

# Lecture: Integer Programming, Spring 2010

Jan Foniok and Komei Fukuda  
Institute for Operations Research [JF, KF]  
and Institute of Theoretical Computer Science [KF]  
ETH Zentrum, Zurich, Switzerland

May 31, 2010

## Contents

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Euclidean Algorithm and Hermite Normal Form</b>	<b>9</b>
2.1	Sizes of Rational Numbers and Polynomial Complexity . . . . .	9
2.2	Computing the GCD . . . . .	10
2.3	Computing the Hermite Normal Form . . . . .	12
2.4	Lattices and the Hermite Normal Form . . . . .	15
2.5	Dual Lattices . . . . .	17
<b>3</b>	<b>Linear Inequalities, LP and Polyhedra</b>	<b>18</b>
3.1	Systems of Linear Inequalities . . . . .	18
3.2	The Fourier-Motzkin Elimination . . . . .	18
3.3	LP Duality . . . . .	20
3.4	Three Theorems on Convexity . . . . .	22
3.5	Representations of Polyhedra . . . . .	24
3.6	The Structure of Polyhedra . . . . .	26
3.7	Some Basic Polyhedra . . . . .	28
<b>4</b>	<b>Integer Hull and the Complexity of IP</b>	<b>29</b>
4.1	Hilbert Basis . . . . .	29
4.2	The Structure of Integer Hull . . . . .	30
4.3	Further Results on Lattice Points in Polyhedra . . . . .	33
<b>5</b>	<b>Well Solved Problems</b>	<b>34</b>
5.1	Good Description and $NP \cap coNP$ . . . . .	34
5.2	Optimal Matching Problem . . . . .	35
5.3	Max-Flow Problem . . . . .	36
5.4	Minimum Cost Flow Problem . . . . .	38
5.5	A Min-Max Relation for Submodular Functions . . . . .	39

5.6	Total Dual Integrality . . . . .	42
<b>6</b>	<b>Diophantine Approximation and Lattice Reduction</b>	<b>43</b>
6.1	Diophantine Approximation . . . . .	43
6.2	Lattice Reduction . . . . .	45
6.3	Applications of Lattice Reduction . . . . .	49
<b>7</b>	<b>Integer programming in fixed dimension</b>	<b>51</b>
7.1	Maximum volume inscribed ellipsoid . . . . .	51
7.2	Lenstra's algorithm . . . . .	53
<b>8</b>	<b>Integral polyhedra and totally unimodular matrices</b>	<b>57</b>
8.1	Integral polyhedra . . . . .	57
8.2	Total unimodularity . . . . .	57
8.3	Incidence matrices of graphs . . . . .	58
<b>9</b>	<b>Valid inequalities</b>	<b>61</b>
9.1	Valid inequalities for polyhedra . . . . .	61
9.2	Integer rounding . . . . .	61
9.3	Gomory cutting planes . . . . .	63
9.4	Disjunctive constraints . . . . .	64
9.5	All valid inequalities for IPs . . . . .	65
<b>10</b>	<b>General IP algorithms</b>	<b>69</b>
10.1	A fractional cutting plane algorithm . . . . .	69
10.2	Mixed integer programs . . . . .	75
10.3	Branch and bound . . . . .	76
<b>11</b>	<b>Lagrangian duality</b>	<b>79</b>
11.1	Lagrangian relaxation and Lagrangian dual . . . . .	79
11.2	How to compute $z(u)$ ? . . . . .	79
11.3	How good is the upper bound? . . . . .	81
11.4	How to compute $z_{LD}$ ? . . . . .	82
<b>12</b>	<b>References</b>	<b>85</b>

# 1 Overview

## Linear Programming

- A *linear programming problem* (an *LP*) is to find an optimal solution  $x \in \mathbb{R}^n$  (if exists) to

$$\begin{aligned} \max \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq \mathbf{0}. \end{aligned} \tag{1.1}$$

Here,  $A \in \mathbb{Q}^{m \times n}$ ,  $b \in \mathbb{Q}^m$  and  $c \in \mathbb{Q}^n$  are given (inputs). An LP may not have an optimal solution if it is infeasible (i.e., the constraints are inconsistent) or unbounded (i.e., the objective function is not bounded above in the feasible region  $\{x : Ax = b, x \geq \mathbf{0}\}$ .)

One of the most important facts on LP is that there is an easy way to verify the optimality of a solution to an LP, using a dual optimal solution. It is the essence of the LP duality.

The LP problem is known to be in class P, that is, there is a polynomial-time algorithm.

## Integer (Linear) Programming

- An *integer (linear) programming problem* (an *IP* or *ILP*) is to find an optimal solution  $x \in \mathbb{Z}^n$  (if exists) to

$$\begin{aligned} \max \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq \mathbf{0} \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{1.2}$$

The only difference from an LP is the additional integer constraint  $x \in \mathbb{Z}^n$ . This innocent-looking condition makes IP very hard to solve. In fact, the IP is NP-hard in general. The linear inequality constraints  $Ax = b, x \geq \mathbf{0}$  can be replaced by  $Ax \leq b$  without changing the complexity of the problem. There are simple reductions between two different forms of IP.

Contrary to LP, there is no known succinct certificate for the optimality of an optimal solution to an IP, in general. On the positive side, if an IP has an optimal solution, it has an optimal solution whose (binary encoding) size is bounded polynomially by the sizes of inputs ( $A$ ,  $b$  and  $c$ ).

There are many known special cases of the IP that are polynomially solvable, such as the assignment problem and the optimal perfect matching problem. On the other hand, there are numerous special cases of the IP that are NP-hard, such as the knapsack problem, the set cover problem and the max-cut problem.

When the integer constraint  $x \in \mathbb{Z}^n$  is replaced by a partial integer constraints  $x_1, \dots, x_k \in \mathbb{Z}$  with  $k < n$ , the problem becomes a *mixed integer programming problem* or simply

*MIP*. It does not seem possible to reduce an MIP to an IP without introducing many new variables and constraints. However, some techniques for solving the IP can be extended to the MIP problems.

## Assignment Problem

Given an  $n \times n$  matrix  $W = [w_{ij}]$ , select  $n$  entries one from each row and each column so as to maximize (or minimize) the sum.

Equivalently, given a weighted complete bipartite graph  $G = (V_1, V_2, E)$  with  $|V_1| = |V_2| = n$ , find a perfect matching  $M \subseteq E$  of maximum total weight.

- IP Formulation of the Assignment Problem

Let  $X = [x_{ij}]$  be an  $n \times n$  variable matrix. One can also consider  $X$  as a vector of size  $n \times n$ . The meaning of each variable  $x_{ij}$  is:

$$x_{ij} = \begin{cases} 1 & \text{if the position } (i, j) \text{ is chosen} \\ 0 & \text{if the position } (i, j) \text{ is not chosen} \end{cases}$$

(IP-A)	$\max w(X) := \sum_{i,j} w_{ij}x_{ij}$
subject to	$\sum_{j=1}^n x_{ij} = 1, \quad \forall i = 1, \dots, n$
	$\sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, n$
	$x_{ij} = 0 \text{ or } 1, \quad \forall i, j = 1, \dots, n.$

- LP Relaxation of the Assignment Problem

$$\begin{array}{ll}
 \text{(LP-A)} & \max w(X) := \sum_{i,j} w_{ij}x_{ij} \\
 & \text{subject to} \quad \sum_{j=1}^n x_{ij} = 1, \quad \forall i = 1, \dots, n \\
 & \quad \quad \quad \sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, n \\
 & \quad \quad \quad x_{ij} \geq 0, \quad \forall i, j = 1, \dots, n.
 \end{array}$$

Since (LP-A) has a less restricted constraint set, its optimal value is at least as large as that of (IP-A). An important fact is

**Theorem 1.1** *There is an integer optimal solution to the linear programming problem (LP-A). Consequently, the optimal values of (LP-A) and (IP-A) are equal.*

- The feasible region of (LP-A):

$$\begin{aligned}
 P(\text{LP-A}) = \{x \in \mathbb{R}^{n \times n} : & \sum_{j=1}^n x_{ij} = 1, \quad \forall i = 1, \dots, n \\
 & \sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, n \\
 & x_{ij} \geq 0, \quad \forall i, j = 1, \dots, n\}.
 \end{aligned}$$

- Theorem 1.1 can be stated more geometrically in terms of the polytope  $P(\text{LP-A})$ :

**Theorem 1.2**  *$P(\text{LP-A})$  is a 0/1 polytope, that is, every extreme point has only 0 or 1 components.*

Here, a point of the region  $P(\text{LP-A})$  is called *extreme* (or a *vertex*) if it cannot be written as the middle point of two other points of the set.

**Exercise 1.1** Prove Theorem 1.2. (Hint: Show that every fractional point  $x \in P(\text{LP-A})$  can be written as  $\frac{x'+x''}{2}$  for some points  $x', x'' \in P(\text{LP-A})$  different from  $x$ .)

**Exercise 1.2** Explain how Theorem 1.1 follows from the proof above.

**Exercise 1.3** Let  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ , and  $P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ . Show that if the polyhedron  $P$  is nonempty, then it has an extreme point. (Hint: Given a nonextreme point  $x \in P$ , design a finite algorithm to find an extreme point. Is it a polynomial-time algorithm when all inputs are rational?)

**Exercise 1.4** Write  $P(\text{LP-A})$  in form  $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  for a suitable  $A$  and  $b$  (a vector of all 1's)? Show that the matrix  $A$  is totally unimodular, i.e., every square submatrix has determinant equal to 0, 1 or  $-1$ .

Use this fact to derive a different proof of Theorem 1.2.

## Optimal Perfect Matching Problem

Given a (undirected) graph  $G = (V, E)$  with edge weight  $w_{ij}$  for each  $(i, j) \in E$ , find a perfect matching  $M$  with maximum total weight. Here  $G$  is undirected and thus we identify  $(i, j)$  with  $(j, i)$ . Similarly, we identify  $w_{ij}$  with  $w_{ji}$ , etc.

- IP Formulation of the Optimal Perfect Matching Problem

Let  $x_{ij} (\equiv x_{ji})$  denote a variable for each edge  $(i, j) \in E$  to represent a matching. The meaning of each variable  $x_{ij}$  is:

$$x_{ij} = \begin{cases} 1 & \text{if } (i, j) \text{ is a matching edge} \\ 0 & \text{otherwise} \end{cases}$$

$  \begin{aligned}  \text{(IP-M)} \quad & \max w(x) := \sum_{(i,j) \in E} w_{ij} x_{ij} \\  & \text{subject to} \quad \sum_{(i,j) \in E} x_{ij} = 1, \quad \forall i \in V \\  & \quad \quad \quad x_{ij} = 0 \text{ or } 1, \quad \forall (i, j) \in E.  \end{aligned}  $
--

- The LP relaxation of (IP-M):

$  \begin{aligned}  \text{(LP-M)} \quad & \max w(x) := \sum_{(i,j) \in E} w_{ij} x_{ij} \\  & \text{subject to} \quad \sum_{(i,j) \in E} x_{ij} = 1, \quad \forall i \in V \\  & \quad \quad \quad x_{ij} \geq 0, \quad \forall (i, j) \in E.  \end{aligned}  $
---

- The feasible region of (LP-M):

$$P(\text{LP-M}) = \left\{ x \in \mathbb{R}^{n \times n} : \sum_{(i,j) \in E} x_{ij} = 1, \quad \forall i \in V, \right. \\
 \left. x_{ij} \geq 0, \quad \forall (i, j) \in E \right\}.$$

One can easily see that  $P(\text{LP-M})$  can have fractional extreme points.

**Exercise 1.5** Prove that  $P(\text{LP-M})$  is half-integer, that is, the components of every extreme point are half-integer, namely, in  $\{0, 1/2, 1\}$ . (Hint: Show that every non-half-integer fractional point is not extreme.)

## 2-Factor Problem and Symmetric TSP

Given a (undirected) graph  $G = (V, E)$  with edge weight  $w_{ij}$ , the optimal 2-factor problem is to find a 2-factor  $M$  with minimum total weight if exists, where a 2-factor is a set of edges which meets each node exactly twice.

- IP Formulation of the 2-Factor Problem (in Minimization Form)

$$\begin{array}{ll}
 \text{(IP-2F)} & \min w(x) := \sum_{(i,j) \in E} w_{ij} x_{ij} \\
 & \text{subject to} \quad \sum_{(i,j) \in E} x_{ij} = 2, \quad \forall i \in V \\
 & \quad \quad \quad x_{ij} = 0 \text{ or } 1, \quad \forall (i,j) \in E.
 \end{array}$$

This problem can be solved in polynomial time using any efficient algorithm for the optimal matching problem. The trivial LP relaxation has fractional optimal solutions in general. However one can eliminate all fractional extreme points by adding some new constraints. Note that if the box constraints  $0 \leq x_{ij} \leq 1$  in the relaxation are replaced by  $0 \leq x_{ij} \leq 2$ , then the resulting feasible region has no fractional extreme point. Why?

- A *traveling-salesman tour* is simply a 2-factor that is connected.
- IP Formulation of the TSP

By adding so-called subtour elimination constraints, we have an IP formulation of the symmetric (i.e. undirected) TSP

$$\begin{array}{ll}
 \text{(IP-2F)} & \min w(x) := \sum_{(i,j) \in E} w_{ij} x_{ij} \\
 & \text{subject to} \quad \sum_{(i,j) \in E} x_{ij} = 2, \quad \forall i \in V \\
 & \quad \quad \quad \sum_{i,j \in S, (i,j) \in E} x_{ij} \leq |S| - 1, \quad \emptyset \neq \forall S \subsetneq V \\
 & \quad \quad \quad x_{ij} = 0 \text{ or } 1, \quad \forall (i,j) \in E.
 \end{array}$$

Remark that the number of subtour elimination constraints is exponential in  $n$ ,  $2^n - 2$ .

- The subtour elimination constraints above can be replaced by equivalent constraints expressing that every TSP tour uses at least two edges from any nontrivial cutset, namely,  $\sum_{i \in S, j \notin S} x_{ij} \geq 2$ , for  $\emptyset \neq \forall S \subsetneq V$ .

## Preview of Basic Topics in Integer Programming

- Solving a System of Linear Equations over Integers

An equation is equivalent to two opposite inequalities. Solving a system of equations is often much simpler than solving a system of inequalities. This is certainly the case when one is considering integer programming.

How difficult is to solve a system of linear equations over integers? More explicitly, for given  $A \in \mathbb{Q}^{m \times n}$  and  $b \in \mathbb{Q}^m$ ,

$$\text{find } x \in \mathbb{Z}^n \text{ such that } Ax = b \text{ or show that such } x \text{ does not exist.} \quad (1.3)$$

Such a system is known as a *linear diophantine* (matrix) equation. It turns out that solving the linear diophantine equation is polynomially solvable, by using a natural extension of Euclid's algorithm for computing GCD (greatest common divisor) of integers. It terminates either to find a solution  $x \in \mathbb{Z}^n$  satisfying  $Ax = b$  or a certificate for infeasibility, namely,  $\lambda \in \mathbb{R}^m$  such that  $A^T \lambda$  is an integer vector and  $b^T \lambda$  is fractional.

