

# CO 450/650 Combinatorial Optimization

Keven Qiu  
Instructor: Bill Cook  
Fall 2023

# Contents

<b>I</b>	<b>Introduction</b>	<b>5</b>
<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Linear Programming</b>	<b>7</b>
2.1	Farkas' Lemma . . . . .	7
2.2	Duality . . . . .	8
2.3	Complementary Slackness . . . . .	9
<b>3</b>	<b>Graph Theory</b>	<b>10</b>
3.1	Undirected Graphs . . . . .	10
3.2	Directed Graphs . . . . .	12
<b>4</b>	<b>Complexity Classes</b>	<b>13</b>
<b>II</b>	<b>Polyhedral Combinatorics</b>	<b>14</b>
<b>5</b>	<b>Integrality of Polyhedra</b>	<b>15</b>
5.1	Convex Hull . . . . .	15
5.2	Polytopes . . . . .	16
5.3	Integral Polytopes . . . . .	18
5.4	Total Unimodularity . . . . .	19
5.5	Separation and Optimization . . . . .	20
5.6	Total Dual Integrality . . . . .	21

<b>III</b>	<b>Optimal Trees and Paths</b>	<b>23</b>
<b>6</b>	<b>Minimum Spanning Trees</b>	<b>24</b>
6.1	Problem . . . . .	24
6.2	Kruskal's Algorithm . . . . .	24
6.3	Linear Programming . . . . .	25
6.3.1	Complementary Slackness Conditions . . . . .	26
6.4	Spanning Tree Polytope . . . . .	27
<b>7</b>	<b>Shortest Paths</b>	<b>28</b>
7.1	Ford's Algorithm . . . . .	29
7.2	Linear Programming . . . . .	29
<b>IV</b>	<b>Network Flows</b>	<b>31</b>
<b>8</b>	<b>Maximum Flow</b>	<b>32</b>
8.1	Problem . . . . .	32
8.2	Augmenting Path Algorithm . . . . .	32
8.2.1	Shortest Augmenting Paths . . . . .	33
8.3	Linear Programming . . . . .	34
<b>V</b>	<b>Matchings</b>	<b>38</b>
<b>9</b>	<b>Matchings</b>	<b>39</b>
9.1	Bipartite Matching . . . . .	40
9.2	Alternating Paths . . . . .	40
9.3	Matching LP . . . . .	41
9.4	Tutte's Theorem . . . . .	42
9.5	Maximum Matching . . . . .	43
9.6	Perfect Matching . . . . .	44
9.6.1	Blossom Algorithm for Perfect Matching . . . . .	44

9.7 Blossom Algorithm for Maximum Matching . . . . .	47
<b>10 Minimum-Cost Perfect Matching</b>	<b>48</b>
10.1 Linear Programming and Matching Polytope . . . . .	48
<b>11 <math>T</math>-Joins and Postman Problems</b>	<b>53</b>
11.1 Postman Problem . . . . .	53
11.2 $T$ -Joins . . . . .	54
11.3 Optimal $T$ -Join Algorithm . . . . .	55
11.4 $T$ -Join LP . . . . .	55
 <b>VI Matroids</b>	 <b>57</b>
12 Matroid Theory	58
 <b>VII Traveling Salesman Problem</b>	 <b>59</b>
13 The Traveling Salesman Problem	60

# List of Algorithms

1	Kruskal's Algorithm (Minimum Spanning Tree) . . . . .	25
2	Ford's Algorithm (Shortest Path) . . . . .	29
3	Ford-Fulkerson Algorithm (Maximum Flow/Minimum Cut) . . . . .	33
4	Blossom Algorithm (Perfect Matching) . . . . .	46
5	Blossom Algorithm (Maximum Matching) . . . . .	47
6	Blossom Algorithm (Minimum-Cost Perfect Matching) . . . . .	51
7	Optimal $T$ -Join Algorithm . . . . .	55
8	Greedy Algorithm . . . . .	58

# Part I

## Introduction

# Chapter 1

## Introduction

**Definition: Combinatorial Optimization**

A subfield of mathematical optimization which involves searching for an optimal object in a finite collection of objects.

Typically, the collection has a concise representation, while the number of objects is large. Objects include graphs, networks, and matroids.

The main tool in combinatorial optimization is linear programming duality.

# Chapter 2

## Linear Programming

### Definition: Linear Programming

The problem of finding a vector  $x$  that maximizes a given linear function  $c^T x$ , where  $x$  ranges over all vectors satisfying a given system  $Ax \leq b$  of linear inequalities.

### 2.1 Farkas' Lemma

#### Lemma (Farkas' Lemma for Inequalities)

The system  $Ax \leq b$  has a solution  $x$  if and only if there is no vector  $y$  satisfying  $y \geq 0$ ,  $y^T A = 0$ , and  $y^T b < 0$ .

**Proof.** Suppose  $Ax \leq b$  has a solution  $\bar{x}$  and suppose there exists a vector  $\bar{y} \geq 0$  satisfying  $\bar{y}^T A = 0$  and  $\bar{y}^T b < 0$ . Then we obtain the contradiction

$$0 > \bar{y}^T b \geq \bar{y}^T (A\bar{x}) = (\bar{y}^T A)\bar{x} = 0$$

Now suppose that  $Ax \leq b$  has no solution. If  $A$  has only one column, then the result is easy. Otherwise, apply Fourier-Motzkin elimination to obtain a system  $A'x' \leq b'$  with one less variable. Since  $A'x' \leq b'$  also has no solution, we can assume by induction that there exists a vector  $y' \geq 0$  satisfying  $y'^T A' = 0$  and  $y'^T b' < 0$ . Now since each inequality in  $A'x' \leq b'$  is the sum of positive multiples of inequalities in  $Ax \leq b$ , we can use  $y'$  to construct a vector  $y$  satisfying the conditions in the theorem.  $\square$

#### Lemma (Farkas' Lemma)

The system  $Ax = b$  has a nonnegative solution if and only if there is no vector  $y$  satisfying  $y^T A \geq 0$  and  $y^T b < 0$ .



**Proof.** Define

$$A' = \begin{bmatrix} A \\ -A \\ -I \end{bmatrix}, b' = \begin{bmatrix} b \\ -b \\ 0 \end{bmatrix}$$

Then  $Ax = b$  has a nonnegative solution  $x$  if and only if  $A'x' \leq b'$  has a solution  $x'$ . Applying Farkas' Lemma for Inequalities to  $A'x' \leq b'$  gives the result.  $\square$

### Corollary

Suppose the system  $Ax \leq b$  has at least one solution. Then every solution  $x$  of  $Ax \leq b$  satisfies  $c^T x \leq \delta$  if and only if there exists a vector  $y \geq 0$  such that  $y^T A = c$  and  $y^T b \leq \delta$ .

## 2.2 Duality

Consider the LP:

$$\begin{array}{ll} \max & c^T x \\ \text{s.t.} & Ax \leq b \end{array}$$

and dual LP

$$\begin{array}{ll} \min & y^T b \\ \text{s.t.} & y^T A = c^T \\ & y \geq 0 \end{array}$$

### Theorem (Weak Duality)

Let  $A$  be an  $m \times n$  matrix,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ . Suppose that  $\bar{x}$  is a feasible solution to  $Ax \leq b$  and  $\bar{y}$  is a feasible solution to  $y \geq 0, y^T A = c^T$ . Then

$$c^T \bar{x} \leq \bar{y}^T b$$

**Proof.**

$$c^T \bar{x} = (\bar{y}^T A) \bar{x} = \bar{y}^T (A \bar{x}) \leq \bar{y}^T b$$

$\square$

### Theorem (Duality Theorem)

Let  $A$  be an  $m \times n$  matrix,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ , then

$$\max\{c^T x : Ax \leq b\} = \min\{y^T b : y^T A = c^T, y \geq 0\}$$

provided that both sets are nonempty.

### Corollary

Let  $A$  be an  $m \times n$  matrix,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ , then

$$\max\{c^T x : Ax \leq b, x \geq 0\} = \min\{y^T b : y^T A \geq c^T\}$$

provided that both sets are nonempty.

## 2.3 Complementary Slackness

Consider the LP and dual LP

$$\max\{c^T x : Ax \leq b, x \geq 0\} = \min\{y^T b : y^T A \geq c^T, y \geq 0\}$$

### Definition: Complementary Slackness Conditions

Suppose  $\bar{x}, \bar{y}$  are feasible solutions to the primal and dual.

- If a component of  $\bar{x} > 0$ , then the corresponding inequality in  $y^T A \geq c^T$  is satisfied by  $\bar{y}$  with equality, i.e.

$$(\bar{y}^T A - c^T)\bar{x} = 0$$

- If a component of  $\bar{y} > 0$ , then the corresponding inequality in  $Ax \leq b$  is satisfied by  $\bar{x}$  with equality, i.e.

$$\bar{y}^T(b - A\bar{x}) = 0$$

### Theorem (Complementary Slackness Theorem)

Let  $x^*$  be a feasible solution of  $\max\{c^T x : Ax \leq b, x \geq 0\}$  and let  $y^*$  be a feasible solution of  $\min\{y^T b : y^T A \geq c^T, y \geq 0\}$ . Then the following are equivalent:

- $\bar{x}, \bar{y}$  are optimal solutions.
- $c^T \bar{x} = \bar{y}^T b$
- The complementary slackness conditions hold.

**Proof.** ((a)  $\iff$  (b)) By Duality Theorem.

((b)  $\iff$  (c)) By Weak Duality, we have  $c^T \bar{x} \leq \bar{y}^T A\bar{x} \leq \bar{y}^T b$ . So,

$$\begin{aligned} c^T \bar{x} = \bar{y}^T b &\iff \bar{y}^T A\bar{x} = c^T \bar{x} \text{ and } \bar{y}^T b = \bar{y}^T A\bar{x} \\ &\iff (\bar{y}^T A - c^T)\bar{x} = 0 \text{ and } \bar{y}^T(b - A\bar{x}) = 0 \end{aligned}$$

□

# Chapter 3

## Graph Theory

### 3.1 Undirected Graphs

**Definition: Graph**

A graph  $G = (V, E)$  is a set of vertices/nodes  $V$  and a set of edges  $E$ . We define  $n = |V|$  and  $m = |E|$ .

**Definition: Degree**

The degree of a vertex  $v$  of a graph  $G$  is the number of edges incident with  $v$ , denoted  $\deg_G(v)$ .

**Definition: Subgraph**

$H$  is a subgraph of  $G$  if  $E(H) \subseteq E(G)$  and  $V(H) \subseteq V(G)$ .

**Definition: Spanning Subgraph**

$H$  is spanning if  $V(H) = V(G)$ .

**Definition: Path**

A sequence  $P = v_0, e_1, v_1, \dots, e_k, v_k$  where  $v_0, \dots, v_k \in V(G)$ ,  $e_1, \dots, e_k \in E(G)$ , and  $e_i = v_{i-1}v_i$ .

We call  $P$  a  $v_0v_1$ -path. The length of  $P$  is the number of edges in  $P$ .

**Definition: Simple Path**

A path  $v_0, e_1, v_1, \dots, e_k, v_k$  where all  $v_i$  are distinct.

**Definition: Edge-Simple Path**

A path  $v_0, e_1, v_1, \dots, e_k, v_k$  where all  $e_i$  are distinct.

**Definition: Closed Path**

A path  $v_0, e_1, v_1, \dots, e_k, v_k$  where  $v_0 = v_k$ .

**Definition: Circuit/Cycle**

An edge-simple, closed path where  $v_0, \dots, v_{k-1}$  are distinct.

**Definition: Connected**

A graph is connected if every pair of vertices is joined by a path.

**Theorem**

A graph  $G$  is connected if and only if there is no set  $A \subseteq V$  where  $\emptyset \neq A \neq V$  with  $\delta(A) = \emptyset$ .

**Definition: Connected Component**

A maximal connected subgraph.

**Definition: Cut Vertex**

A vertex  $v$  of a connected graph  $G$  where  $G - v$  is not connected.

**Definition: Forest**

A graph with no circuits.

**Definition: Tree**

A connected forest.

