the dual vector associated with the lower level program. It is in a certain way surprising, and certainly of theoretical interets that, through standard manipulations, one achieves the mixed integer program Note that this program is nothing but the lifted formulation of the Miller–Tucker–Zemlin constraints derived by Desrochers and Laporte (1991), where the three constraints involving the vector u are facet-defining.

In the symmetric case, the analysis supports a multicommodity extension, where each commodity is assigned to a subtour between two prespecified vertices. More precisely, let $[v_1, v_2, \ldots, v_{|K|}]$ be a sequence of vertices. Then, the flow for commodity $k \in K$ must follow a path from vertex v_k to v_{k+1},[7] and the sequence of such paths must form a

[7]By convention, $v_{K+1} \equiv v_1$.

Hamiltonian circuit. If the number of commodity is 3 or less, the ordering of the vertices is irrelevant. In the Euclidean case and if $|K|$ is more than 3, it is yet possible to find a set of vertices that are extreme points of the convex hull of vertices, together with the order in which they must be visited in some optimal tour (see Flood, 1956).

When applied to graphs from the TSPLIB library (TSPLIB), the linear relaxation of the three-commodity reformulation provides lower bounds of quality comparable to those obtained by the relaxation proposed by Dantzig et al. (1954). This is all the more surprising in the view that the latter formulation is exponential, while the former is in $O(n^2)$.

3.4 Final considerations

This chapter has provided a very brief overview of two important classes of bilevel programs, from the perspective of combinatorial optimization. Those classes are not the only ones to possess a combinatorial nature. Indeed, let us consider a bilevel program (or an MPEC) where the induced region is the union of polyhedral faces.[8] A sufficient condition that an optimal solution be attained at an extreme point of the induced region is then that the upper level objective be concave in both upper and lower level variables. An interesting situation also occurs when the upper level objective is quadratic and *convex*. In this case, the solution of the problem restricted to a polyhedral face occurs at an extreme point of the primal-dual polyhedron, and it follows that the problem is also combinatorial.

Actually, bilevel programs almost always integrate a combinatorial element. For instance, let us consider the general bilevel program:

$$\begin{aligned} &\min_{x,y} f(x,y) \\ &\text{s.t. } y \in \arg\min_{y} g(x,y) \\ &\qquad G(x,y) \leq 0. \end{aligned}$$

Under suitable constraints (differentiability, convexity and regularity of the lower level problem), one can replace the lower level problem by its

[8]This situation is realized when the lower level is a linear, a convex quadratic, or a linear complementarity problem, and joint constraints, whenever they exist, are linear.

Kuhn-Tucker conditions and obtain the equivalent program

$$\begin{aligned} \mathsf{BLKKT}: \quad & \min_{x,y} f(x,y) \\ & \text{s.t. } G(x,y) \leq 0 \\ & \quad \nabla_y g(x,y) + \lambda \nabla_y G(x,y) = 0 \\ & \quad \lambda G(x,y) = 0. \end{aligned}$$

If the set of active constraints were known a priori, BLKKT would reduce to a standard nonlinear program. Provided that f, g and each of the G_i's be convex, the last constraint could yet make it non-convex, albeit "weakly," in the sense that replacing all functions by their quadratic approximations would make the bilevel problem convex. The main computational pitfall is actually the identification of the active set. This two-sided nature of bilevel programming and MPEC is well captured in the formulation proposed by Scholtes (2004), which distinguishes between the continuous and combinatorial natures of MPECs. By rearranging variables and constraints, one can reformulate BLKKT as the generic program

$$\begin{aligned} & \min_x f(x) \\ & \text{s.t. } G(x) \in \mathcal{Z}. \end{aligned}$$

If $\mathcal{Z}$ is the negative orthant, this is nothing more than a standard nonlinear program. However, special choices of $\mathcal{Z}$, may force pairs of variables to be complementary. It is then ill-advised to linearize $\mathcal{Z}$, and the right approach is to develop a calculus that does not sidestep the combinatorial nature of the set $\mathcal{Z}$. Along that line of reasoning, Scholtes proposes an SQP (Sequential Quadratic Programming) algorithm that leaves $\mathcal{Z}$ untouched and is guaranteed, under mild assumptions, to converge to a strong stationary solution. While this approach is satisfactory from a local analysis point of view, it does not settle the main challenge, that is, aiming for an optimal or near-optimal solution. In our view, progress in this direction will be achieved by addressing problems with specific structures, such as the BBBP.

Acknowledgments We would like to thank the following collaborators with whom most of the results have been obtained: Charles Audet, Luce Brotcorne, Benoît Colson, Pierre Hansen, Brigitte Jaumard, Joachim Júdice, Martine Labbé, Sébastien Roch, Frédéric Semet and Luís Vicente.

References

Alarie, S., Audet, C., Jaumard, B., and Savard, G. (2001). Concavity cuts for disjoint bilinear programming. *Mathematical Programming*, 90:373–398.

Audet, C. (1997). Optimisation globale structurée : propriétés, équivalences et résolution. Ph.D. thesis, École Polytechnique de Montréal.

Audet, C., Hansen, P., Jaumard, B., and Savard, G. (1997). Links between linear bilevel and mixed 0-1 programming problems. *Journal of Optimization Theory and Applications*, 93:273–300.

Audet, C., Hansen, P., Jaumard, B., and Savard, G. (1999). A Symmetrical linear maxmin approach to disjoint bilinear programming. *Mathematical Programming*, 85:573–592.

Bard, J.F. (1998). *Practical Bilevel Optimization — Algorithms and Applications*. Kluwer Academic Publishers.

Beale, E. and Small, R. (1965). Mixed integer programming by a branch and bound technique. In: *Proceedings of the 3rd IFIP Congress*. pp. 450–451.

Candler, W. and Norton, R. (1977). Multilevel programing. Technical Report 20, World Bank Development Research Center, Washington, DC.

Côté, J.-P., Marcotte, P., and Savard, G. (2003). A bilevel modeling approach to pricing and fare optimization in the airline industry. *Journal of Revenue and Pricing Management*, 2:23–36.

Dantzig, G., Fulkerson, D., and Johnson, S. (1954). Solution of a large-scale traveling salesman problem. *Operations Research*, 2:393–410.

Dempe, S. (2002). *Foundations of Bilevel Programming*. Nonconvex Optimization and Its Applications, vol. 61. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Desrochers, M. and Laporte, G. (1991). Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. *Operations Research Letters*, 10:27–36.

Flood, M. (1956). The traveling salesman problem. *Operations Research*, 4:61–75.

Fukushima, M. (1992). Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Mathematical Programming*, 53:99–110.

Fukushima, M. and Pang, J. (1999). Complementarity constraint qualifications and simplified B-stationarity conditions for mathematical programs with equilibrium constraints. *Computational Optimization and Applications*, 13:111–136.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, New York.

Grigoriev, A., van Hoesel, S., van der Kraaij, A., Uetz, M., and Bouhtou, M. (2004). Pricing network edges to cross a river. Technical Report RM04009, Maastricht Economic Research School on Technology and Organisation, Maastricht, The Netherlands.

Hansen, P., Jaumard, B., and Savard, G. (1992). New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13:1194–1217.

Hogan, W. (1973). Point-to-set maps in mathematical programming. *SIAM Review*, 15:591–603.

Jeroslow, R. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32:146–164.

Konno, H. (1971). Bilinear Programming. II. Applications of Bilinear Programming. Technical Report Technical Report No. 71-10, Operations Research House, Department of Operations Research, Stanford University, Stanford.

Kočvara, M. and Outrata, J. (1994). On optimization of systems governed by implicit complementarity problems. *Numerical Functional Analysis and Optimization*, 15:869–887.

Labbé, M., Marcotte, P., and Savard, G. (1998). A bilevel model of taxation and its applications to optimal highway pricing. *Management Science*, 44:1608–1622.

Loridan, P. and Morgan, J. (1996). Weak via strong Stackelberg problem: New results. *Journal of Global Optimization*, 8:263–287.

Luo, Z., Pang, J., and Ralph, D. (1996). *Mathematical Programs with Equilibrium Constraints.* Cambridge University Press.

Marcotte, P., Savard, G., and Semet, F. (2003). A bilevel programming approach to the travelling salesman problem. *Operations Research Letters*, 32:240–248.

Marcotte, P., Savard, G., and Zhu, D. (2001). A trust region algorithm for nonlinear bilevel programming. *Operations Research Letters*, 29:171–179.

Mills, H. (1960). Equilibrium points in finite games. *Journal of the Society for Industrial and Applied Mathematics*, 8:397–402.

Nash, J. (1951). Noncooperative games. *Annals of Mathematics*, 14:286–295.

Pardalos, P. and Schnitger, G. (1988). Checking local optimality in constrained qu adratic programming is NP-hard. *Operations Research Letters*, 7:33–35.

Roch, S., Savard, G., and Marcotte, P. (2004). An Approximation Algorithm for Stackelberg Network Pricing.

Savard, G. (1989). *Contribution à la programmation mathématique à deux niveaux.* Ph.D. thesis, Ecole Polytechnique de Montréal, Université de Montréal.

Scholtes, S. (2004). Nonconvex structures in nonlinear programming. *Operations Research*, 52:368–383.

Scholtes, S. and Stöhr, M. (1999). Exact penalization of mathematical programs with equilibrium constraints. *SIAM Journal on Control and Optimization*, 37:(2):617–652.

Shimizu, K., Ishizuka, Y., and Bard, J. (1997). *Nondifferentiable and Two-Level Mathematical Programming.* Kluwer Academic Publishers.

Stackelberg, H. (1952). *The Theory of Market Economy.* Oxford University Press, Oxford.

TSPLIB. A library of traveling salesman problems. Available at `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95`.

van Hoesel, S., van der Kraaij, A., Mannino, C., Oriolo, G., and Bouhtou. M. (2003). Polynomial cases of the tarification problem. Technical Report RM03053, Maastricht Economic Research School on Technology and Organisation, Maastricht, The Netherlands.

Vicente, L., Savard, G., and Júdice, J. (1994). Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, 81:379–399.

Vicente, L., Savard, G., and Júdice, J. (1996). The discrete linear bilevel programming problem. *Journal of Optimization Theory and Applications*, 89:597–614.

Chapter 8

VISUALIZING, FINDING AND PACKING DIJOINS

F.B. Shepherd
A. Vetta

Abstract We consider the problem of making a directed graph *strongly connected*. To achieve this, we are allowed for assorted costs to add the reverse of any arc. A successful set of arcs, called a *dijoin*, must intersect every directed cut. Lucchesi and Younger gave a min-max theorem for the problem of finding a minimum cost dijoin. Less understood is the extent to which dijoins pack. One difficulty is that dijoins are not as easily visualized as other combinatorial objects such as matchings, trees or flows. We give two results which act as visual certificates for dijoins. One of these, called a lobe decomposition, resembles Whitney's ear decomposition for 2-connected graphs. The decomposition leads to a natural optimality condition for dijoins. Based on this, we give a simple description of Frank's primal-dual algorithm to find a minimum dijoin. Our implementation is purely primal and only uses greedy tree growing procedures. Its runtime is $O(n^2m)$, matching the best known, due to Gabow. We then consider the function $f(k)$ which is the maximum value such that every weighted directed graph whose minimum weight of a directed cut is at least k, admits a weighted packing of $f(k)$ dijoins (a weighted packing means that the number dijoins containing an arc is at most its weight). We ask whether $f(k)$ approaches infinity. It is not yet known whether $f(k_0) \geq 2$ for some constant k_0. We consider a concept of *skew submodular flow polyhedra* and show that this dijoin-pair question reduces to finding conditions on when their integer hulls are non-empty. We also show that for any k, there exists a half-integral dijoin packing of size $k/2$.

1. Introduction

We consider the basic problem of *strengthening* a network $D = (V, A)$ so that it becomes strongly connected. That is, we require that there be a directed path between any pair of nodes in both directions. To

achieve this goal we are allowed to add to the graph (at varying costs) the reverse of some of the network arcs. Equivalently, we are searching for a collection of arcs that induce a strongly connected graph when they are either contracted or made bi-directional. It is easy to see that our problem is that of finding a *dijoin*, a set of arcs that intersects every directed cut.

Despite its fundamental nature, the *minimum cost dijoin problem* is not a standard modelling tool for the combinatorial optimizer in the same way that shortest paths, matchings and network flows are. We believe that widespread adoption of these other problems stems from the fact that they can be tackled using standard concepts such as dynamic programming, greedy algorithms, shrinking, and tree growing. Consequently, simple and efficient algorithms can be implemented and also, importantly, taught. One objective of this paper, therefore, is to examine dijoins using classical combinatorial optimization techniques such as decomposition, arborescence growing and negative cycle detection. Under this framework, we present a primal version of Frank's seminal primal-dual algorithm for finding a minimum cost dijoin. The running time of our implementation is $O(n^2m)$, which matches the fastest known running time (due to Gabow, 1995). We also consider the question of packing dijoins and along the way we present open problems which hopefully serve to show that the theory of dijoins is still a rich and evolving topic. We begin, though, with some dijoin history.

1.1 Background

We start by discussing the origins of the first polytime algorithm (based on the Ellipsoid Method) for this problem: the Lucchesi – Younger Theorem. A natural lower bound on the number of arcs in a dijoin is the size of any collection of disjoint directed cuts. The essence of Lucchesi-Younger is to show that such lower bounds are strong enough to certify optimality. A natural generalization also holds when an integer cost vector c is given on the arcs. A collection $\mathcal{C}$ of directed cuts is a *c-packing* if for any arc a, at most c_a of the cuts in $\mathcal{C}$ contain a. The *size* of a packing is $|\mathcal{C}|$.

THEOREM 8.1 (LUCCHESI AND YOUNGER, 1978) *Let D be a digraph with a cost c_a on each arc. Then*

$$\min\left\{\sum_{a\in T} c_a : T \text{ is a dijoin}\right\} = \max\{|\mathcal{C}| : \mathcal{C} \text{ is a c-packing}\}.$$

This was first conjectured for planar graphs in the thesis of Younger (1963). It was conjectured for general graphs in Younger (1969) and

independently by Robertson (cf. Lucchesi and Younger, 1978). (A preliminary result for bipartite graphs appeared in McWhirter and Younger, 1971.) This theorem generated great interest after it was announced at the 1974 International Congress of Mathematicians in Vancouver. It proved also to be a genesis of sorts; we describe now some of the historical developments and remaining questions in combinatorial optimization which grew from it.

Consider the $0-1$ matrix C whose rows correspond to the incidence vectors of directed cuts in D. Theorem 8.1 implies that the dual of the linear program $\min\{cx : Cx \geq 1, x \geq 0\}$ always has an integral optimum for each integral vector c. Obviously, the primal linear program also always has integral optimal solutions since any $0-1$ solution identifies a dijoin. Matrices with this primal integrality property are called *ideal*. As discussed shortly, this new class of ideal matrices did not behave as well as its predecessors, such as matrices arising from bipartite matchings and network flows, each of which consisted of totally unimodular matrices.

Consider next an *integer dual* of the minimum dijoin problem where we reverse the roles of what is being minimized with what is being packed. Formally, for a $0-1$ matrix C, its *blocking matrix* is the matrix whose rows are the minimal $0-1$ solutions x to $Cx \geq 1$. A result of Lehman (1990) states that a matrix C is ideal if and only if its blocking matrix is ideal. Note that the blocking matrix of our directed cut incidence matrix C, is just the dijoin incidence matrix, which we denote by M. It follows that a minimum cost directed cut in a graph is obtained by solving the linear program $\min\{cx : Mx \geq 1, x \geq 0\}$.