When there is no integer solution to the system  $Ax = b$ , one can still look for a good approximation. Namely, the *closest vector problem* looks for a point  $Ax$  closest to  $b$  in Euclidean distance, more explicitly,

$$\min\{\|b - y\| : y = Ax, x \in \mathbb{Z}^n\}. \quad (1.4)$$

A closely related problem is the *shortest vector problem* to find a shortest nonzero vector in the lattice generated by  $A$ ,

$$\min\{\|y\| : \mathbf{0} \neq y = Ax, x \in \mathbb{Z}^n\}. \quad (1.5)$$

The shortest vector problem (1.5) can be polynomially reducible to the closest vector problem (1.4). The decision problems of these problems are known to be NP-complete. Approximation algorithms have been proposed for both problems. One powerful technique is Lovász's basis reduction method.

The problem can be casted more geometrically. The *lattice*  $L(A)$  generated by the *matrix*  $A$  is defined by

$$L(A) = \{y : y = Ax, x \in \mathbb{Z}^n\}. \quad (1.6)$$

Figure 1.1 shows the lattice generated by the matrix  $A = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}$ . The linear diophantine equation asks whether a given vector  $b$  is a lattice point in  $L(A)$ . The shortest vector problem asks for a lattice point closest to the origin, while the closest point problem asks for a lattice point closest to a given point  $b$ .

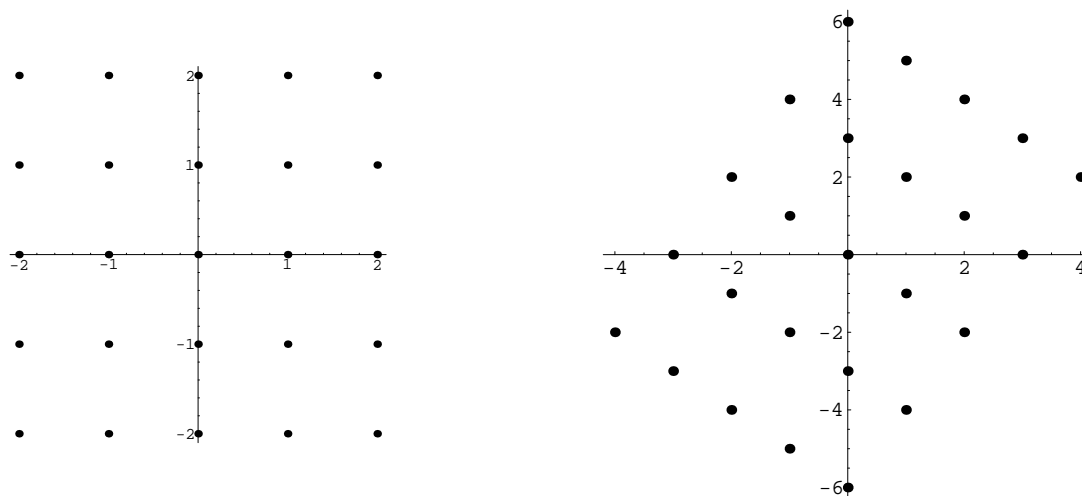


Figure 1.1: The Standard Lattice  $\mathbb{Z}^2$  and the Lattice  $L(A)$  generated by  $A$ .

- Integer Hull

There are two convex polyhedra associated with the integer programming problem

$$\max c^T x \text{ subject to } Ax \leq b, x \geq \mathbf{0}, x \in \mathbb{Z}^n.$$

The first one is simply the feasible region of its LP relaxation:

$$P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}\}, \quad (1.7)$$

and the second one is the *integer hull* polyhedron, i.e., the convex hull of the set of all integer feasible solutions:

$$P_I = \text{conv}\{x \in \mathbb{Z}^n : Ax \leq b, x \geq \mathbf{0}\}. \quad (1.8)$$

Clearly,  $P_I \subseteq P$  and the inclusion is proper in general. In some very special cases, the equality is satisfied like in the case of the optimal assignment problem. While  $P$  is a polyhedron by definition, it is not clear as to whether  $P_I$  is a polyhedron, i.e., it is representable as the solution set to a system of finite linear inequalities. We will show that it is a polyhedron with a reasonable complexity. What is reasonable? For example, all extreme points of  $P_I$  are small in the sense that the binary size of each coordinate is polynomially bounded by the binary sizes of  $A$  and  $b$ .

- Cutting Planes

One common way is to repeat the following procedure:

- solve the LP relaxation by any efficient algorithm,
- if the optimal solution to the LP is integer, then it solves the IP,
- otherwise, introduce a new linear inequality (“a cutting plane”) which is valid for all integer feasible solutions and violated by the LP optimal solution.

There are different ways to generate cutting planes. The best-known is Gomory's cutting plane which is based on the LP tableau/dictionary. It is important to note that the procedure may not terminate in a finite steps unless some clever tactics are incorporated. Lagrangian

- Lagrangian Relaxation

Consider the IP with two sets of linear inequalities:

$$\begin{aligned} z^* = \max \quad & c^T x \\ \text{subject to} \quad & A_1 x \leq b_1 \\ & A_2 x \leq b_2 \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{1.9}$$

Here, the first set  $A_1 x \leq b_1$  of, say  $m_1$  constraints, is meant to be easy to deal with under the integer restrictions  $x \in \mathbb{Z}^n$ , and the second set  $A_2 x \leq b_2$  of  $m_2$  constraints is somehow very hard to deal with. For example, for the symmetric TSP, the first set with  $x \in \mathbb{Z}^n$  may represent the 2-factor constraints and the second set represents the subtour elimination constraints.

The Lagrangian relaxation is a technique to relax the original problem by removing the second set of hard constraints from optimization and by adding a penalty term in the objective in the way that the optimal value does not decrease. More explicitly, for any  $y \geq \mathbf{0}$  (where  $y \in \mathbb{R}^{m_2}$ ),

$$\begin{aligned} f(y) := \max \quad & c^T x + y^T (b_2 - A_2 x) \\ \text{subject to} \quad & A_1 x \leq b_1 \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{1.10}$$

Observe that for any  $y \geq \mathbf{0}$ ,  $f(y)$  is an upper bound of  $z^*$ , and because of our assumption, the problem can be solved efficiently. Moreover, the function  $f(y)$  is convex over  $y \geq \mathbf{0}$  (why?). Because of this convexity, finding the best (least) upper bound is generally a well behaving problem (where the local optimality ensures the global optimality):

$$\min_{y \geq \mathbf{0}} \{ y^T b_2 + \max_{A_1 x \leq b_1, x \in \mathbb{Z}^n} (c^T - y^T A_2) x \}. \tag{1.11}$$

Since the function  $f(y)$  is piecewise-linear and in particular nondifferentiable, a standard technique for solving (1.11) is the subgradient method.

## 2 Euclidean Algorithm and Hermite Normal Form

### 2.1 Sizes of Rational Numbers and Polynomial Complexity

Whenever we evaluate the complexity of an algorithm, we use the **binary encoding length of input data** as input size and we bound the number of arithmetic operations necessary to solve the worst-case problem instance of the same input size. Also, in order to claim a **polynomial complexity**, it is not enough that the number of required arithmetic operations is bounded by a polynomial function in the input size, but also, the largest size of numbers generated by the algorithm must be bounded by a polynomial function in the input size. Here we formally define the sizes of a rational number, a rational vector and a rational matrix.

Let  $r = p/q$  be a rational number with canonical (i.e. relatively prime) representation with  $p \in \mathbb{Z}$  and  $q \in \mathbb{N}$ . We define the *binary encoding size* of  $r$  as

$$\text{size}(r) := 1 + \lceil \log_2(|p| + 1) \rceil + \lceil \log_2(q + 1) \rceil. \quad (2.1)$$

The *binary encoding size* of a rational vector  $v \in \mathbb{Q}^n$  and that of a rational matrix  $A \in \mathbb{Q}^{m \times n}$  are defined by

$$\text{size}(v) := n + \sum_{j=1}^n \text{size}(v_j), \quad (2.2)$$

$$\text{size}(A) := mn + \sum_{i=1}^m \sum_{j=1}^n \text{size}(a_{ij}). \quad (2.3)$$

**Exercise 2.1** For any two rational numbers  $r$  and  $s$ , show that

$$\begin{aligned} \text{size}(r \times s) &\leq \text{size}(r) + \text{size}(s), \\ \text{size}(r + s) &\leq 2(\text{size}(r) + \text{size}(s)). \end{aligned}$$

Can one replace the constant 2 by 1 in the second inequality?

**Theorem 2.1** *Let  $A$  be a rational square matrix. Then the size of its determinant is polynomially bounded, and more specifically,  $\text{size}(\det(A)) < 2 \text{size}(A)$ .*

**Proof.** Let  $p/q$  be the canonical representation of  $\det(A)$ , let  $p_{ij}/q_{ij}$  denote that of each entry  $a_{ij}$  of  $A$ , and let  $\delta$  denote  $\text{size}(A)$ .

First, we observe

$$q \leq \prod_{i,j} q_{ij} < 2^{\delta-1}, \quad (2.4)$$

where the last inequality can be verified by taking  $\log_2$  of the both sides. By the definition of determinant, we have

$$|\det(A)| \leq \prod_{i,j} (|p_{ij}| + 1). \quad (2.5)$$

Combining (2.4) and (2.5),

$$|p| = |\det(A)|q \leq \prod_{i,j} (|p_{ij}| + 1)q_{ij} < 2^{\delta-1}, \quad (2.6)$$

where the last inequality again is easily verifiable by taking  $\log_2$  of both sides. Then it follows from (2.4) and (2.6) that

$$\text{size}(\det(A)) = 1 + \lceil \log_2(|p| + 1) \rceil + \lceil \log_2(q + 1) \rceil \leq 1 + (\delta - 1) + (\delta - 1) < 2\delta. \quad (2.7)$$

■

**Corollary 2.2** *Let  $A$  be a rational square matrix. Then the size of its inverse is polynomially bounded by its size  $\text{size}(A)$ .*

**Corollary 2.3** *If  $Ax = b$ , a system of rational linear equations, has a solution, it has one polynomially bounded by the sizes of  $A$  and  $b$ .*

Here is a theorem due to Jack Edmonds (1967) who gave an elegant proof for it.

**Theorem 2.4** *Let  $Ax = b$  be a system of rational linear equations. Then, there is a polynomial-time algorithm (based on the Gaussian or the Gauss-Jordan elimination) to find either a solution  $x$  or a certificate of infeasibility, namely,  $\lambda$  such that  $\lambda^T A = \mathbf{0}$  and  $\lambda^T b \neq 0$ .*

## 2.2 Computing the GCD

For given two positive integers  $a$  and  $b$ , the following iterative procedure finds the greatest common divisor (GCD):

```

procedure EuclideanAlgorithm( $a, b$ );
begin
  if  $a < b$  then swap  $a$  and  $b$ ;
  while  $b \neq 0$  do
    begin
       $a := a - \lfloor a/b \rfloor \times b$ ;
      swap  $a$  and  $b$ ;
    end;
  output  $a$ ;
end.

```

Note that the algorithm works not only for integers but rational  $a$  and  $b$ .

**Exercise 2.2** Apply the algorithm to  $a = 212$  and  $b = 288$ .

**Exercise 2.3** Explain why it is a correct algorithm for GCD. Analyze the complexity of the algorithm in terms of the sizes of inputs  $a$  and  $b$ . How can one extend the algorithm to work with  $k$  positive integers?

This algorithm does a little more than computing the GCD. By looking at the matrix operations associated with this, we will see that the algorithm does much more. Let's look at the two operations the algorithm uses in terms of matrices:

$$[a, b] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = [b, a] \quad (2.8)$$

$$[a, b] \begin{bmatrix} 1 & 0 \\ -\lfloor a/b \rfloor & 1 \end{bmatrix} = [a - \lfloor a/b \rfloor \times b, b]. \quad (2.9)$$

It is important to note that the two elementary transformation matrices

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ (Swapping),} \quad \begin{bmatrix} 1 & 0 \\ -\lfloor a/b \rfloor & 1 \end{bmatrix} \text{ (Remainder)} \quad (2.10)$$

are integer matrices and have determinant equal to  $+1$  or  $-1$ . This means that the transformations preserve the existence of an integer solution to the following linear equation:

$$[a, b] \begin{bmatrix} x \\ y \end{bmatrix} = c \quad (\text{i.e., } ax + by = c). \quad (2.11)$$

Let  $T \in \mathbb{Z}^{2 \times 2}$  be the product of all transformation matrices occurred during the Euclidean algorithm applied to  $a$  and  $b$ . This means that  $|\det T| = 1$  and

$$[a, b] T = [a', 0], \quad (2.12)$$

where  $a'$  is the output of the Euclidean algorithm and thus  $\text{GCD}(a, b)$ .

Now, we see how the algorithm finds the general solution to the linear diophantine equation:

$$\text{find } \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{Z}^2 \text{ satisfying } [a, b] \begin{bmatrix} x \\ y \end{bmatrix} = c. \quad (2.13)$$

Once the transformation matrix  $T$  is computed, the rest is rather straightforward. Since  $T$  is integral and has determinant  $-1$  or  $1$ , the following equivalence follows.

$$\begin{aligned} \left\langle \exists \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{Z}^2 : [a, b] \begin{bmatrix} x \\ y \end{bmatrix} = c \right\rangle &\Leftrightarrow \left\langle \exists \begin{bmatrix} x' \\ y' \end{bmatrix} \in \mathbb{Z}^2 : [a, b] T \begin{bmatrix} x' \\ y' \end{bmatrix} = c \right\rangle \\ &\Leftrightarrow \left\langle \exists \begin{bmatrix} x' \\ y' \end{bmatrix} \in \mathbb{Z}^2 : [a', 0] \begin{bmatrix} x' \\ y' \end{bmatrix} = c \right\rangle \Leftrightarrow \langle a' | c \text{ (} a' \text{ divides } c) \rangle. \end{aligned}$$

Finally, when  $a' | c$ , let  $x' := c/a'$  and  $y'$  be any integer. Then,

$$\begin{bmatrix} x \\ y \end{bmatrix} := T \begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} c/a' \\ y' \end{bmatrix} \quad (2.14)$$

with  $y' \in \mathbb{Z}$  is the general solution to the diophantine equation (2.13).

### 2.3 Computing the Hermite Normal Form

By extending the Euclidean algorithm (in matrix form) to a system of linear equations in several integer variables, we obtain a procedure to solve the linear diophantine problem:

$$\text{find } x \in \mathbb{Z}^n \text{ satisfying } Ax = b, \quad (2.15)$$

where  $A \in \mathbb{Z}^{m \times n}$  and  $b \in \mathbb{Z}^m$  are given. We assume that  $A$  is full row rank. (Otherwise, one can either reduce the problem to satisfy this condition or show that there is no  $x \in \mathbb{R}^n$  satisfying  $Ax = b$ . How?)

Note that a seemingly more general problem of rational inputs  $A$  and  $b$  can be easily scaled to an equivalent problem with integer inputs.