**Definition: Cut**

Let  $R \subseteq V$ , then

$$\delta(R) = \{vw : vw \in E, v \in R, w \notin R\}$$

**Definition:  $rs$ -Cut**

A cut for which  $r \in R, s \notin R$ .

## 3.2 Directed Graphs

### Definition: Digraph

A digraph  $G = (V, E)$  is a set of vertices/nodes  $V$  and a set of edges  $E$ , sometimes called arcs, where each  $e \in E$  has two ends, one called the head  $h(e)$  and the other called the tail  $t(e)$ .

### Definition: Forward Arc

In a path  $v_0, e_1, v_1, \dots, e_k, v_k$ ,  $e_i \in P$  is called forward if  $t(e_i) = v_{i-1}$  and  $h(e_i) = v_i$ .

### Definition: Backward Arc

In a path  $v_0, e_1, v_1, \dots, e_k, v_k$ ,  $e_i \in P$  is called backward if  $t(e_i) = v_i$  and  $h(e_i) = v_{i-1}$ .

### Definition: Dipath

If all arcs in a path  $P$  are forward, then  $P$  is a dipath.

### Definition: Dicircuit

A dipath that is a circuit.

### Definition: Directed Spanning Tree

A directed spanning tree rooted at  $r$  is a spanning tree that contains a dipath from  $r$  to each  $v \in V$ .

# Chapter 4

## Complexity Classes

**Definition: Decision Problem**

A problem with a yes-no answer.

**Definition:  $\mathcal{P}$** 

Decision problems that can be solved in polynomial time.

**Definition:  $\mathcal{NP}$** 

Decision problems in which we can certify the answer is yes in polynomial time.

**Definition:  $\text{co-}\mathcal{NP}$** 

Decision problems in which we can certify the answer is no in polynomial time.

A good characterization means the problem is in  $\mathcal{NP} \cap \text{co-}\mathcal{NP}$ .

**Definition:  $\mathcal{NP}$ -Hard**

A problem  $X$  is  $\mathcal{NP}$ -hard if every other problem  $Y$  in  $\mathcal{NP}$  can be reduced to  $X$ .

S. Cook (1971) proved that the satisfiability problem (SAT) is  $\mathcal{NP}$ -hard. R. Karp (1972) used Cook's result to show 21 well-known combinatorial optimization problems are also  $\mathcal{NP}$ -hard.

To show that the traveling salesman problem (TSP) is  $\mathcal{NP}$ -hard, we show that any example of SAT can be formulated as a TSP, of size polynomial in the size of SAT. Then, since Cook shows SAT is  $\mathcal{NP}$ -hard, TSP is also  $\mathcal{NP}$ -hard.

## Part II

# Polyhedral Combinatorics

# Chapter 5

## Integrality of Polyhedra

### 5.1 Convex Hull

#### Definition: Convex Combination

$x = \lambda_1 v_1 + \cdots + \lambda_k v_k$  for some vectors  $v_1, \dots, v_k$  and nonnegative scalars  $\lambda_1, \dots, \lambda_k$  such that  $\lambda_1 + \cdots + \lambda_k = 1$ .

#### Definition: Convex Hull

The convex hull of a finite set  $S$ , denoted  $\text{conv.hull}(S)$ , is the set of all vectors that can be written as a convex combination of  $S$ .

It is also defined as the smallest convex set containing  $S$ .

#### Proposition

Let  $S \subseteq \mathbb{R}^n$  be a finite set and let  $w \in \mathbb{R}^n$ . Then

$$\max / \min \{w^T x : x \in S\} = \max / \min \{w^T x : x \in \text{conv.hull}(S)\}$$

#### Theorem (Minkowski)

If  $S$  is finite, then  $\text{conv.hull}(S)$  is a polyhedron.

$$\begin{aligned} \max \{w^T x : x \in S\} &= \max \{w^T x : x \in \text{conv.hull}(S)\} \\ &= \max \{w^T x : Ax \leq b\} \\ &= \min \{y^T b : y^T A = w^T, y \geq 0\} \end{aligned}$$

So we can use LP duality to attack combinatorial problems. If we understand  $Ax \leq b$ , then the problem is in  $\text{co-}\mathcal{NP}$ . Thus, if we have an algorithm to produce the inequalities in  $Ax \leq b$  (separation), then the problem is in  $\mathcal{P}$  (Ellipsoid method).



## 5.2 Polytopes

### Definition: Polyhedron

A set of the form  $\{x : Ax \leq b\}$ .

In combinatorial optimization, we typically have  $x \geq 0$  as a constraint, so we have polyhedra of the form  $\{x : Ax \leq b, x \geq 0\}$ .

### Definition: Polytope

A polyhedron  $P \subseteq \mathbb{R}^n$  is a polytope if there exists  $\ell, u \in \mathbb{R}^n$  such that  $\ell \leq x \leq u$  for all  $x \in P$ .

### Definition: Convex Set

Let  $P$  be a polyhedron,  $x_1, x_2 \in P$ , and  $0 \leq \lambda \leq 1$ . If  $\lambda x_1 + (1 - \lambda)x_2 \in P$ , then  $P$  is a convex set.

### Definition: Valid Inequality

An inequality  $w^T x \leq t$  is valid for a polyhedron  $P$  if  $P \subseteq \{x : w^T x \leq t\}$ .

### Definition: Hyperplane

The solution set of  $w^T x = t$  where  $w \neq 0$ .

### Definition: Supporting Hyperplane

With respect to a polyhedron  $P$ , a hyperplane is supporting if  $w^T x \leq t$  is valid for  $P$  and  $P \cap \{x : w^T x = t\} \neq \emptyset$ .

### Definition: Face

The intersection of a polyhedron with one of its supporting hyperplanes.

The null set and the polyhedron itself is a face.

### Definition: Proper Face

Faces which are not the null set or the polyhedron itself.

### Proposition

A nonempty set  $F \subseteq P = \{x : Ax \leq b\}$  is a face of  $P$  if and only if for some subsystem  $A^\circ x \leq b^\circ$  of  $Ax \leq b$ , we have  $F = \{x \in P : A^\circ x = b^\circ\}$ .

**Proof.** ( $\implies$ ) Suppose  $F$  is a face of  $P$ . Then there exists a valid inequality  $w^T x \leq t$  such that  $F = \{x \in P : w^T x = t\}$ .

Consider the LP problem  $\max\{w^T x : Ax \leq b\}$ . The set of optimal solutions is precisely  $F$ . Now let  $y^*$  be an optimal solution to the dual problem  $\min\{y^T b : y^T A = w, y \geq 0\}$  and let  $A^\circ x \leq b^\circ$  be those inequalities  $a_i^T x \leq b_i$  whose corresponding dual variable  $y_i^*$  is positive. By complementary slackness, we have  $F = \{x : Ax \leq b, A^\circ x = b^\circ\}$  as required.

( $\Leftarrow$ ) Conversely, if  $F = \{x \in P : A^\circ x = b^\circ\}$  for some subsystem  $A^\circ x \leq b^\circ$  of  $Ax \leq b$ , then add the inequalities  $A^\circ x \leq b^\circ$  to obtain an inequality  $w^T x \leq t$ . Every  $x \in F$  satisfies  $w^T x = t$  and every  $x \in P \setminus F$  satisfies  $w^T x < t$  as required.  $\square$

### Proposition

Let  $F$  be a minimal nonempty face of  $P = \{x : Ax \leq b\}$ . Then  $F = \{x : A^\circ x = b^\circ\}$  for some subsystem  $A^\circ x \leq b^\circ$  of  $Ax \leq b$ .  
Moreover, the rank of the matrix  $A^\circ$  is equal to the rank of  $A$ .

### Definition: Vertex/Extreme Point

A vector  $x \in P$  is called a vertex/extreme point if  $\{x\}$  is a face of  $P$ .  
Equivalently,  $x \in P$  is a vertex/extreme point if  $x$  cannot be written as  $\frac{1}{2}x_1 + \frac{1}{2}x_2$  for points  $x_1, x_2 \in P, x_1 \neq x_2$ .

Note: Not all polyhedra have vertices, but if  $P \subseteq \mathbb{R}_+^n$ , then  $P$  has vertices.

### LP Fact

If a polyhedron  $P$  has vertices, then the set of optimal LP solutions contains at least one vertex of  $P$ .  
Moreover, if all vertices of  $P$  are integral, then the LP always has an integral optimal solution.

### Definition: Pointed Polyhedron

A polyhedron  $P$  is pointed if it has at least one vertex.

$\{(x_1, x_2) \in \mathbb{R}^2 : x_1 \geq 0\}$  is a polyhedron with no vertex.

### Proposition

If a polyhedron  $P$  is pointed, then every minimal nonempty face of  $P$  is a vertex.

### Proposition

Let  $P = \{x : Ax \leq b\}$  and  $v \in P$ . Then  $v$  is a vertex of  $P$  if and only if  $v$  cannot be written as a convex combination of vectors in  $P \setminus \{v\}$ .

### Theorem

A polytope is equal to the convex hull of its vertices.

**Proof.** Let  $P$  be a nonempty polytope. Since  $P$  is bounded,  $P$  must be pointed. Let

$v_1, \dots, v_k$  be the vertices of  $P$ . Clearly,  $\text{conv.hull}(\{v_1, \dots, v_k\}) \subseteq P$ . So suppose there exists

$$u \in P \setminus \text{conv.hull}(\{v_1, \dots, v_k\})$$

Then by proposition, there exists an inequality  $w^T x \leq t$  that separates  $u$  from

$$\text{conv.hull}(\{v_1, \dots, v_k\})$$

Let  $t^* = \max\{w^T x : x \in P\}$  and consider the face  $F = \{x \in P : w^T x = t^*\}$ . Since  $u \in P$ , we have  $t^* > t$ . So  $F$  contains no vertex of  $P$ , a contradiction.  $\square$

### Theorem

A set  $P$  is a polytope if and only if there exists a finite set  $V$  such that  $P$  is the convex hull of  $V$ .

## 5.3 Integral Polytopes

### Definition: Rational Polyhedron

A polyhedron that can be defined by rational linear systems.

### Definition: Integral Polyhedron

A rational polyhedron where every nonempty face contains an integral vector.

### Definition: Pointed Integral Polyhedron

A pointed rational polyhedron is integral if and only if all its vertices are integral.

### Theorem

A rational polytope  $P$  is integral if and only if for all integral vectors  $w$ , the optimal value of  $\max\{w^T x : x \in P\}$  is an integer.

**Proof.** To prove sufficiency, suppose that for all integral vectors  $w$ , the optimal value of  $\max\{w^T x : x \in P\}$  is an integer. Let  $v = (v_1, \dots, v_n)^T$  be a vertex of  $P$  and let  $w$  be an integral vector such that  $v$  is the unique optimal solution to  $\max\{w^T x : x \in P\}$ . By multiplying  $w$  by a large positive integer if necessary, we may assume  $w^T v > w^T u + u_1 - v_1$  for all vertices  $u$  of  $P$  other than  $v$ . This implies that if we let  $\bar{w} = (w_1 + 1, w_2, \dots, w_n)^T$ , then  $v$  is an optimal solution to  $\max\{\bar{w}^T x : x \in P\}$ . So  $\bar{w}^T v = w^T v + v_1$ . But, by assumption,  $w^T v$  and  $\bar{w}^T v$  are integers. Thus,  $v_1$  is an integer. We can repeat this for each component of  $v$ , so  $v$  must be integral.  $\square$

## 5.4 Total Unimodularity

### Proposition

Let  $A$  be an integral, nonsingular,  $m \times n$  matrix. Then  $A^{-1}b$  is integral for every integral vector  $b \in \mathbb{R}^m$  if and only if  $\det(A) = 1$  or  $-1$ .

**Proof.** (  $\Leftarrow$  ) Suppose  $\det(A) = \pm 1$ . By Cramer's Rule, we know that  $A^{-1}$  is integral, which implies  $A^{-1}b$  is integral for every integral  $b$ .