Unlike the directed cut matrix, however, Schrijver (1980), showed that the dual of this linear program does not always possess an integral optimum. In particular, this implies that M is not totally unimodular. Figure 8.1 depicts the example of Schrijver. Note that it is a $0-1$-weighted digraph whose minimum weight directed cut is 2, but for

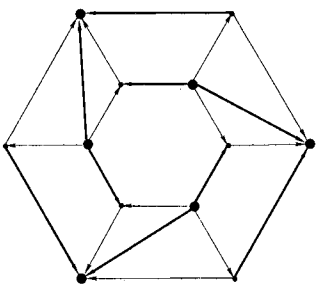


Figure 8.1. The Schrijver Example (bold arcs have weight 1)

which there does not exist a pair of disjoint dijoins amongst the arcs of positive weight. The example depends critically on the use of zero-weight arcs. Indeed, the following long-standing unweighted conjecture is still unresolved.

WOODALL'S CONJECTURE *The minimum cardinality of a directed cut equals the maximum cardinality of a collection of disjoint dijoins.*

Schrijver (1982) and Feofiloff and Younger (1987) verified the conjecture (even in the weighted case) if the digraph is *source-sink connected*, that is, there is a directed path from each source to each sink. Note that restricting the conjecture to planar graphs, and then translating to the dual map, one has the following question which is also open. In a planar digraph with no directed cycle of length less than k, there is a partition of its arcs into k disjoint *feedback arc sets* (collections of arcs whose deletion destroys all directed cycles).[1] Observe that two disjoint feedback arc sets can be trivially found in any graph. If we take an ordering of the nodes $v_1, v_2, \ldots, v_n$ then $A' = \{(v_i, v_j) \in A : i < j\}$ and $A - A'$ are feedback arc sets. Nothing is evidently known for $k > 2$, except in the case of series-parallel digraphs for which the problem was recently settled by Lee and Wakabayashi (2001).

We define a function $f(k)$ which is the maximum value such that every weighted digraph, whose minimum weight directed cut is at least k, contains a *weighted packing* of $f(k)$ dijoins. By weighted packing, we mean that the number of dijoins containing an arc is at most the arc's weight. We ask whether $f(k)$ goes to infinity as k increases. Currently it is not even known if $f(k_0) \geq 2$ for some k_0. As suggested by Pulleyblank (1994), one tricky aspect in verifying Woodall's Conjecture is that dijoins are not as easily visualized as directed cuts themselves or other combinatorial objects such as trees, matchings, flows, etc. For a given subset T of arcs, one must resort to checking whether each directed cut does indeed include an element of T. Motivated by this, in Section 2, we devise two "visual" certificates for dijoins. One of these, called a *lobe decomposition*, resembles Whitney's well-known ear decompositions for 2-connected and strongly connected graphs. This decomposition is used later to define augmenting structures for dijoins, and it also immediately implies the following.

THEOREM 8.5 *Let D be a digraph with arc weights w whose minimum directed cut is of weight at least 2. If each component of the graph induced*

[1] It was actually the feedback arc problem which originally motivated Younger (1963).

by the support of w induces a 2-edge-connected undirected graph, then the set of positive weight arcs contains two disjoint dijoins.

In Section 4 we discuss an approach to determining whether there exists a constant k_0 such that $f(k_0) \geq 2$. We conjecture that dijoins have an "Erdös-Posa property," that is, $f(k)$ approaches infinity. Even more strongly we propose:

CONJECTURE 8.1 *Let D be a weighted digraph whose minimum weight directed cut is of size k. Then there is a weighted packing of dijoins of size $\Omega(k)$.*

We prove the weaker result that there always exists such a "large" half-integral packing.

THEOREM 8.2 *Let D be a weighted digraph whose minimum weight dicut is of size k. Then there is a half-integral packing of dijoins of size $k/2$.*

We also discuss a possible approach for finding integral packings based on a notion of skew supermodularity. This is related to a number of other recent results on generalized Steiner problems in undirected graphs and on network design problems with orientation constraints. Skew submodular flows may themselves be an interesting direction for future research.

1.2 Algorithms

Frank's original $O(n^3m)$ combinatorial algorithm (Frank, 1981) for the minimum dijoin problem[2] is a primal-dual algorithm which essentially looks for an augmenting cycle of negative cost much like Klein's cycle cancelling algorithm for minimum cost network flows. It differs in two main respects. First, in some iterations, no progress is made in the primal, but rather some of the dual variables are altered. Second, the negative cycles are not computed in the residual digraph associated with the current dijoin. Rather, such cycles are computed in an extended graph which contains new arcs called *jumping arcs*; these arcs may not correspond to any arc in the original digraph. Avoiding jumping arcs altogether is a difficult task, but there are two reasons to attempt this. The first is that it is more natural to work in a residual digraph associated with the original digraph; for example, this is what is done for minimum cost network flows. The second is that computation of these arcs has proved to be the bottleneck operation in terms of running time.

[2]More complex algorithms for the problem were found also by Lucchesi (1976) and Karzanov (1979).

Frank's original algorithm computed these arcs in time $O(n^2m)$ per iteration. Gabow developed a sophisticated theory of centroids and used it to compute these arcs in $O(nm)$ time. In Section 3.2 we discuss a simple primal $O(n^2m)$ implementation of Frank's algorithm. There we give a simple $O(nm)$ primal algorithm for computing the jumping arcs. Putting this together, we exhibit an $O(n^2m)$ algorithm which is based only on negative cycle detection and arborescence growing routines.

The Lucchesi – Younger Theorem led Edmonds and Giles (1977) to develop *submodular flows*, a common generalization of network flows and dijoins. Frank's algorithm for dijoins proved to be the prototype for the original "combinatorial" algorithms for submodular flows, see for example Cunningham and Frank (1985). In particular, the notion of jumping arc carried over to that of an *exchange capacity* which is at the heart of every algorithm for submodular flows. Computation of exchange capacities corresponds to the problem of minimizing a submodular function (a problem for which combinatorial algorithms have only recently been devised Iwata et al., 2001; Schrijver, 2000).

Recently, it was shown by Fleischer and Iwata (2000) how to compute minimum cost submodular flows without an explicit call to a submodular flow minimization routine; instead they capitalize on a structural result of Schrijver (2000) on submodular flow extreme points. In a similar vein, we seek optimality conditions in terms of the original topology given by D. In this direction, we describe a cycle *flushing* operation (similar to augmentations), inspired by the above-mentioned lobe decomposition, which allows one to work in a modified auxiliary graph whose arcs are parallel to those of D (and so no expensive computation is required to build it). We show that any pair of minimal dijoins can be obtained from one another by a sequence of cycle flushings. Unlike network flows, however, we may not always be guaranteed to be able to improve the cost on each flushing operation. That is, we may not restrict to negative cost cycle augmentations. We show instead that a dijoin is optimal if and only if there is no negative cost strongly connected subgraph structure in the modified auxiliary graph. Specifically, for a dijoin T we define a residual graph $\mathcal{D}(T)$ which includes the arcs $T \cup (\overline{A - T})$ each with cost zero, the arcs $A - T$ each with their original cost, and the arcs $\overline{T}$ each with the negative of their original cost (for a set X of arcs, $\overline{X}$ denotes the set of arcs that are the reverse of arcs in X).

THEOREM 8.7 *A dijoin T is optimal if and only if $\mathcal{D}(T)$ contains no negative cost strongly connected subgraph, without any negative cycle of length two.*

Detecting such negative cost subgraphs is NP-hard in general, as shown in Section 3.1.2, although in this specific setting there is evidently a polytime algorithm to find such subgraphs. This result effectively determines a "test set" for local search. In other words, call two dijoins *neighbourly* whenever one can be obtained from the other by flushing along the cycles in such a strong strongly connected subgraph. Then the result implies that the set of neighbourly pairs includes all adjacent dijoins on the dijoin polyhedron. We have not characterized the adjacent pairs however (cf. Chvátal, 1975, where adjacent sets on the stable set polytope are characterized).

2. Visualizing and certifying dijoins

In this section we describe two "visual certificates" concerning whether a set of arcs forms a dijoin. They have a similar flavour but are of independent use depending on the setting. First we introduce the necessary notation and definitions. We consider a digraph $D = (V, A)$. For a nonempty, proper subset $S \subseteq V$, we denote by $\delta^+(S)$ the set of arcs with tail in S and head in $V - S$. We let $\delta^-(S) = \delta^+(V - S)$. We also set $\delta(S) = \delta^+(S) \cup \delta^-(S)$ and call $\delta(S)$ the cut *induced* by S. A cut is *directed* if $\delta^+(S)$, or $\delta^-(S)$, is empty. We then refer to the set of arcs as a *directed cut* and call S its *shore*. The shore of a directed cut induces an *in-cut* if $\delta^+(S) = \varnothing$, and an *out-cut* otherwise. Clearly if S induces an out-cut, then $V - S$ induces an in-cut. We may abuse terminology and refer to the cut $\delta(S)$ by its *shore* S.

A *dijoin* T is a collection of arcs that intersects every directed cut, i.e., at least one arc from each cut is present in the dijoin. An arc, $a \in T$, is said to be *critical* if, for some directed cut $\delta^+(S)$, it is the only arc in T belonging to the cut. In this case we say that $\delta^+(S)$ is a *justifying cut* for a. A dijoin is *minimal* if all its arcs are critical.

Observe that if $a \in A$ induces a cut edge in the *underlying undirected graph* then a set of arcs A' is a dijoin if and only if $a \in A'$ and $A' - \{a\}$ is a dijoin of $D - a$. Thus, we make the assumption throughout that the underlying undirected graph is 2-edge connected. Notice, also, that we may assume that the graph is acyclic since we may contract the nodes in a directed cycle without affecting the family of directed cuts.

A (simple) *path* P in an undirected graph is defined as an alternating sequence $v_0, e_0, v_1, e_1, \ldots, e_{l-1}, v_l$ of nodes and edges such that each e_i has endpoints v_i and v_{i+1}. We also require that none of the nodes are repeated, except possibly $v_0 = v_l$ in which case the path is called a *cycle*. Note that the path $v_l, e_{l-1}, v_{l-1}, \ldots, e_0, v_0$ is distinct from P, and is called the *reverse* of P, denoted by P^-. A *path* in a digraph D is

defined similarly as a sequence $P = v_0, a_0, v_1, a_1, \ldots, a_{l-1}, v_l$ of nodes and arcs where for each i, the head and tail of a_i are $\{v_i, v_{i+1}\}$. If the head of a_i is v_{i+1}, then it is called a *forward* arc of P. Otherwise, it is called a *backward* arc. Such a path is called a cycle if $v_0 = v_l$. The path (cycle) is *directed* if every arc is forward. Finally, for an arc a with tail s and head t, we denote by $\bar{a}$ a new arc associated to a with tail t and head s. For a subset $F \subseteq A$, we let $\overline{F} = \{\bar{a} : a \in F\}$. We also let $\overline{D}$ denote the digraph $(V, \bar{A})$.

2.1 A decomposition via cycles

A cycle (or path), C, is *complete* with respect to a dijoin, T, if all of its forward arcs lie in T. In McWhirter and Younger (1971) the notion of a complete path (called *minus paths*) is already used. A complete cycle (or path) is *flush* if, in addition, none of its backward arcs is in the dijoin.

LEMMA 8.1 *Let T be a dijoin of D and C be a complete cycle in D. Then either C is flush or T is not minimal.*

Proof. Suppose that $a \in T$ is a reverse arc of C and $\delta^+(S)$ is a justifying cut for a. Since $\delta^+(S)$ is a directed cut, C intersects it in an equal number of forward and reverse arcs. In particular, $\delta^+(S) \cap T$ contains some forward arc of C. It follows that $T - \{a\}$ is a dijoin. □

THEOREM 8.3 *Every dijoin contains a complete cycle.*

Proof. We actually show that every non-dijoin arc is contained in a complete cycle. Since the underlying graph is 2-edge connected there is a non-dijoin arc (otherwise any cycle in D is complete). Let $a = (s, t)$ be such an arc. We grow a tree $\mathcal{T}$ rooted at s. Initially, let $V(\mathcal{T}) = s$ and $A(\mathcal{T}) = \varnothing$. At each step we consider a node $v \in \mathcal{T}$ which we have yet to examine and consider the arcs in $\delta(v) \cap \delta(\mathcal{T})$. We add such an arc a' to $\mathcal{T}$ if $a' \in \delta^-(\mathcal{T})$ or if $a' \in \delta^+(\mathcal{T}) \cap T$. It follows that this process terminates either when $\mathcal{T} = V$ or when $\delta(\mathcal{T})$ consists only of out-going, non-dijoin arcs. This later case can not occur, otherwise $\delta(\mathcal{T})$ is a directed cut that does not intersect the dijoin T. So we have $t \in T$. In addition $a = (s, t)$ is not in $\mathcal{T}$ as a was an out-going, non-dijoin arc when it was examined in the initial stage. Now take the path $P \in \mathcal{T}$ connecting s and t. All of its forward arcs are dijoin arcs and by adding a we obtain a complete cycle, C, in which a is a backward arc. □

The preceding tree algorithm also gives the following useful check of whether or not a dijoin arc is critical.

LEMMA 8.2 *A dijoin arc $a = (s,t)$ is critical if and only if there is no flush path from s to t path in $D - \{a\}$. Moreover, any such arc lies on a flush cycle.*

Proof. This is equivalent to the following. A dijoin arc $a = (s,t)$ is critical if and only if the tree algorithm fails to find an $s - t$ path when applied to $D - \{a\}$. To prove this, remove a from D and grow $\mathcal{T}$ from s as before. If $t \in \mathcal{T}$ upon termination of the algorithm then let P be the path from s to t in $\mathcal{T}$. All the forward arcs in P are in the dijoin. So $C = P \cup \{a\}$ is a complete cycle. However, a is a backward arc with respect to C and it too is in the dijoin. Thus, by Lemma 8.1, a is not critical. Suppose then, on termination, $t \notin \mathcal{T}$ (notice that since we have removed a dijoin arc it is possible that the algorithm terminates with $\mathcal{T} \neq V$). Clearly $\delta(\mathcal{T})$ is a justifying cut for a, and so a is critical. □

COROLLARY 8.1 *A dijoin T is minimal if and only if every complete cycle in T is flush.*

Consider a complete cycle C with respect to T. We denote by $D \star C$ (there is an implied dependency on T) the digraph obtained by contracting C to a single node, and then contracting the strong component containing that node to make a new node v_C. The subgraph of D corresponding to v_C is denoted by $H(C)$. Hence, $D \star C = D/H(C)$, where / denotes the contraction operation. Observe that any cut of D that splits the nodes of $H(C)$ either contains a dijoin arc $a \in T \cap C$ or is not a directed cut. This observation lies behind our consideration of the digraph $D \star C$.

LEMMA 8.3 *Let D be an acyclic digraph and T a dijoin. If C is a complete cycle, then $T - H(C)$ is a dijoin in $D \star C$. If T is minimal in D then $T - A(C)$ is also minimal in $D \star C$.*

Proof. First, note that any directed cut in $D \star C$ corresponds to a directed cut, $\delta^+(S)$, in D (with $V(H(C)) \subseteq S$ or $V(H(C)) \subseteq V - S$). It follows that $T - V(C)$ is a dijoin in $D \star C$. Now suppose that T is minimal. We first show that every arc $a \in H(C) - A(C)$ is contained in a complete cycle C_a whose forward arcs are a subset of C's forward arcs. For if a is such an arc, then since $H(C)/V(C)$ is strongly connected, there is a directed path P which contains a, is internally disjoint from $V(C)$, and whose endpoints lie in $V(C)$; the endpoints, say x and y, are distinct as P is itself not a directed cycle in D. We may then identify a complete cycle by traversing a subpath of C from x to y, and then traversing P in the reverse direction. Lemma 8.1 now implies that no arc of P lies in T. In particular, this shows that $T \cap H(C) - A(C) = \varnothing$.