**Theorem 2.5** *There is a finite algorithm to find an  $n \times n$  integer matrix  $T$  with  $|\det T| = 1$  such that  $AT$  is of form  $[B \mathbf{0}]$ , where  $B = [b_{ij}]$  is an  $m \times m$  nonnegative nonsingular lower-triangular integer matrix with  $b_{ii} > 0$  and  $b_{ij} < b_{ii}$  for all  $i = 1, \dots, m$  and  $j = 1, \dots, i - 1$ .*

This matrix  $[B \mathbf{0}]$  is known as the *Hermite normal form*, and it will be shown that it is unique.

**Corollary 2.6** *The linear diophantine problem (2.15) has no solution  $x$  if and only if there is  $z \in \mathbb{Q}^n$  such that  $z^T A$  is integer and  $z^T b$  is fractional.*

**Proof.** The “if” part is trivial. To prove the “only if” part, we assume that  $\exists x \in \mathbb{Z}^n : Ax = b$ . Let  $T$  be the integer matrix given by Theorem 2.5. Because  $|\det T| = 1$ , we have the following equivalence:

$$\begin{aligned} \langle \exists x \in \mathbb{Z}^n : Ax = b \rangle &\Leftrightarrow \langle \exists x' \in \mathbb{Z}^n : ATx' = b \rangle \Leftrightarrow \langle \exists x' \in \mathbb{Z}^n : [B \mathbf{0}]x' = b \rangle \\ &\Leftrightarrow \langle B^{-1}b \text{ is not integer} \rangle. \end{aligned}$$

Since  $B^{-1}b$  is not integer, there is a row vector  $z^T$  of  $B^{-1}$  such that  $z^T b$  is fractional. Since  $B^{-1}AT = [I \mathbf{0}]$ , we know that  $B^{-1}A = [T^{-1} \mathbf{0}]$  and it is an integer matrix as  $|\det T| = 1$ . This implies that  $z^T A$  is integer. This completes the proof. ■

As for the single diophantine equation, one can write the general solution to the linear diophantine problem (2.15).

**Corollary 2.7** *Let  $A \in \mathbb{Q}^{m \times n}$  be a matrix of full row rank,  $b \in \mathbb{Q}^m$ , and  $AT = [B \mathbf{0}]$  be an Hermite normal form of  $A$ . Then the following statements hold.*

- (a) *The linear diophantine problem (2.15) has a solution if and only if  $B^{-1}b$  is integer.*
- (b) *If  $B^{-1}b$  is integer, then the general solution  $x$  to (2.15) can be written as*

$$x = T \begin{bmatrix} B^{-1}b \\ z \end{bmatrix}, \quad (2.16)$$

*for any  $z \in \mathbb{Z}^{n-m}$ .*

Now, we are going to prove the main theorem, Theorem 2.5.

**Proof.** (of Theorem 2.5). Extending the operations we used in (2.10), our proof of Theorem 2.5 involves three elementary matrix (column) operations on  $A$ :

(c-0) multiplying a column of  $A$  by  $-1$ ;

(c-1) swapping the positions of two columns of  $A$ ;

(c-2) adding an integer multiple of a column to another column of  $A$ .

Both (c-1) and (c-2) were already used and (c-0) is merely to deal with negative entries which are allowed in  $A$ . Each operation can be written in form  $A T$ , where  $T$  is a *unimodular* matrix, i.e., an integer matrix of determinant equal to  $-1$  or  $1$ .

The algorithm operates on the first row, the second and to the last row. We may assume that we have already transformed the first  $k$  ( $\geq 0$ ) rows properly, namely, we have a sequence of matrices  $T_1, T_2, \dots, T_s$  such that  $T = T_1 T_2 \cdots T_s$  and

$$A_k := A T = \begin{bmatrix} B' & \mathbf{0} \\ C & A' \end{bmatrix} \quad (2.17)$$

where  $B'$  is a  $k \times k$  matrix which is already in the form required for the final  $B$ , namely, it is a nonnegative nonsingular lower-triangular integer matrix with  $b'_{ii} > 0$  and  $b'_{ij} < b'_{ii}$  for all  $i = 1, \dots, k$  and  $j = 1, \dots, i-1$ . Now we process the  $k$ th row of  $A_k$ , and essentially the first row of  $A'$ . The first row of  $A'$  contains  $n - k$  integers and the rest is written by  $A''$ :

$$A' = \begin{bmatrix} a'_{11}, a'_{12}, \dots, a'_{1(n-k)} \\ A'' \end{bmatrix} \quad (2.18)$$

Now we apply the Euclidean algorithm to find the GCD, say  $\alpha$ , of the  $n - k$  integers. For this, we first use (c-0) operations to make the numbers all nonnegative. Then remaining operations are straightforward with (c-1) and (c-2) to convert the first row to  $[\alpha, 0, 0, \dots, 0]$ . This in the form of the whole matrix  $A_k$  looks like (for some  $T'$ ),

$$A_k T' = \begin{bmatrix} B' & \mathbf{0} \\ C & \alpha, 0, 0, \dots, 0 \\ & A''' \end{bmatrix}. \quad (2.19)$$

Note that  $\alpha$  is strictly positive because of the full row rank assumption. Finally, we can reduce the entries in the first row of  $C$  by adding some integer multiples of  $\alpha$  in the  $(k+1)$ st column (for some  $T''$ ) as

$$A_k T' T'' = \begin{bmatrix} B' & \mathbf{0} \\ c'_{11}, \dots, c'_{1k} & \alpha, 0, 0, \dots, 0 \\ C'' & A''' \end{bmatrix}, \quad (2.20)$$

so that all entries  $c'_{11}, \dots, c'_{1k}$  are smaller than  $\alpha$ . Now we have made the principal  $(k+1) \times (k+1)$  matrix in the form of the final  $B$ . This completes the proof.  $\blacksquare$

While the algorithm is finite, it is not clear how good this is. It has been observed that the largest size of numbers appearing during the course of the algorithm grows rapidly. There are ways to make the procedure polynomial. We will discuss this issue later.

**Example 2.1** The following small example is simple enough to calculate by hand.

$$A = \begin{bmatrix} -8 & 10 & -4 \\ -4 & -2 & 8 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 2 & 0 & 0 \\ 14 & 20 & -36 \end{bmatrix},$$

$$A_2 = [B \mathbf{0}] = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 4 & 0 \end{bmatrix}.$$

The transformation matrix  $T$  with  $A_2 = A T$  is

$$T = \begin{bmatrix} 6 & -2 & 9 \\ 7 & -2 & 10 \\ 5 & -1 & 7 \end{bmatrix}.$$

**Example 2.2** The following small example (randomly generated) shows how numbers grow rapidly. Of course, this example is not meant to be computed by hand.

$$A = \begin{bmatrix} -100 & -32 & 140 & 168 & 147 \\ 68 & -16 & -125 & 168 & 7 \\ -12 & -28 & -50 & 147 & -133 \\ -60 & 64 & -65 & 28 & 28 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -523 & 944 & 159 & 320 & 1976 \\ -115 & 604 & 54 & 151 & 388 \\ 976 & -2080 & -565 & -156 & -3652 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1489619 & -2848 & -495 & 5739 & -1722 \\ -37305137 & -71331 & -6180 & 143636 & -29988 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 \\ -299296004657 & -572624931 & -602688 & 309680 & 1400700 \end{bmatrix},$$

$$A_4 = [B \mathbf{0}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 \\ 43 & 129 & 12 & 140 & 0 \end{bmatrix}.$$

The transformation matrix  $T$  with  $A_4 = A T$  is

$$T = \begin{bmatrix} -807814365429333 & -1545542680854 & -1626716396 & -377867 & 1101240 \\ -1448925874428057 & -2772142804282 & -2917738997 & -677754 & 1975225 \\ -731120268289411 & -1398808473625 & -1472275536 & -341992 & 996688 \\ -365381147997122 & -699061793372 & -735777339 & -170912 & 498100 \\ 248937455097979 & 476277073276 & 501291704 & 116444 & -339360 \end{bmatrix}.$$

**Observation 2.8** *In the example above, the numbers appearing during the course of the algorithm seem to grow. This is a fact commonly observed. Yet, it is not known if our algorithm is exponential. There are ways to modify the algorithm so that it runs in polynomial-time.*

**Exercise 2.4** Write a computer program to compute a Hermite normal form of any integer matrix  $A$  of full row rank. For this, one needs to use an environment where infinite precision integer arithmetic is supported, e.g., C/C++ with GNU gmp, Mathematica, Maple, and Sage.

## 2.4 Lattices and the Hermite Normal Form

There is a closed relationship between the Hermite normal form of a matrix  $A \in \mathbb{Q}^{m \times n}$  and the lattice generated by (the columns of)  $A$ . Recall that the lattice  $L(A)$  generated by  $A$  is defined by

$$L(A) = \{y : y = Ax, x \in \mathbb{Z}^n\}. \quad (2.21)$$

The lattice  $L(A)$  is *full dimensional* if it spans the whole space  $\mathbb{R}^m$ , or equivalently,  $A$  is full row rank.

**Lemma 2.9** *Let  $A$  be rational matrix of full row rank with Hermite normal form  $[B \mathbf{0}]$ . Then,  $L(A) = L([B \mathbf{0}])$ .*

**Proof.** This follows directly from the fact that  $AT = [B \mathbf{0}]$  for some unimodular matrix  $T$ . ■

**Theorem 2.10** *Let  $A$  and  $A'$  be rational matrices of full row rank with Hermite normal forms  $[B \mathbf{0}]$  and  $[B' \mathbf{0}]$ , respectively. Then the matrices  $A$  and  $A'$  generate the same lattice (i.e.  $L(A) = L(A')$ ) if and only if  $B = B'$ .*

**Proof.** Clearly, the sufficiency is clear: if  $B = B'$ ,  $L(A) = L(A')$ .

Assume that  $L := L(A) = L(A')$ . Then, by Lemma 2.9,  $L = L(B) = L(B')$ . Now we show that the  $k$ th columns  $B_{\cdot k}$  and  $B'_{\cdot k}$  are equal for  $k = 1, \dots, n$ . First, observe that  $B_{\cdot k}$  and  $B'_{\cdot k}$  are in  $L$  with the property (\*) that the first  $(k - 1)$  components are all zero and the  $k$ th component is positive. Because  $B$  is in Hermite normal form, it follows that  $B_{\cdot k}$  is a vector in  $L$  satisfying (\*) with its  $k$ th component being smallest possible. Since the same thing can be said about  $B'_{\cdot k}$ ,  $b_{kk} = b'_{kk}$  for  $k = 1, \dots, n$ . In addition, because  $B$  is in Hermite normal form,  $B_{\cdot k}$  is a lexicographically smallest vector of nonnegative components satisfying (\*), and so is  $B'_{\cdot k}$ . Such a vector is unique, and thus  $B_{\cdot k} = B'_{\cdot k}$ . ■

**Corollary 2.11** *Every rational matrix of full row rank has a unique Hermite normal form.*

A *basis* of a full dimensional lattice  $L(A)$  is a nonsingular matrix  $B$  such that  $L(A) = L(B)$ . A direct consequence of Theorem 2.5 is

**Corollary 2.12** *Every rational matrix of full row rank has a basis.*

**Exercise 2.5** Let  $A$  be a rational matrix of full row rank, let  $B$  be a bases of  $L(A)$  and let  $B'$  be a nonsingular  $n \times n$  matrix whose column vectors are points in  $L(A)$ . Show the following statements are valid:

(a)  $|\det(B)| \leq |\det(B')|$ .

(b)  $B'$  is a basis of  $L(A)$  if and only if  $|\det(B)| = |\det(B')|$ .

Using the fact (a) above, we show that the size of the Hermite normal form is small.

**Theorem 2.13** *The size of the Hermite normal form of a rational matrix  $A$  of full row rank is polynomially bounded by  $\text{size}(A)$ .*

**Proof.** Let  $[B \ \mathbf{0}]$  be the Hermite normal form of  $A$ , and let  $B'$  be any basis (i.e. nonsingular  $m \times m$  submatrix) of  $A$  (which is not necessarily a basis of  $L(A)$ ). First of all,  $\det(B) > 0$ . By Exercise 2.5 (a), we have  $\det(B) \leq |\det(B')|$ . By Theorem 2.1, this inequality implies that the size of  $\det(B)$  is polynomially bounded by  $\text{size}(A)$ .

Since  $B$  is lower triangular,  $\det(B)$  is the product of diagonal entries  $b_{ii}$ , ( $i = 1, \dots, n$ ). It follows that the size of each entry  $b_{ii}$  is less than the size of  $\det(B)$ , and thus is polynomially bounded by  $\text{size}(A)$ . Since  $B$  is in Hermite normal form, each nondiagonal entry  $b_{ij}$  is less than or equal to  $b_{ii}$  and has the same property. ■

The theorem above suggests that the Hermite Normal Form might be computable in polynomial time. In fact, there are methods to control the largest size of numbers generated during the course of the algorithm given in the previous section.

One such algorithm is as follows. First of all, a given matrix  $A$  of full row rank, is enlarged to

$$\widehat{A} := \left[ A \left| \begin{array}{cccc} M & 0 & \cdots & 0 \\ 0 & \ddots & & 0 \\ 0 & \cdots & 0 & M \end{array} \right. \right], \quad (2.22)$$

where  $M$  is set to be the positive integer  $|\det(B')|$  for an arbitrarily chosen basis  $B'$  of  $A$ . The first observation is that this new matrix generates the same lattice as  $A$ .

**Exercise 2.6** Show that  $L(\widehat{A}) = L(A)$ .

Thus, computing the Hermite normal form of  $\widehat{A}$  is equivalent to that of  $A$ . Now, since we have the added columns, it is possible to reduce the entries appearing in the first  $n$  columns by adding proper multiples of the last  $m$  columns, so that all entries are nonnegative and at most  $M$ . This reduction should be applied before the Euclidean algorithm is applied to each row. Since  $\text{size}(M)$  is polynomially bounded by  $\text{size}(A)$ , one can control the number of arithmetic operations and the size of numbers appearing in the application of Euclidean algorithm applied to each row of  $\widehat{A}$ .

The sources of the ideas and more detailed discussion can be found in Shrijver's book [11, Section 5.3]. One important consequence of Theorem 2.13, Corollary 2.3 and Theorem 2.4 is the following.

**Corollary 2.14** *For any rational matrix  $A$  of full row rank, there is a transformation matrix  $T$  such that  $AT$  is the Hermite normal form of  $A$  and  $\text{size}(T)$  is polynomially bounded by the size of  $A$ . Furthermore, such a matrix can be computed in polynomial time.*

## 2.5 Dual Lattices

For a lattice  $L$  in  $\mathbb{R}^m$ , its *dual lattice*  $L^*$  is defined by

$$L^* := \{y \in \mathbb{Q}^m : y^T z \in \mathbb{Z}, \forall z \in L\}. \tag{2.23}$$

If  $L$  is generated by a matrix  $A$ ,

$$L^* = (L(A))^* = \{y \in \mathbb{Q}^m : y^T A \in \mathbb{Z}^n\}. \tag{2.24}$$

In terms of the Hermite normal form  $[B \ \mathbf{0}]$  of  $A$  (assuming  $A$  is full row rank), the dual lattice is generated by the transpose (and the rows) of  $B^{-1}$ , that is,

$$L^* = L((B^{-1})^T). \tag{2.25}$$

Why is it so? To see that, set  $L' = L((B^{-1})^T)$  and we will show  $L' = L^*$ . Since  $B^{-1}B = I$  and  $L$  is generated by (the columns of)  $B$ , each row of  $B^{-1}$  has an integer inner product with any vector in  $L$ . This shows  $L' \subseteq L^*$ . For the converse, take any vector  $y$  in  $L^*$ . Let  $s^T = y^T B$ . By definition of  $L^*$ ,  $s \in \mathbb{Z}^m$ . Observing that  $y^T = s^T B^{-1}$ ,  $y$  is an integer linear combination of the rows of  $B^{-1}$ . This proves  $L' \supseteq L^*$  and completes the proof.