(  $\Rightarrow$  ) Conversely, suppose  $A^{-1}b$  is integral for all integral vectors  $b$ . Then, in particular,  $A^{-1}e_i$  is integral for all  $i = 1, \dots, m$ . This means that  $A^{-1}$  is integral. So  $\det(A)$  and  $\det(A^{-1})$  are both integers. But,  $\det(A) \cdot \det(A^{-1}) = 1$ , this implies  $\det(A) = \pm 1$ .  $\square$

### Definition: Unimodular

A matrix  $A$  of full row rank is unimodular if  $A$  is integral and each basis of  $A$  has determinant  $\pm 1$ .

### Theorem (Veinott & Dantzig 1968)

Let  $A$  be an integral  $m \times n$  matrix of full row rank. Then the polyhedron defined by  $Ax = b, x \geq 0$  is integral for every integral vector  $b \in \mathbb{R}^m$  if and only if  $A$  is unimodular.

**Proof.** (  $\Leftarrow$  ) Suppose  $A$  is unimodular. Let  $b \in \mathbb{R}^m$  be an integral vector and let  $\bar{x}$  be a vertex of  $\{x : Ax = b, x \geq 0\}$ . The nonnegativity constraints implies the polyhedron has vertices. Then there are  $n$  linearly independent constraints satisfied by  $\bar{x}$  with inequality. It follows that the columns of  $A$  corresponding to the nonzero components of  $\bar{x}$  are linearly independent. Extending these columns to a basis  $B$  of  $A$ , we have the nonzero components of  $\bar{x}$  are contained in the integral vector  $B^{-1}b$ . So  $\bar{x}$  is integral.

(  $\Rightarrow$  ) Conversely, suppose  $\{x : Ax = b, x \geq 0\}$  is integral for all integral vectors  $b$ . Let  $B$  be a basis of  $A$  and let  $v$  be an integral vector in  $\mathbb{R}^m$ . By previous proposition, it suffices to show that  $B^{-1}v$  is integral. Let  $y$  be an integral vector such that  $y + B^{-1}v \geq 0$  and let  $b = B(y + B^{-1}v)$ . Note  $b$  is integral. Furthermore, by adding zero components to the vector  $y + B^{-1}v$ , we can obtain a vector  $z \in \mathbb{R}^n$  such that  $Az = b$ . Then,  $z$  is a vertex of  $\{x : Ax = b, x \geq 0\}$ , since  $z$  is a polyhedron and satisfies  $n$  linearly independent constraints with equality: the  $m$  equations  $Ax = b$  and the  $n - m$  equations  $x_i = 0$  for the columns  $i$  outside  $B$ . So  $z$  is integral, and thus,  $B^{-1}v$  is integral.  $\square$

### Definition: Totally Unimodular (TU)

A matrix is totally unimodular if all of its square submatrices have determinant 0, 1, or  $-1$ .

It is easy to see that  $A$  is totally unimodular if and only if  $\begin{bmatrix} A & I \end{bmatrix}$  is unimodular where  $I \in \mathbb{R}^{m \times m}$ .

### Theorem (Hoffman-Kruskal)

Let  $A$  be an  $m \times n$  integral matrix. Then the polyhedron defined by  $Ax \leq b, x \geq 0$  is integral for every integral vector  $b \in \mathbb{R}^m$  if and only if  $A$  is totally unimodular.

**Proof.** Applying the linear programming trick of adding slack variables, we have that for any integral  $b$ , the polyhedron  $\{x : Ax \leq b, x \geq 0\}$  is integral if and only if the polyhedron  $\{z : \begin{bmatrix} A & I \end{bmatrix} z = b, z \geq 0\}$  is integral. So the result follows from previous theorem.  $\square$

### Theorem

Let  $A$  be an  $m \times n$  totally unimodular matrix and let  $b \in \mathbb{R}^m$  be an integral vector. Then the polyhedron defined by  $Ax \leq b$  is integral.

**Proof.** Let  $F$  be a minimal face of  $\{x : Ax \leq b\}$ . Then, by proposition,  $F = \{x : A^\circ x = b^\circ\}$  for some subsystem  $A^\circ x \leq b^\circ$  of  $Ax \leq b$ , with  $A^\circ$  having full row rank. By reordering the columns, if necessary, we may write  $A^\circ$  as  $\begin{bmatrix} B & N \end{bmatrix}$  where  $B$  is a basis of  $A^\circ$ . It follows

$$\bar{x} = \begin{bmatrix} B^{-1}b^\circ \\ 0 \end{bmatrix}$$

is an integral vector in  $F$ .  $\square$

### Theorem

Let  $A$  be a  $0, \pm 1$  valued matrix where each column has at most one  $+1$  and at most  $-1$ . Then  $A$  is totally unimodular.

**Proof.** Let  $N$  be a  $k \times k$  submatrix of  $A$ . If  $k = 1$ , then  $\det(N)$  is either 0 or  $\pm 1$ . So we may suppose that  $k \geq 2$  and proceed by induction on  $k$ . If  $N$  has a column having at most one nonzero, then expanding the determinant along this column, we have that  $\det(N)$  is either 0 or  $\pm 1$ , by induction. On the other hand, if every column of  $N$  has both a  $+1$  and a  $-1$ , then the sum of the rows of  $N$  is 0 and hence  $\det(N) = 0$ .  $\square$

### Proposition

$A$  is totally unimodular if and only if  $A^T$  is totally unimodular.

## 5.5 Separation and Optimization

Recall that the plan for polyhedral combinatorics is to formulate the problem as optimizing over a finite set of vectors  $S$ , find a linear description of  $\text{conv.hull}(S)$ , and apply the Duality Theorem of Linear Programming. This gives us a min-max relation for the combinatorial problem.

### Separation Problem

Given a bounded rational polyhedron  $P \subseteq \mathbb{R}^n$  and a rational vector  $v \in \mathbb{R}^n$ , either conclude that  $v \in P$  or, if not, find a rational vector  $w \in \mathbb{R}^n$  such that  $w^T x < w^T v$  for all  $x \in P$ .

### Optimization Problem

Given a bounded rational polyhedron  $P \subseteq \mathbb{R}^n$  and a rational objective vector  $w \in \mathbb{R}^n$ , either find  $x^* \in P$  that maximizes  $w^T x$  over all  $x \in P$  or conclude that  $P$  is empty.

### Definition: Classes of Polyhedra

$\mathcal{P} = \{P_t : t \in \mathcal{O}\}$  where  $\mathcal{O}$  is some collection of objects and for each  $t \in \mathcal{O}$ ,  $P_t$  is a bounded rational polyhedron.

E.g.  $\mathcal{O}$  is the collection of all graphs and  $P_t$  is the perfect matching polytope for the graph  $t$ .

### Definition: Proper Class

For each object  $t \in \mathcal{P}$ , we can compute in polynomial time (with respect to size of  $t$ ) positive integers  $n_t$  and  $s_t$  such that  $P_t \subseteq \mathbb{R}^{n_t}$ . and such that  $P_t$  can be described by a linear system where each inequality has size at most  $s_t$ .

### Definition: Polynomially Solvable

A separation/optimization problem is polynomially solvable over the class  $\mathcal{P}$  if there exists a polynomial time algorithm to solve the problem.

### Theorem (Separation $\equiv$ Optimization)

For any proper class of polyhedra, the optimization problem is polynomially solvable if and only if the separation problem is polynomially solvable.

## 5.6 Total Dual Integrality

### Definition: Totally Dual Integral

A rational linear system  $Ax \leq b$  is totally dual integral if the minimum of

$$\max\{w^T x : Ax \leq b\} = \min\{y^T b : y^T A = w^T, y \geq 0\}$$

can be achieved by an integral vector  $y$  for each integral  $w$  for which the optima exist.

**Theorem (Hoffman 1974)**

Let  $Ax \leq b$  be a totally dual integral system such that  $P = \{x : Ax \leq b\}$  is a rational polytope and  $b$  is integral. Then  $P$  is an integral polytope.

**Proof.** Since  $b$  is integral, the duality equation implies  $\max\{w^T x : x \in P\}$  is an integer for all integral vectors  $w$ . Thus, by theorem for integral polytopes,  $P$  is integral.  $\square$

**Theorem**

Let  $P$  be a rational polyhedron. Then there exists a totally dual integral system  $Ax \leq b$ , with  $A$  integral, such that  $P = \{x : Ax \leq b\}$ . Furthermore, if  $P$  is a integral polyhedron, then  $b$  can be chosen to be integral.

## Part III

# Optimal Trees and Paths



# Chapter 6

## Minimum Spanning Trees

### 6.1 Problem

**Definition: Spanning Tree**

A subgraph  $T \subseteq G$  where  $V(T) = V(G)$ ,  $T$  is connected, and  $T$  is acyclic.

**Lemma**

An edge  $e = uv$  of  $G$  is an edge of a circuit of  $G$  if and only if there is a path in  $G \setminus e$  from  $u$  to  $v$ .

**Minimum Spanning Tree Problem (MST)**

Given a connected graph  $G$  and a real cost  $c_e$  for each  $e \in E$ , find a minimum cost spanning tree of  $G$ .

**Lemma**

A spanning connected subgraph of  $G$  is a spanning tree if and only if it has exactly  $n - 1$  edges.

### 6.2 Kruskal's Algorithm

**Theorem**

Kruskal's algorithm finds a MST.

This is a polynomial time algorithm and is very fast in practice for sparse graphs. We can maintain  $F$  with a union-find data structure.

---

**Algorithm 1** Kruskal's Algorithm (Minimum Spanning Tree)

---

```
1: Sort  $E$  to  $\{e_1, \dots, e_m\}$  so that  $c_{e_1} \leq \dots \leq c_{e_m}$ 
2:  $F = \emptyset, H = (V, F)$ 
3: for  $i = 1$  to  $m$  do
4:   if ends of  $e_i$  are in different components of  $H$  then
5:      $F \leftarrow F \cup \{e_i\}$ 
6: return  $H$ 
```

---

## 6.3 Linear Programming

**Definition:**  $\kappa : E \rightarrow \mathbb{N}$

For  $A \subseteq E$ ,  $\kappa(A)$  is the number of components in the subgraph  $(V, A)$  of  $G$ .

The maximum number of tree edges in  $A$  is  $|V| - \kappa(A)$ .

We can formulate the MST problem as an ILP.

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \sum (x_e : e \in A) \leq |V| - \kappa(A), \quad \forall A \subsetneq E \\ & \sum (x_e : e \in E) = |V| - 1 \\ & x_e \in \{0, 1\}, \quad \forall e \in E \end{aligned}$$

We can relax the integer program to get the following linear program.

**Definition: MST LP**

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x(A) \leq |V| - \kappa(A), \quad \forall A \subsetneq E \\ & x(E) = |V| - 1 \\ & x_e \geq 0, \quad \forall e \in E \end{aligned}$$

We replace the minimization with a maximization in the primal to write the dual.

**Definition: MST Dual LP**

$$\begin{aligned} \min \quad & \sum ((|V| - \kappa(A))y_A : A \subseteq E) \\ \text{s.t.} \quad & \sum (y_A : e \in A) \geq -c_e, \quad \forall e \in E \\ & y_A \geq 0, \quad \forall A \subsetneq E \end{aligned}$$

### 6.3.1 Complementary Slackness Conditions

Let  $T$  be a tree found by Kruskal's algorithm. Define the characteristic vector of  $T$

$$x_e^0 = \begin{cases} 1 & \text{if } e \in E(T) \\ 0 & \text{if } e \notin E(T) \end{cases}$$

#### Definition: MST Complementary Slackness Conditions

- (i) For all  $e \in E$ , if  $x_e^0 > 0$ , then  $\sum(y_A^0 : e \in A) = -c_e$ .
- (ii) For all  $A \subsetneq E$ , if  $y_A^0 > 0$ , then  $\sum(x_e^0 : e \in A) = |V| - \kappa(A)$ .

#### Theorem (Edmonds 1971)

Let  $x^0$  be the characteristic vector of an MST with respect to costs  $c_e$ . Then  $x^0$  is an optimal solution to the MST LP.

**Proof.** We show that  $x^0$  is optimal for the LP and  $x^0$  is the characteristic vector generated by Kruskal's algorithm.