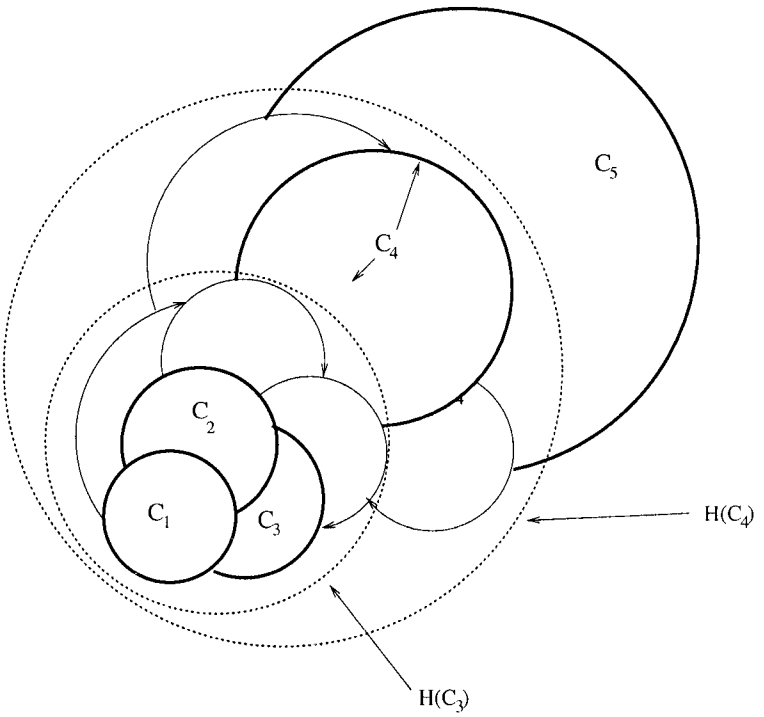


Figure 8.2. A lobe decomposition

Now consider $a \in T - A(C)$ and let S_a be a justifying cut in D for a. If $V(H(C)) \subseteq S_a$ or $V(H(C)) \cap S_a = \varnothing$, then it is a justifying cut for a (with respect to $T - A(C)$) in $D \star C$. Otherwise, $\delta^+(S_a)$ must contain an arc $a' \in H(C) - A(C)$. But then $C_{a'}$ must intersect $\delta^+(S_a)$ in some arc of $T \cap C$, contradicting the fact that $\delta^+(S_a) \cap T = \{a\}$. □

Set $D_0 = D$ and let C_0 be a complete cycle in D_0. A *lobe decomposition* (which supports T) is a sequence $\mathcal{S} = \{C_0, \mathcal{P}^0, C_1, \mathcal{P}^1, \ldots, C_k, \mathcal{P}^k\}$ such that

- For each $i > 0$, C_i is a complete cycle for T in $D_i = D_{i-1} \star C_{i-1}$ containing the special node $v_{C_{i-1}}$.
- For each $i \geq 0$, $\mathcal{P}^i = \{P_0^i, P_1^i, \ldots, P_{n_i}^i\}$ is a directed ear decomposition of $H_i/(H_{i-1} \cup C_i)$. Here $H_i = H(C_i)$, that is, $V(H_i) = \bigcup_{j=0}^{i}(\cup_t P_t^j \cup C_j)$.
- D_{k+1} consists of a single node ($H_{k+1} = V$).

Alternatively, we could replace the first condition, by one which does not look for a complete cycle in $D_{i-1} \star C_{i-1}$ but rather for a complete path with distinct endpoints $x, y \in H_{i-1}$ and whose internal nodes lies in

$V - H_{i-1}$. An example of a lobe decomposition is shown in Figure 8.2. We refer to the C_i's as the *lobes* of the decomposition. The *ears* for some particular lobe are the directed paths P_j^i in any directed ear decomposition for the subgraph $H(C_i)$. Each P_i or C_j is called a *segment* of the decomposition. We say that a lobe decomposition $\mathcal{S}$ is *flush* if C_i is a flush for each i. We remark that, whilst each C_i is only a complete cycle in D_i and not necessarily D, it can easily be extended to give a complete cycle C_i' in $H_i \subseteq D$ using arcs of H_{i-1} (possibly using arcs in some $C_j, j < i$). Thus, we may also think of T as being generated by a sequence of $C_0', C_1', \ldots, C_k'$ of flush cycles in D.

This leads to the following decomposition theorem.

THEOREM 8.4 (LOBE DECOMPOSITION) *A set of arcs T is a dijoin if and only if it has a lobe decomposition. A dijoin T is minimal if and only if every such decomposition is flush with respect to T.*

Proof. Suppose that $T = T_0$ is a dijoin in $D = D_0$. Then we may find a complete cycle C_0 in D_0 by Lemma 8.3. Now set $D_1 = D_0 \star C_0$. By Lemma 8.3, $T_1 = T - A(C_0)$ is again a dijoin in D_1. Thus we may find a complete cycle C_1 in D_1. We may repeat this process on $D_{i+1} = D_i \star C_i$ until some D_k consists of a single node. Hence the C_i give rise to a lobe decomposition. Conversely, first suppose that T has a lobe decomposition $\mathcal{S} = \{C_0, \mathcal{P}^0, C_1, \mathcal{P}^1, \ldots, C_k, \mathcal{P}^k\}$ and let $\delta(S)$ be a directed cut. Since D_k consists of a single node, the nodes in S and $V - S$ must have been contracted together at some point. Suppose this occurs for the first time in stage j. Thus there is some node $x \in S$ and $y \in V - S$ which are 'merged' at this point by contracting the arcs within C_j or one of its ears. Since $\delta(S)$ is a directed cut and since D_j either lies entirely in S or entirely in $V - S$, none of C_j's ears may intersect the cut $\delta^+(S)$. Hence x and y were merged via the complete cycle C_j. Hence T intersects $\delta(S)$ and is, therefore, a dijoin.

Now suppose that T is minimal but there is a lobe decomposition for which some C_i is not flush. Let i be the smallest index of such a cycle. Lemma 8.3 implies that T_i is minimal in D_i. However, C_i is complete but not flush, contradicting Corollary 8.1 applied to T_i. Conversely, suppose that every lobe decomposition is flush. If T is not minimal, then we may find a non-flush complete cycle C by Corollary 8.1. But then we may start with this cycle $C_0 = C$, and proceed to obtain a non-flush lobe decomposition, a contradiction. □

This theorem immediately implies:

THEOREM 8.5 *Let D be a digraph with arc weights w whose minimum directed cut is of weight at least 2. If each nontrivial component of the*

graph induced by the support of w induces a 2-edge-connected undirected graph, then D contains two disjoint dijoins made up of positive weight arcs.

Proof. Let $H_1, H_2, H_3, \ldots, H_t$ be the nontrivial connected components of the subgraph induced by positive weight arcs. For each component, create a lobe decomposition and let F_i be those arcs that are traversed in the forward direction of this decomposition. Now set $F := \bigcup_i F_i$, and $F' := \bigcup_i (A(H_i) - F_i)$. We claim that F, F' are the desired dijoins. This is because if $\delta^+(S)$ is a directed cut, then since it has positive w-weight, for some i, $S \cap V(H_i)$ is a proper, nonempty subset of $V(H_i)$. One easily sees that F_i, and $A(H_i) - F_i$ intersect this cut, hence the result. □

This clearly extends to the cardinality case, for which two disjoint dijoins was first observed by Frank (cf. Schrijver, 2003).

COROLLARY 8.2 *If D is a connected digraph with no directed cut of size 1, then it contains two disjoint dijoins.*

2.2 A decomposition via trees

We now discuss a decomposition based on building a connected subgraph on the underlying undirected graph. We begin with several more definitions. Given our acyclic digraph $D = (V, A)$, we denote by V^+ and V^- the set of sources and sinks, respectively. An ordered pair of nodes (u, v) is *legal* if u is a source, v is a sink, and there is a directed path from u to v in D. A (not necessarily directed) path is called a *source-sink* path if its start node is a source node u of D and its terminating node v is a sink of D. A source-sink path is *legal* if the pair (u, v) is legal.

A *cedar* is a connected (not necessarily spanning) subgraph $\mathcal{K}$ that contains every source and sink, and can be written in the form

$$A(\mathcal{K}) = \bigcup_{P \in \mathcal{P}} A(P)$$

where $\mathcal{P}$ is a collection of legal source-sink paths. We call $\mathcal{P}$ the *decomposition* of the cedar. Given a cedar $\mathcal{K}$, we denote by $F(\mathcal{K})$ the set arcs that are oriented in a forward direction along some path in the source-sink decomposition.

We start with several lemmas.

LEMMA 8.4 *If $\mathcal{K}$ is a cedar, then $F(\mathcal{K})$ is a dijoin.*

Proof. Suppose that $\delta^+(S)$ is a directed cut. Note that since D is acyclic, there must be some source in S and some sink in $V - S$. Since $\mathcal{K}$ is a

cedar, it is connected and hence there is a path $P \in \mathcal{P}$ joining a node of S to some node of $V - S$. Let u and v be the source and sink of this path. Since P is legal, it can not be the case that $u \in V - S$ and $v \in S$. It is easy to see that, in each of the remaining three cases, P must have traversed an arc of $\delta^+(S)$ in the forward direction. Thus $F(\mathcal{K})$ does indeed cover the cut. □

Thus, the complete paths induced by a cedar form a dijoin. Now, for each source v, we denote by T_v the maximal out-arborescence rooted at v in D. Let $\mathcal{R}$ be the subgraph obtained as the union of the T_v's. The following lemma follows trivially from the acyclicity of D.

LEMMA 8.5 *There is a directed path from any node to some sink in D. There is also a dipath to any node from some source. In particular, each node lies in some T_v.*

LEMMA 8.6 *The digraph $\mathcal{R}$ is connected.*

Proof. Suppose that $S \neq V$ is a connected component of $\mathcal{R}$. By the connectivity of D, we may assume that there is an arc (x, y) such that $x \in S$ and $y \notin S$. By Lemma 8.5, there exists a source $v \in \mathcal{R}$ such that $x \in T_v$. The existence of the arc (x, y) then contradicts the maximality of T_v. □

Associated with a digraph D is a *derived digraph*, denoted by D'. The node set of D' is $V^+ \cup V^-$ and there is an arc (u, v) for each legal pair (u, v) such that $u \in V^+$ and $v \in V^-$.

LEMMA 8.7 *The derived graph D' is connected.*

Proof. Let S be a nonempty proper subset of $V^+ \cup V^-$. It is enough to show that $\delta_{D'}(S)$ is nonempty. By Lemma 8.5, we may assume that both S and $V - S$ contain a source nodes. Let $v_1, \ldots, v_p$ be those sources in S, and let $w_1, \ldots, w_q$ be those sources in $V - S$. Evidently, no T_{v_i} contains a sink in $V - S$ and no T_{w_j} contains a sink in S. On the other hand, by Lemma 8.6, there exists some i and j for which T_{v_i} and T_{w_j} share a common node, x say. Hence, by Lemma 8.5, there is a dipath from x to some sink node y. Therefore, there is a dipath to y from both v_i and w_j. It follows that $\delta_{D'}(S) \neq \varnothing$ as required. □

LEMMA 8.8 *If T is a minimal dijoin, then there exists a cedar $\mathcal{K}$ such that $T = F(\mathcal{K})$.*

Proof. Given a minimal dijoin T, we construct the desired cedar $\mathcal{K}$. For any legal pair (u, v) in distinct components of $\mathcal{K}$, there is a complete

path, P_{uv}, from u to v. Suppose that such a complete path does not exist. Then there is a subset S such that $u \in S$ and $v \notin S$ with $\delta^-(S) = \varnothing$ and that $\delta^+(S) \cap T = \varnothing$. That is, S defines a directed cut which is not covered by T, a contradiction. Now we add the arcs of P_{uv} to the cedar. The desired cedar is then obtained from the final subgraph $\mathcal{K}$ by throwing out any singleton nodes. The resulting digraph is necessarily connected by Lemma 8.7 and, hence, is a cedar. Moreover, by Lemma 8.4, $F(\mathcal{K})$ is a dijoin. Since $F(\mathcal{K}) \subseteq T$, we have that $F(\mathcal{K}) = T$, by the minimality of T, □

One simple consequence is the following.

COROLLARY 8.3 (CEDAR DECOMPOSITION) *A set of arcs T is a dijoin if and only if there is a cedar $\mathcal{K}$ with $T \subseteq F(\mathcal{K})$. Moreover, T is minimal if and only if $T = F(\mathcal{K})$ for every cedar $\mathcal{K}$ with $T \subseteq F(\mathcal{K})$.*

3. Finding minimum cost dijoins

In this section, we consider the problem of finding minimum cost dijoins. We begin by presenting a "flushing" operation that can be used to transform one dijoin into another. We then use this operation to characterize when a dijoin is optimal. Finally, we give a simple efficient primal implementation of Frank's algorithm.

3.1 Augmentation and optimality in the original topology

Our approach, in searching for a minimum cost dijoin, is to transform a non-optimal dijoin into an instance of lower cost by augmenting along certain cycle structures. The augmentation operation is motivated by the lobe decomposition in Section 2.1. In due course, we will develop a primal algorithm for the dijoin problem along the lines of that given by Klein (1967) for minimum cost flows.

Given a dijoin T and a cycle C, let $T' = T \otimes C$ be the resultant graph where C is made into a flush cycle. We call this operation *flushing* the cycle C. We may make the resultant flush cycle have either clockwise or anti-clockwise orientation by adjusting whether we flush on C or $\overline{C}$. These two possibilities are shown in Figure 8.3. Similar operations have been applied in various ways previously to paths instead of cycles (see, for example, Frank, 1981; Fujishige, 1978; Lucchesi and Younger, 1978; McWhirter and Younger, 1971; Zimmermann, 1982). One key difference is that we introduce the reverse of arcs from outside the current dijoin.

We now formalize this operation and introduce a cost structure. Given T, we construct an auxiliary digraph, $\mathcal{D}(T)$, as follows. Add each arc

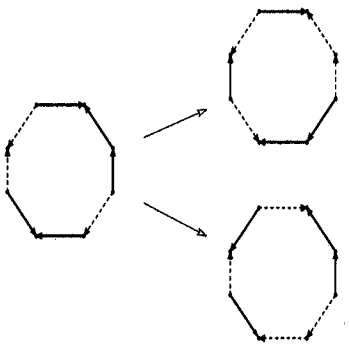


Figure 8.3. Flushing a cycle

$a \in T$ to $\mathcal{D}(T)$ and give it cost 0; also add a reverse copy $\bar{a}$ with cost $-c_a$ (corresponding to the action of removing a from T). For each $a \notin T$, add arc a to $\mathcal{D}(T)$ with cost c_a (corresponding to the action of adding a to T), and include a reverse copy with cost 0. We call the arcs $T \cup \overline{(A-T)}$ the *zero arcs* (or benign arcs) of the auxiliary digraph. Now for a directed cycle C in $\mathcal{D}(T)$, we define $T \otimes C$ as $T \cup \{a \in C : a \notin T\} - \{a \in T : \bar{a} \in C\}$); we also define $T \otimes A'$ in the obvious fashion for an arbitrary set of arcs A'. The value of this operation is illustrated by the following simple lemma.

LEMMA 8.9 *Let T be a dijoin in D, and let C be a cycle of length at least 3 in $\mathcal{D}(T)$. Then $T \otimes C$ is also a dijoin.*

Proof. If S induces an out-cut, then if C crosses this cut, then it crosses it at least once in the forward direction, and hence an arc of $T \otimes C$ intersects this cut. If C did not cross this cut, then $T \cap \delta^+(S) = (T \otimes C) \cap \delta^+(S)$. Thus $T \otimes C$ is indeed a dijoin. □

We now see that one may move from one dijoin to another by repeatedly flushing along cycles. To this end, consider a directed graph $\mathcal{G}$ whose nodes correspond to the dijoins of D and for which there is an arc (T, T') whenever there is a cycle C, of length at least 3 in $\mathcal{D}(T)$, such that $T' = T \otimes C$. We have the following result.