**Example 2.3** Figure 1.1 depicts the lattice generated by  $A = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}$ . The Hermite normal form is  $B = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}$  and its inverse is  $B^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{2}{3} & \frac{1}{3} \end{bmatrix}$ . While the primal lattice  $L(A)$  is generated by the columns of  $B$ , the dual lattice  $(L(A))^*$  is generated by the rows of  $B^{-1}$ . Figure 2.1 depicts both the primal and the dual lattices (partially) on the same scale.

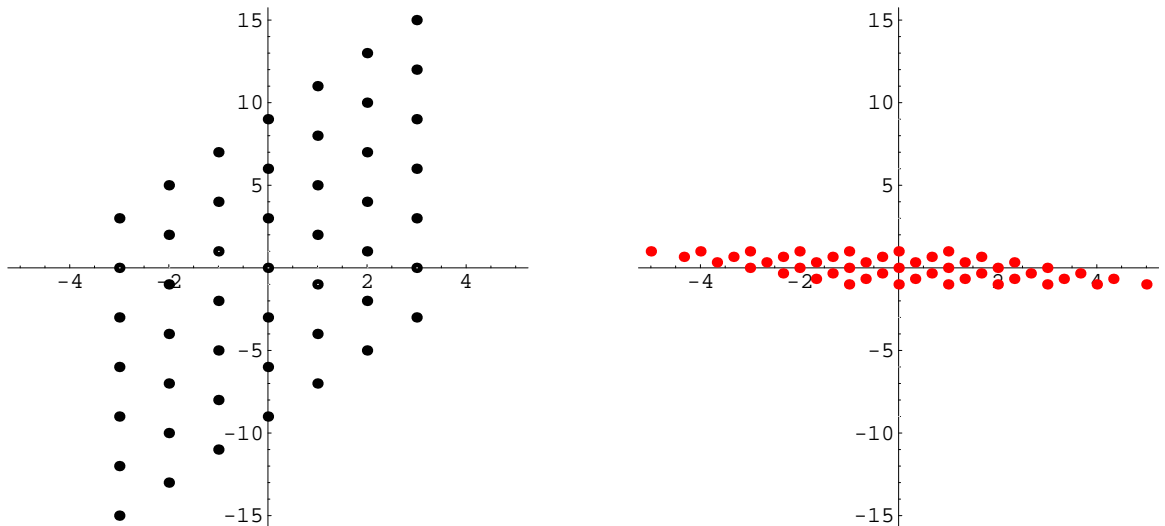


Figure 2.1: Lattice  $L(A)$  and its Dual Lattice.

## 3 Linear Inequalities, LP and Polyhedra

### 3.1 Systems of Linear Inequalities

Consider a system of rational linear inequalities

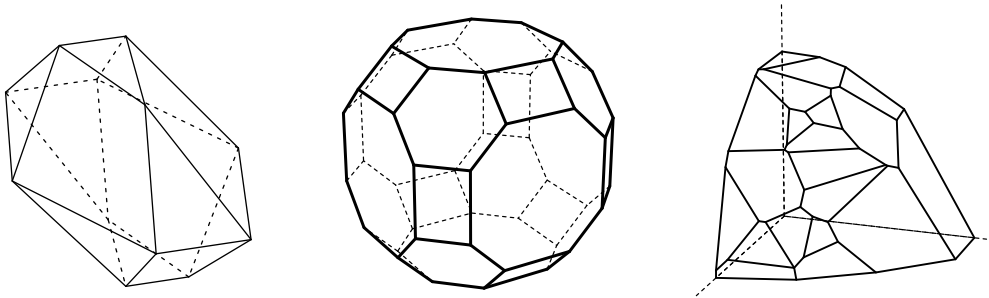
$$Ax \leq b, \quad (3.1)$$

where  $A \in \mathbb{Q}^{m \times n}$  and  $b \in \mathbb{Q}^m$  are given. The set

$$P = P(A, b) := \{x \in \mathbb{R}^n : Ax \leq b\} \quad (3.2)$$

of solutions to the system is a subset of  $\mathbb{R}^n$ , known as a *convex polyhedron*. It is in fact a convex set: a subset  $C$  of  $\mathbb{R}^n$  is said to be *convex* if the line segment  $[u, v] := \{x : x = \alpha u + (1 - \alpha)v, 0 \leq \alpha \leq 1\}$  between any two points  $u$  and  $v$  in  $C$  is entirely contained in  $C$ . A bounded convex polyhedron is called a *convex polytope*.

Below, the first two are centrally symmetric polytopes in  $\mathbb{R}^3$ , and the third one is randomly generated. One can interpret the third one as a bounded polyhedron (i.e. polytope) contained in the nonnegative orthant or as an unbounded polyhedron having only its non-negative orthant part drawn.



### 3.2 The Fourier-Motzkin Elimination

Consider a system (3.1) of  $m$  linear inequalities in  $n$  variables. Solving such a system means either to find a rational vector  $x$  satisfying the system or to detect inconsistency of the system. The latter can be proven by a certificate given by the well-known theorem of Gale:

**Theorem 3.1 (Gale's Theorem)** *For any  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$ , exactly one of the following statements holds:*

- (a) *there exists  $x \in \mathbb{R}^n$  such that  $Ax \leq b$ ;*
- (b) *there exists  $z \in \mathbb{R}^m$  such that  $z \geq \mathbf{0}$ ,  $z^T A = \mathbf{0}$  and  $z^T b < 0$ .*

It is easy to see that both statements cannot hold simultaneously as it would mean 0 is less than or equal to some (strictly) negative number. Thus the nontrivial part of the theorem is that one of (a) and (b) is always valid. There are several constructive proofs of the theorem.

Here we present an arguably simplest constructive proof due to Fourier and Motzkin. The main idea is to transform the system (3.1) to an equivalent system of the same form with one less variables.

First we rewrite the system (3.1) by looking at the coefficients  $a_{in}$ 's for the last variable  $x_n$ . Let us define a partition of row indices into the three sets:

$$I^+ := \{i \mid a_{in} > 0\}, \quad I^- := \{i \mid a_{in} < 0\} \quad \text{and} \quad I^0 := \{i \mid a_{in} = 0\}.$$

The system (3.1) can be rewritten by solving each inequality with respect to  $x_n$ :

$$\begin{array}{ll} x_n \leq f_i(x') & \forall i \in I^+ \\ g_j(x') \leq x_n & \forall j \in I^- \\ h_k(x') \leq 0 & \forall k \in I^0, \end{array}$$

where  $x'$  is the vector  $x$  with the last component eliminated, i.e.  $x' = (x_1, \dots, x_{n-1})^T$  and each functions  $f_i$ ,  $g_j$  and  $h_k$  denote some affine functions in  $n - 1$  variables.

It is not difficult to show (Exercise 3.1) that the system (3.1) is equivalent to the new system in  $n - 1$  variables:

$$\begin{array}{ll} g_j(x') \leq f_i(x') & \forall (i, j) \in I^+ \times I^- \\ h_k(x') \leq 0 & \forall k \in I^0. \end{array}$$

This system can thus be written as

$$A'x' \leq b'. \tag{3.3}$$

**Exercise 3.1** Prove the equivalence:  $Ax \leq b \Leftrightarrow A'x' \leq b'$ .

No reason to stop here. Let's continue to eliminate the variable  $x_{n-1}$ , then  $x_{n-2}$  and so forth until all variables are eliminated. This generates a sequence of equivalent systems of linear inequalities:

$$\begin{array}{l} A^{(0)}x^{(0)} \leq b^{(0)} \text{ (This is the original system } Ax \leq b.) \\ \Downarrow \\ A^{(1)}x^{(1)} \leq b^{(1)} \text{ (This is } A'x' \leq b' \text{ above.)} \\ \Downarrow \\ A^{(2)}x^{(2)} \leq b^{(2)} \\ \Downarrow \\ \vdots \\ \Downarrow \\ A^{(n)}x^{(n)} \leq b^{(n)}, \end{array}$$

where  $A^{(k)}x^{(k)} \leq b^{(k)}$  denotes the  $k$ th system where the last  $k$  variables have been eliminated.

**Exercise 3.2** Show that the elimination step as a matrix transformation. More precisely, there is a matrix  $T$  (depending on  $A$ ) such that the system  $A'x' \leq b'$  is the same system as  $TAx \leq Tb$  up to positive scaling. Note that the last column of the product  $TA$  is totally zero and thus  $TAx$  does not involve the last variable  $x_n$ .

By the exercise above, the last system  $A^{(n)}x^{(n)} \leq b^{(n)}$  can be now written as  $T^{(n)}Ax \leq T^{(n)}b$ . Of course, the left hand side  $T^{(n)}Ax$  is a vector of zero's.

**Exercise 3.3** Prove Theorem 3.1 by using the matrix  $T^{(n)}$ .

**Exercise 3.4** Prove the following forms of alternative theorem using Gale's Theorem. Below, the statement is read as "exactly one of the two statements (a) or (b) holds."

**The Farkas Lemma**

- (a)  $\exists x : Ax = b$  and  $x \geq \mathbf{0}$ ;
- (b)  $\exists z : z^T A \geq \mathbf{0}$  and  $z^T b < 0$ .

**Gordan's Theorem**

- (a)  $\exists x : Ax = \mathbf{0}$  and  $x \gneq \mathbf{0}$ ;
- (b)  $\exists z : z^T A > \mathbf{0}$ .

### 3.3 LP Duality

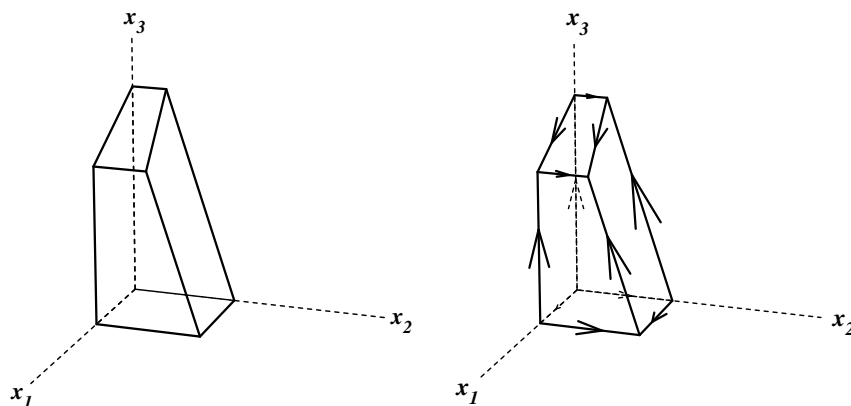
For a given  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ , the linear programming problem (in canonical form) is

$$(P): \quad \begin{array}{l} \max \quad c^T x \\ \text{subject to} \quad Ax \leq b \\ \quad \quad \quad x \geq \mathbf{0}. \end{array} \quad \left| \quad \begin{array}{l} = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i, \forall i = 1, \dots, m \\ x_j \geq 0, \forall j = 1, \dots, n. \end{array} \right.$$

We often abbreviate a linear programming problem as an *LP*. A vector  $x$  satisfying all the constraints  $Ax \leq b$  and  $x \geq \mathbf{0}$  is called a *feasible solution*. An *optimal solution* is a feasible solution that attains the largest objective value. In the case of minimization problem, an optimal solution attains the smallest value.

The set of feasible solutions  $\{x : Ax \leq b, x \geq 0\}$  is called the *feasible region*. An LP is called *feasible* if the feasible region is not empty. An LP is called *unbounded* if the objective function  $c^T x$  is not bounded above (below for the minimization case) over the feasible region.

Geometrically, the feasible region is a *convex polyhedron*.



In general, maximizing or minimizing a linear function subject to a system of linear inequality constraints in  $n$  variables can be reduced to an optimization in the form above. Also, note that no strict inequality constraint such as  $x_1 > 0$  is allowed in linear programming.

There are two fundamental theorems, the weak duality theorem and the strong duality theorem. To state these theorems, we need to define the *dual problem*

$$(D): \quad \min \quad b^T y \\ \text{subject to} \quad A^T y \geq c \\ \quad \quad \quad y \geq \mathbf{0}$$

which is a linear programming problem itself. The original problem (P) is called the *primal problem* when we need to distinguish it from the dual problem.

**Theorem 3.2 (LP Weak Duality Theorem)** *For any feasible solution  $x$  for the primal problem (P) and for any feasible solution  $y$  for the dual problem (D),  $c^T x \leq b^T y$ .*

**Theorem 3.3 (LP Strong Duality Theorem)** *If both the primal problem (P) and the dual problem (D) are feasible, there exist a dual pair  $(x^*, y^*)$  of feasible solutions such that  $c^T x^* = b^T y^*$ . (By the previous theorem, they are both optimal.)*

The first theorem is very easy to prove. Thus it may not be appropriate to call it a theorem, but since it is widely accepted to be so called. Let's prove it.

**Proof.** (of the Weak Duality Theorem, Theorem 3.2) Let  $x$  and  $y$  be a dual pair of feasible solutions. Then,

$$\begin{aligned} c^T x &\leq (A^T y)^T x && \text{(because } A^T y \geq c \text{ and } x \geq \mathbf{0}) \\ &= y^T Ax \\ &\leq y^T b && \text{(because } Ax \leq b \text{ and } y \geq \mathbf{0}) \\ &= b^T y. \end{aligned}$$

This completes the proof. ■

Now, we are ready to prove the second theorem which is much harder to prove.