Let  $e_1, \dots, e_m$  be the order in which Kruskal's algorithm considers the edges. Let  $R_i = \{e_1, \dots, e_i\}$  for  $1 \leq i \leq m$ . Let  $y^0$  be the dual solution.

- $y_A^0 = 0$  if  $A$  is not one of the  $R_i$ 's.
- $y_{R_i}^0 = c_{e_{i+1}} - c_{e_i}$  for  $1 \leq i \leq m-1$ .
- $y_{R_m}^0 = -c_{e_m}$

It follows from the ordering of the edges,  $y_A^0 \geq 0$  for  $A \neq E$ . Now consider the first constraint, then where  $e = e_i$ , we have

$$\sum(y_A^0 : e \in A) = \sum_{j=i}^m y_{R_j}^0 = \sum_{j=i}^{m-1} (c_{e_{j+1}} - c_{e_j}) - c_{e_m} = -c_{e_i} = -c_e$$

All of the inequalities hold with equality. So  $y^0$  is a feasible dual solution and complementary slackness condition (i) holds.

Now suppose  $y_A^0 > 0$  for some  $A \subsetneq E$ . Thus,  $A = R_i$  for some  $i$ . Consider the constraint

$$\sum(x_e^0 : e \in R_i) \leq |V| - \kappa(R_i)$$

If this does not hold with equality, then there is some edge of  $R_i$  having ends in two different components of  $(V, R_i \cap T)$  and this would have been added to  $T$  by Kruskal's algorithm. So  $(x^0, y^0)$  satisfy the complementary slackness conditions, which means they are optimal solutions to their LPs. Therefore,  $T$  is an MST.  $\square$

## 6.4 Spanning Tree Polytope

### Definition: Spanning Tree Polytope

$\text{conv.hull}\{x^H : H \text{ is a spanning tree}\}$  where

$$x_e^H = \begin{cases} 1 & \text{if } e \in E(H) \\ 0 & \text{if } e \notin E(H) \end{cases}$$

### Theorem

The spanning tree polytope is the solution set to the following linear system:

$$\begin{aligned} \sum(x_e : e \in A) &\leq |V| - \kappa(A), \quad \forall A \subsetneq E \\ \sum(x_e : e \in E) &= |V| - 1 \\ x_e &\geq 0, \quad \forall e \in E \end{aligned}$$

**Proof.** Let  $P$  be the solution set of the linear system. We showed that for any edge costs  $(c_e : e \in E)$ , the LP

$$\max\{\sum(-c_e x_e : e \in E) : x \in P\}$$

has an integral optimal solution. So every vertex of  $P$  is integral.  $\square$

# Chapter 7

## Shortest Paths

### Shortest Path Problem

Given a digraph  $G$ , a vertex  $r \in V$ , and a real cost vector  $(c_e : e \in E)$ , find for each  $v \in V$ , a minimum-cost dipath from  $r$  to  $v$ .

Note: You can provide a solution to the shortest path problem for  $r$  by giving a directed spanning tree rooted at  $r$ .

**Proof.** For each  $v \in V \setminus \{r\}$ , all shortest paths have at most one arc having head  $v_j$ , since the only such arc we need is the last arc of one min-cost  $rv$ -dipath.

So the union of the arc sets of all the shortest paths has exactly  $|V| - 1$  arcs and thus is a tree.  $\square$

### Important Case

If  $c_e \geq 0$  for all  $e \in E$ , then this problem is handled by Dijkstra's algorithm, which starts at  $r$  and grows the tree vertex by vertex.

However, the Hamiltonian dipath problem (does  $G$  have a simple dipath  $P$  with  $V(P) = V(G)$ ) is  $\mathcal{NP}$ -hard.

When  $G$  has negative-cost dicircuits, this is a problem, since there is no shortest path as we can loop around the dicircuit an infinite amount of times. There do exist polynomial time algorithms that either finds a shortest path or detect a negative-cost dicircuit.

### Definition: Feasible Potential

$y = (y_v : v \in V)$  is a feasible potential if it satisfies  $y_v + c_{vw} \geq y_w$  for all  $vw \in E$ .

### Proposition

Let  $y$  be a feasible potential and let  $P$  be an  $rs$ -dipath. Then  $c(P) \geq y_s - y_r$ .

**Proof.** Suppose that  $P$  is  $v_0, e_1, v_1, \dots, e_k, v_k$  where  $v_0 = r$  and  $v_k = s$ . Then

$$c(P) = \sum_{i=1}^k c_{e_i} \geq \sum_{i=1}^k (y_{v_i} - y_{v_{i-1}}) = y_{v_k} - y_{v_0} = y_s - y_r$$

□

So a potential  $y$  provides a stopping rule. A dipath  $P$  and a potential  $y$  with  $c(P) = y_s - y_r$  implies  $P$  is optimal.

## 7.1 Ford's Algorithm

### Definition: Incorrect

Given vertex values  $(y_v : v \in V)$ , the edge  $vw$  is incorrect if  $y_v + c_{vw} < y_w$ .

To correct  $vw$ , we set  $y_w = y_v + c_{vw}$  and  $\text{predecessor}(w) = v$ .

---

### Algorithm 2 Ford's Algorithm (Shortest Path)

---

- 1:  $y_r = 0, y_v = \infty$  for all  $v \in V \setminus \{r\}$
  - 2:  $\text{predecessor}(r) = \emptyset, \text{predecessor}(v) = -1$  for all  $v \in V \setminus \{r\}$
  - 3: **while**  $y$  is not a feasible potential **do**
  - 4:     Find an incorrect arc  $vw$  and correct  $vw$
- 

### Theorem

If there are no negative-cost dicircuits, then Ford's algorithm terminates in a finite number of steps.

At termination, for each  $v \in V$ , the predecessors define a shortest  $rv$ -dipath of cost  $y_v$ .

Specialized versions like Ford-Bellman run in polynomial time.

## 7.2 Linear Programming

### Definition: Shortest Path LP

$$\begin{array}{ll} \max & y_s - y_r \\ \text{s.t.} & y_w - y_v \leq c_{vw}, \quad \forall vw \in E \end{array}$$

**Definition: Shortest Path Dual LP**

$$\begin{aligned}
& \min \quad \sum (c_e x_e : e \in E) \\
& \text{s.t.} \quad \sum (x_{wv} : wv \in E) - \sum (x_{vw} : vw \in E) = \begin{cases} 0 & \text{if } v \in V \setminus \{r, s\} \\ -1 & \text{if } v = r \\ 1 & \text{if } v = s \end{cases} \\
& \quad \quad x_{vw} \geq 0, \forall vw \in E
\end{aligned}$$

Any  $rs$ -dipath is a solution to the dual LP, so if the dual LP has an optimal solution, then it has an optimal solution that is an  $rs$ -dipath.

The constraint matrix for the LP is totally unimodular.

**Theorem**

Let  $G$  be a digraph,  $r, s \in V$ , and  $c \in \mathbb{R}^E$ . If there exists a minimum-cost dipath from  $r$  to  $v$  for every  $v \in V$ , then

$$\min\{c(P) : P \text{ an } rs\text{-dipath}\} = \max\{y_s : y \text{ a feasible potential}\}$$

The vertices of the polyhedron defined by the dual LP constraints are the vectors  $x^P$  of simple dipaths.

$$x_e^P = \begin{cases} 1 & \text{if } e \in E(P) \\ 0 & \text{if } e \notin E(P) \end{cases}$$

Note: This is *not* the convex hull of simple dipaths.

Since the matrix is totally unimodular, we could add  $x_{vw} \leq 1$  for all  $vw \in E$ , but this will not give simple dipaths.

# **Part IV**

## **Network Flows**



# Chapter 8

## Maximum Flow

### 8.1 Problem

**Definition: Net Flow/Excess**

$$f_x(v) = x(\delta(\bar{v})) - x(\delta(v)) = \sum(x_{wv} : w \in V, wv \in E) - \sum(x_{vw} : w \in V, vw \in E)$$

**Definition:  $rs$ -Flow**

A vector  $x$  that satisfies  $f_x(v) = 0$  for all  $v \in V$ .

**Definition: Value of  $rs$ -Flow**

$$f_x(s)$$

**Maximum Flow Problem**

Given a digraph  $G = (V, E)$ , with source  $r$  and sink  $s$ , find an  $rs$ -flow of maximum value.

### 8.2 Augmenting Path Algorithm

**Definition: Augmenting Path**

An  $rs$ -path  $P$  is  $x$ -augmenting if for all forward arcs  $e$  we have  $x_e < u_e$ , and for all reverse arcs  $e$  we have  $x_e > 0$ .

Given a flow  $x = (x_e : e \in E)$  and augmenting path  $P$ , we can augment flow  $x$  by the largest  $\varepsilon$ . There is some forward arc with  $x_e + \varepsilon = u_e$  or some reverse arc has  $x_e - \varepsilon = 0$ . The value  $\varepsilon$  is called the  $x$ -width of  $P$ .

---

**Algorithm 3** Ford-Fulkerson Algorithm (Maximum Flow/Minimum Cut)

---

```
1:  $x = 0$ 
2: while there is an  $x$ -augmenting path  $P$  do
3:    $\varepsilon_1 = \min(u_e - x_e : e \text{ forward in } P)$ 
4:    $\varepsilon_2 = \min(x_e : e \text{ reverse in } P)$ 
5:    $\varepsilon = \min(\varepsilon_1, \varepsilon_2)$  //  $x$ -width of  $P$ 
6:   if  $\varepsilon = \infty$  then
7:     No maximum flow
8: return  $x$  is maximum flow, set  $R$  of vertices reachable by an  $x$ -augmenting path from  $r$ 
    is minimum cut
```

---

**Definition: Auxiliary Digraph**

$G(x)$ , depending on  $G, u, x$ , where  $V(G(x)) = V$  and  $vw \in E(G(x))$  if and only if  $vw \in E$  and  $x_{vw} < u_{vw}$  or  $wv \in E$  and  $x_{wv} > 0$ .

$rs$ -dipaths in  $G(x)$  corresponding to  $x$ -augmenting paths in  $G$ . Each iteration of Ford-Fulkerson can be performed in  $O(m)$  time, using breadth-first search.

**Theorem**

If  $u$  is integral and the maximum flow value is  $K < \infty$ , then the maximum flow algorithm terminates after at most  $K$  augmentations.

### 8.2.1 Shortest Augmenting Paths

**Theorem (Dinits 1970, Edmonds & Karp 1972)**

If each augmentation of the augmenting path algorithm on a shortest augmenting path, then there are at most  $nm$  augmentations.

**Corollary**

The augmenting path algorithm with breadth-first search solves the maximum flow problem in time  $O(nm^2)$ .

Let  $d_x(v, w)$  be the least length of a  $vw$ -dipath in  $G(x)$ .  $d_x(v, w) = \infty$  if no  $vw$ -dipath exists.

Consider a typical augmentation from flow  $x$  to flow  $x'$  determined by the augmenting path  $P$  having vertex-sequence  $v_0, \dots, v_k$ .

**Lemma**

For each  $v \in V$ ,  $d_{x'}(r, v) \geq d_x(r, v)$  and  $d_{x'}(v, s) \geq d_x(v, s)$ .

**Proof.** Suppose that there exists a vertex  $v$  such that  $d_{x'}(r, v) < d_x(r, v)$  and choose such  $v$  so that  $d_{x'}(r, v)$  is as small as possible. Clearly,  $d_{x'}(r, v) > 0$ . Let  $P'$  be a  $rv$ -dipath in  $G(x')$

of length  $d_{x'}(r, v)$  and let  $w$  be the second-last vertex of  $P'$ . Then

$$d_x(r, v) > d_{x'}(r, v) = d_{x'}(r, w) + 1 \geq d_x(r, w) + 1$$

It follows that  $wv$  is an arc of  $G(x')$ , but not of  $G(x)$ , otherwise  $d_x(r, v) \leq d_x(r, w) + 1$ , so  $w = v_i$  and  $v = v_{i-1}$  for some  $i$ . But, this implies that  $i - 1 > i + 1$ , a contradiction. The second statement is similar.  $\square$

**Definition:**  $\tilde{E}(x)$

$$\tilde{E}(x) = \{e \in E : e \text{ is an arc of a shortest } x\text{-augmenting path}\}$$

**Lemma**

If  $d_{x'}(r, s) = d_x(r, s)$ , then  $\tilde{E}(x') \subsetneq \tilde{E}(x)$ .