THEOREM 8.6 *Given a dijoin T and a minimal dijoin T^* there is a directed path in $\mathcal{G}$ from T to T^*. In particular, there is a directed path in $\mathcal{G}$ from a minimal dijoin to any other minimal dijoin.*

Proof. Let T^* have a flush lobe decomposition that is generated by the flush cycles $C_0, C_1, \ldots, C_k$. Take $T_0 = T$ and let $T_{i+1} = T_i \otimes C_i$. Note

that $T^* \subseteq T_{k+1}$. If $T^* = T_{k+1}$ then we are done. Otherwise take an arc $a \in T_{k+1} - T^*$. Since T^* is a dijoin, the arc a is not critical. So there is a complete cycle C_a for which a is a backward arc. Applying $T_{k+2} = T_{k+1} \otimes C_a$ removes the arc a. We may repeat this process for each arc in $T_{k+1} - T^*$. The result follows. □

Observe that there is a cost associated with flushing along the cycle C. This cost is precisely $\sum_{a \in C: a \notin T} c_a - \sum_{a \in T: \bar{a} \in C} c_a$. We call a directed cycle C, of length at least 3 in $\mathcal{D}(T)$, *augmenting* (with respect to c) if it has negative cost. (We note that the general problem of detecting such a negative cycle is NP-hard as is shown in Section 3.1.2.) Clearly, if C is augmenting, then $c(T \otimes C) < c(T)$. Hence if $\mathcal{D}(T)$ contains an augmenting cycle, then T can not be optimal.

3.1.1 Auxiliary networks and optimality certificates.

Ideally one could follow the same lines as Klein's negative cycle cancelling algorithm for minimum cost flows. He uses the well-known residual (auxiliary) digraph where for each arc a, we include a reverse arc if it has positive flow, and a forward arc if its capacity is not yet saturated. A current network flow is then optimal if and only if there is no negative cost directed cycle in this digraph. This auxiliary digraph is not well enough endowed, however, to provide such optimality certificates for the dijoin problem. Instead Frank introduces his notion of jumping arcs which are added to the auxiliary digraph. In this augmented digraph the absence of negative cost directed cycles does indeed characterize optimality of a dijoin.

We attempt now to characterize optimality of a dijoin working only with the auxiliary digraph $\mathcal{D}(T)$ since it does not contain any jumping arcs. We describe an optimality certificate in this auxiliary digraph. Conceptually this avoids having to compute or visualize jumping arcs; note that $\mathcal{D}(T)$ is trivial to compute since all its arcs are parallel to those of D. This comes at a cost, however, in that the new certificate gives rise to a computational task which seems not as simple as detecting a negative cycle (at least not without adding the jumping arcs!). This is forewarned by several complexities possessed by the dijoin problem which are absent for network flows. For instance, if a network flow f is obtained from a flow f' by augmenting on some cycle C, then we may obtain f' back again, by augmenting the (topologically) same cycle. The same is not true for dijoins. Figure 8.4 shows gives two examples in which two dijoins can each be obtained from the other in a single cycle flushing. Any such cycles, however, are necessarily topologically distinct.

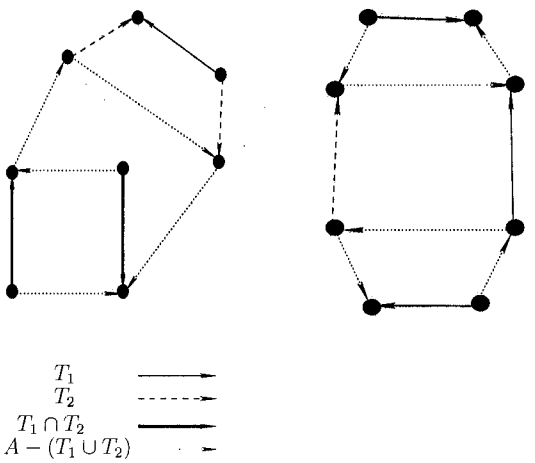


Figure 8.4. The asymmetry of flushing

One might hope that any non-optimal dijoin T, could be witnessed by the existence in $\mathcal{D}(T)$ of an augmenting cycle. Unfortunately, Figure 8.5 shows that this is not the case. The dijoin T_1 is non-optimal; there is, however, no augmenting cycle in $\mathcal{D}(T_1)$. We remark that **we do not know whether the non-existence of an augmenting cycle guarantees any approximation from an optimal dijoin.** (Neither do we know, whether this approach leads to a reasonable heuristic algorithm in practice.)

An alternative optimality condition is suggested by the fact that, whereas any network flow can be decomposed into directed flow cycles, dijoins admit a lobe decomposition. Thus we focus instead on strongly connected subgraphs of the auxiliary graph. We say that a subgraph is *clean* if it contains no digons (directed cycles of length two). Now take a dijoin T and consider a clean strongly connected subgraph H of $\mathcal{D}(T)$. Since H can be written as the union of directed cycles, it follows by Lemma 8.9 that the flushing operation produces another dijoin $T' = T \otimes H$. The resulting dijoin has cost $c(T') = c(T) + \sum_{a \in H} c_a$. Consequently, if H has negative cost then we obtain a better dijoin. The absence of a negative cost clean strongly connected subgraph will, in fact, certify optimality. In order to prove this, we need one more definition. Given a clean subgraph H and a directed cycle C, we define $H \diamond C$ as follows: firstly, take $H \cup C$ and remove any multiple copies of any arc; secondly, if a and $\bar{a}$ are in $H \cup C$ then keep only the arc from C. Our certificate of optimality now follows.

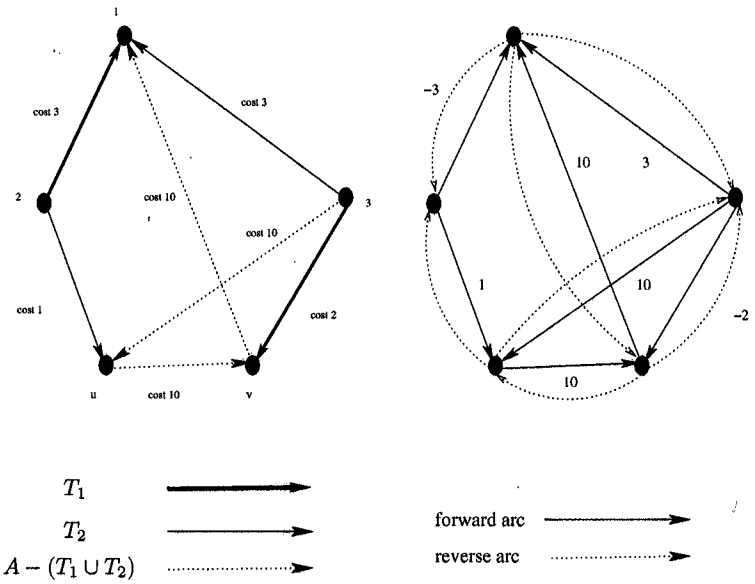


Figure 8.5. A non-optimal dijoin with no augmenting cycle

THEOREM 8.7 *A dijoin T is of minimum cost if and only if $\mathcal{D}(T)$ induces no negative cost clean strongly connected subgraph.*

Proof. As we have seen, if T is of minimum cost, then clearly $\mathcal{D}(T)$ contains no negative cost clean strongly connected subgraph. So suppose that T is not a minimum cost dijoin and let T^* be an optimal dijoin. Then, by Theorem 8.6, there are cycles $C_1, \ldots, C_t$ such that $T^* = T \otimes C_1 \otimes C_2 \cdots \otimes C_t$. Let $H^1 = C_1$ and $H^{r+1} = H^r \diamond C_{r+1}$. We now show that H^t is a negative cost, strongly connected, clean subgraph. The theorem will then follow as $T^* = T \otimes H^t$.

(i) By the use of the operation $\diamond$, no digons may be present in H^i and, hence, H^i is clean.

(ii) Since $T^* = T \otimes H^t$, we have that $c(H^t) = c(T^*) - c(T) < 0$. Therefore the cost of H^t is negative.

(iii) We prove that H^i is a strongly connected subgraph by induction. $H^1 = C_1$ is strongly connected since C_1 is a directed cycle. Assume that H^{i-1} is strongly connected. Then, clearly, $H^{i-1} \cup C_i$ is also strongly connected. Now H^i is just $H^{i-1} \cup C_i$ minus, possibly, the complements of some of the arcs in C_i. Suppose, $a = (u, v) \in H^{i-1}$ is the complement of an arc in in C_i. Now $a \notin H^i$ but since all the arcs in C^i are in H^i there is still a directed path from u to v in H^i. Thus, H^i is a strongly connected subgraph. □

COROLLARY 8.4 *A dijoin T is of minimum cost if and only if $\mathcal{D}(T)$ induces no negative cost strongly connected subgraph that contains no negative cost digon.*

Proof. If T is not optimal then by Theorem 8.7 there is a negative cost clean strongly connected subgraph H in $\mathcal{D}(T)$. Clearly, H is a negative cost strongly connected subgraph containing no negative cost digon. Conversely, take a negative cost strongly connected subgraph H that contains no negative cost digon. Suppose that H contains a digon $C = (a, \bar{a})$. This digon has non-negative cost and, therefore a is a non-dijoin arc. If we then flush along H (insisting here that for a digon, the non-digon arc a is flushed after $\bar{a}$) we obtain a dijoin of lower cost. □

These results can be modified slightly so as to insist upon spanning subgraphs. For example we obtain:

THEOREM 8.8 *A dijoin T is of minimum cost if and only if $\mathcal{D}(T)$ contains no spanning negative cost clean strongly connected subgraph.*

Theorem 8.8 follows directly from Theorem 8.7 and the following lemma.

LEMMA 8.10 *If T is a minimal dijoin, then the zero arcs induce a strongly connected subgraph.*

Proof. Recall that the zero arcs are those arcs in $T \cup \overline{(A - T)}$. Now, consider any pair of nodes u and v. If there is no dipath consisting only of zero arcs, then there is a subset $S \subseteq V$ containing u but not v such that $\delta^+(S) \subseteq A - T$ and $\delta^-(S) \subseteq T$. Moreover, $\delta^-(S) \neq \varnothing$ and so contains some arc a. One now sees that a cannot be contained on a flush cycle, contradicting Lemma 8.2. □

3.1.2 A hardness result. We have now developed our optimality conditions. In this section, however, we present some bad news. The general problem of finding negative cost clean subgraphs is hard. We consider a digraph D and cost function $c\colon A \to \mathbb{Q}$. Consider the question of determining whether D contains a negative cost directed cycle whose length is at least three. The corresponding dual problem is to find shortest length clean paths. As we now show, this problem is difficult.

We note that the question of whether there exists any cycle of length greater than 2 is answered in polytime as follows. A *bi-tree* is a directed graph obtained from a tree by replacing each edge by a directed digon. A *long acyclic order* for digraph D is an ordered partition $V_1, V_2, \ldots, V_q$

such that each V_i induces a bi-tree and any arc (u, v) with $u \in V_i$ and $v \in V_j$ with $i \neq j$, satisfies $i < j$. One may prove the following fact

PROPOSITION 8.1 *A digraph D has no directed cycle of length at least 3 if and only if it has a long acyclic order.*

We now return to examine the complexity of finding a negative cost such circuit.

THEOREM 8.9 *Given a digraph D with a cost function c. The task of finding shortest path distances is NP-hard, even in the absence of negative cost clean cycles.*

Proof. So our directed graph D may contain negative cost digons but no negative cost clean cycles. By a reduction from 3-SAT we show that the problem of finding shortest path distances is NP-hard. Given an instance, $C_1 \wedge C_2 \wedge \cdots \wedge C_n$, of 3-SAT we construct a directed graph D as follows. Associated with each clause C_j is a *clause gadget.* Suppose $C_j = (x \vee y \vee z)$ then the clause gadget consists of 8 disjoint directed paths from a node s_j to a node t_j. This is shown in Figure 8.6.

Each of the paths contains 5 arcs. For each path, the three interior path arcs represent one of the 8 possible true-false assignments for the variables x, y and z. The two end arcs associated with each path have a cost 0, except for the path corresponding to the variable assignment $(\bar{x}, \bar{y}, \bar{z})$. Here the final arc has a cost 1. Assume for a moment that the three interior path arcs also have cost 0. Then the 7 variable assignments that satisfy the clause C_j correspond to paths of cost 0, whilst the non-satisfying assignment corresponds to a path of cost 1.

Now, for $1 \leq j \leq n-1$, we identify the node t_j of clause C_j with the node s_{j+1} of clause C_{j+1}. Our goal then is to find a shortest path from $s = s_1$ to $t = t_n$. Such a path will correspond to an assignment of the variables that satisfies the maximum possible number of clauses. We do, though, need to ensure that the variables are assigned consistently throughout the path. This we achieve as follows. Each arc representing a variable will in fact be a structure, called a *variable gadget.*

Note that a variable appears an equal number of times in its unnegated form x and its negated form $\bar{x}$. This is due to the fact that we have four occurrences of x and four of $\bar{x}$ for each clause containing x or $\bar{x}$. Thus if x and $\bar{x}$ appear in a total of n_x clauses we have $4n_x$ occurrences of x and of $\bar{x}$. The variable gadget representing x or $\bar{x}$ will be a directed path consisting of $8n_x$ arcs, with the arcs alternating in cost between L and $-L$. We link together the $4n_x$ directed paths corresponding to the x assignments with the $4n_x$ directed paths corresponding to the $\bar{x}$ assignments as follows. Each of the $4n_x$ negative cost arcs in an x structure

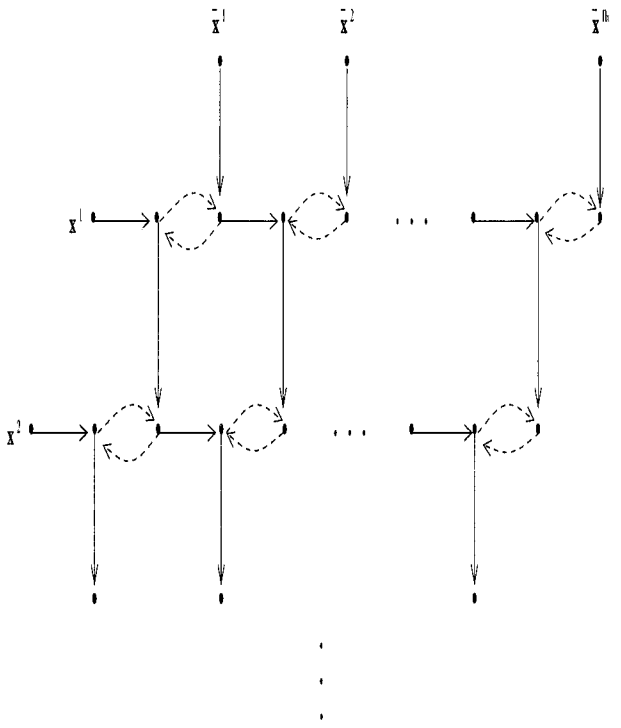


Figure 8.6. The clause gadget for a $C_j = (x \vee y \vee z)$

forms a digon with a negative cost arc from one of the $\bar{x}$ structures, and vice versa. This is shown in Figurew 8.7. Here the positive cost arcs are solid whilst the negative cost arcs are dashed. In addition, the $4n_x$ gadgets corresponding to x are labelled $x^1, x^2, \ldots, x^{n_x}$ etc. Hence each pair of variable gadgets x^i and $\bar{x}^j$ meet at a unique negative cost digon.

This completes the description of the directed graph corresponding to our 3-SAT instance. Note that any consistent assignment of the variables corresponds to a unique clean $s - t$ path. This path has the property that it either traverses every arc in a variable gadget or none of the arcs. Note that if a gadget corresponding to x is completely traversed, then none of the gadgets corresponding to $\bar{x}$ may be completely traversed, since our shortest paths must be clean.

We say that a variable gadget is *semi-traversed* if some but not all of the arcs in the gadget are traversed. We do not require any gadget to explicitly enforce consistency within the variables. Instead we show that in an optimal path, none of the variable gadgets may be semi-traversed.