**Proof.** (of the Strong Duality Theorem, Theorem 3.3) Assume that both the primal problem (P) and the dual problem (D) are feasible. We have to show that

$$\begin{aligned} \exists(x, y) : Ax \leq b, x \geq \mathbf{0} \\ \quad \quad \quad A^T y \geq c, y \geq \mathbf{0} \\ \quad \quad \quad c^T x = b^T y. \end{aligned} \tag{3.4}$$

Now, we verify the statement (3.4) is always valid under the assumption.

$$\begin{aligned}
(3.4) &\Leftrightarrow \left\langle \begin{array}{l} \exists(x, y) : Ax \leq b, x \geq \mathbf{0} \\ A^T y \geq c, y \geq \mathbf{0} \\ c^T x \geq b^T y \end{array} \right\rangle \quad (\text{by the Weak Duality}) \\
&\Leftrightarrow \left\langle \exists \begin{bmatrix} x \\ y \end{bmatrix} \geq \mathbf{0} : \begin{bmatrix} A & \mathbf{0} \\ -I & \mathbf{0} \\ \mathbf{0} & -A^T \\ \mathbf{0} & -I \\ -c^T & b^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} b \\ \mathbf{0} \\ -c \\ \mathbf{0} \\ 0 \end{bmatrix} \right\rangle \\
&\Leftrightarrow \left\langle \nexists \begin{bmatrix} s \\ t \\ u \\ v \\ w \end{bmatrix} \geq \mathbf{0} : \begin{bmatrix} s \\ t \\ u \\ v \\ w \end{bmatrix}^T \begin{bmatrix} A & \mathbf{0} \\ -I & \mathbf{0} \\ \mathbf{0} & -A^T \\ \mathbf{0} & -I \\ -c^T & b^T \end{bmatrix} = \mathbf{0} \text{ and } \begin{bmatrix} s \\ t \\ u \\ v \\ w \end{bmatrix}^T \begin{bmatrix} b \\ \mathbf{0} \\ -c \\ \mathbf{0} \\ 0 \end{bmatrix} < 0 \right\rangle \quad (\text{by Gale's Thm}) \\
&\Leftrightarrow \left\langle \nexists \begin{bmatrix} s \\ u \\ w \end{bmatrix} \geq \mathbf{0} : A^T s \geq cw, Au \leq bw, b^T s < c^T u \right\rangle \\
&\Leftrightarrow \left\langle \nexists \begin{bmatrix} s \\ u \end{bmatrix} \geq \mathbf{0} : A^T s \geq \mathbf{0}, Au \leq \mathbf{0}, b^T s < c^T u \right\rangle \quad (\text{by the Weak Duality}) \\
&\Leftrightarrow \langle A^T s \geq \mathbf{0}, Au \leq \mathbf{0}, s \geq \mathbf{0}, u \geq \mathbf{0} \Rightarrow b^T s \geq c^T u \rangle.
\end{aligned}$$

Now the last step of the proof is to show the last statement above is always true which implies the theorem. Assume

$$A^T s \geq \mathbf{0}, Au \leq \mathbf{0}, s \geq \mathbf{0}, u \geq \mathbf{0}.$$

By the assumption, we have a dual pair  $(x, y)$  of feasible solutions. Thus, we have

$$b^T s - c^T u \geq (Ax)^T s - (A^T y)^T u = x^T A^T s - y^T Au \geq 0 - 0 = 0.$$

This completes the proof. ■

### 3.4 Three Theorems on Convexity

Before we discuss the theory of representations and combinatorial structure of convex polyhedron, it is good to mention some basic facts about convexity.

For any subset  $S$  of  $\mathbb{R}^n$ , the *convex hull*  $\text{conv}(S)$  of  $S$  is the intersection of all convex sets containing  $S$ . Since the intersection of two convex sets is convex, it is the smallest convex set containing  $S$ .

**Proposition 3.4** *Let  $S$  be a subset  $\mathbb{R}^n$ . Then*

$$\text{conv}(S) = \left\{ x : x = \sum_{i=1}^k \lambda_i p_i, \sum_{i=1}^k \lambda_i = 1, \lambda_i \geq 0 \forall i = 1, \dots, k, \right. \quad (3.5)$$

*for some finite points  $p_1, \dots, p_k \in S$ .*

**Proof.** Let the RHS of (3.5) be  $T$ . One has to show both inclusions  $\text{conv}(S) \supseteq T$  and  $\text{conv}(S) \subseteq T$ . Both inclusions are elementary and left to the reader. ■

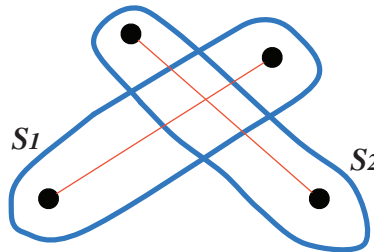
One basic theorem on convexity is Carathéodory's theorem, saying that the finiteness condition on  $k$  in (3.5) can be much more restrictive, namely,  $k \leq n + 1$ .

**Theorem 3.5 (Carathéodory's Theorem)** *Let a point  $p$  be in the convex hull of a set  $S$  of  $k$  points  $p_1, \dots, p_k$  in  $\mathbb{R}^n$ . Then  $p$  is in the convex hull of at most  $n + 1$  points in  $S$ .*

**Proof.** Left to the reader. Hint: When  $k \geq n + 2$ , the points  $p_1, \dots, p_k$  are *affinely dependent*, i.e., there exist  $\alpha_1, \dots, \alpha_k$  not all zero such that  $\sum_i \alpha_i = 0$  and  $\sum_i \alpha_i p_i = \mathbf{0}$ . Use this to show that at least one point in  $S$  is unnecessary to represent  $x$ . ■

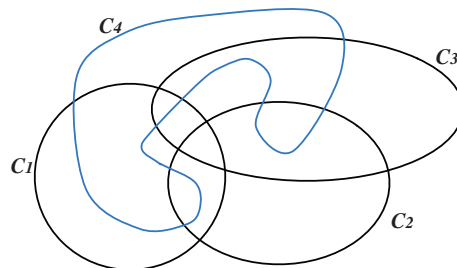
Here are two more basic theorems on convexity.

**Theorem 3.6 (Radon's Theorem)** *Let  $S$  be a subset of  $\mathbb{R}^n$  with  $|S| \geq n + 2$ . Then  $S$  can be partitioned into two sets  $S_1$  and  $S_2$  so that  $\text{conv}(S_1) \cap \text{conv}(S_2) = \emptyset$ .*



**Proof.** Since  $|S| \geq n + 2$ , the points in  $S$  are affinely dependent. Use this fact to find a natural partition. ■

**Theorem 3.7 (Helly's Theorem)** *Let  $C_1, C_2, \dots, C_h$  be convex sets in  $\mathbb{R}^n$  such that  $C_1 \cap C_2 \cap \dots \cap C_h = \emptyset$ . Then, there are at most  $n + 1$  of them whose intersection is empty.*



**Proof.** (Clearly, the convexity assumption on  $C_j$ 's is important as the theorem fails with only one nonconvex  $C_j$  above.) Use induction on  $h$ . If  $h \leq n + 1$ , the theorem is trivial. Assume that the theorem is true for  $h < k (\geq n + 1)$  and prove (\*) the theorem holds for  $h = k$ . Note that  $h \geq n + 2$ . Suppose the statement (\*) does not hold, namely,  $S_j := \bigcap_{j \neq i} C_j \neq \emptyset$  for all  $j = 1, \dots, h$ . Apply Radon's theorem to get a contradiction. ■

### 3.5 Representations of Polyhedra

For a set  $\{v_1, \dots, v_k\}$  of vectors in  $\mathbb{R}^n$ , define their *cone* (or *nonnegative hull*) as

$$\text{cone}(\{v_1, \dots, v_k\}) := \{x : x = \sum_i \lambda_i v_i, \lambda_i \geq 0 \forall i = 1, \dots, k\}. \quad (3.6)$$

For subsets  $P$  and  $Q$  of  $\mathbb{R}^n$ , their *Minkowski sum*  $P + Q$  is defined as

$$P + Q := \{p + q : p \in P \text{ and } q \in Q\}. \quad (3.7)$$

**Theorem 3.8 (Minkowski-Weyl's Theorem for Polyhedra)** *For  $P \subseteq \mathbb{R}^n$ , the following statements are equivalent:*

- (a)  $P$  is a polyhedron, i.e., there exist  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$  for some  $m$  such that  $P = \{x : Ax \leq b\}$ ;
- (b)  $P$  is finitely generated, i.e., there exist (finite) vectors  $v_i$ 's and  $r_j$ 's in  $\mathbb{R}^n$  such that  $P = \text{conv}(\{v_1, \dots, v_s\}) + \text{cone}(\{r_1, \dots, r_t\})$ .

The statement (b) above can be written in matrix form as follows. Here,  $\mathbf{1}$  denotes a vector of all 1s.

- (b)  $P$  is finitely generated, i.e., there exist two matrices  $V \in \mathbb{R}^{n \times s}$  and  $R \in \mathbb{R}^{n \times t}$  for some  $s$  and  $t$  such that  $P = \{x : x = V\mu + R\lambda, \mu \geq \mathbf{0}, \mathbf{1}^T \mu = 1, \lambda \geq \mathbf{0}\}$ .

Theorem 3.8 actually consists of two theorems. The direction from (a) to (b) is Minkowski's Theorem, while the reverse direction from (b) to (a) is Weyl's Theorem.

When a polyhedron  $P$  is bounded (thus a polytope), the minimal representation consists of all extreme points  $v_1, \dots, v_s$  and no rays. Another special case of  $b = \mathbf{0}$  leads to a homogeneous version of the theorem. It is a special case but it is actually as powerful as the nonhomogeneous version above (Exercise 3.5).

**Theorem 3.9 (Minkowski-Weyl's Theorem for Cones)** *For  $P \subseteq \mathbb{R}^n$ , the following statements are equivalent:*

- (a)  $P$  is a polyhedral cone, i.e., there exist  $A \in \mathbb{R}^{m \times n}$  for some  $m$  such that  $P = \{x : Ax \leq \mathbf{0}\}$ ;
- (b)  $P$  is a finitely generated cone, i.e., there exists a matrix  $R \in \mathbb{R}^{n \times t}$  for some  $t$  such that  $P = \{x : x = R\lambda, \lambda \geq \mathbf{0}\}$ .

We first show one direction which follows almost immediately by the Fourier-Motzkin elimination.

**Proof.** (for Theorem 3.9 (b)  $\implies$  (a)). Assume that  $P$  is a finitely generated cone and there exists a matrix  $R \in \mathbb{R}^{n \times t}$  such that  $P = \{x : x = R\lambda, \lambda \geq \mathbf{0}\}$ . The conditions  $x = R\lambda, \lambda \geq \mathbf{0}$  can be considered a system of linear inequalities in variables  $x$  and  $\lambda$ . Thus one can apply the Fourier-Motzkin elimination to eliminate all variables  $\lambda_1, \dots, \lambda_t$  from this system. The result is an equivalent system of inequalities in  $x$  variables. This is a representation of form (a). ■

Let us say that a pair  $(A, R)$  of matrices is a *double description pair* or simply a *DD-pair* if they represent the same polyhedron, namely,

$$Ax \leq \mathbf{0} \Leftrightarrow x = R\lambda, \text{ for some } \lambda \geq \mathbf{0}. \quad (3.8)$$

With this language, the Minkowski theorem says for any matrix  $A$ , there exists  $R$  such that  $(A, R)$  is a DD-pair. The Weyl theorem states that for any  $R$ , there exists  $A$  such that  $(A, R)$  is a DD-pair.

**Lemma 3.10** *For two matrices  $A \in \mathbb{R}^{m \times n}$  and  $R \in \mathbb{R}^{n \times t}$ , the pair  $(A, R)$  is a DD-pair if and only if  $(R^T, A^T)$  is a DD-pair.*

**Proof.** Because of symmetry, we only need to show one direction. Assume the pair  $(A, R)$  is a DD-pair, namely (3.8) is valid. Now we have to show  $(R^T, A^T)$  is also a DD-pair. Now we have a sequence of equivalences

$$\begin{aligned} R^T y \leq \mathbf{0} & \\ \Leftrightarrow \lambda^T R^T y \leq \mathbf{0}, \forall \lambda \geq \mathbf{0} & \\ \Leftrightarrow (R\lambda)^T y \leq \mathbf{0}, \forall \lambda \geq \mathbf{0} & \\ \Leftrightarrow Ax \leq \mathbf{0} \text{ implies } x^T y \leq 0 & \quad (\text{by the assumption (3.8)}) \\ \Leftrightarrow \nexists x : Ax \leq \mathbf{0} \text{ and } y^T x > 0 & \\ \Leftrightarrow y = A^T \mu, \text{ for some } \mu \geq \mathbf{0} & \quad (\text{by Farkas' Lemma}). \end{aligned}$$

The equivalence of the first and the last statement is exactly what we needed to prove. ■

This lemma has a very useful consequence in computation, namely, no one needs to implement both transformations between (a) and (b) but only one.

**Proof.** (for Theorem 3.9). We have already proved Weyl's theorem. On the other hand, Lemma 3.10 says that showing one direction is sufficient to prove both directions. This completes the proof. ■

As Lemma 3.10 indicates, there is a polyhedron associated with the pair  $(R^T, A^T)$ . Namely, if  $(A, R)$  is a DD-pair, the polyhedral cone

$$P^* := \{y \in \mathbb{R}^n : R^T y \leq \mathbf{0}\} \quad (3.9)$$

$$= \{y \in \mathbb{R}^n : y = A^T \mu, \mu \geq \mathbf{0}\} \quad (3.10)$$

is known as the *dual* or the *dual cone* of  $P = \{x : Ax \leq \mathbf{0}\} = \{x : x = R\lambda, \lambda \geq \mathbf{0}\}$ .

**Exercise 3.5** Derive the nonhomogeneous Theorem 3.8 from the homogeneous Theorem 3.9. Hint: Homogenize a given nonhomogeneous system with an extra dimension, convert it by the homogeneous theorem, and then get a nonhomogeneous representation.

The Fourier-Motzkin elimination is not practical for converting between two representations of polyhedra, due to the explosion of the size of intermediate systems. Methods known as the *double description method* and the *reverse search method* are both much more practical and used in many existing implementations (e.g., LRS LIB, CDD LIB).

### 3.6 The Structure of Polyhedra

For a nonempty polyhedron  $P$  in  $\mathbb{R}^n$ , we define two sets the *linearity space* and the *recession cone*.

$$\text{lin. space}(P) := \{z : x + \lambda z \in P, \forall x \in P \text{ and } \forall \lambda \in \mathbb{R}\} \quad (3.11)$$

$$\text{rec. cone}(P) := \{z : x + \lambda z \in P, \forall x \in P \text{ and } \forall \lambda \geq 0\}. \quad (3.12)$$

The recession cone is also known as the *characteristic cone*. Both sets contain the origin, and in general  $\text{lin. space}(P) \subseteq \text{rec. cone}(P)$ .

A polyhedron  $P$  is called *pointed* if it contains an extreme point. Here are some structural properties of polyhedra.

**Theorem 3.11** *Let be  $P$  be a nonempty polyhedron in  $\mathbb{R}^n$ , the following statements hold:*

- (a) *If  $P$  is written as  $P = Q + C$  for some polytope  $Q$  and some polyhedral cone  $C$ , then  $C = \text{rec. cone}(P)$ .*
- (b) *If  $P$  is represented as  $P = \{x : Ax \leq b\}$ , then  $\text{rec. cone}(P) = \{z : Az \leq \mathbf{0}\}$  and  $\text{lin. space}(P) = \{z : Az = \mathbf{0}\}$ ;*
- (c)  *$P$  is pointed if and only if  $\text{lin. space}(P)$  is trivial, i.e.,  $\text{lin. space}(P) = \{\mathbf{0}\}$ ;*
- (d)  *$P$  is bounded if and only if  $\text{rec. cone}(P)$  is trivial, i.e.,  $\text{rec. cone}(P) = \{\mathbf{0}\}$ .*

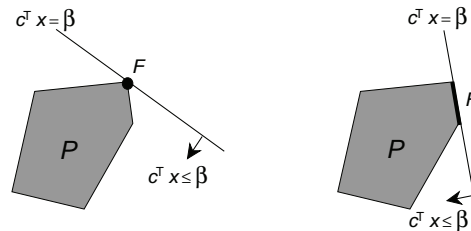
**Proof.** Left to the reader. ■

The statement (a) of the theorem above implies that in the generator representation in Minkowski-Weyl's Theorem, Theorem 3.8 (b), the cone part  $\text{cone}(\{r_1, \dots, r_t\})$  is unique while the convex hull part  $\text{conv}(\{v_1, \dots, v_s\})$  is clearly not.