**Proof.** Let  $k = d_x(r, s)$  and suppose that  $e \in \tilde{E}(x')$ . Then  $e$  induces an arc  $vw$  of  $G(x')$  and  $d_{x'}(r, v) = i - 1$ ,  $d_{x'}(ws) = k - i$  for some  $i$ . Therefore,  $d_x(r, v) + d_x(w, s) \leq k - 1$  by previous lemma. Now suppose that  $e \notin \tilde{E}(x)$ , then  $x_e \neq x'_e$ , so  $e$  is an arc of  $P$ , a contradiction. This proves  $\tilde{E}(x') \subseteq \tilde{E}(x)$ .

There is an arc  $e$  of  $P$  such that  $e$  is forward and  $x'_e = u_e$  or  $e$  is reverse and  $x'_e = 0$ . Therefore, any  $x'$ -augmenting path using  $e$  must use it in the opposite direction from  $P$ , so its length, for some  $i$ , will be at least  $i + k - i + 1 + 1 = k + 23$ , so  $e \notin \tilde{E}(x')$ .  $\square$

**Proof.** (Dinitz, Edmonds, Karp) It follows from previous lemma that there can be at most  $m$  augmentations per stage. Since there are at most  $n - 1$  stages, there are at most  $nm$  augmentations in all.  $\square$

## 8.3 Linear Programming

**Definition: Maximum Flow LP**

$$\begin{aligned} \max \quad & f_x(s) \\ \text{s.t.} \quad & f_x(v) = 0, \forall v \in V \setminus \{r, s\} \\ & 0 \leq x_e \leq u_e, \forall e \in E \end{aligned}$$

We give a different LP approach to this problem.

**Definition: Minimum Cut LP**

$$\begin{aligned} \min \quad & \sum (u_e y_e : e \in E) \\ \text{s.t.} \quad & \sum (y_e : e \in E(P)) \geq 1, \forall \text{ } rs\text{-simple dipaths } P \\ & y_e \geq 0, \forall e \in E \end{aligned}$$

Every  $rs$ -cut  $\delta(R)$  gives a feasible solution

$$y_e^R = \begin{cases} 1 & \text{if } e \in \delta(R) \\ 0 & \text{if } e \notin \delta(R) \end{cases}$$

**Definition: Maximum Flow LP (Dual Minimum Cut LP)**

$$\begin{aligned} \max \quad & \sum (w_P : P \text{ a simple } rs\text{-dipath}) \\ \text{s.t.} \quad & \sum (w_P : e \in E(P)) \leq u_e, \forall e \in E \\ & w_P \geq 0, \forall \text{ simple dipaths } P \end{aligned}$$

Let  $x$  be a max flow. We want to find a simple  $rs$ -dipath  $P$  such that  $x_e > 0$  for each  $e \in E(P)$ . Set  $w_P = \min\{x_e : e \in E(P)\}$  and let  $x_e = x_e - w_P$  for all  $e \in E(P)$ . We repeat until  $\sum (x_{rv} : rv \in E) = 0$ .

$\sum (w_P : P \text{ a simple } rs\text{-dipath})$  is equal to the original value of the flow. Therefore, the max flow equals the min cut which implies the two LPs have integral optimal solutions if  $u_e$  is integral for all  $e \in E$ .

**Proposition**

There exists a family  $(P_1, \dots, P_k)$  of  $rs$ -dipaths such that  $|\{i : e \in P_i\}| \leq u_e$  for all  $e \in E$  if and only if there exists an integral feasible  $rs$ -flow of value  $k$ .

**Proof.** ( $\implies$ ) We have seen family of dipaths determines a corresponding flow.

( $\impliedby$ ) Let  $x$  be a flow. We assume that  $x$  is acyclic, that is, there is no dicircuit  $C$ , each of whose arcs  $e$  has  $x_e > 0$ . If a dicircuit does exist, we can decrease  $x_e$  by 1 on all arcs of  $C$ . The new  $x$  remains feasible of value  $k$ .

If  $k \geq 1$ , we can find an arc  $vs$  with  $x_{vs} \geq 1$ . Then, if  $v \neq r$ , it follows that there is an arc  $wv$  with  $x_{wv} \geq 1$  by the constraint  $f_x(v) = 0$ . If  $w \neq r$ , then the argument can be repeated producing distinct vertices, since  $x$  is acyclic, so we get a simple  $rs$ -dipath  $P_k$  on each arc  $e$  with  $x_e \geq 1$ . We can decrease  $x_e$  by 1 for each  $e \in P_k$ . The new  $x$  is an integral feasible flow of value  $k - 1$ , and the process is repeated.  $\square$

**Definition: Path Flow**

A vector  $x \in \mathbb{R}^E$  such that for some  $rs$ -dipath  $P$  and some  $\alpha \in \mathbb{R}$ ,  $x_e = \alpha$  for each  $e \in P$  and  $x_e = 0$  for every other arc of  $G$ .

**Definition: Circuit Flow**

A vector  $x \in \mathbb{R}^E$  such that for some  $rs$ -dicircuit  $C$  and some  $\alpha \in \mathbb{R}$ ,  $x_e = \alpha$  for each  $e \in C$  and  $x_e = 0$  for every other arc of  $G$ .

### Proposition

Every  $rs$ -flow of nonnegative value is the sum of at most  $m$  flows, each of which is a path flow or a circuit flow.

### Proposition

For any  $rs$ -cut  $\delta(R)$  and any  $rs$ -flow  $x$ , we have

$$f_x(s) = x(\delta(R)) - x(\delta(\bar{R}))$$

**Proof.** We add the equations  $f_x(v) = 0$  for all  $v \in \bar{R} \setminus \{s\}$  as well as the identity  $f_x(s) = f_x(s)$ . The right hand side sums to  $f_x(s)$ .

For any arc  $vw$  with  $v, w \in R$ ,  $x_{vw}$  occurs in none of the equations, so it does not occur in the sum. If  $v, w \in \bar{R}$ , then  $x_{vw}$  occurs in the equation for  $v$  with a coefficient of  $-1$ , and in the equation for  $w$  with a coefficient of  $+1$ , so it has a coefficient of  $0$  in the sum. If  $v \in R, w \notin R$ , then  $x_{vw}$  occurs in the equation for  $w$  with a coefficient of  $1$ , and so has coefficient  $1$  in the sum. If  $v \notin R, w \in R$ , then  $x_{vw}$  occurs in the sum with a coefficient of  $-1$ . So, the left hand side sums to  $x(\delta(R)) - x(\delta(\bar{R}))$ , as required.  $\square$

### Corollary

For any feasible  $rs$ -flow  $x$  and any  $rs$ -cut  $\delta(R)$ ,

$$f_x(s) \leq u(\delta(R))$$

**Proof.** Using previous proposition, since  $x(\delta(R)) \leq u(\delta(R))$  and  $x(\delta(\bar{R})) \geq 0$ .  $\square$

### Theorem (Max-Flow Min-Cut)

If there is a maximum  $rs$ -flow, then

$$\max\{f_x(s) : x \text{ is a feasible } rs\text{-flow}\} = \min\{u(\delta(R)) : \delta(R) \text{ is an } rs\text{-cut}\}$$

**Proof.** By previous corollary, we need only show that there exists a feasible flow  $x$  and a cut  $\delta(R)$  such that  $f_x(s) = u(\delta(R))$ . Let  $x$  be a flow of maximum value. Let  $R = \{v \in V : \text{there exists an } x\text{-augmenting } rv\text{-path}\}$ . Clearly  $r \in R$  and  $s \notin R$ , since there can be no  $x$ -augmenting path.

For every arc  $vw \in \delta(R)$ , we must have  $x_{vw} = u_{vw}$ , since otherwise adding  $vw$  to the  $x$ -augmenting  $rv$ -path would yield such a path to  $w$ , but  $w \notin R$ . Similar, for every arc  $vw \in \delta(\bar{R})$ , we have  $x_{vw} = 0$ . Then by proposition,  $f_x(s) = x(\delta(R)) - x(\delta(\bar{R})) = u(\delta(R))$ .  $\square$

### Theorem

A feasible flow  $x$  is maximum if and only if there is not  $x$ -augmenting path.

**Proof.** ( $\implies$ ) If  $x$  is maximum, there is no  $x$ -augmenting path.

(  $\Leftarrow$  ) If there is no  $x$ -augmenting path, then the construction of the proof of Max-Flow Min-Cut yields a cut  $\delta(R)$  with  $f_x(s) = u(\delta(R))$ , so  $x$  is maximum, by corollary.  $\square$

### Theorem

If  $u$  is integral and there exists a maximum flow, then there exists a maximum flow that is integral.

**Proof.** Choose an integral flow  $x$  of maximum value. If there is an  $x$ -augmenting path, then since  $x$  and  $u$  are integral, the new flow can be chosen integral, contradicting the choice of  $x$ . Hence there is no  $x$ -augmenting path, so  $x$  is a maximum flow, by previous theorem.  $\square$

### Corollary

If  $x$  is a feasible  $rs$ -flow and  $\delta(R)$  is an  $rs$ -cut, then  $x$  is maximum and  $\delta(R)$  is minimum if and only if  $x_e = u_e$  for all  $e \in \delta(R)$  and  $x_e = 0$  for all  $e \in \delta(\overline{R})$ .

**Proof.** Combine Max-Flow Min-Cut theorem with the proof of corollary.  $\square$

# Part V

## Matchings

# Chapter 9

## Matchings

**Definition: Matching**

A set  $M \subseteq E$  such that no vertex of  $G$  is incident with more than one edge in  $M$ .

**Definition:  $M$ -Covered**

A vertex  $v$  is covered by  $M$  if some edge of  $M$  is incident with  $v$ .

**Definition:  $M$ -Exposed**

A vertex  $v$  is exposed if  $v$  is not  $M$ -covered.

The number of vertices covered by  $M$  is  $2|M|$  and number of  $M$ -exposed vertices is  $|V| - 2|M|$ .

**Definition: Maximum Matching**

A matching of maximum cardinality, denoted  $\nu(G)$ .

**Definition: Deficiency**

The minimum number of exposed vertices for any matching of  $G$ , denoted by  $\text{def}(G)$ .

Note  $\text{def}(G) = |V| - 2\nu(G)$ .

**Definition: Perfect Matching**

A matching that covers all vertices.



## 9.1 Bipartite Matching

### Definition: Bipartite

$G = (V, E)$  is bipartite if  $V = V_1 \cup V_2$ , where  $V_1, V_2$  disjoint and every edge has one end in  $V_1$  and the other end in  $V_2$ .

### Definition: Vertex Cover

A set  $C \subseteq V$  such that every edge has at least one in  $C$ .

### Lemma

If  $M$  is a matching and  $C$  is a cover, then  $|M| \leq |C|$ .

**Proof.** Every  $e \in M$  has at least one end in  $C$ . No vertex in  $C$  meets more than one edge in  $M$ .  $\square$

### Definition: Minimum Cover

A cover of minimum cardinality, denoted  $\tau(G)$ .

### Theorem (König)

If  $G$  is bipartite,  $\nu(G) = \tau(G)$ .

**Proof.** We note that  $\nu(G) \leq \nu^*(G)$  and  $\tau(G) \geq \tau^*(G)$ . By using LP duality and the matching LP (*Matching LP*), we show that  $\nu(G) = \nu^*(G)$ . We also have the matching LP in the form of  $Mx^+ = (1, \dots, 1)^T$ . Since  $M$  is totally unimodular, then  $M^T$  is also totally unimodular. So the dual LP has all integral vertices, implying  $\tau(G) = \tau^*(G)$ . So,

$$\nu(G) = \nu^*(G) = \tau^*(G) = \tau(G)$$

$\square$

## 9.2 Alternating Paths

### Definition: $M$ -Alternating

A path  $P$  is  $M$ -alternating if its edges are alternately in and not in  $M$ .