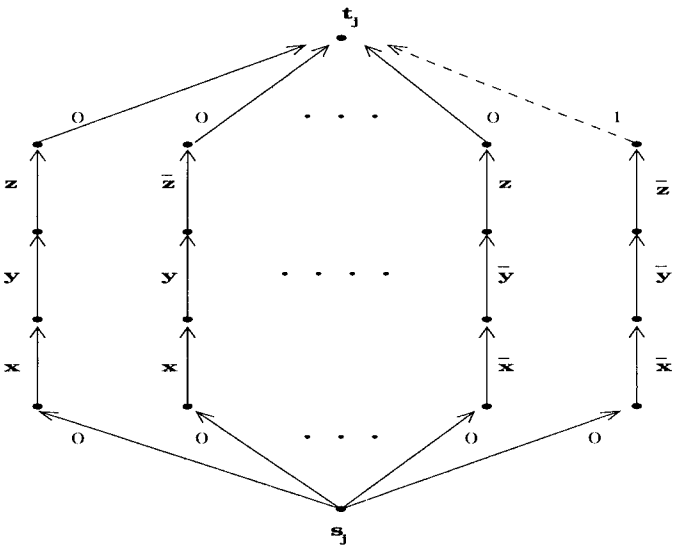


Figure 8.7. Interleaving variable gadgets.

By our previous observation, this implies that the path corresponds to a consistent variable assignment. To see that none of the variable gadgets are semi-traversed note that, in any path, the arcs immediately before and after an arc of cost $-L$ must have cost L. This follows as we may not use both arcs in a negative cost digon. However, if we have a semi-traversed gadget, then locally, with respect to that gadget, the path must contain two consecutive arcs of cost L. As a result the overall cost of the path is at least L. For $L > n$, though, clearly there are paths corresponding to consistent variable assignments with smaller cost. The result follows. □

Fortunately, as we will now see, the residual graph (associated with a dijoin) that we consider has enough structure to allow us to find negative cost clean strongly connected subgraphs efficiently.

3.2 A primal algorithm

We now present a simple primal implementation of Frank's primal-dual algorithm for the minimum cost dijoin problem. We also show how this algorithm fits into the framework we have just formulated. As discussed in the introduction, Frank's algorithm looks for augmentations of a dijoin in a residual digraph that contains a collection of *jumping arcs* that are not necessarily parallel to any arc of D. We begin by defining this residual digraph. To do this we require the concept of a *strict set.*

For a dijoin T, a directed cut $\delta(S)$ is strict if the number of dijoin arcs in $\delta(S)$ equals the number of weakly connected components in $D - S$. The *minimal strict set* containing a node v is defined as

$$P(v) = \{x : \nexists \text{ a directed cut } \delta^+(S) \text{ s.t. } x \in S, v \notin S \text{ and } |\delta^+(S) \cap T| = 1\}.$$

From these minimal strict sets, we obtain a collection of jumping arcs $\mathcal{J} = \{(u, v) : u \in P(v)\}$. The residual digraph is then $\mathcal{F}(T) = (V, A^F)$, where $A^F = A_r \cup A_f \cup \mathcal{J}$ and $A_r = \{\bar{a} : a \in T\}$ and $A_f = \{a : a \notin T\}$. Frank actually generates dual variables and uses them to select a subgraph of (V, A^F) to work on.[3] We add the following cost structure to (V, A^F). Each arc a of A_r receives a cost of $-c_{\bar{a}}$, each arc $a \in A_f$ receives a cost c_a, and all jumping arcs have cost 0. Given these costs, the following result is then implied by the analysis of Frank's algorithm.[4]

THEOREM 8.10 (OPTIMALITY THEOREM, FRANK) *A dijoin T is optimal if and only if $\mathcal{F}(T)$ has no negative cycle.*

Frank does not actually look for negative cycles but rather finds node potentials in the aforementioned subgraph of $\mathcal{F}(T)$. He then either improves the primal dijoin, or updates his dual variables. He shows that this need be done at most $O(n)$ times. The running time of each iteration is dominated by the time to build the jumping arcs, which he does in $O(n^2m)$ time. Gabow showed how to compute these arcs in $O(nm)$ time and, thus, improved the total running time of Frank's algorithm from $O(n^3m)$ to $O(n^2m)$. Gabow's approach is based on so-called centroids of a poset and is applicable to a broader class of problems. His method is very general and also rather complex to describe and implement. In Section 3.2.1 we give an elementary $O(nm)$ algorithm for building the jumping arcs for a given dijoin.

We now describe a simple primal implementation of Frank's algorithm that builds directly on the Optimality Theorem. We call this the PENDING-ARC algorithm. The algorithm assumes the availability of a subroutine (such as is given in the next section) for computing the jumping arcs of a dijoin.

The pending-arc algorithm. In each iteration i, we have a current dijoin T_i and subgraph G_i of $\mathcal{F}(T_i)$ that contains no negative cost cycles.

[3]Frank works in the digraph where all arcs are reversed.

[4]We remark that phrasing optimality in terms of negative cycles was first done by Fujishige (1978) for independent-flows and Zimmermann (1982) for submodular flows.

The (negative cost) arcs in $\mathcal{F}(T_i) - G_i$ are called the PENDING ARCS at time i; we denote by P_i this set of arcs. Initially we find any (minimal) dijoin T_0, and let G_0 consist of the jumping and forward arcs of $\mathcal{F}(T_0)$.

Iteration $i+1$ consists of adding some arc a of P_i to G_i. We then look for a negative cycle containing a. If none exists, then $G_{i+1} = G_i + a$ has no negative cost cycles and $P_{i+1} = P_i - a$ is a smaller set of pending arcs. Otherwise, we find a negative cost cycle C containing a and we augment on it (dijoin and non-dijoin arcs have their status reversed, jumping arcs are ignored) to obtain T_{i+1}. Provided that this cycle is chosen (as detailed below) to be a minimal length cycle of minimum cost, one can show that $G_{i+1} = \mathcal{F}(T_{i+1}) - (P_i - a)$ also has no negative cost cycle. Since $|P_0| = |T_0| \leq n$ and the number of pending arcs decreases by one in each iteration, we have that $P_n = \varnothing$ and so $G_n = \mathcal{F}(T_n)$. Since G_n has no negative cycles, the Optimality Theorem implies that T_n is a minimum cost dijoin.

Note that, in each iteration, we may clearly determine whether a lies in a negative cost cycle in time $O(nm)$ (an $O(m)$ implementation is given below). If one is found, then we must recompute the jumping arcs for the new dijoin T_{i+1}. This can also be done in $O(nm)$ time, and hence the total running time of the algorithm is $O(n^2m)$.

Since one of our aims is to motivate thought on solving dijoin optimization in the original topology, without explicit reference to jumping arcs, we now show how the augmenting structure from Section 3.1.1 is obtained directly from a negative cycle C in $\mathcal{D}(T)$.

LEMMA 8.11 *The cycle C gives rise to a negative cost clean strongly connected subgraph in $\mathcal{D}(T_i)$.*

Proof. The arcs in $C - \mathcal{J}$ are present in $\mathcal{D}(T_i)$ and have a negative total cost. In addition, it follows from Frank that augmenting on C produces a dijoin T_{i+1} of lower cost than T_i. Hence, repeating the arguments used in the proof of Theorem 8.7, we see that $T_{i+1} = T \otimes H$, where H is a negative cost clean strongly connected subgraph. Moreover, H is the union of $C - \mathcal{J}$ and a collection of zero arcs, and is therefore easily obtained. □

We now describe a faster implementation of (and fill in some technical details concerning) the PENDING-ARC algorithm. In particular, we show how to compute a suitable cycle C in $O(m)$ time. Therefore, letting $J(n)$ denote the worst case time needed to update the set of jumping arcs in an n-node graph after a cycle augmentation is performed, we obtain the following result.

THEOREM 8.11 *The algorithm* PENDING-ARC *solves the minimum dijoin problem in* $O(nm + nJ(n))$ *time.*

Theorem 8.11 suggests there may be hope of finding even more efficient methods for making a digraph strongly connected. It clearly levels the blame for inefficiency squarely on the shoulders of computation of the jumping arcs. We discuss the implications of this in the next section. Here, however, we prove Theorem 8.11, that is, correctness of the PENDING-ARC algorithm. This amounts to showing two things: (1) After each iteration, T_{i+1} is again a dijoin and (2) G_{i+1} has no negative cycle. Clearly if we do not augment on a negative cycle, then both these conditions hold. So we assume that we do apply an augmentation. Frank's work indeed shows (see also Lemma 8.13) that if C is chosen as a minimal length minimum cost cycle, then T_{i+1} is again a dijoin. In order to prove (2) we need to more completely describe how to find the cycle C.

The simplest way to establish the result, is to maintain shortest path distances (from an arbitrary fixed node r) in each G_i as we go along. Let $d_i(x)$, or simply $d(x)$, denote this distance for each node x. Let $a = (u, v)$ be the new pending arc to be included. We are looking for a shortest path from v to u in the digraph G_i. This can be achieved in $O(m)$ time by allowing Dijkstra's algorithm to revisit some nodes which were already placed in the shortest path tree (this is described in Bhandari (1994) in a special setting arising from minimum cost flows, but his technique works for any digraph without negative cycles). A more traditional approach, however, is as follows. For each arc $b = (x, y) \in G_i$ let $\hat{c}_b$ denote its cost in this auxiliary digraph, and set $c'_b = d(x) + \hat{c}_b - d(y)$. Consequently, each c'_b is non-negative since the shortest path values satisfy Bellman's inequalities: $d(y) \leq d(x) + c_{(x,y)}$ for each arc (x, y). In the following, we refer to an arc as *tight*, relative to the d-values, if $c'_b = 0$.

Grow a shortest path tree F from v using the costs c'. If the shortest path to u is at least the auxiliary cost $|\hat{c}_a|$ of a then there is no negative cost cycle in $G_i \cup \{a\}$. In this case, there may be a shorter path from r to v using the arc (u, v). We can then easily update the rest of the distance labels simply by growing an arborescence from v. So assume there is a negative cost cycle in $G_i \cup \{a\}$. We obtain such a cycle C in $\mathcal{F}(T_i)$ via the union of a with a $v - u$ path in F. Note that since $c' \geq 0$, we may assume without loss of generality that C does not contain any shortcuts via a jumping arc (these have auxiliary cost 0). We now perform an augmentation on T_i along C, and let T_{i+1} be the resulting set of arcs. We also set $P_{i+1} = P_i - \{a\}$ and $G_{i+1} = \mathcal{F}(T_{i+1}) - P_{i+1}$.

It remains to show that G_{i+1} has no negative cost cycle. In order to do this we show how to update the shortest path distances $d(x)$. Note that it is sufficient to find new values that (i) satisfy the Bellman inequalities

and (ii) for each node $x \neq r$, there is a tight arc entering x. In order to establish these facts we need one technical lemma, whose proof is deferred to the end of the section.

LEMMA 8.12 *If (x, y) is a jumping arc on C, then after augmenting T_i along C, there is a jumping arc (y, x).*

Let δ be the length of the path in F from v to u. For each integer $i \leq \delta \leq |\hat{c}_a|$, and each node $x \in F$ at distance i from v, we set $d(x) = d(x) - (\delta - i)$. In particular, we reduce $d(v)$ by δ. One easily checks that after performing this change, every arc of $C - a$ becomes tight under the new $d(x)$ values. Thus, using Lemma 8.12, the reverse of all of these arcs occurs in G_{i+1} and each of these arcs is tight. One also checks that the Bellman inequalities still hold for all arcs of $G_i - C = G_{i+1} - \overline{C}$. We have thus established (i). Moreover, one checks that the above facts imply that (ii) holds for every node except possibly u. In addition, it fails at u only if there was only a single tight arc into u previously and this arc was on C.

To correct this, we grow an arborescence R from u as follows. First increase $d(u)$ until $\bar{a}$ becomes tight; that is, $d(u) = d(v) + |\hat{c}_a|$. Now u has a tight arc entering it, but this may have destroyed condition (ii) for some other node if its only tight arc came from u (and, in fact, was the reverse of an arc on C!). We scan u's out-neighbours to check for this. If we find such a neighbour x, we add it and the tight arc to R. We then update x's label; this involves searching x's in-neighbours and, possibly, discovering a new tight arc. We then repeat this process until no further growing is possible. The process terminates in $O(m)$ time provided we do not revisit a node. Such a revisitation would correspond to a directed cycle of tight arcs. This, however, could not be the case, since each of these tight arcs was the unique such arc entering its head and every node originally had a directed path of tight arcs from the source node r. Thus after completing this process, we have amended the labels $d(x)$ so that every node satisfies (ii), and every arc satisfies (i). Thus we have new shortest path labels.

We now prove Lemma 8.12 and, thus, Theorem 8.11. To do so, we invoke the following result from the work of Frank.

LEMMA 8.13 (FRANK) *Let S induce a directed in-cut in D and let j be the number of jumping arcs in $C \cap \delta^-(S)$. Then $|\delta^-(S) \cap T_i| \geq 1 + j$.*

One easily checks that this lemma implies that T_{i+1} is again a dijoin. From this we also obtain the following structure on directed cuts that become "strict" after augmentation.

LEMMA 8.14 *Suppose that S induces a directed in-cut with $|\delta^-(S) \cap T_{i+1}| = 1$. Then either C did not cross the cut induced by S, or*

(i) Every arc of $C \cap \delta^+_{G_i}(S)$ is the reverse of an arc of T_i.

(ii) Every arc of $C \cap \delta^-_{G_i}(S)$ is a jumping arc except possibly one which is an arc of $A - T_i$.

Proof. Since S induces an in-cut, each arc of C in $\delta^+(S)$ is either a jumping arc or the reverse of an arc of T_i. Let j_1 be the number of jumping arcs, and t be the number of negative cost arcs. Similarly, each arc of C in $\delta^-(S)$ must be either a jumping arc or an arc of $A - T_i$. Let j be the number of jumping arcs and k be the number of arcs of $A - T_i$. Note that $|\delta^-(S) \cap T_{i+1}| \geq k$ and hence $k \in \{0, 1\}$. Note next that $j_1 + t = j + k$. Moreover, by the previous lemma we have that $|\delta^-(S) \cap T_i| \geq 1 + j$. Thus $1 = |\delta^-(S) \cap T_{i+1}| \geq j + 1 - t + k$, and so $j + k - t \leq 0$ which implies $j_1 = 0$. This completes the proof. □

Using this, our desired result follows.

Proof of Lemma 8.12. Suppose that (y, x) is not a jumping arc after T_i is augmented along C. Then there is a directed cut induced by a set S such that $x \in S$, $y \notin S$ and $|\delta^-(S) \cap T_{i+1}| = 1$. But then by Lemma 8.14 every arc of $\delta^+(S) \cap C$ must be the reverse of a dijoin arc, contradicting the fact that (x, y) lies in this set. □

3.2.1 Computing the jumping arcs. The bottleneck operation in obtaining a minimum cost dijoin is obtaining the set $\mathcal{J}$ of jumping arcs. In order to find the jumping arcs we have to find the minimal strict set, denoted $P(v)$, with respect to each node v. Frank (1981) showed how to construct all the minimal strict sets in $O(n^3m)$ time, and also proved that this need be done at most n times, giving a total running time of $O(n^2m)$. He then asserted that "it would be useful to have a procedure with running time $O(n^3)$ (or perhaps $O(n^2)$)."

Gabow (1995) described such a procedure that runs in $O(nm)$, improving the running time of Frank's algorithm to $O(n^2m)$. Underlying Gabow's result is the following simple observation. Consider the digraph obtained from D by replacing each arc in D by two parallel arcs and adding the reverse of every dijoin arc. Then a node u is in $P(v)$ if and only if there are two arc-disjoint paths from v to u in this digraph. Gabow developed a general method which builds efficient representations of arbitrary posets of strict sets. This is achieved by repeatedly finding a centroid (like a separator) of the poset. He then uses the final representation to answer 2-arc connection queries and, thus, find the jumping arcs.