**Corollary 3.12** *If  $P$  is a cone  $\{x : Ax \leq \mathbf{0}\}$  and pointed, then there exists a vector  $c$  such that  $c^T x > 0$  for all nonzero  $x \in P$ .*

**Proof.** Set  $c^T = -\mathbf{1}^T A$ . Show that this vector satisfies  $c^T x > 0$  for all nonzero  $x \in P$  (Exercise). ■

For  $c \in \mathbb{R}^n$  and  $\beta \in \mathbb{R}$ , an inequality  $c^T x \leq \beta$  is called *valid* for a polyhedron  $P$  if  $c^T x \leq \beta$  holds for all  $x \in P$ . A subset  $F$  of a polyhedron  $P$  is called a *face* of  $P$  if it is represented as  $F = P \cap \{x : c^T x = \beta\}$  for some valid inequality  $c^T x \leq \beta$ .



Note that both  $\emptyset$  and  $P$  are faces, called *trivial faces*. The faces of dimension 0 are called *vertices* (or extreme points) and the faces of dimension  $\dim(P) - 1$  are called *facets*. The first important fact on faces is that there are only finitely many of them. It follows from the following.

**Theorem 3.13** *Let  $P = \{x \in \mathbb{R}^d : Ax \leq b\}$ . Then a nonempty subset  $F$  of  $P$  is a face of  $P$  if and only if  $F$  is represented as the set of solutions to an inequality system obtained from  $Ax \leq b$  by setting some of the inequalities to equalities, i.e.,*

$$F = \{x : A^1x = b^1 \text{ and } A^2x \leq b^2\}, \quad (3.13)$$

where  $A = \begin{bmatrix} A^1 \\ A^2 \end{bmatrix}$  and  $b = \begin{bmatrix} b^1 \\ b^2 \end{bmatrix}$ .

**Proof.** Let  $F$  be a nonempty face. Then,  $F = P \cap \{x : c^T x = \beta\}$  for some valid inequality  $c^T x \leq \beta$ . The set  $F$  is the set of optimal solutions to the LP of  $\max c^T x$  subject to  $Ax \leq b$ . Since the LP has an optimal solution, the dual LP of  $\min b^T y$  subject to  $A^T y = c, y \geq \mathbf{0}$  has an optimal solution, say  $y^*$ , by the strong duality. Then, put  $A_i x \leq b_i$  to be in the equality part  $A^1 x = b^1$  if and only if  $y_i^* > 0$ . Then the resulting set  $\{x : A^1 x = b^1 \text{ and } A^2 x \leq b^2\}$  coincides with  $F$ .

The converse follows immediately by setting  $c^T = \mathbf{1}^T A^1$  and  $\beta = \mathbf{1}^T b^1$ , for a nonempty set  $F$  of form (3.13). ■

**Corollary 3.14** *Every minimal nonempty face of  $P = \{x \in \mathbb{R}^d : Ax \leq b\}$  is an affine subspace of form  $\{x : A^1 x = b^1\}$  where  $A^1 x = b^1$  is a subsystem of  $Ax = b$ .*

**Proof.** By Theorem 3.13, every nonempty face  $F$  has a representation of form

$$F = \{x : A^1 x = b^1 \text{ and } A^2 x \leq b^2\}.$$

Assume  $F$  is minimal. Set  $F' = \{x : A^1 x = b^1\}$ . We will show that  $F = F'$ . We claim that the inequality part  $A^2 x \leq b^2$  must be redundant in the representation of  $F$ . Suppose some of the inequalities can be violated by a point in  $F'$ . Then,  $F'$  is not a minimal nonempty face (why?), a contradiction. ■

**Corollary 3.15** *Let  $P = \{x : Ax \leq b\}$  be a rational polyhedron. Then, every nonempty face of  $P$  contains a point of size polynomially bounded by the sizes of  $A$  and  $b$ .*

**Proof.** It is enough to show the claim for every nonempty minimal face of  $P$ . Then, the result follows from Corollary 3.14 and Corollary 2.3. ■

This corollary also implies that every extreme point of a polyhedron has size polynomially bounded.

**Theorem 3.16** *Let  $P = \{x : Ax \leq b\}$  be a rational polyhedron. Then, it has a generator representation  $P = \text{conv}(\{v_1, \dots, v_s\}) + \text{cone}(\{r_1, \dots, r_t\})$  such that each generator  $v_i$  or  $r_j$  is of size polynomially bounded by the size of the matrix  $[A, b]$ .*

**Proof.** Let  $P = \{x : Ax \leq b\}$  be a rational polyhedron, and  $A$  and  $b$  be integer.

If  $P$  is bounded, the minimal generator representation is the set of extreme points and it is polynomially bounded by Corollary 3.15.

If  $P$  is a pointed cone, by Corollary 3.12, the intersection of  $P$  with the hyperplane  $\mathbf{1}^T Ax = -1$  is a polytope. Clearly the extreme points of this polytope constitute a minimal representation of  $P$  and have size polynomially bounded by the sizes of  $A$  and  $b$ . (The halfline generated by each extreme point is called an *extremal ray* of this pointed cone.)

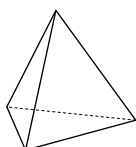
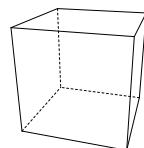
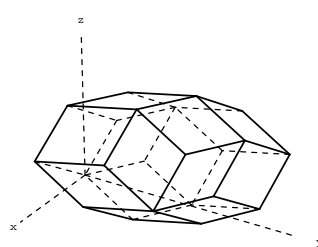
If  $P$  is a pointed polyhedron, it is the Minkowski sum of a polytope and a pointed cone, and the first two cases imply that a minimal representation is polynomially bounded.

If  $P$  is not pointed, the linearity space is given by the system of linear equations  $Ax = 0$  and it has a generator representation by linearly independent vectors  $\{b_1, \dots, b_k\}$  of this space, by Corollary 2.3. Then, setting  $Q = P \cap \{x : b_i^T x = 0, i = 1, \dots, k\}$ ,  $P = Q + \text{lin. space}(P)$ . Now  $Q$  is pointed and it has a generator representation of polynomially bounded size. ■

**Remark 3.17** *From the proof of Theorem 3.16, a minimal representation of a polyhedron  $P$  consists of a set of points each of which is from a minimal nonempty face of  $P$ , a set of vectors to generate the pointed cone which is the intersection of  $P$  with the linear subspace orthogonal to the linearity space of  $P$ , and a set of vectors to generate the linearity space.*

### 3.7 Some Basic Polyhedra

An  $n$ -simplex is the convex hull of  $n + 1$  affinely independent points  $v_0, v_1, \dots, v_n$  in  $\mathbb{R}^n$ . The *standard  $n$ -cube* is the convex hull of  $2^n$  0/1 points in  $\mathbb{R}^n$ , and an  $n$ -cube is any full-rank affine transformation of the standard  $n$ -cube. A *zonotope* in  $\mathbb{R}^n$  (generated by  $k$  generators) is the Minkowski sum of  $k$  line segments in  $\mathbb{R}^n$ . The standard cube is a special zonotope generated by the  $n$  line segments  $[0, e_j]$ , where  $e_j$  denotes the  $j$ th unit vector.

Type	Figure	# Vertices	# Facets	# $i$ -Faces
Simplex( $n$ )		$n + 1$	$n + 1$	$\binom{n+1}{i}$
Cube( $n$ )		$2^n$	$2n$	$\binom{n}{i} 2^{n-i}$
Zonotope( $n, k$ )	 $n = 3$ and $k = 5$	$\leq 2 \sum_{i=0}^{n-1} \binom{k-1}{i}$	$\leq 2 \binom{k}{n-1}$	$O(k^{n-1})$

## 4 Integer Hull and the Complexity of IP

### 4.1 Hilbert Basis

The Hermite normal form  $[B \ \mathbf{0}]$  of a rational matrix  $A \in \mathbb{R}^{m \times n}$  can be considered as a minimal generating set of the lattice  $L(A)$  generated by  $A$ . Namely, the  $m$  columns of  $B$  form a minimal set of vectors in  $\mathbb{R}^m$  whose integer combinations generate  $L(A)$ .

In this section, we are going to deal with the lattice points in a polyhedral cone. Is there any similar basis for the integer points in a polyhedral cone  $C$  generated by rational vectors  $\{a_1, a_2, \dots, a_t\}$ ? A *Hilbert basis* of a cone  $C$  is a finite set of rational vectors  $b_1, b_2, \dots, b_k$  such that every lattice point in the cone is a nonnegative integer combination of  $b_1, b_2, \dots, b_k$ . Here we are mainly concerned with integral Hilbert basis.

Note that a *Hilbert basis* is sometimes called a *Hilbert finite generating set* and then the term *Hilbert basis* is used only for the ones that are (set-inclusion) minimal.

**Theorem 4.1** *Every rational cone admits an integral Hilbert basis. Furthermore, if it is pointed, a (set-inclusion) minimal integral Hilbert basis is unique.*

**Proof.** Without loss of generality, we assume a rational cone  $C$  is generated by integral vectors  $a_1, a_2, \dots, a_t$  in  $\mathbb{R}^n$ , i.e.,  $C = \text{cone}(\{a_1, a_2, \dots, a_t\})$ . We claim that the finite set  $B = \{b_1, \dots, b_k\}$  of integral vectors contained in the zonotope

$$Z := \{x \in \mathbb{R}^n : x = \sum_{i=1}^t \lambda_i a_i, 0 \leq \lambda_i \leq 1, i = 1, \dots, t\} \tag{4.1}$$

is a Hilbert basis, see Figure 4.1 for an example with  $n = 2$ ,  $t = 2$  and  $k = 8$ .

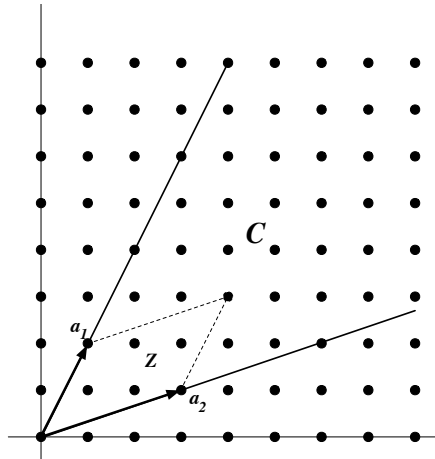


Figure 4.1: The Cone  $C$  and the Polytope  $Z$ .

Let  $p$  be any integral point in  $C$ . Then, we have

$$p = \sum_{i=1}^t \lambda_i a_i, \lambda_i \geq 0, i = 1, \dots, t, \tag{4.2}$$

for some  $\lambda_i$  (not necessarily integer). Furthermore, we have

$$p - \sum_{i=1}^t \lfloor \lambda_i \rfloor a_i = \sum_{i=1}^t (\lambda_i - \lfloor \lambda_i \rfloor) a_i. \quad (4.3)$$

First, the LHS is an integer vector. Secondly, the RHS vector, the same vector as the LHS, is in  $Z$ , because  $0 \leq \lambda_i - \lfloor \lambda_i \rfloor < 1$ . Therefore, it is an integer vector in  $Z$  which is among  $b_1, \dots, b_k$ . Since  $a_1, \dots, a_t$  are contained in  $\{b_1, \dots, b_k\}$ ,  $p$  is a nonnegative integer combination of  $b_1, \dots, b_k$ .

For the second part, assume that the cone  $C$  is pointed. We claim that

$$\widehat{B} := \{x \in B \setminus \{\mathbf{0}\} : x \text{ is not the sum of two other vectors in } B\} \quad (4.4)$$

is a unique minimal Hilbert basis. It is easy to see that every vector in  $\widehat{B}$  must be in any integral Hilbert basis. Now, we need to show that every vector  $b$  in  $B$  not in  $\widehat{B}$  can be represented as nonnegative integer combination of vectors in  $\widehat{B}$ . Suppose there is a vector  $b$  in  $B$  violating this property, and take such a vector  $b$  minimizing  $c^T b$ , where  $c$  is a vector such that  $c^T x > 0$  for all nonzero  $x \in C$ . The existence of  $c$  is guaranteed because  $C$  is pointed, due to Corollary 3.12. Because  $b$  is not in  $\widehat{B}$ ,  $b = b_i + b_j$  for some nonzero vectors  $b_i, b_j$  in  $B$ . Now, we have  $c^T b = c^T b_i + c^T b_j$ , and all terms are positive. This means  $c^T b_i < c^T b$  and  $c^T b_j < c^T b$ . By the assumption that  $c^T b$  is minimized under the condition that  $b$  is not in  $\widehat{B}$ , both  $b_i$  and  $b_j$  must belong to  $\widehat{B}$ , contradicting  $b$  is not a nonnegative integer combination of vectors in  $\widehat{B}$ . ■

**Exercise 4.1** Show that if a rational cone is not pointed, a minimal integral Hilbert basis is not unique.

**Exercise 4.2** In the proof above, assume that  $t = n$  and the rational cone is generated by  $n$  linearly independent vectors  $C = \text{cone}(\{a_1, a_2, \dots, a_n\})$ . Derive a tight lower bound of  $k$  in terms of  $n$  and the absolute value of the determinant  $\det([a_1, a_2, \dots, a_n])$ . Note that  $k$  is the number of lattice points in the zonotope  $Z$  and  $k \geq 2^n$ , because the zonotope  $Z$  is combinatorially a cube.

## 4.2 The Structure of Integer Hull

For a rational polyhedron  $P$  in  $\mathbb{R}^n$ , its integer hull  $P_I$  is defined as the convex hull of all integer points in  $P$ :

$$P_I := \text{conv}\{x : x \in P \cap \mathbb{Z}^n\}. \quad (4.5)$$

It is not clear from the definition that the integer hull is a polyhedron, and in particular finitely generated. We will show that this is the case and the proof uses an argument similar to those used to prove the existence of a Hilbert basis.

There are some obvious facts on the integer hull. First of all the integer hull of every rational cone  $C$  is  $C$  itself:

$$C_I := C. \quad (4.6)$$

**Theorem 4.2** *The integer hull  $P_I$  of a rational polyhedron  $P$  is a polyhedron itself, and if it is nonempty, then  $P_I = B + C$ , where  $B$  is an integer polytope and  $C = \text{rec. cone}(P)$ .*

**Proof.** Assume that  $P$  is a rational polytope with a decomposition  $P = Q + C$  into a polytope  $Q$  and the recession cone  $C$ . Assume that  $P_I$  is nonempty. Let  $a_1, a_2, \dots, a_t$  be integral vectors in  $\mathbb{R}^n$  with  $C = \text{cone}(\{a_1, a_2, \dots, a_t\})$ . Let  $Z$  be the zonotope defined by

$$Z := \{x \in \mathbb{R}^n : x = \sum_{i=1}^t \lambda_i a_i, 0 \leq \lambda_i \leq 1, i = 1, \dots, t\}. \quad (4.7)$$

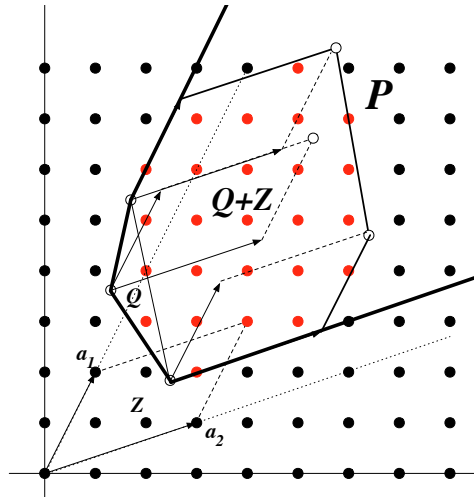


Figure 4.2: The Critical Region  $Q + Z$ .