### Definition: $M$ -Augmenting

An  $M$ -alternating path  $P$  is  $M$ -augmenting if the ends of  $P$  are distinct and are both  $M$ -exposed.

**Definition: Symmetric Difference**

For sets  $S$  and  $T$ , let  $S\Delta T$  denote the symmetric difference, which is defined as

$$S\Delta T = (S \cup T) \setminus (S \cap T)$$

Let a path  $P$  be an  $M$ -augmenting path. Then we can obtain a larger matching  $M' = M\Delta E(P)$  with  $|M'| = |M| + 1$ .

**Theorem (Petersen 1891, Berge 1957)**

A matching  $M$  in a graph  $G$  is maximum if and only if there is no  $M$ -augmenting path.

**Proof.** ( $\implies$ ) Suppose there exists an  $M$ -augmenting path  $P$  joining  $v$  and  $w$ . Then  $N = M\Delta E(P)$  is a matching that covers all vertices covered by  $M$ , plus  $v$  and  $w$ . So,  $M$  is not maximum.

( $\impliedby$ ) Conversely, suppose that  $M$  is not maximum and some other matching  $N$  satisfies  $|N| > |M|$ . Let  $J = N\Delta M$ . Each vertex of  $G$  is incident with at most two edges of  $J$ , so  $J$  is the edge set of some vertex disjoint paths and circuits of  $G$ . For each such path or circuit, the edges alternately belong to  $M$  or  $N$ . Therefore, all circuits are even and contain the same number of edges of  $M$  and  $N$ . Since  $|N| > |M|$ , there must be at least one path with more edges of  $N$  than  $M$ . This path is an  $M$ -augmenting path.  $\square$

## 9.3 Matching LP

**Definition: Matching LP**

$P$  is the set of solutions to

$$\begin{aligned} x(\delta(v)) &\leq 1, \forall v \in V \\ x_e &\geq 0, \forall e \in E \end{aligned}$$

Let  $\bar{x}$  be a vertex of  $P$ . We show that  $\bar{x}$  is integral, which implies that  $M = \{e \in E : \bar{x}_e = 1\}$  is a matching and  $\nu(G) = \nu^*(G)$ .

Recall that for a polyhedron  $P = \{x : Ax \leq b\} \subseteq \mathbb{R}^n$ ,  $\bar{x} \in P$  is a vertex if and only if  $\bar{x}$  is the unique solution to  $A'x = b'$  for some subset of  $n$  inequalities  $A'x \leq b'$  from  $Ax \leq b$ .

For our matching  $P$ , let  $E^+ := \{e : \bar{x}_e > 0\}$  and  $E^0 := \{e : \bar{x}_e = 0\}$ . We write  $\bar{x} = (\bar{x}^+, \bar{x}^0)$  split by  $(E^+, E^0)$ .

Since  $\bar{x}$  is a vertex, there exists  $V^+ \subseteq V$  such that  $\bar{x}$  is the unique solution to

$$\begin{aligned} \sum (x_e : e \in \delta(v) \cap E^+) &= 1, \forall v \in V^+ \\ x_e &= 0, \forall e \in E^0 \end{aligned}$$

Restricting to  $E^+$ , we can write the system of equations as

$$Mx^+ = (1, \dots, 1)^T$$

By Cramer's Rule, the solution to the system is  $(\bar{x}_1^+, \dots, \bar{x}_k^+)$ , where

$$\bar{x}_j^+ = \frac{\det(M^j)}{\det(M)}$$

with  $M^j$  obtained from  $M$  by replacing the  $j$ th column by  $(1, \dots, 1)^T$ .

Claim:  $\det(M) = 1$  or  $\det(M) = -1$ .

This gives that  $\bar{x}_j^+$  is integer for all  $j$ , so  $\bar{x}$  is integer. Thus,  $\nu(G) = \nu^*(G)$ .

### Lemma

Let  $G = (V, E)$  be a bipartite graph. Let  $A$  be the  $|V| \times |E|$  matrix  $[A_{ve}]$  with

$$A_{ve} = \begin{cases} 1 & \text{if } e \in \delta(v) \\ 0 & \text{if } e \notin \delta(v) \end{cases}$$

then  $A$  is totally unimodular.

**Proof.** By induction of the number of rows  $k$  of the submatrix  $B$  of  $A$ . If  $B$  is  $1 \times 1$ , then this is obvious.

Suppose it is true for  $k = 1, \dots, t-1$  and let  $B$  be a  $t \times t$  submatrix of  $A$ .

1. If  $B$  has a column of all 0's, then  $\det(B) = 0$ .
2. If a column of  $B$  has exactly one 1, then we compute  $\det(B)$  by expanding on that column and use induction.
3. Otherwise, every column of  $B$  has exactly two 1's.

We can partition the rows of  $B$  into  $W_1$  and  $W_2$ , so that every column has exactly one 1 in  $W_1$  and exactly one 1 in  $W_2$  ( $W_1$  are vertices in  $V_1$ ,  $W_2$  in  $V_2$  from  $G$  being bipartite).

Now multiplying each row in  $W_1$  by 1 and each row in  $W_2$  by  $-1$  and summing, we get the row vector of all 0's. So  $\det(B) = 0$ .  $\square$

## 9.4 Tutte's Theorem

Let  $A$  be a subset of the vertices which  $G - A$  has  $k$  components  $H_1, \dots, H_k$  having an odd number of vertices. Let  $M$  be a matching of  $G$ . For each  $i$ , either  $H_i$  has an  $M$ -exposed vertex or  $M$  contains an edge having just one end in  $V(H_i)$ . All such edges have their other

ends in  $A$  and since  $M$  is a matching, all these ends must be distinct. Therefore, there can be at most  $|A|$  edges and so the number of  $M$ -exposed vertices is at least  $k - |A|$ .

**Definition: Odd Count  $\text{oc}(H)$**

The number of components of  $H$  that contain an odd number of vertices.

Thus, for any  $A \subseteq V$ ,

$$\nu(G) \leq \frac{1}{2}(|V| - \text{oc}(G - A) + |A|)$$

If  $A$  is a cover of  $G$ , then there are  $|V| - |A|$  odd components of  $G - A$  (each is a single vertex), so the right hand side reduces to  $|A|$ . This bound is at least as strong as that provided by covers.

**Theorem (Tutte-Berge Formula)**

For a graph  $G = (V, E)$ , we have

$$\max\{|M| : M \text{ a matching}\} = \min \left\{ \frac{1}{2}(|V| - \text{oc}(G \setminus A) + |A|) : A \subseteq V \right\}$$

**Theorem (Tutte's Matching Theorem 1947)**

A graph  $G = (V, E)$  has a perfect matching if and only if for all  $A \subseteq V$ ,  $\text{oc}(G \setminus A) \leq |A|$ .

$A$  is called a Tutte set.

## 9.5 Maximum Matching

**Maximum Matching Problem**

Given a graph  $G$ , find a maximum matching of  $G$ .

**Definition: Maximum Matching ILP**

$$\begin{aligned} \max \quad & \sum (x_e : e \in E) \\ \text{s.t.} \quad & x(\delta(v)) \leq 1, \forall v \in V \\ & x_e \geq 0, \forall e \in E \\ & x_e \text{ integer}, \forall e \in E \end{aligned}$$

**Definition: Maximum Matching LP Relaxation**

$$\begin{aligned}
& \max \quad \sum (x_e : e \in E) \\
& \text{s.t.} \quad x(\delta(v)) \leq 1, \forall v \in V \\
& \quad \quad x_e \geq 0, \forall e \in E
\end{aligned}$$

**Definition: Minimum Cover Dual LP**

$$\begin{aligned}
& \min \quad \sum (y_v : v \in V) \\
& \text{s.t.} \quad y_u + y_v \geq 1, \forall e = (u, v) \in E \\
& \quad \quad y_v \geq 0, \forall v \in V
\end{aligned}$$

Let  $M$  be a matching and  $C$  be a cover, then

$$x_e^M = \begin{cases} 1 & \text{if } e \in M \\ 0 & \text{if } e \notin M \end{cases}, y_v^C = \begin{cases} 1 & \text{if } v \in C \\ 0 & \text{if } v \notin C \end{cases}$$

So,  $\nu(G) \leq \nu^*(G)$  and  $\tau(G) \geq \tau^*(G)$ , and by LP duality, we have

$$\nu(G) \leq \nu^*(G) = \tau^*(G) \leq \tau(G)$$

## 9.6 Perfect Matching

### 9.6.1 Blossom Algorithm for Perfect Matching

Suppose we have a matching  $M$  of  $G$  and a fixed  $M$ -exposed vertex  $r$  of  $G$ . We can iteratively build up sets  $A, B$  of vertices such that each vertex in  $A$  is the other end of an odd-length  $M$ -alternating path beginning at  $r$ , and each vertex in  $B$  is the other end of an even-length  $M$ -alternating path beginning at  $r$ .

Begin with  $A = \emptyset, B = \{r\}$ , and use the rule: if  $vw \in E, v \in B, w \notin A \cup B, wz \in M$ , then add  $w$  to  $A$ ,  $z$  to  $B$ . The set  $A \cup B$  and edges in the construction form a tree  $T$  rooted at  $r$ .

**Definition:  $M$ -Alternating Tree**

A tree  $T$  such that

- every vertex of  $T$  other than  $r$  is covered by an edge of  $M \cap E(T)$ ;
- for every vertex  $v$  of  $T$ , the path in  $T$  from  $v$  to  $r$  is  $M$ -alternating.

We let the vertex sets at odd and even distances from the root as  $A(T)$  and  $B(T)$  respectively. Note that  $|B(T)| = |A(T)| + 1$  since all other vertices other than  $r$  come in matched pairs, one in  $A(T)$  and one in  $B(T)$ .

**Algorithm Sketch:** Given a matching  $M$ , if  $M$  is not perfect, search for an augmenting path  $P$ . Recall that  $M' = M \Delta E(P)$  gives us a larger matching since  $|M'| = |M| + 1$ .

If the algorithm does not find an augmenting path, then we need to certify that  $G$  has no perfect matching. We can use Tutte's Matching Theorem to find a Tutte set  $A$  where  $oc(G \setminus A) > |A|$ .

Let  $r \in V$  be  $M$ -exposed. Grow an  $M$ -alternating tree  $T$ .

Choose an edge  $vw \in E$  with  $v \in B(T)$  and  $w \notin A(T)$ .

Case 1:  $w$  is  $M$ -exposed.

We have an augmenting path from  $r$  to  $w$ , so we can augment this path to get a larger matching. We reset  $T$  since  $r$  is now  $M$ -covered.

Case 2:  $w \notin V(T)$  and  $w$  is  $M$ -covered.

We can grow  $T$  by adding  $vw$  and the edge  $e \in M$  having  $w$  as an end.

Case 3:  $w \in B(T)$ .

Let  $C$  be the circuit formed from  $vw$  and the path in  $T$  joining  $v$  and  $w$ .  $|C|$  is odd since vertices in  $B(T)$  are at even distances from  $r$ .

Note:  $|M \cap E(C)| = \frac{|C|-1}{2}$  so  $M$  is a near-perfect matching of  $C$ .

#### **Definition: Blossom**

Let  $v, w \in B(T)$  and  $vw \in E(G)$ . The odd circuit in  $T + vw$  is a blossom.

#### **Definition: Shrink**

Let  $C$  be an odd circuit in  $G$ . Define  $G' = G \times C$  as the subgraph obtained from  $G$  by shrinking  $C$ ;  $G'$  has vertex set  $(V - V(C)) \cup \{C\}$  and edge set  $E \setminus \gamma(V(C))$ .

#### **Definition: Pseudonode**

The vertex after shrinking a blossom.

Let  $G' \times C$  denote the graph obtained by shrinking  $C$ . The near-perfect matching allows us to un-shrink an augmenting path. Note that the pseudonode is in  $B(T)$ .