Here we describe an elementary $O(nm)$ algorithm, based upon arborescence growing, for computing the minimal strict sets. We also discuss the hardness of improving upon this. We use the terminology that, for a dijoin arc a, an out-cut $\delta^+(S)$ is *a-separating (for v)* if a is the unique arc of T in $\delta^+(S)$ and $v \notin S$. We refer to the set S as being *a-separating (for v)* as well. Observe that any node that is not in $P(v)$ is contained in (the shore of) a separating cut for v. If no such cut exists then the node is in $P(v)$. This motivates the following question: given a dijoin arc a and a node v, find the a-separating cut for v (if it exists) which is induced by a maximal set $S_a(v)$. We use the notation S_a instead of $S_a(v)$ if the context is clear. We call such a question a *red query*, and using such queries we can compute each $P(v)$ in $O(m)$ time. To do this we grow an arborescence, called the *jumping tree*, rooted at v in $D \cup \overline{D}$. As we proceed, the nodes are coloured blue and red depending upon the outcome of the query. At the end of the algorithm, the minimal strict cut $P(v)$ consists exactly of those nodes coloured blue.

Growth of the jumping tree $\mathcal{T}$ alternates between blue phases and red phases. A *blue phase* consists of repeatedly finding a blue node u in the tree and a node w not yet in the tree, for which there is an arc $(u, w) \in D$. We then add the arc to the tree and label w as blue. We repeat this until no such arcs exist. At this point the nodes of $\mathcal{T}$ induce an in-cut. We then choose a dijoin arc $a = (u, w)$ in this in-cut, and attempt to determine the colour of v. More precisely, we make a *red query* with respect to a and u, the result of which is an arborescence S_a rooted at u. If S_a is empty, in which case there are no a-separating cuts for v, we colour u blue. Otherwise, S_a identifies a maximal a-separating cut, and all nodes in this set are coloured red. The algorithm is formally described below. Here $\Gamma^+(S)$ denotes the set of out-neighbours of S, that is, those nodes $y \notin S$ such that there is an arc (x, y) for some $x \in S$.

The jumping-tree algorithm.

```
Colour v blue; set T = ({v}, ∅)
While V(T) ≠ V
If there is an arc (u,w) ∈ A(D) such that u ∈ T, w ∉ T
   Colour w blue; add w and (u,w) to T
Otherwise there exists a = (u,w) ∈ T such that u ∉ T, w ∈ T
   Add u, ā to T and let S_a =REDQUERY(a)
   If V(S_a) = ∅ then colour u blue
   Else colour all nodes of S_a red and add S_a to T
      Colour all nodes of Γ+(S_a) − V(T) blue
EndWhile
```

THEOREM 8.12 *The algorithm* JUMPING-TREE *correctly calculates $P(v)$.*

Proof. Notice that a node u can only be coloured red if there is a dijoin arc a for which there is a justifying cut $\delta^+(S)$ with $u \in S$ and $v \in V - S$. Since this is a separating cut, such a node is not contained in $P(v)$ and is, therefore, correctly labelled. Next we show that any blue nodes is in $P(v)$. This we achieve using induction beginning with the observation that the blue node v is correctly labelled. Now a node w may be coloured blue for one of two reasons. First, there is a blue node u and an arc (u, w). Now, if u is correctly labelled then w must be as well. Suppose the converse, and let $\delta^+(S)$ be a separating cut with respect to v that contains w in its shore. Clearly u must then also be in this shore, a contradiction. A node u can also be coloured blue if there is a dijoin arc a inducing a maximal separating cut $\delta^+(S)$ with respect to v, contains a non-dijoin arc (y, u). Suppose that there is a dijoin arc a' inducing a maximal separating cut $\delta^+(S')$ with respect to v whose shore contains u. Since there is an arc (y, u), we have $y \in S'$ and therefore S and S' intersect. It follows that $S \subseteq S'$, otherwise $\delta^+(S \cap S')$ is not a maximal a'-separating cut for v. This implies that the head of arc a, say z is in S'. We obtain a contradiction since z must be coloured blue. □

LEMMA 8.15 *The algorithm* JUMPING-TREE *can be used to find the jumping arcs in* $O(nm)$ *time.*

Proof. The algorithm JUMPING-TREE evidently runs in $O(m)$ time if the red queries can be answered in say $O(|S_a|)$-time. To achieve this, we implement an $O(nm)$ time preprocessing phase. In particular, for each dijoin arc $a = (u, w) \in T$, we spend $O(m)$ time to build a structure that allows us later to find an arbitrary set S_a in time $O(|S_a|)$.

Our method is as follows. Let $D_a = D \cup (\overline{T} - \bar{a})$. Initially, we call node u *alive*. We then repeatedly grow, in D_a, maximal in-arborescences A_x from any existing alive node x. In building such arborescences we may spawn new alive nodes. As we proceed, we let X be the set of *visited* nodes. We add a new node to some A_x only if it is not already in X. If it is added, then it is marked as visited and put in X. In addition, if this node had been labelled alive, then this label is now removed. Each node z in A_x is also marked by an (x) so we know that it was node x's tree that first visited z.

After A_x is completed, observe that X induces a strict cut containing a as its only dijoin arc. All out-neighbours of X, except w, are then marked as alive, i.e., put on a stack of candidates for growing future trees. Upon completion, we have generated a poset structure for the a-strict sets. We use this structure to create an acyclic graph H which has a node y for each arborescence A_y which was grown. There is an arc $(x, y) \in H$ if there is an arc with tail in A_x and head in A_y. In other

words, H is obtained by contracting each A_x to a single node. This graph is easily built as the algorithm runs.

Given this preprocessing, we now show how to find the maximal a-separating sets for v. Consider a partial arborescence $\mathcal{T}$ created during the course of the jumping tree algorithm for v. Observe that if some node $y \in H$ lies in S_a, then $A_y \subseteq V(S_a)$ by construction of A_y. Thus, our task amounts to determining which nodes of H lie in S_a. In addition, observe that if some node $y \in H$ lies in $V - \mathcal{T}$, then either (i) $v \in A_y$ or (ii) $A_y \subseteq V - V(\mathcal{T})$. To see this, suppose that $v \notin A_y$ and that $A_y \cap \mathcal{T} \neq \varnothing$. Then $\mathcal{T}$ must have grown along some arc into $A_y \cap \mathcal{T}$. However, no such arc exists in $D \cup \overline{T - a}$. In particular, this also shows that $S_a \subseteq V - \mathcal{T}$. It then follows that a node $y \in H$ lies in S_a if and only if $y \notin \mathcal{T}$.

Consequently, we may build S_a by starting with the node of H which contains the tail of a and growing a maximal arborescence in $H - \mathcal{T}$. Then S_a is the union of those A_y's which were visited in this process. Given H and the A_y's, this can be done in $O(|S_a|)$ time, as required. □

We remark that Gabow also showed how fast matrix multiplication can be used to find all the strict minimal cuts in $O(n^{\mathrm{MM}(n)})$ time (here, $\mathrm{MM}(n)$ denotes the time to do matrix multiplication). Our algorithm may also be implemented in this time. We end this section by commenting on the hardness of finding a more efficient algorithm for calculating the $P(v)$. In particular, we show that finding all the minimal strict sets is as hard as boolean matrix multiplication. To achieve this we show that *transitive closure* in acyclic graphs is a special case of the minimal strict set problem.

LEMMA 8.16 *The minimal strict set problem is as hard as transitive closure.*

Proof. Given an acyclic graph D we form a new graph D' as follows. Add two new nodes s and t with an arc (s, t). In addition, add an arc from s to every source in D and add an arc to t from every sink in D. Clearly, the arc (s, t) is itself a dijoin T in D'. Now observe that, for each node v in D, there is a correspondence between the reachability sets for v in D and the minimal strict set, with respect to T, containing v in D'. The result follows. □

To see that transitive closure is as hard as Boolean matrix multiplication. Take two matrices A and B and consider the acyclic graph defined

by the adjacency matrix

$$M = \begin{pmatrix} I & A & 0 \\ 0 & I & B \\ 0 & 0 & I \end{pmatrix}.$$

Now the transitive closure of M is then:

$$\mathrm{Cl}(M) = \begin{pmatrix} I & A & AB \\ 0 & I & B \\ 0 & 0 & I \end{pmatrix}.$$

Therefore, trying to speed up the minimum cost dijoin algorithm by finding more efficient methods to calculate the minimal strict cuts may be difficult. One possible way around this would be to avoid calculating the minimal strict cuts from scratch at each iteration. Instead, it may well be possible to more efficiently update the minimal strict cuts after each augmentation.

4. Packing dijoins

As mentioned in the introduction, an important question regarding dijoins is the extent to which they pack. Here, we discuss this topic further. We consider a digraph D with a non-negative integer vector u of arc weights, and denote by $\omega_D(u)$ the minimum weight of a directed cut in D. An initial question of interest is: determine the existence of a constant k_0 such that every weighted digraph D, with $\omega_D(u) \geq k_0$, admits a pair of disjoint dijoins contained in the support of u. (The unweighted case was discussed in Section 2.) We approach this by considering submodular network design problems with associated orientation constraints (as were recently studied in Frank, Király, and Király, 2001, and Khanna et al., 1999). First, however, we look at the problem of fractionally packing dijoins.

4.1 Half-integral packing of dijoins

By blocking theory and the Lucchesi–Younger Theorem (see Section 1.1), any digraph D with arc weighting vector u has a fractional u-packing of dijoins (a packing such that each arc a is in at most u_a of the dijoins) of size $\omega_D(u)$. We show now that there is always a large $\frac{1}{2}$-integral packing of dijoins.

THEOREM 8.13 *For any digraph D and non-negative arc vector u, there is a half-integral u-packing of dijoins of size $\omega_D(u)/2$.*

Proof. Let $k = \omega_D(u) \geq 2$. We now fix a node v and consider a partition of the set of directed cuts $\delta^+(S)$ into two sets. Following a well-known

trick, let $\mathcal{O} = \{S : \delta^-(S) = \varnothing, v \in S\}$ and $\mathcal{I} = \{S : \delta^+(S) = \varnothing, v \in S\}$. Next note that an arc vector x identifies a (fractional) dijoin if and only if $x(\delta^+(S)) \geq 1$ for each $S \in \mathcal{O}$ and $x(\delta^-(S)) \geq 1$ for each $S \in \mathcal{I}$.

Consider now the digraph H obtained from D by deleting any zero-weight arcs and adding infinitely many copies of each reverse arc. It follows that for each proper $S \subseteq V$ containing v we have $|\delta^+_H(S)| \geq k$. The Branching Theorem of Edmonds (1973) then implies that H contains k disjoint spanning out-arborescences rooted at v. Call these arborescences $O_1, O_2, \ldots, O_k$, and for each i, let x^i be the $0-1$ incidence vector in $\mathbb{R}^A$ of the set of forward arcs $O_i \cap A$. Obviously for each arc a, we have $\sum_i x^i_a \leq 1$. Similarly, there are k disjoint spanning in-arborescences rooted at v and an associated sequence y^i of arc vectors. We thus have, for each i and j, that $x^i(\delta^+(S)) \geq 1$ for each $S \in \mathcal{O}$ and $y^j(\delta^+(S)) \geq 1$ for each $S \in \mathcal{I}$. Hence, the support of the integral vector $x^i + y^j$ identifies a dijoin $T_{i,j}$ for each i, j pair. Since any arc is in at most two of the dijoins $T_{1,1}, T_{2,2}, \ldots, T_{k,k}$, a $\frac{1}{2}$-integral dijoin-packing of size $k/2$ is obtained by giving each such dijoin weight one half. □

We speculate that the structure of the above proof may also be used in order to settle Conjecture 8.1. Namely, consider the digraph H obtained in the proof, and let v be any node in H. Is it the case that for each $r = 0, 1, \ldots, \omega_D(u)$ there exists r arc-disjoint out-arborescences $T_1, T_2, \ldots, T_r$ in H rooted at v with the following property? In $H' - (\bigcup_i A(T_i))$ each cut $\delta_{H'}(S)$ with $v \in S$, contains at least $\omega_D(u)$ incoming arcs. Thus we could also pack $\omega_D(u) - r$ incoming arborescences at v. If such an out-and-in arborescence packing result holds, then Conjecture 8.1 holds by taking $r = \lfloor \omega_D(u)/2 \rfloor$ and then combining each out-arborescence with each in-arborescence to obtain a dijoin.

4.2 Skew supermodularity and packing dijoins

We begin this section by recalling some definitions. Two sets A and B are *intersecting* if each of $A - B, B - A, A \cap B$ are non-empty; the sets are *crossing* if, in addition, $V - (A \cup B)$ is non-empty. A family of sets $\mathcal{F}$ is a *crossing family* (respectively, *intersecting family*) if for any pair of crossing (respectively, intersecting) sets $A, B \in \mathcal{F}$ we have $A \cap B, A \cup B \in \mathcal{F}$. The submodular flow polyhedra of Edmonds and Giles are given in terms of set functions defined on such crossing families of sets. We consider a larger family of set functions based on a notion of skew submodularity, a concept introduced for undirected graphs in Williamson et al. (1995). A set family $\mathcal{F}$ is *skew crossing* if for each intersecting pair A and B either $A \cap B, A \cup B \in \mathcal{F}$ or $A - B, B - A \in \mathcal{F}$.

A real-valued function f defined on $\mathcal{F}$ is *skew supermodular* if for each intersecting pair $A, B \in \mathcal{F}$ one of the following holds:

(i) $A \cap B, A \cup B \in \mathcal{F}$ and $f(A) + f(B) \leq f(A \cap B) + f(A \cup B)$.
(ii) $A - B, B - A \in \mathcal{F}$ and $f(A) + f(B) \leq f(A - B) + f(B - A)$.

We claim that for an arbitrary digraph D, the family $\mathcal{F}^{\pm}(D) = \{S : \delta^+(S) = \varnothing \text{ or } \delta^-(S) = \varnothing\}$ is skew crossing. Indeed, consider intersecting members A and B of $\mathcal{F}^{\pm}$. Suppose first that both A and B induce out-cuts (or in-cuts). If A and B are crossing then $A \cap B, A \cup B$ are also out-cuts (respectively in-cuts). If A and B are not crossing then both $A - B$ and $B - A$ induce in-cuts (respectively out-cuts). Finally, suppose that A induces an out-cut and B induces an in-cut. Then $A - B$ is an out-cut and $B - A$ is an in-cut. Thus the claim is verified.

We are interested in *skew supermodular network design problems* with *orientation constraints*. That is, we have a digraph $D = (V, A)$ and a skew supermodular function f defined on a skew crossing family $\mathcal{F}$ of subsets of V. We are also given a partition of arcs into pairs a and $\bar{a}$, and for each such pair there is a capacity u_a. We then define $\mathcal{P}_D(f, u)$ to be the polyhedron of all non-negative vectors $x \in \mathbb{Q}^A$ such that $x(\delta^+(S)) \geq f(S)$ for each $S \in \mathcal{F}$ and $x_a + x_{\bar{a}} \leq u_a$ for each arc pair a and $\bar{a}$. In general, the region $\mathcal{P}_D(f, u)$ need not be integral and we are interested in finding minimum cost vectors in its integer hull. There are a number of related results in this direction. Notably, Melkonian and Tardos (1999) show that if f is crossing supermodular and if the orientation constraints are dropped, then each extreme point of this polyhedron contains a component of value at least $\frac{1}{4}$. Khanna et al. (1999) describe a 4-approximation algorithm for the case with orientation constraints provided that $f(S) = 1$ for every proper subset S. They also show that the polyhedron is integral in the case that f is intersecting supermodular (see Frank, Király, and Király, 2001, for generalizations of this latter result).