For the proof of the theorem, it suffice to show that

$$P_I = (Q + Z)_I + C. \quad (4.8)$$

To see  $(Q + Z)_I + C \subseteq P_I$ , observe

$$(Q + Z)_I + C \subseteq P_I + C = P_I + C_I \subseteq (P + C)_I = P_I.$$

For the reverse inclusion, take any integer point  $p \in P_I$  and we show that  $p \in (Q + Z)_I + C$ . This is sufficient because  $(Q + Z)_I + C$  is convex. Now  $p = q + c$ , for some  $q \in Q$  and  $c \in C$ . Now, we have  $c = \sum_i \lambda_i a_i = \sum_i [\lambda_i] a_i + \sum_i (\lambda_i - [\lambda_i]) a_i$ , where the first term is denoted by  $c'$  and the second by  $z$ . Clearly,  $c' \in C \cap \mathbb{Z}^n$  and  $z \in Z$ . It follows that  $p = q + c' + z = (q + z) + c'$  which implies that  $q + z$  is integer and thus  $q + z \in (Q + Z)_I$ . Since  $c' \in C$ , we have  $p \in (Q + Z)_I + C$ . ■

**Theorem 4.3** *The integer hull  $P_I$  of a rational polyhedron  $P = \{x : Ax \leq b\}$  given by an integer matrix  $A$  and an integer vector  $b$  has a generator representation  $P_I = \text{conv}(\{z_1, \dots, z_k\}) + \text{cone}(\{r_1, \dots, r_h\})$  such that the size of each generator  $z_i$  or  $r_j$  is polynomially bounded by the size of the matrix  $[A, b]$ .*

**Proof.** Let  $P = \{x : Ax \leq b\}$  be a rational polyhedron and let  $\delta$  denote the size of  $[A, b]$ .

By Theorem 3.16, a rational polyhedron  $P$  has a generator representation  $Q + C$  with  $Q = \text{conv}(\{v_1, \dots, v_s\})$  and  $C = \text{cone}(\{r_1, \dots, r_h\})$ , where each of  $v_i$ 's and  $r_j$ 's has size polynomially bounded by  $\delta$ . We may also assume that all  $r_j$ 's are integer vectors. By Theorem 4.2,  $\text{rec. cone}(P_I) = \{r_1, \dots, r_h\}$ . We shall show that

$$P_I = \text{conv}(\{z_1, \dots, z_k\}) + C, \quad (4.9)$$

where  $z_1, \dots, z_k$  are the integer points in the set  $Q + Y$  and

$$Y = \{y : y = \sum_{j=1}^h \lambda_j r_j, 0 \leq \lambda_j \leq 1, j = 1, \dots, h, \quad (4.10)$$

$$\text{at most } n \text{ of } \lambda_j \text{'s are positive}\}. \quad (4.11)$$

Each vector  $z_i$  has size polynomially bounded by  $\delta$ , because all  $r_j$ 's are integer, polynomially bounded by  $\delta$  in size, at most  $n$  of them are used to represent  $z_i$ , and every integer point in  $Q$  has size polynomially bounded by  $\delta$ .

We are left to show the equation (4.9). For this, it is sufficient to show that each minimal nonempty face  $F$  of  $P_I$  contains at least one point from  $\{z_1, \dots, z_k\}$ , see Remark 3.17. Let  $z$  be an integer point of  $F$ . Because  $z \in P$

$$z = q + \sum_{j=1}^h \mu_j r_j, \quad (4.12)$$

for some  $\mu_j \geq 0$ . By Carathéodory's Theorem (Theorem 3.5), one may assume that at most  $n$  of  $\mu_j$ 's are positive. Let  $z'$  be the vector

$$z' := q + \sum_{j=1}^h (\mu_j - \lfloor \mu_j \rfloor) r_j. \quad (4.13)$$

It follows that  $z'$  is an integer vector and thus it is one of the vectors from  $\{z_1, \dots, z_k\}$ . It is easy to see that  $z' \in F$ . This completes the proof. ■

**Corollary 4.4** *The integer linear programming (IP) is in NP.*

**Proof.** Consider the IP decision problem: does an IP  $\max\{c^T x : x \in \mathbb{Z}^n, Ax \leq b\}$  have a feasible solution with the objective value greater than a given constant  $K$ ? We need to show that the affirmative answer to the decision problem has a succinct certificate. There are two cases, (1) the IP has an optimal solution and (2) the IP is unbounded.

If the IP has an optimal solution, by Theorem 4.3, there is an integer point  $z$  of size polynomially bounded by the size  $[A, b]$  which attains the objective value greater than  $K$ . This is a succinct certificate.

If the IP is unbounded, again by Theorem 4.3, there are an integer feasible point  $z$  and a direction  $r$  of sizes both polynomially bounded by  $[A, b]$  such that  $c^T r > 0$ . These constitute a succinct certificate.

The case when  $P = \{x \in \mathbb{Z}^n : Ax = b, x \geq \mathbf{0}\}$  is polynomially reducible to the case above. ■

**Corollary 4.5** *The integer linear programming (IP) is in NPC.*

**Proof.** Since, for example, the knapsack problem, a well-known NPC problem, is polynomially reducible to the IP, the IP is at least as hard as the problems in NPC. By Corollary 4.4 which states the IP is in NP, it is in NPC. ■

### 4.3 Further Results on Lattice Points in Polyhedra

Here, we mention some important results on lattice points in polyhedra without proofs. The original proofs are not particularly difficult but beyond the scope of this lecture notes.

In Section 4.1, we proved that there is a Hilbert basis for any rational cone. This means that every integer vector in a rational cone is a nonnegative integer combination of the vectors in a Hilbert basis. Now, an integer analogue of Carathéodory's theorem states that only a small number of integer vectors in the cone is necessary to represent a given integer vector.

**Theorem 4.6 (Cook, Fonlupt and Schrijver (1983))** *Let  $C$  be a pointed rational cone and let  $\widehat{B}$  be the minimal integral Hilbert basis. Then, every integer point  $p$  in  $\widehat{B}$  is an integer nonnegative combination of at most  $2n - 1$  of the vectors in  $\widehat{B}$ .*

Later this bound  $2n - 1$  was improved to  $2n - 2$  by Sebő (1990).

Another interesting result is on the distance between the integer programming solutions and the solutions to the linear programming relaxation. It shows that if both problems have an optimal solution, then for any given LP solution  $x^*$ , there is an optimal solution  $z^*$  to the IP such that the size of  $|x_i^* - z_i^*|$  is polynomially bounded by the size of input, for each  $i$ .

**Theorem 4.7 (Cook, Gerards, Schrijver and Tardos (1986))** *For a given matrix  $A \in \mathbb{Z}^{m \times n}$ , vectors  $b \in \mathbb{Z}^m$  and  $c \in \mathbb{Z}^n$ , let (IP) be the integer programming problem  $\max c^T x$  subject to  $Ax \leq b$  and  $x \in \mathbb{Z}^n$ , and let (LP) be its LP relaxation (without the  $x \in \mathbb{Z}^n$  constraints). Let  $D$  denote the largest absolute value of subdeterminants of  $A$ . Assume that both problems admit an optimal solution. Then the following statements hold.*

- (a) *For any optimal solution  $x^*$  of (LP), there exists an optimal solution  $z^*$  of (IP) such that  $|x_i^* - z_i^*| \leq nD$  for all  $i$ .*
- (b) *For any optimal solution  $z^*$  of (IP), there exists an optimal solution  $x^*$  of (LP) such that  $|x_i^* - z_i^*| \leq nD$  for all  $i$ .*

By Theorem 2.1, the size of  $D$  is at most twice the size of the matrix  $A$ . The theorem above thus shows that there exists an optimal solution to (IP) in a hypercube of a width of polynomial size centered at a given LP optimal solution, if both admit an optimal solution.

## 5 Well Solved Problems

In this section, we present a few classes of integer programming that can be solved in polynomial time. These problems are mostly arising from combinatorial optimization in graphs and digraphs. We focus on how the convex hull of all integral solutions can be described by a set of inequalities. It appears that when the feasible solutions of a combinatorial optimization problem admits a “good” description (to be defined below) by a set of inequalities, the description often leads to an algorithm which runs in polynomial time. This applies for example to the assignment problem and the optimal (perfect) matching problem. The description of the latter involves a large number of inequalities, exponential in the input size.

Note that there are some combinatorial optimization problems that are polynomially solvable but not known to have a good description by inequalities. For example, the optimal stable set problem in claw-free graphs (first studied by Minty with many follow-up results) is polynomially solvable but still resists an explicit description.

### 5.1 Good Description and $\text{NP} \cap \text{coNP}$

Polyhedral combinatorics is an ingenious way of looking at combinatorial optimization from the viewpoint of certificates of optimality with the basic facts of linear programming. Its foundation is due to Jack Edmonds who carried the strong belief that if the convex hull of the feasible region of an LP problem admits a “good” description, the IP over the feasible region should be polynomially solvable. Behind this is a more general belief that if a problem belongs to both NP and coNP, it must be polynomially solvable. While this belief is not easy to justify formally, it has guided many researchers to discover new algorithms in combinatorial optimization and beyond.

Consider a class  $\mathcal{I}$  of rational IPs of form  $\max\{c^T x : x \in \mathbb{Z}^n, Ax \leq b\}$  such that for each instance  $I(A, b, c) \in \mathcal{I}$ , the integer hull  $P_I$  of the feasible region  $P = \{x : Ax \leq b\}$  of the LP relaxation admits a *good description*, that is,  $P_I$  can be represented in form  $P_I = \{x : \hat{A}x \leq \hat{b}\}$  such that the validity of each inequality  $\hat{A}_i x \leq \hat{b}_i$  is verifiable in polynomial time (that implies for example that the size of the vector  $[\hat{A}_i, \hat{b}_i]$  is polynomially bounded by the size of the input matrix  $[A, b]$ ). It is important to note that a good description does not require the number of inequalities in  $\hat{A}x \leq \hat{b}$  to be polynomial.

The following theorem can be considered as the key logic behind this story.

**Theorem 5.1** *Let  $\mathcal{I}$  be a class of rational IPs  $\max\{c^T x : x \in \mathbb{Z}^n, Ax \leq b\}$  such that every instance admits a good description. Then the decision problem of the class  $\mathcal{I}$  of integer programming is in  $\text{NP} \cap \text{coNP}$ .*

**Proof.** Let  $\mathcal{I}$  be a class as described. Since the integer programming was shown to be in NP (Corollary 4.4), we only need to show that it is in coNP. For an instance  $I(A, b, c)$  in this class and a constant  $K$ , suppose the decision problem of checking whether the IP  $I(A, b, c)$  has a feasible solution  $z$  such that  $c^T z > K$  is “no.” Without loss of generality, we may assume  $c$  is integer. Then, the answer being “no” is equivalent to the statement that  $Q := \{x : \hat{A}x \leq \hat{b}, c^T x \geq K + 1\}$  is empty.

By Helly’s Theorem, Theorem 3.7, there are at most  $n + 1$  of the inequalities in  $\hat{A}x \leq \hat{b}, c^T x \geq K + 1$  that constitute an inconsistent system. By Gale’s Theorem (Theorem 3.1)

and Corollary 3.15, there is a certificate for the inconsistency that is polynomially bounded.

This completes the proof.  $\blacksquare$

A convex polyhedron  $P = \{x : Ax \leq b\}$  is called *integral* if every nonempty face  $P$  contains an integer point. By definition, the integer hull  $P_I$  of a convex polyhedron  $P$  is an integer polyhedron. In the following sections we will see various examples of integral polyhedra. Some of them are easily seen to be integral and others require rather involved analysis.

Good description has been proven to be extremely useful in discovering efficient algorithms in combinatorial optimization. Yet, there is an even stronger notion that can lead us to further results. A good description of the integer hull guarantees the existence of a dual solution that is a succinct certificate for optimality. When a dual solution can be chosen to be integral, then we can often interpret a pair of primal and dual integer solutions as a certificate for certain min-max relation satisfied by seemingly unrelated families of combinatorial objects. This leads to the notion of total dual integrality (TDI), to be studied in Section 5.9. The max-flow-min-cut theorem in network flow theory is a typical min-max relationship.

## 5.2 Optimal Matching Problem

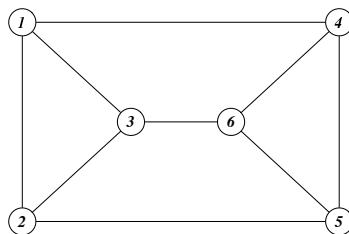
We observed in Example 1.5 that the feasible region of the LP relaxation of the optimal perfect matching problem is not integral.

$$\begin{aligned}
 \text{(LP-M)} \quad & \max w(x) := \sum_{(i,j) \in E} w_{ij} x_{ij} \\
 & \text{subject to} \quad \sum_{(i,j) \in E} x_{ij} = 1 \quad \forall i \in V \\
 & \quad \quad \quad x_{ij} \geq 0, \quad \forall (i,j) \in E.
 \end{aligned}$$

In particular, the feasible region is half-integral.

### Example 5.1 Small Example

Consider the following graph. Clearly there are perfect matchings in this graph.



On the other hand, there are fractional solutions such as the one with weight  $1/2$  for the edges on two triangles formed by nodes 1, 2, 3 and 4, 5, 6, respectively. This half-integral solution is easily seen to be an extreme point of the (LP-M) feasible region.

The small example shows that the LP relaxation (LP-M) is not very useful (yet) to solve the optimal matching problem. Edmonds showed that we only need to add one extra class of constraints. The idea can be seen in the example. Since there are three nodes, the sum of all variables for the edges cannot be more than 1, if  $x$  represents a matching.

In general, if we pick up any subset  $S$  of nodes with odd cardinality  $2k + 1$  for some  $k$ , any vector  $x$  representing a matching must satisfy the inequality:

$$\sum_{i,j \in S, (i,j) \in E} x_{ij} \leq k. \quad (5.1)$$

This observation leads to an appropriate LP relaxation of the optimal perfect matching problem.

$$\begin{aligned} \text{(LP-M2)} \quad \max w(x) &:= \sum_{(i,j) \in E} w_{ij} x_{ij} \\ \text{subject to} \quad \sum_{(i,j) \in E} x_{ij} &= 1 \quad \forall i \in V \\ \sum_{i,j \in S, (i,j) \in E} x_{ij} &\leq \frac{|S| - 1}{2} \quad \forall \text{ odd set } S \subseteq V \\ x_{ij} &\geq 0, \quad \forall (i,j) \in E. \end{aligned}$$

**Theorem 5.2** (IP-M) has an optimal solution if and only if (LP-M2) has an optimal solution. Furthermore, if (IP-M) has an optimal solution, the optimal values of (IP-M) and (LP-M2) are equal. In particular, there is an integral optimal solution to the linear programming problem (LP-M2), if it has an optimal solution.