#### **Definition: Frustrated**

An  $M$ -alternating tree  $T$  in a graph  $G$  is frustrated if every edge of  $G$  has one end in  $B(T)$  and the other end in  $A(T)$ .

**Definition:**  $S(v)$ 

Given a vertex  $v$  of  $G'$ , there corresponds a set  $S(v)$  of vertices of  $G$ , where

$$S(v) = \begin{cases} v & \text{if } v \in V(G) \\ \bigcup_{w \in V(C)} S(w) & \text{if } v = C \text{ is a pseudonode} \end{cases}$$

**Lemma**

Let  $M'$  be a matching of  $G'$  and let  $T$  be an  $M'$ -alternating tree of  $G'$  such that no element of  $A(T)$  is a pseudonode. If  $T$  is frustrated, then  $G$  has no perfect matching.

**Proof.** When we delete  $A(T)$  from  $G'$ , we get a component with vertex set  $S(v)$  for each  $v \in B(T)$ . By construction,  $|S(v)|$  is odd since it is the union of an odd number of vertices and pseudonodes  $u$  each having  $|S(u)|$  odd. Since  $|B(T)| = |A(T)| + 1$ , then  $\text{oc}(G \setminus A(T)) > |A(T)|$ .  $A(T)$  is therefore a Tutte set.  $\square$

**Algorithm 4** Blossom Algorithm (Perfect Matching)

- 1: **Input:** Graph  $G$  and matching  $M$  of  $G$
- 2: Choose an  $M$ -exposed vertex  $r$
- 3:  $T = (\{r\}, \emptyset)$
- 4: **while** there exists  $vw \in E$  with  $v \in B(T), w \notin A(T)$  **do**
- 5:     **Case:**  $w$  is  $M'$ -exposed
- 6:         Let  $P$  be the augmenting path from  $r$  to  $w$ ,  $M = M \Delta E(P)$
- 7:         **if** there is no  $M$ -exposed vertex in  $G$  **then**
- 8:             **return** Perfect matching  $M$
- 9:         **else**
- 10:              $T = (\{r\}, \emptyset)$ , where  $r$  is  $M$ -exposed
- 11:     **Case:**  $w \notin V(T)$ ,  $w$  is  $M$ -covered
- 12:         Let  $wz \in M$  and  $z \notin V(T)$ ,  $E(T) = E(T) \cup \{vw, wz\}$
- 13:     **Case:**  $w \in B(T)$
- 14:         Let  $C$  be the circuit formed from  $vw$  and the  $vw$ -path in  $T$
- 15:          $G = G \times C$  // Shrink  $C$
- 16: **return**  $G$  has no perfect matching,  $A(T)$  is the Tutte set

**Theorem**

The Blossom Algorithm terminates after  $O(n)$  augmentations,  $O(n^2)$  shrinking steps, and  $O(n^2)$  tree-extension steps.  
Moreover, it determines correctly whether  $G$  has a perfect matching.

## 9.7 Blossom Algorithm for Maximum Matching

We can extend the Blossom algorithm for perfect matchings to maximum matchings.

---

### Algorithm 5 Blossom Algorithm (Maximum Matching)

---

```

1: Input: Graph  $G$  and matching  $M$  of  $G$ 
2:  $\mathcal{T} = \emptyset$ 
3: Choose an  $M$ -exposed vertex  $r$ 
4:  $T = (\{r\}, \emptyset)$ 
5: while there exists  $vw \in E$  with  $v \in B(T), w \notin A(T)$  do
6:   Case:  $w$  is  $M$ -exposed
7:     Let  $P$  be the augmenting path from  $r$  to  $w$ ,  $M = M \Delta E(P)$ 
8:     if there is no  $M$ -exposed vertex then
9:       return Perfect matching  $M$ 
10:    else
11:       $T = (\{r\}, \emptyset)$ , where  $r$  is  $M$ -exposed
12:    Case:  $w \notin V(T)$ ,  $w$  is  $M$ -covered
13:      Let  $wz \in M$  and  $z \notin V(T)$ ,  $E(T) = E(T) \cup \{vw, wz\}$ 
14:    Case:  $w \in B(T)$ 
15:      Let  $C$  be the circuit formed from  $vw$  and the  $vw$ -path in  $T$ 
16:       $G = G \times C$  // Shrink  $C$ 
17:  $\mathcal{T} = \mathcal{T} \cup \{T\}$ ,  $G = G \setminus V(T)$ ,  $M = M \setminus E(T)$ 
18: if there exists an  $M$ -exposed vertex then
19:   Go to line 5
20: Restore the matching  $M$ 
21: return  $M$ 

```

---

#### Theorem

The Blossom Algorithm can be implemented to run in time  $O(nm \log n)$ .



# Chapter 10

## Minimum-Cost Perfect Matching

### 10.1 Linear Programming and Matching Polytope

By Minkowski's theorem, we know that there is a system of inequalities for the convex hull of a set, but we typically do not know what the system is. However, Edmonds founded the matching polytope theory in the 1960s.

The integer program for the minimum-cost perfect matching problem is

$$\begin{aligned} \min \quad & \sum (c_e x_e : e \in E) \\ \text{s.t.} \quad & x(\delta(v)) = 1, \forall v \in V \\ & x_e \geq 0, \forall e \in E \\ & x_e \text{ integer}, \forall e \in E \end{aligned}$$

If  $G$  is bipartite, then we do not need the integrality constraint. If  $G$  is non-bipartite, then there exists a circuit  $C$  with  $|E(C)|$  odd.

For a general graph, the original minimum-weight perfect matching LP can achieve an optimal solution, which may be a fractional vertex of the polyhedron defined by the LP. This is caused by the odd circuits in the graph.

#### Definition: Perfect Matching Polytope

$$\mathcal{PM}(G) = \text{conv.hull}(\{x^M : M \text{ a perfect matching}\})$$

Let  $S \subseteq V$  with  $|S|$  odd and  $|S| \geq 3$ . For an undirected graph, every perfect matching must contain at least one edge in  $\delta(S)$ , so we arrive at the blossom inequality constraint.

#### Definition: Blossom Inequality

For  $S \subseteq V$  with  $|S|$  odd and  $|S| \geq 3$ ,

$$x(\delta(S)) \geq 1$$

### Theorem (Perfect Matching Polytope Theorem – Edmonds)

For any graph  $G = (V, E)$ ,  $\mathcal{PM}(G)$  is the solution set of the linear system

$$\begin{aligned} x(\delta(v)) &= 1, \quad \forall v \in V \\ x(\delta(S)) &\geq 1, \quad \forall S \subseteq V, |S| \text{ odd}, |S| \geq 3 \\ x_e &\geq 0, \quad \forall e \in E \end{aligned}$$

**Proof.** (Schrijver) Let  $Q$  denote the solution set of the linear system. Clearly  $\mathcal{PM}(G) \subseteq Q$ . Suppose  $Q \not\subseteq \mathcal{PM}(G)$  and let  $x$  be a vertex of  $Q$  with  $x \notin \mathcal{PM}(G)$ . Choose this counterexample  $G$  such that  $|V| + |E|$  is as small as possible.

**Claim 1:**  $0 < x_e < 1$  for all  $e \in E$ .

**Proof.** (Claim 1) Otherwise, if  $x_e = 0$ , then delete  $e$ . If  $x_e = 1$ , then delete  $e$  and the ends of  $e$ .  $\square$

So each vertex of  $G$  has degree at least 2, which implies  $|E| \geq |V|$ .

**Claim 2:**  $|E| > |V|$ .

**Proof.** (Claim 2) If  $|E| = |V|$ , then  $G$  is a circuit and the theorem is true.  $\square$

Since  $x$  is a vertex of  $Q$ , there are  $|E|$  constraints of the linear system satisfied as an equation by  $x$ . Thus, there exists an odd  $S \subseteq V$  with  $3 \leq |S| \leq |V| - 3$  and  $x(\delta(S)) = 1$ .

Let  $G'$  be the graph obtained by shrinking  $S$  to a single vertex and  $G''$  be obtained by shrinking  $V \setminus S$  to a single vertex. Let  $x'$  and  $x''$  be obtained by shrinking  $x$  to  $G'$  and  $G''$  respectively. So  $x'$  and  $x''$  satisfy the linear system for  $G'$  and  $G''$ . By induction,  $x' \in \mathcal{PM}(G')$  and  $x'' \in \mathcal{PM}(G'')$ .

Since  $x$  is rational,  $x'$  and  $x''$  are rational convex combinations of perfect matchings in  $G'$  and  $G''$ , i.e.

$$x' = \frac{1}{k} \sum_{i=1}^k x^{M'_i}, \quad x'' = \frac{1}{k} \sum_{i=1}^k x^{M''_i}$$

for some  $k$  (the common denominator of the multipliers  $\lambda'_i$  and  $\lambda''_i$  in the convex combinations).

For each edge  $e \in \delta(S)$ , the number of indices  $i$  with  $e \in M'_i$  is  $kx'_e = kx_e = kx''_e$  which is equal to the number of indices  $i$  with  $e \in M''_i$ .

We may assume that for each  $i$ , the two matchings  $M'_i$  and  $M''_i$  have an edge in  $\delta(S)$  in common. So  $M_i = M'_i \cup M''_i$  is a perfect matching of  $G$  and

$$x = \frac{1}{k} \sum_{i=1}^k x^{M_i}$$

and thus  $x \in \mathcal{PM}(G)$ , a contradiction.  $\square$

**Definition: ODD**

$$\text{ODD} = \{S \subseteq V : |S| \text{ odd}, |S| \geq 3, |S| \leq |V| - 3\}$$

**Definition: Minimum-Cost Perfect Matching LP**

$$\begin{aligned} \min \quad & \sum (c_e x_e : e \in E) \\ \text{s.t.} \quad & x(\delta(v)) = 1, \forall v \in V \\ & x(\delta(S)) \geq 1, \forall S \in \text{ODD} \\ & x_e \geq 0, \forall e \in E \end{aligned}$$

**Definition: Minimum-Cost Perfect Matching Dual LP**

$$\begin{aligned} \max \quad & \sum (y_v : v \in V) + \sum (Y_S : S \in \text{ODD}) \\ \text{s.t.} \quad & y_v + y_w + \sum (Y_S : e \in \delta(S), S \in \text{ODD}) \leq c_e, \forall e = vw \in E \\ & Y_U \geq 0, \forall S \in \text{ODD} \end{aligned}$$

**Theorem (Edmonds 1965)**

Let  $G$  be a graph and let  $c \in \mathbb{R}^E$ . Then  $G$  has a perfect matching if and only if the Minimum-Cost Perfect Matching LP has a feasible solution.  
Moreover, if  $G$  has a perfect matching, then the minimum cost of a perfect matching is equal to the optimal value of the LP.

**Definition: Complementary Slackness Conditions**

Let  $x^* = (x_e^* : e \in E)$ ,  $y^* = (y_v^* : v \in V)$ ,  $Y^* = (Y_S^* : S \in \text{ODD})$ .

- (i) If  $x_e^* > 0$ , then  $y_v^* + y_w^* + \sum (Y_S^* : e \in \delta(S), S \in \text{ODD}) = c_e$ .
- (ii) If  $Y_S^* > 0$ , then  $x^*(\delta(S)) = 1$ .

**Definition: Reduced Cost**

Given a dual solution  $(y, Y)$ , the reduced cost of an edge  $e = vw$  is

$$\bar{c}_e = c_e - y_v - y_w - \sum (Y_S : e \in \delta(S), S \in \text{ODD})$$

By dual feasibility, we have that  $\bar{c}_e \geq 0$  for all  $e \in E$ . To satisfy the complementary slackness conditions, we want a perfect matching  $M$  such that if  $e \in M$ , then  $\bar{c}_e = 0$  and if  $Y_S > 0$ , then  $|\delta(S) \cap M| = 1$ .

In other words, we only want edges having reduced cost 0 and only use dual variables  $Y_S$  if  $\delta(S)$  contains exactly one matching edge.

**Definition: Equality Subgraph  $E^=$** 

$$E^= = \{e \in E : \bar{c}_e = 0\}$$

We need a perfect matching in  $E^=$  such that if  $Y_S \geq 0$ , then  $|\delta(S) \cap M| = 1$ .