In terms of packing dijoins, we are interested in the polyhedron $\mathcal{P}_H(f, u)$ where $H = D \cup \overline{D}$ and $f(S) = 1$ for each $S \in \mathcal{F}^{\pm}$. Observe that, for any digraph D and weighting u with $\omega_D(u) \geq 2$, we have $\mathcal{P}_H(f, u)$ is non-empty since we may assign $\frac{1}{2}$ to each arc variable. For now, we may as well assume u is a $0 - 1$ vector. Suppose that $\mathcal{P}_H(f, u)$ has an integral solution x. Let F be those arcs $a \in D$ such that $x_a = 1$, and let $K = A - F$. We claim that the arc sets of D associated with both F and K are dijoins. For suppose that S induces an out-cut. Then the constraint $x(\delta^+(S)) \geq 1$ implies that F contains an arc from this cut, whereas the constraint $x(\delta^+(V - S)) \geq 1$ implies that K intersects this

cut.[5] The example of Schrijver (1980) implies that there is a weighted digraph with $\omega_D(u) = 2$ for which $\mathcal{P}_H(f, u)$ has no integral point (since D does not have a pair of disjoint dijoins amongst the support of u). We ask:

CONJECTURE 8.2 *Is there a constant k_0 such that, for every weighted digraph D, if $\omega_D(u) \geq k_0$ then $\mathcal{P}_H(f, u)$ contains an integral point? Is this true for $k_0 = 4$?*

We note that, if $\omega_D(u) \geq 4k$ were to imply that $\mathcal{P}_H(kf, u)$ contains an integral point, then the methods used at the beginning of this section can be made to yield an $\Omega(k)$ packing of dijoins.

Acknowledgments. The first author is grateful to Dan Younger for the most pleasurable introduction to this topic. The authors are also grateful to Bill Pulleyblank for his insightful suggestions.

References

Bhandari, R. (1994). Optimal diverse routing in telecommunication fiber networks. *Proceedings of IEEE Infocom*, pp. 1498–1508.

Chvátal, V. (1975). On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18:138–154.

Cook, W., Cunningham, W., Pulleyblank, W., and Schrijver, A. (1998). *Combinatorial Optimization.* Wiley-Interscience, New York.

Cui, W. and Fujishige, S. (1998). A primal algorithm for the submodular flow problem with minimum-mean cycle selection. *Journal of the Operations Research Society of Japan*, 31:431–440.

Cunningham, W. and Frank, A. (1985). A primal-dual algorithm for submodular flows. *Mathematics of Operations Research*, 10(2):251–261.

Edmonds, J. (1973). Edge-disjoint branchings. In: R. Rustin (ed.), *Combinatorial Algorithms*, pp. 91–86, Alg. Press, New York.

Edmonds, J. and Giles, R. (1977). A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, 1:185–204.

Frank, A. (1981). How to make a digraph strongly connected. *Combinatorica*, 1:145–153.

Frank, A., Király, T., and Király, Z. (2001). *On the Orientation of Graphs and Hypergraphs.* Technical Report TR-2001-06 of the Egerváry Research Group, Budapest, Hungary.

[5] We remark, that we could have also cast this as a "skew" submodular flow model as well; this could be achieved by defining $f(S) = |\delta(S)| - 1$ and then requiring $x(\delta^+(S)) - x(\delta^-(S)) \leq f(S)$ for each $S \in \mathcal{F}$.

Feofiloff, P. and Younger, D. (1987). Directed cut transversal packing for source-sink connected graphs. *Combinatorica*, 7:255 – 263.

Fleischer, L. and Iwata, S. (2000). Improved algorithms for submodular function minimization and submodular flow. *STOC*, 107 – 116.

Fujishige, S. (1978). Algorithms for solving the independent flow problem. *Journal of the Operations Research Society of Japan*, 21(2):189 – 204.

Fujishige, S. (1991). Submodular functions and optimization. *Annals of Discrete Mathematics*, 47, Monograph, North Holland Press, Amsterdam.

Gabow, H. (1995). Centroids, representations, and submodular flows. *Journal of Algorithms*, 18:586 – 628.

Iwata, S., Fleischer, L., and Fujishige, S. (2001). A combinatorial strongly polynomial time algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761 – 777.

Iwata, S., McCormick, S., and Shigeno, M. (2003). Fast cycle canceling algorithms for minimum cost submodular flow. *Combinatorica*, 23:503 – 525.

Jain, K. (2001). A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39 – 60.

Karzanov, A.V. (1979). On the minimal number of arcs of a digraph meeting all its directed cutsets. Abstract in *Graph Theory Newsletter*, 8.

Khanna, S., Naor, S., and Shepherd, F.B. (1999). Directed network design problems with orientation constraints. *SODA*, 663 – 671.

Klein, M. (1967). A primal method for minimal cost flows. *Management Science*, 14:205 – 220.

Lee O. and Wakabayashi, Y. (2001). Note on a min-max conjecture of Woodall. *Journal of Graph Theory*, 14:36 – 41.

Lehman, A. (1990). Width-length inequality and degenerate projective planes. In: P.D. Seymour and W. Cook (eds.), *Polyhedral Combinatorics*, pp. 101 – 106, *Proceedings of the DIMACS Workshop*, Morristown, New Jersey, June 1989. American Mathematical Society.

Lovász, L. (1976). On two minmax theorems in graphs. *Journal of Combinatorial Theory, Series B*, 21:96 – 103.

Lucchesi, C. (1976). *A Minimax Equality for Directed Graphs.* Ph.D. thesis, University of Waterloo.

Lucchesi, C. and Younger, D. (1978). A minimax theorem for directed graphs. *Journal of the London Mathematical Society*,17:369 – 374.

McWhirter, I. and Younger, D. (1971). Strong covering of a bipartite graph. *Journal of the London Mathematical Society*,3:86 – 90.

Melkonian, V. and Tardos, É. (1999). Approximation algorithms for a directed network design problem. *IPCO*, 345–360.

Pulleyblank, W.R. (1994). Personal communication.

Schrijver, A. (1980). A counterexample to a conjecture of Edmonds and Giles. *Discrete Mathematics*, 32:213–214.

Schrijver, A. (1982). Min-max relations for directed graphs. *Annals of Discrete Mathematics*, 16:261–280.

Schrijver, A. (2000). A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80:346–355.

Schrijver, A. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer Verlag, Berlin.

Wallacher, C. and Zimmermann, U. (1999). A polynomial cycle canceling algorithm for submodular flows. *Mathematical Programming, Series A*, 86(1):1–15.

Williamson, D., Goemans, M., Mihail, M., and Vazirani, V. (1995). A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15:435–454.

Younger, D. (1963). *Feedback in a directed graph.* Ph.D. thesis, Columbia University.

Younger, D. (1969). Maximum families of disjoint directed cuts. In: *Recent Progress in Combinatorics*, pp. 329–333, Academic Press, New York.

Zimmermann, U. (1982). Minimization on submodular flows. *Discrete Applied Mathematics*, 4:303–323.

Chapter 9

HYPERGRAPH COLORING BY BICHROMATIC EXCHANGES

Dominique de Werra

Abstract A general formulation of hypergraph colorings is given as an introduction. In addition, this note presents an extension of a known coloring property of unimodular hypergraphs; in particular it implies that a unimodular hypergraph with maximum degree d has an equitable k-coloring $(S_1, \ldots, S_k)$ with $1 + (d-1)|S_k| \geq |S_1| \geq \cdots \geq |S_k|$. Moreover this also holds with the same d for some transformations of H (although the maximum degree may be increased). An adaptation to balanced hypergraphs is given.

1. Introduction

This paper presents some basic concepts on hypergraph coloring and in particular it will use the idea of bichromatic exchanges to derive some results on special classes of hypergraphs. In de Werra (1975) a result on coloring properties of unimodular hypergraphs is formulated in terms of "parallel nodes"; it is a refinement of a basic property of unimodular hypergaphs given in de Werra (1971). As observed recently by Bostelmann (2003), the proof technique given in de Werra (1975) may fail in some situations. The purpose of this note is to provide a stronger version of the result in de Werra (1975) together with a revised proof technique.

We will use the terminology of Berge (see Berge, 1987, where all terms not defined here can be found).

A hypergraph $H = (X, \mathcal{E})$ is characterized by a set X of nodes and a family $\mathcal{E} = (E_i \mid i \in I)$ of edges $E_i \subset X$; if $|E_i| = 2$ for all $i \in I$ H is a graph.

In order to extend in a non trivial way the concepts of node coloring (and also of edge coloring) of graphs to hypergraphs, we shall define a *k-coloring* $C = (S_1, \ldots, S_k)$ of a hypergraph $H = (X, \mathcal{E})$ as a partition

of the node set X of H into subsets $S_1, \ldots, S_k$ (called *stable* sets) such that no S_j contains all nodes of an edge E_i with $|E_i| \geq 2$.

Notice that if $|E_i| = 2$ for each edge E_i in $\mathcal{E}$, then this defines precisely classical node k-colorings in graphs: the two end nodes of an edge must receive different colors.

The *adjacency matrix* $A = (a_{ij} \mid i \in I, j \in X)$ of a hypergraph is defined by setting $a_{ij} = 1$ if node j is in edge E_i or $a_{ij} = 0$ else. So a k-coloring of H may be viewed as a partition of the column set into subsets $S_1, \ldots, S_k$ such that for each row i with at least two non zero entries, there are at least two subsets S_p, S_q of columns for which $\sum(a_{ij} \mid j \in S_p) \geq 1$, $\sum(a_{ij} \mid j \in S_q) \geq 1$.

Notice that if we consider the transposed matrix A^T of A, it may be viewed as the adjacency matrix of some hypergraph $H^* = (\mathcal{J}, \mathcal{X})$ called the *dual* of H; it is obtained by interchanging the roles of nodes and edges of H. In other words each edge E_i of H becomes a node e_i of H^*; each node j of H becomes an edge X_j of H^*. Edge X_j contains all nodes e_i such that $E_i \ni j$ in H.

The dual of a hypergraph is another hypergraph, so coloring the edges of H is equivalent to coloring the nodes of its dual H^* (in an edge coloring of a hypergraph, we would require that for each node j contained in more than two edges, not all edges containing j have the same color).

So edge colorings and node colorings are equivalent concepts for general hypergraphs.

Notice however that the dual of a graph G is generally a hypergraph. So coloring the edges of G in such a way that no two adjacent edges have the same color is equivalent to coloring the nodes of hypergraph G^* in such a way that in each edge of G^* all nodes have different colors. But this is simply a node coloring problem in the graph $\widetilde{G}$ obtained by replacing each edge of G^* by a clique.

So in the remainder of this note we shall consider only node colorings of hypergraphs without loss of generality. We will review some special types of colorings (which are more restricted than usual colorings of hypergraphs in the sense that the subsets S_i have to satisfy some additional conditions) and this will lead us to consider some classes of hypergraphs in which such colorings may be constructed.

This will provide some opportunity to illustrate how some classical coloring techniques like bichromatic exchanges can be extended to hypergraphs.

The results to be derived by such procedures are simple generalizations of some edge coloring problems in graphs and the reader is encouraged to derive them from their hypergraph theoretical formulations.

We also refer the reader to the seminal book of Berge (Berge, 1987) for additional illustrations and basic properties of hypergraphs.

H is *unimodular* if its adjacency matrix $A = (a_{ij} \mid i \in I, j \in X)$ is totally unimodular.

For $k \geq 2$, an *equitable k-coloring* of H is a partition of the node set X into k subsets $S_1, \ldots, S_k$ such that for each color r and for each edge E_i, we have

$$\lfloor |E_i|/k \rfloor \leq |E_i \cap S_r| \leq \lceil |E_i|/k \rceil \tag{9.1}$$

It is known (see Chapter 5 in Berge, 1987) that H is unimodular if and only if every subhypergraph H' of H has an equitable bicoloring. From this it can be seen (see de Werra, 1971) that for any $k \geq 2$ a unimodular H has an equitable k-coloring.

Let d be the maximum degree of the nodes of H, i.e., $d = \max_j \sum_i a_{ij}$ where A is the edge-node incidence matrix of H. Two nodes of H are *parallel* if they are contained in exactly the same edges; parallelism is an equivalence relation on X; let $N_1, \ldots, N_p$ be its classes.

In de Werra (1975) the following result was given:

PROPOSITION 9.1 *Let H be a unimodular hypergraph with maximum degree d. Then for any $k \geq 2$ H has an equitable k-coloring $C = (S_1, \ldots, S_k)$ satisfying*

(a) $\max_r |S_r| \leq 1 + (d-1) \min_r |S_r|$

(b) $-1 \leq |N_s \cap S_r| - |N_s \cap S_t| \leq 1$ $(r, t \leq k)$

for every class N_s of parallel nodes.

We will state a simple extension of this result and give a proof technique which can be used to derive Proposition 9.1.

2. An extension and a revised proof

Let $H = (X\mathcal{E})$ be a hypergraph; we call *x-augmented hypergraph* of H any hypergraph $H(x) = (X', \mathcal{E}')$ obtained from H as follows:

$$X' = (X \backslash x) \cup \{x_1, \ldots, x_q\} \quad \text{where } x_1, \ldots, x_q \text{ are new nodes} \tag{9.2}$$

$$E_i' = \begin{cases} E_i & \text{if } E_i \not\ni x \\ (E_i \backslash x) \cup \{x_1, \ldots, x_q\} & \text{if } E_i \ni x. \end{cases} \tag{9.3}$$

Let $F_0 = \{x_1, \ldots, x_q\}$.

Furthermore, let $\mathcal{E}'' = (F_s \mid s \in J)$ be any family of edges $F_s \subseteq F_0$ such that:

(a) $F_0 \in \mathcal{E}''$

(b) $(F_0, \mathcal{E}'')$ is unimodular

Then for $H(x)$ we set

$$\mathcal{E}' = (E'_i \mid i \in I) \cup \mathcal{E}''.$$

PROPOSITION 9.2 *If H is unimodular and x is a node of H, then $H(x)$ is also unimodular.*

Proof. We simply have to prove that $H(x)$ has an equitable bicoloring (it will follow that every subhypergraph of $H(x)$ also has an equitable bicoloring).

This can be seen as follows: let us assume that we have a hypergraph H' obtained from H by simply considering that in $H(x)$ we have $\mathcal{E}'' = \{F_0\}$ (i.e., $|J| = 1$).

CLAIM 9.1 *The hypergraph $H' = H(x)$ with $J = \{0\}$ and all its subhypergraphs have an equitable bicoloring.*

Proof of the Claim. Let $C = (S_a, S_b)$ be an equitable bicoloring of H; we now extend C to a bicoloring C' of H' as follows: assume x had color a in C; we then color $x_1, x_2, \ldots, x_q$ alternately with colors a and b while starting with color a. Since $x_1, \ldots, x_q$ are contained in exactly the same edges of H', we notice that the bicoloring C' is equitable for H'.

This ends the proof of the claim. □

CLAIM 9.2 *Any equitable bicoloring C' of H' can be extended to an equitable bicoloring of $H(x)$.*

Proof of the Claim. Let $C' = (S'_a, S'_b)$ be an equitable bicoloring of H' which exists from Claim 9.1. We clearly have $-1 \leq |F_0 \cap S'_a| - |F_0 \cap S'_b| \leq 1$.

Assume w.l.o.g. that $|S'_a \cap F_0| \geq |S'_b \cap F_0|$.

Now $(F_0, \mathcal{E}'')$ has an equitable bicoloring (S''_a, S''_b) since by assumption (b) it is unimodular. This coloring satisfies $-1 \leq |S''_a \cap F_0| - |S''_b \cap F_0| \leq 1$ since by (a) F_0 is an edge and we may also assume w.l.o.g that $|S''_a \cap F_0| \geq |S''_b \cap F_0|$.