**Algorithm Sketch:** Grow an alternating tree using only  $E^=$  edges, starting at an  $M$ -exposed vertex. Unlike the unweighted algorithm, we need to keep the shrunk graph from iteration to iteration, because of the CS conditions on the  $Y_S$  variables. Similar to the unweighted version, we choose an edge  $e = vw$  with  $v \in B(T), w \notin A(T)$ . If none exist, we will possibly change the dual solution to create new edges in  $E^=$ .

**Algorithm 6 Blossom Algorithm (Minimum-Cost Perfect Matching)**


---

```

1: Input: Graph  $G$ 
2:  $M = \emptyset, Y_S = 0 : \forall S \in \text{ODD}, y_v = 0 : \forall v \in V$ 
3:  $T = (\{r\}, \emptyset)$ , where  $r$  is an  $M$ -exposed vertex
4: while true do
5:   Case:  $e = vw \in E^=$  with  $v \in B(T), w \notin A(T)$ 
6:     Case:  $w$  is  $M$ -exposed
7:       Let  $P$  be the augmenting path from  $r$  to  $w$ ,  $M = M \Delta E(P)$ 
8:       if there is no  $M$ -exposed vertex then
9:         return Perfect matching  $M$ 
10:      else
11:         $T = (\{r\}, \emptyset)$ , where  $r$  is  $M$ -exposed.
12:      Case:  $w \notin V(T)$ ,  $w$  is  $M$ -covered
13:        Let  $wz \in M$  and  $z \notin V(T)$ ,  $E(T) = E(T) \cup \{vw, wz\}$ 
14:      Case:  $w \in B(T)$ 
15:        Let  $C$  be the circuit formed from  $vw$  and the  $vw$ -path in  $T$ 
16:         $G = G \times C$  // Shrink  $C$ 
17:      Case: No edge  $e = vw$  available
18:         $\varepsilon_1 = \min(\bar{c}_e : e \text{ joins a vertex in } B(T) \text{ to a vertex not in } V(T))$  // next iteration
        will augment  $M$  or grow  $T$ 
19:         $\varepsilon_2 = \min(\bar{c}_e/2 : e \text{ joins two vertices in } B(T))$  // next iteration will shrink the odd
        circuit
20:         $\varepsilon_3 = \min(y_v : v \in A(T), v \text{ a pseudonode})$ 
21:         $\varepsilon = \min(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ 
22:        if  $\varepsilon = \infty$  then
23:          return  $G$  has no perfect matching; there exists a Tutte set
24:         $y_v = y_v + \varepsilon$  if  $v \in B(T)$  //  $y_v$  may be a pseudonode, so  $y_v$  is actually  $Y_S$ 
25:         $y_v = y_v - \varepsilon$  if  $v \in A(T)$  //  $y_v$  may be a pseudonode, so  $y_v$  is actually  $Y_S$ 
26:        if  $\varepsilon = \varepsilon_3$  then
27:          Unshrink the pseudonode  $v \in A(T)$  that attained the minimum
28:          Let  $s, t$  be the vertices adjacent to  $v$ 
29:          Keep the even-length  $st$ -path in  $T$ 

```

---

The Blossom Algorithm for Minimum-Cost Perfect Matching above has runtime complexity  $O(mn^2)$ . This can be improved to  $O(n^3)$ .

# Chapter 11

## $T$ -Joins and Postman Problems

### 11.1 Postman Problem

**Definition: Postman Tour**

A closed path where each edge is traversed at least once.

**Definition: Euler Tour**

A closed edge-simple path  $P$  such that  $E(P) = E(G)$ .

Note that if a graph  $G$  has an Euler tour, then the optimal postman tour is the Euler tour.

**Theorem**

A connected graph  $G$  has an Euler tour if and only if every vertex of  $G$  has even degree.

**Definition: Postman Set**

A set  $J \subseteq E$  is a postman set of  $G$  if for every  $v \in V$ ,  $v$  is incident with an odd number of edges from  $J$  if and only if  $v$  has odd degree in  $G$ .

**Postman Problem**

Given a graph  $G = (V, E)$  and  $c \in \mathbb{R}^E$  such that  $c \geq 0$ , find a postman set  $J$  such that  $c(J)$  is minimum.

**Definition: Postman Problem LP**

$$\begin{aligned}
\min \quad & \sum (c_e x_e : e \in E) \\
\text{s.t.} \quad & x(\delta(v)) \equiv |\delta(v)| \pmod{2}, \forall v \in V \\
& x_e \geq 0, \forall e \in E \\
& x_e \text{ integer}, \forall e \in E
\end{aligned}$$

## 11.2 $T$ -Joins

**Definition:  $T$ -Join**

Let  $G = (V, E)$  be a graph and let  $T \subseteq V$  such that  $|T|$  is even. A  $T$ -join is a set  $J \subseteq E$  such that

$$|J \cap \delta(v)| \equiv |T \cap \{v\}| \pmod{2}, \forall v \in V$$

In other words,  $J$  is a  $T$ -join if and only if the odd-degree vertices of the subgraph  $(V, J)$  are exactly the elements of  $T$ .

**Optimal  $T$ -Join Problem**

Given a graph  $G = (V, E)$ , a set  $T \subseteq V$  such that  $|T|$  is even, and a cost vector  $c \in \mathbb{R}^E$ , find a  $T$ -join  $J$  of  $G$  such that  $c(J)$  is minimum.

Examples:

- Postman sets: Let  $T = \{v \in V : |\delta(v)| \text{ is odd}\}$ . Then the  $T$ -joins are precisely the postman sets. Finding an optimal  $T$ -join solves the postman problem.
- Even set: Let  $T = \emptyset$ . Then a  $T$ -join is exactly an even set, that is, a set  $A \subseteq E$  such that every vertex of  $(V, A)$  has even degree. A set is even if and only if it can be decomposed into edge sets of edge-disjoint circuits.
- $rs$ -paths: Let  $r, s \in V$  and let  $T = \{r, s\}$ . Every  $T$ -join  $J$  contains the edge-set of an  $rs$ -path. (**Proof.** If not, the component of the subgraph  $(V, J)$  that contains  $r$  has only one vertex of odd degree.)

**Proposition**

Let  $J'$  be a  $T'$ -join of  $G$ . Then  $J$  is a  $T$ -join of  $G$  if and only if  $J \Delta J'$  is a  $(T \Delta T')$ -join of  $G$ .

**Proof.** It is enough to prove the “only if” part, since the other part can be deduced by applying this one with  $J$  replaced by  $J \Delta J'$  and  $T$  replaced by  $T \Delta T'$ .

Suppose that  $J$  is a  $T$ -join and  $J'$  is a  $T'$ -join. Let  $v \in V$ . Then  $|(J \Delta J') \cap \delta(v)|$  is even if and only if  $|J \cap \delta(v)| \equiv |J' \cap \delta(v)| \pmod{2}$ , which is true if and only if  $v$  is an element of neither or both of  $T$  and  $T'$ , that is, if and only if  $v \notin T \Delta T'$ .  $\square$

## 11.3 Optimal $T$ -Join Algorithm

### Proposition

Every minimal  $T$ -join is the union of the edge sets of  $\frac{|T|}{2}$  edge-disjoint simple paths, which join the vertices in  $T$  in pairs.

### Proposition

Suppose that  $c \geq 0$ . Then there is an optimal  $T$ -join that is the union of  $\frac{|T|}{2}$  edge-disjoint shortest paths joining the vertices of  $T$  in pairs.

---

### Algorithm 7 Optimal $T$ -Join Algorithm

---

- 1: Identify the set  $N$  of edges having negative cost and let the set  $T'$  of vertices incident with an odd number of edges from  $N$
  - 2:  $c = |c|$ ,  $T = T \Delta T'$
  - 3: Find a least-cost  $uv$ -path  $P_{uv}$  with respect to  $c$  for each pair  $u, v$  of vertices from  $T$  and let  $d(u, v)$  be the cost of  $P_{uv}$
  - 4: Form a complete graph  $\hat{G} = (T, \hat{E})$  with  $uv$  having weight  $d(u, v)$  for each  $uv \in \hat{E}$
  - 5: Find a minimum-weight perfect matching  $M$  in  $\hat{G}$
  - 6: Let  $J$  be the symmetric difference of  $E(P_{uv})$  for  $uv \in M$
  - 7:  $J = J \Delta N$
- 

## 11.4 $T$ -Join LP

### Definition: $T$ -Odd

A set  $S \subseteq V$  is  $T$ -odd if  $|S \cap T|$  is odd.

### Definition: $T$ -Cut

The set  $\delta(S)$  where  $S \subseteq V$  is  $T$ -odd.

### Definition: $T$ -Join LP

$$\begin{aligned} \min \quad & \sum (c_e x_e : e \in E) \\ \text{s.t.} \quad & x(D) \geq 1, \quad \forall T\text{-cuts } D \\ & x_e \geq 0, \quad \forall e \in E \end{aligned}$$



**Definition:  $T$ -Join Dual LP**

$$\begin{aligned} \max \quad & \sum (Z_D : D \text{ a } T\text{-cut}) \\ \text{s.t.} \quad & \sum (Z_D : e \in D, D \text{ a } T\text{-cut}) \leq c_e, \forall e \in E \\ & Z_D \geq 0, \forall T\text{-cuts } D \end{aligned}$$

**Theorem**

If  $G = (V, E)$  is a graph,  $T \subseteq V$  with  $|T|$  even, and  $c \in \mathbb{R}^E$  with  $c \geq 0$ , then the minimum cost of a  $T$ -join of  $G$  is equal to the optimal value of the  $T$ -join LP.

**Theorem**

Let  $G = (V, E)$  be a graph and  $c \in \mathbb{Z}^E$ . Suppose that every circuit of  $G$  has even  $c$ -cost. Then the  $T$ -join dual LP has an optimal solution that is integral.

# Part VI

## Matroids

# Chapter 12

## Matroid Theory

Recall Kruskal's algorithm to find a maximum-weight spanning tree. A slight variant of the algorithm is to find a maximum-weight forest. Let  $\mathcal{I} = \{J \subseteq E : J \text{ is a forest}\}$ .

---

**Algorithm 8** Greedy Algorithm

---

- 1:  $J = \emptyset$
  - 2: **while** there exists  $e \notin J$  with  $c_e > 0$  and  $J \cup \{e\} \in \mathcal{I}$  **do**
  - 3:     Choose  $e$  with  $c_e$  maximum
  - 4:      $J = J \cup \{e\}$
  - 5: **return**  $J$
- 

The family  $\mathcal{I}$  of forests of a graph has the property that the Greedy Algorithm finds a maximum-weight independent set. Families like forests for which the Greedy Algorithm always returns an optimal solution are called matroids.

**Definition: Matroid**

Let  $S$  be a finite set (*ground set*) and  $\mathcal{I}$  be a family of subsets of  $S$  (*independent sets*).  $M = (S, \mathcal{I})$  is a matroid if the following axioms are satisfied:

(M0)  $\emptyset \in \mathcal{I}$ .

(M1) If  $J' \subseteq J \in \mathcal{I}$ , then  $J' \in \mathcal{I}$ .

(M2) For every  $A \subseteq S$ , every maximal independent subset of  $A$  has the same cardinality.

**Definition: Basis**

A maximal independent subset of a set  $A \subseteq S$  is a basis of  $A$ .

**Definition: Rank**

The size (which depends only on  $A$  by (M2)) of the basis of  $A$ , denoted  $r(A)$ .

## Part VII

# Traveling Salesman Problem

# Chapter 13

## The Traveling Salesman Problem

**Definition: Tour**

A circuit that passes exactly once through each vertex.

This is also known as a Hamiltonian circuit.

**Traveling Salesman Problem (TSP)**

Given a finite set of points  $V$  and a cost  $c_{uv}$  of travel between each pair  $u, v \in V$ , find a tour of minimal cost.

The TSP can be modeled as a graph problem by considering the complete graph. TSP belongs to class of  $\mathcal{NP}$ -hard problems.