Let

$$\overline{S}_a = (S_a - F_0) \cup (F_0 \cap S''_a)$$
$$\overline{S}_b = (S_b - F_0) \cup (F_0 \cap S''_b).$$

Then it is easy to see that (since all nodes of F_0 are contained in exactly the same edges of $\mathcal{E}' - \mathcal{E}''$) $\overline{C} = (\overline{S}_a, \overline{S}_b)$ is an equitable bicoloring of $H(x)$. This ends the proof of the claim. □

Now the result follows since the construction of equitable bicolorings given above is trivially valid for any subhypergraph. □

We will say that $\widehat{H}$ is an *augmentation* of H if it is obtained by successively choosing distinct nodes x (of the initial H) and constructing an x-augmented hypergraph. Then we have from Proposition 9.2:

COROLLARY 9.1 *If $\widehat{H}$ is an augmentation of a unimodular hypergraph H, then $\widehat{H}$ is also unimodular.*

We can now state the announced extension.

PROPOSITION 9.3 *Let H be a unimodular hypergraph with maximum degree d and let $\widehat{H}$ be an augmentation of H; then for any $k \geq 2$ $\widehat{H}$ has an equitable k-coloring $(S_1, \ldots, S_k)$ satisfying*

$$\max_r |S_r| \leq 1 + (d - 1) \min_r |S_r|.$$

Notice that the maximum degree of $\widehat{H}$ may be strictly larger than d when introducing the subhypergraphs $(F_0^s, (\mathcal{E}'')^s)$ according to the augmentation procedure.

Proof. From Corollary 9.1, we know that $\widehat{H}$ is unimodular; so let $C = (S_1, \ldots, S_k)$ be an equitable k-coloring of $\widehat{H}$. Let $F_0^1, F_0^2, \ldots, F_0^t$ be the disjoint sets of nodes which have been introduced consecutively in the augmentation operations. We remove all but the edge F_0^s from each family $(\mathcal{E}'')^s$ introduced during the augmentations. Let H' be the resulting hypergraph; clearly C is also an equitable k-coloring for H'.

Then we show there exists an equitable k-coloring $C' = (S_1', \ldots, (S_k'))$ of H' with:

$$1 + (d-1)s_k' \geq s_1' \geq s_2' \geq \cdots \geq s_k'$$

where $s_r' = |S_r'|$ for all $r \leq k$. Here d is the maximum degree of H.

Let us assume that if $s_i = |S_i|$ $(i = 1, \ldots, k)$ the colors are ordered in such a way that $s_1 \geq \cdots \geq s_k$.

(1) So we assume $s_1 = s_k + K > 1 + (d-1)s_k$. We construct a simple graph G' whose nodes are those of $S_1 \cup S_k$; its edges are obtained in the following way: we examine consecutively all edges E_i' of H' (except the edges $F_0^1, F_0^2, \ldots, F_0^t$); in each E_i' we join by an edge as many disjoint pairs of nodes x, y with $x \in S_1$, $y \in S_k$ provided they have not been joined yet. For doing this we associate to each F_0^s an edge $E_i'(s)$ such that $E_i'(s) \supset F_0^s$. Such edges do exist by construction. We consider consecutively all edges E_i' (starting by the edges $E_i'(s)$; when we consider $E_i'(s)$ we first join as many pairs x, y within F_0^s and we continue with the nodes in $E_i'(s) - F_0^s$. For the other edges, we join the pairs x, y in any order.

We recall that the sets F_0^s are disjoint; so the construction is possible.

We will get a simple bipartite graph G' with maximum degree at most d since each node in H' belongs to at most d edges E'_i (not considering the edges F_0^s).

(2) Now G' has at most $d.s_k$ edges and since $s_1+s_k \geq 2+d.s_k$, G' is not connected. So there exists a connected component $G^* = (S_1^* \cup S_k^*, E^*)$ of G' for which $s_k^* < s_1^* = s_k^* + L \leq 1 + (d-1)s_k^*$. We have $0 < L \leq 1+(d-2)s_k^* \leq 1+(d-2)s_k < K$. Interchanging the nodes of S_1^* and S_k^* we get a partition $(\overline{S}_1, \overline{S}_k)$ of the nodes of the subhypergraph induced by $S_1 \cup S_k$. If (possibly after permutation of indices 1 and k) we still have $\overline{s}_1 > 1 + (d-1)\overline{s}_k$ we repeat the procedure. Finally we get an equitable bicoloring (S'_1, S'_k) with $s'_k \leq s'_1 \leq 1 + (d-1)s'_k$.

Now letting $S'_r = S_r$ for $r \neq 1, k$ we have an equitable k-coloring; after permuting the indices if necessary we have $s'_1 \geq s'_2 \geq \cdots \geq s'_k$ but the number of pairs a, b of colors for which $s'_a - s'_b = \max_{c,d}(s'_c - s'_d) = s_1 - s_k$ has decreased by at least one.

Repeating this we will finally get an equitable k-coloring $C'' = (S''_1, \ldots, S''_k)$ of H' with $1 + (d-1)s''_k \geq s''_1 \geq \cdots \geq s''_k$.

(3) We now have to transform C'' into an equitable k-coloring $\widehat{C} = (\widehat{S}_1, \ldots, \widehat{S}_k)$ of $\widehat{H}$ satisfying the same cardinality constraints, i.e., $1 + (d-1)\widehat{s}_k \geq \widehat{s}_1 \geq \cdots \geq \widehat{s}_k$.

Consider now the first edge $F_0^1 = \{x_1, \ldots, x_q\}$ introduced into $(\mathcal{E}'')^1$; since C'' is an equitable k-coloring of H' (which contains F_0^1 as an edge) we have

$$\lfloor |F_0^1|/k \rfloor \leq |S''_r \cap F_0^1| \leq \lceil |F_0^1|/k \rceil \quad \text{for } r = 1, \ldots, k.$$

Let $c_r^s = |S''_r \cap F_0^s|$ for $r = 1, \ldots, k$; $s = 1, \ldots, t$.

Now construct any equitable k-coloring of $(F_0^1, (\mathcal{E}'')^1)$; such a coloring $C^1 = (S_1^1, \ldots, S_k^1)$ exists since by (b) $(F_0^1, (\mathcal{E}'')^1)$ is unimodular. Since by (a) (F_0^1 is an edge of this hypergraph, the values $|S_1^1 \cap F_0^1|$, $|S_2^1 \cap F_0^1|$, $\ldots, |S_k^1 \cap F_0^1|$ are a permutation of $c_1^1, c_2^1, \ldots, c_k^1$. So we may reorder the colors in C' in such a way that we have an equitable k-coloring $\widetilde{C}^1 = (\widetilde{S}_1^1, \widetilde{S}_2^1, \ldots, \widetilde{S}_k^1)$ of $(F_0^1, (\mathcal{E}'')^1)$ with $|\widetilde{S}_r^1) \cap F_0^1| = c_r^1$ for $r = 1, \ldots, k$.

Now, setting $\overline{S}''_r = (S''_r - F_0^1) \cup \widetilde{S}_r^1$ for $r = 1, \ldots, k$.

We get an equitable k-coloring $\overline{C}'' = (\overline{S}''_1, \ldots, \overline{S}''_k)$ of H', which is also equitable for the edges in $(\mathcal{E}'')^1$.

Moreover we have $|\overline{S}''_r| = |S''_r|$ for $r = 1, \ldots, k$.

Repeating this procedure for $F_0^2, \ldots, F_0^t$ we will get the required equitable k-coloring $\widehat{C} = (\widehat{S}_1, \ldots, \widehat{S}_k)$; it will satisfy $|\widehat{S}_r| = |S''_r|$ for $r = 1, \ldots, k$ so that we will have:

$$1 + (d-1)\widehat{s}_k \geq \widehat{s}_1 \geq \cdots \geq \widehat{s}_k. \qquad \square$$

COROLLARY 9.2 (DE WERRA, 1975) *Let H be a unimodular hypergraph with maximum degree d; let $N_1, N_2, \ldots, N_p$ be the maximal subsets of nodes such that all nodes in N_1 are exactly in the same edges $(s = 1, \ldots, p)$. For any $k \geq 2$, H as an equitable k-coloring $(S_1, \ldots, S_k)$ such that*

(a) $\max_r |S_r| \leq 1 + (d-1) \min_r |S_r|$

(b) $\lfloor |N_s|/k \rfloor \leq |N_s \cap S_r| \leq \lceil |N_s|/k \rceil$, $r = 1, \ldots, k$, $s = 1, \ldots, p$.

Proof. H can be viewed as an augmentation of a unimodular hypergraph $\widetilde{H}$ with maximum degree d where $|\widetilde{N}_s| = 1$ for each s; we transform $\widetilde{H}$ into H by replacing each $\widetilde{N}_s$ by $F_0^s = N_s$ (if $|N_s| \geq 2$) so that in $(F_0^s, (\mathcal{E}'')^s)$ we have a single edge F_0^s (for $s = 1, \ldots, t$). □

REMARK Consider the unimodular hypergraph with edges $\{1234\}$, $\{3456\}$, $\{17\}$, $\{18\}$. In the original proof of Corollary 9.2, (de Werra, 1975) one starts from a bicoloring which is equitable except possibly for some of the subsets N_i: here $N_1 = \{34\}$, $N_2 = \{56\}$. For instance $S_1 = \{1256\}$, $S_2 = \{3478\}$; one removes the largest possible even number of nodes in each N_i (here all 4 nodes are removed). Then one constructs an equitable bicoloring of the remaining hypergraph. $\overline{S}_1 = \{1\}$, $\overline{S}_2 = \{278\}$ from which one gets $S_1' = \{135\}, S_2' = \{24678\}$. But one has $|S_2'| - |S_1'| > |S_1| - |S_2| = 0$. So it may happen that $\max_r |S_r'| - \min_r |S_r'|$ does increase at some iteration of the recoloring process.

If $H = (X, \mathcal{E})$ is the dual of a graph G (the edge-node incidence matrix A of H is the node-edge incidence matrix of G), then coloring the nodes of H is equivalent to coloring the edges of G. In such a case, the maximum degree d of H is at most two. So that the "balancing" inequalities $\max_r |S_r| \leq 1 + (d-1) \min_r |S_r|$ become simply

$$\max_{r,s} (|S_r| - |S_s|) \leq 1.$$

The "parallel nodes" in H correspond to parallel edges and so Corollary 9.2 states that for any k a bipartite multigraph has an equitable edge k-coloring such that in each family of parallel edges the coloring is equitable and furthermore the cardinalities of the different color classes are all within one (see de Werra, 1975).

The above properties have been extended to totally unimodular matrices with entries $0, +1, -1$; but in these formulations the conditions on the cardinalities $|S_r|$ are not as immediate as in the $0, 1$ case.

Balanced hypergraphs have been defined in Berge (1987) as hypergraphs which have a balanced $0, 1$-matrix as edge-node incidence matrix. A $0, 1$ matrix is *balanced* if it does not contain any square submatrix of odd order with exactly two ones in each row and in each column.

It is known (Berge, 1987) that balanced hypergraphs have a *good* k-coloring $(S_1, \ldots, S_k)$ for any $k \geq 2$, i.e., a partition of the node set X into k subsets S_r such that for each edge E and for each color r

$$|S_r \cap E| \begin{cases} \leq 1 & \text{if } |E| \leq k, \\ \geq 1 & \text{if } |E| \geq k. \end{cases}$$

Now Proposition 9.3 can be formulated as follows for balanced hypergraphs:

PROPOSITION 9.4 *Let H be a balanced hypergraph with maximum degree d; then for any $k \geq 2$, H has a good k-coloring $(S_1, \ldots, S_k)$ satisfying*

$$\max_r |S_r| \leq 1 + (d-1) \min_r |S_r|.$$

Proof. We use the same technique as in the proof of Proposition 9.3. We start from a good k-coloring $(S_1, \ldots, S_k)$ with $s_1 \geq s_2 \geq \cdots \geq s_k$ and $s_1 > 1 + (d-1)s_k$. We consider the subhypergraph H' generated by $S_1 \cup S_k$; in each edge E with $|E| \geq 2$ we link one pair x, y of nodes with $x \in S_1, y \in S_k$.

As before we get a bipartite graph G' with maximum degree d; we can interchange colors as in the proof of Proposition 9.3 and we finally get a good bicoloring (S'_1, S'_k) with $s'_k \leq s'_1 \leq 1 + (d-1)s'_k$.

Setting $S'_r = S_r$ for $r \neq 1, k$ we get again a good k-coloring. The number of pairs a, b of colors for which $s'_a - s'_b = \max_{c,d}(s'_c - s'_d) = s_1 - s_k$ has decreased by at least one.

Repeating this will finally give the required k-coloring. □

3. Final remarks

One should notice that the augmentation operation described here is the analogous of transformations which are known for perfect graphs: replacement of a node x in a perfect graph G by a perfect graph G' whose nodes are linked to all neighbors of x in G (see for instance Schrijver (1993) for a review of results and for references). Here we replace a node by a unimodular hypergraph $(F_0, \mathcal{E}'')$ but in order to have a simple recoloring procedure in the transformed hypergraph which is still unimodular, the set F_0 of all new nodes is introduced as an edge of $(F_0, \mathcal{E}'')$.

So one may furthermore wonder whether such an augmentation procedure can be defined for balanced hypergraphs. However in the case where $|F_0| > k$, we cannot use the same procedure as for unimodular hypergraphs: while for unimodular hypergraphs, it is always possible to extend an equitable k-coloring C'' of H' to an equitable k-coloring $\widehat{C}$ of $\widehat{H}$ without changing the cardinalities of the color classes, this is a priori not

possible for balanced hypergraphs. The reason lies in the fact that for a given F_0 and a given k, there is a unique vector $s_1 \geq s_2 \geq \cdots \geq s_k$ which gives the cardinalities of the color classes of an equitable k-coloring, while for good colorings (associated to balanced hypergraphs) it may not be unique. For instance for $|F_0| = 5, k = 3$, we have $(s_1, s_2, s_3) = (2, 2, 1)$ for any equitable 3-coloring, while we may have (2,2,1) or (3,1,1,) for good 3-colorings. So the proof technique used in Proposition 9.2 cannot be used in the same way.

Finally one should recall that these chromatic properties have been extended to the case of $0, +1, -1$ balanced matrices. These can be characterized by a bicoloring property in a similar way to totally unimodular matrices. Such matrices correspond to "oriented balanced hypergraphs"; they have been extended to a class called r-balanced matrices (see Conforti et al. (2005)) which is also characterized by a bicoloring property. We just mention the basic definitions: a $0, \pm 1$ matrix A has an *r-equitable bicoloring* if its columns can be partitioned into 2 color classes in such a way that:

(i) The bicoloring is equitable for the row submatrix determined by all rows with at most $2r$ non zero entries.

(ii) Every row with more than $2r$ non zero entries contains r pairwise disjoint pairs of non zero entries such that each pair contains either entries of opposite sign in columns of the same color class or entries of the same sign in columns of different color classes.

In Conforti et al. (2005) it is shown that a $0, \pm 1$ matrix A is r-balanced if and only if every submatrix of A has an r-equitable coloring. Clearly 1-balanced matrices are precisely the balanced matrices. Also if $r \geq \lceil n/2 \rceil$ (where A has n columns) the r-balanced matrices are the totally unimodular matrices.

Acknowledgments. The author would like to express his gratitude to Ulrike Bostelmann for having raised a stimulating question related to the original proof of Corollary 9.2.

References

Berge, C. (1987). *Hypergraphes.* Gauthier-Villars, Paris.

Bostelmann, U. (2003). Private communication.

Conforti, M., Cornuejols, G., and Zambelli, G. (2005). Bi-colorings and equitable bi-colorings of matrices. Forthcoming.

Schrijver, A. (1993). *Combinatorial Optimization.* Springer Verlag, New York.

de Werra, D. (1971). Equitable colorations of graphs. *Revue française d'informatique et de recherche opérationnelle*, R-3:3–8.

de Werra, D. (1975). A few remarks on chromatic scheduling. In: B. Roy (ed.), *Combinatorial Programming, Methods and Applications*, pp. 337–342, D. Reidel Publishing Company, Dordrecht.