**Theorem 5.3** A matching  $x^* = (x_{ij}^*)$  is optimal if and only if there exists a feasible solution to the dual of (LP-M2) whose objective value is equal to  $w(x^*)$ .

Edmonds found an ingenious algorithm, known as the Blossom Algorithm, for the optimal matching problem which finds an optimal matching with a dual feasible solution in polynomial time. Unlike the assignment problem, one should not apply the simplex method to the LP relaxation (LP-M2), because there are just too many (exponential in  $n$ ) odd set inequalities and therefore too many dual variables. Edmonds' algorithm considers only a small set of such inequalities at a time, and update the set whenever necessary. Fortunately it is possible to show that there is a dual optimal solution with a polynomial number of nonzero components.

### 5.3 Max-Flow Problem

The word “network” is often used for a graph whose edges or nodes have certain capacities or costs. These extra information typically indicates some physical or economic characteristics of the represented components in a practical application. For example, in a network of oil pipelines, the capacity of a pipe connection from one location  $a$  and another  $b$  indicates the maximum volume of oil that can be sent per hour from  $a$  to  $b$ .

Consider a directed graph  $G = (V, E)$  with the node set  $V = \{1, 2, \dots, n\}$  and the edge set  $E$  with edge capacity  $u_e \in \mathbb{R} \cup \{+\infty\}$  for each  $e \in E$ . There are two fixed nodes, the *source node*  $s$  and the *sink node*  $t$ , in  $V$ .

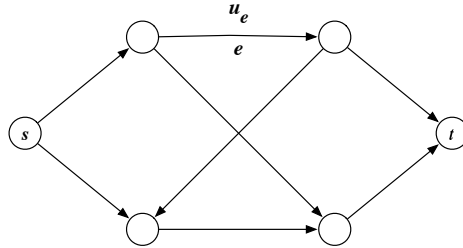
We use the following general notations for directed graphs. For each edge  $e \in E$ , its tail node is denoted by  $t(e)$  and its head by  $h(e)$ . For any subset  $S \subset V$  of nodes, we define two subsets  $\delta^+(S)$  and  $\delta^-(S)$  by

$$\delta^+(S) := \{e \in E : t(e) \in S\}, \tag{5.2}$$

$$\delta^-(S) := \{e \in E : h(e) \in S\}. \tag{5.3}$$

Namely,  $\delta^+(S)$  ( $\delta^-(S)$ , respectively) is the set of *out-going* (*in-coming*) edges. For simplicity, for  $v \in V$ , we use  $\delta^+(v)$  to denote  $\delta^+(\{v\})$  and similarly  $\delta^-(v)$  to denote  $\delta^-(\{v\})$ .

Intuitively, the max-flow problem is to find the largest amount of flow one can send from  $s$  to  $t$  in the graph  $G$  assuming that there is no leak in any of the intermediate nodes of  $V \setminus \{s, t\}$ .



We use the variable  $x_e$  to denote the amount of a flow on the edge  $e$  and let  $x = (x_e : e \in E) \in \mathbb{R}^E$ . For a subset  $F$  of edges, we denote

$$x(F) := \sum_{e \in F} x_e. \tag{5.4}$$

The *max-flow problem* is

$$\max \quad f \tag{5.5}$$

$$\text{subject to} \quad x(\delta^+(v)) - x(\delta^-(v)) = \begin{cases} f, & i = s \\ 0, & \forall i \in V \setminus \{s, t\} \\ -f, & i = t \end{cases} \tag{5.5a}$$

$$0 \leq x_e \leq u_e, \quad \forall e \in E. \tag{5.5b}$$

The equations (5.5a) with RHS being 0 are called the *conservation* constraints, indicating no leaks occur. By this, one of the remaining two equations in (5.5a) is redundant. When  $x$  satisfies all the constraints, it is called a *feasible flow* and the associated value  $f$  is the *value* of the flow  $x$ .

**Exercise 5.1** If all capacities  $u_e$ 's are nonnegative integers, there exists an integral optimal solution. Prove this by an elementary argument similar to that for the assignment problem (Theorem 1.2).

**Exercise 5.2** Write the equation (5.5a) in matrix form as  $Ax = b$ . The matrix  $A$  is known as the incidence matrix of the directed graph  $G$ . Show that  $G$  is totally unimodular.

In fact, all extreme solutions are integral, and the feasible region is an integral polyhedron. Thus, for example, the simplex method finds an integral optimal solution.

There are various polynomial-time algorithms for the max-flow problem that are more (both theoretically and practically) efficient than any general methods for linear programming. In fact, there are *strongly polynomial algorithms* that require in the worst case a number of arithmetic operations polynomially bounded by the dimensions of the input matrix  $A$  (these are the numbers of nodes and edges in the present case). No such algorithm is known in general for linear programming.

A partition  $(S, \bar{S} = V \setminus S)$  of the node set  $V$  is called an  $(s, t)$ -cut if  $s \in S$  and  $t \in \bar{S}$ . The *capacity*  $g(S, \bar{S})$  of an  $(s, t)$ -cut  $(S, \bar{S})$  is defined as the sum of capacities of the edges directed from  $S$  to its complement  $\bar{S}$ .

$$g(S, \bar{S}) = \sum_{e \in \delta^+(S)} u_e. \tag{5.6}$$

It is easy to show that for any feasible flow of value  $f$  and any  $(s, t)$ -cut of value  $g$ ,  $f \leq g$ . The following theorem shows the equality for max-flow and min-cut pairs (a min-max theorem).

**Theorem 5.4** A feasible flow  $x^* = (x_{ij}^*)$  with value  $f^*$  is optimal if and only if there exists an  $(s, t)$ -cut with the capacity equal to  $f^*$ .

**Exercise 5.3** Write the dual LP of the max-flow problem (5.5). Be careful about the RHS variable  $f$  which is not constant. How can you interpret an  $(s, t)$ -cut as a feasible solution to the dual LP? What does the dual objective value represent?

## 5.4 Minimum Cost Flow Problem

Consider a directed graph  $G = (V, E)$  where each edge  $e \in E$  has a capacity  $u_e \in \mathbb{R} \cup \{+\infty\}$  and a *cost*  $c_e$ , and each node  $v$  is assigned a *demand*  $b_v$ . The cost  $c_e$  can be interpreted as the cost of using the edge  $e$  per unit flow. The demand  $b_v$  can be interpreted as follows. If  $b_v > 0$ , the node  $v$  is a *supply node* and supplies a flow of value  $b_v$ . If  $b_v < 0$ ,  $v$  is a *demand node* and it demands a flow of value  $-b_v$ .

The *minimum cost flow problem* is to find a flow of minimum cost satisfying all capacity constraints and demand constraints.

$$\max \quad c^T x = \sum_{e \in E} c_e x_e \tag{5.7}$$

$$\text{subject to} \quad x(\delta^+(v)) - x(\delta^-(v)) = b_v, \quad \forall v \in V \tag{5.7a}$$

$$0 \leq x_e \leq u_e, \quad \forall e \in E. \tag{5.7b}$$

It is easy to see that for a feasible flow to exist, the capacities  $u_e$  must be nonnegative and the demands  $b_v$  must sum up to zero:  $\sum_{v \in V} b_v = 0$ . This problem includes the maximum flow problem as a special case. (How?)

**Theorem 5.5** *If  $u_e$ 's and  $b_v$ 's are integers and the minimum cost flow problem has an optimal solution then it has an integral optimal solution.*

**Exercise 5.4** Prove Theorem 5.5 by an elementary argument similar to that for the assignment problem (Theorem 1.2).

It follows from Theorem 5.5 that all extreme solutions are integral. Moreover, the feasible region is pointed (due to the nonnegativity constraints) and thus it is an integral polyhedron.

**Exercise 5.5** The minimum cost flow problem can be written as  $\max c^T x$  subject to  $Ax = b$  and  $\mathbf{0} \leq x \leq u$ . Here  $A$  is the incidence matrix of  $G$ . Argue that the feasible region  $P = \{x : Ax = b, \mathbf{0} \leq x \leq u\}$  is integral by using the fact that  $A$  is totally unimodular.

The minimum cost flow problem is an LP with a special structure. Thus the certificate of optimality using the dual optimal solution (Theorem 3.3) applies here.

Like the max-flow problem, there are polynomial and strongly polynomial algorithms known for this problem. Designing a strongly polynomial algorithm turned out to be very hard. The strongly polynomial method due to E. Tardos uses some sophisticated repeated rounding techniques. It is not yet known to be practical, but the existence is in great contrast to the general LP case for which no strong polynomial algorithms are known.

## 5.5 A Min-Max Relation for Submodular Functions

In this section, we present a very general min-max relation due to Edmonds and Giles (1977). The min-max relation is a common generalization of many nontrivial min-max relations in graphs and matroids (which is an abstraction of graphs and linear independence).

Let  $G = (V, E)$  be a directed graph. A family  $\mathcal{F}$  of subsets of  $V$  is called a *crossing family on  $V$*  if  $S \cap T \in \mathcal{F}$ ,  $S \cup T \in \mathcal{F}$  for any  $S \in \mathcal{F}$  and  $T \in \mathcal{F}$  such that  $S \cap T \neq \emptyset$  and  $S \cup T \neq V$ . A function  $f : \mathcal{F} \rightarrow \mathbb{R}$  is called *submodular on  $\mathcal{F}$*  if

$$f(S \cap T) + f(S \cup T) \leq f(S) + f(T). \quad (5.8)$$

For a directed graph  $G = (V, E)$ , a crossing family  $\mathcal{F}$  on  $V$ , a submodular function  $f$  on  $\mathcal{F}$ , and two vectors  $l, u \in (\mathbb{R} \cup \{\pm\infty\})^E$ , consider the LP

$$\max \quad c^T x = \sum_{e \in E} c_e x_e \quad (5.9)$$

$$\text{subject to} \quad x(\delta^+(S)) - x(\delta^-(S)) \leq f(S), \quad \forall S \in \mathcal{F} \quad (5.9a)$$

$$l_e \leq x_e \leq u_e, \quad \forall e \in E. \quad (5.9b)$$

The dual LP is

$$\min \quad \sum_{S \in \mathcal{F}} f(S) y_S + u^T z - l^T w \quad (5.10)$$

$$\text{subject to} \quad F(y, e) + z_e - w_e = c_e, \quad \forall e \in E \quad (5.10a)$$

$$y, z, w \geq \mathbf{0}, \quad (5.10b)$$

where  $F(y, e) = \sum \{y_S : e \in \delta^+(S)\} - \sum \{y_S : e \in \delta^-(S)\}$ . By considering  $f$  as vector  $(f_S : S \in \mathcal{F})$ , the first term in the objective function can be written as  $f^T y$ .

**Theorem 5.6** *If  $c$  is integer and the dual LP (5.10) has an optimal solution, then it has an integer optimal solution.*

**Theorem 5.7** *If  $l$ ,  $u$  and  $f$  are integer and the LP (5.9) has an optimal solution, then it has an integer optimal solution.*

Theorem 5.7 implies that the feasible region of the primal LP is integral if  $l$ ,  $u$  and  $f$  are integer. Theorem 5.7 is actually a corollary of Theorem 5.6, by the fundamental result of total dual integrality in Section 5.6.

## Network Flows

The minimum cost flow is a special case of (5.9). To see that, set

$$\mathcal{F} = \{\{v\} : v \in V\} \cup \{V \setminus \{v\} : v \in V\}, \quad (5.11)$$

$f(\{v\}) = b_v$ ,  $f(V \setminus \{v\}) = -b_v$ , and  $l_e = 0, \forall e \in E$ . It is easy to show that  $f$  is submodular on  $\mathcal{F}$  and the LP (5.10) coincides with the minimum cost flow problem (5.7).

## Matroids and Greedy Algorithm

For a undirected connected graph  $G' = (V', E)$  with edge weights  $c_e, e \in E$ , the *minimum spanning tree problem* is to find a spanning tree of minimum total weight (known as an *MST*). It is well-known that the greedy algorithm finds an MST, where the algorithm starts with the empty tree  $T = \emptyset$ , and selects an edge with the smallest weight among the edges which can be added to a subgraph  $T$  of the already selected edges.

A simple reason why the greedy algorithm works is that the set  $\mathcal{B}$  of all spanning trees (represented as the sets of its edges) of  $G'$  satisfies the *matroid basis axiom*:

**(B)** If  $B$  and  $B'$  are in  $\mathcal{B}$  and  $e \in B \setminus B'$ , then there is  $f \in B' \setminus B$  such that  $B \setminus \{e\} \cup \{f\}$  is in  $\mathcal{B}$ .

If a family  $\mathcal{B}$  of subsets of  $E$  satisfies (B), then  $\mathcal{B}$  is called the set of *bases* of a matroid.

Let  $M = (E, \mathcal{B})$  be a matroid on  $E$  with the set  $\mathcal{B}$  of its bases. A subset  $S$  of  $E$  is called *independent* if  $S \subseteq B$  for some  $B \in \mathcal{B}$ . The *rank*  $r(S)$  of a subset  $S$  of  $E$  is the largest cardinality of independent subsets of  $S$ . It can be shown that the rank function  $r$  is submodular on the family  $2^E$  of all subsets of  $E$ . The most important fact behind this is that every independent subset  $F$  of a subset  $S$  of  $E$  is extendable to an independent subset of  $S$  of largest cardinality.

The *matroid polytope* of  $M = (E, \mathcal{B})$  is

$$P_M := \{x : \mathbf{0} \leq x \leq \mathbf{1}, x(S) \leq r(S) \forall S \subseteq E\}, \quad (5.12)$$

where  $x(S) := \sum_{e \in S} x_e$ . It is known that the extreme points of  $P_M$  are the incidence vectors of the independent sets of  $M$ . Here, the *incidence vector*  $\chi(S)$  of  $S \subseteq E$  is the vector in  $\{0, 1\}^E$  such that  $\chi(S)_e = 1$  if and only if  $e \in S$ . In particular,  $P_M$  is an integral polytope. By this integrality, the LP

$$\max c^T x : x \in P_M \quad (5.13)$$

attains maximum at an integer solution. Moreover, an integer optimal solution can be found by the greedy algorithm.

To see the integrality of the matroid polytope as a special case of Theorem 5.7, let  $G = (V, E)$  be a graph with the heads and the tails of its edges all distinct. Note that both graphs  $G$  and  $G'$  have the same set of edges but their vertices are disjoint. We make this fact clear by distinguishing the head/tail functions  $h(e), t(e)$  for  $G$  from the head/tail functions  $h'(e), t'(e)$  for  $G'$ , for  $e \in E$ .

$$\mathcal{F} := \{\{t(e) : e \in S\} : S \subseteq E\}. \quad (5.14)$$

Set the submodular function  $f$  on  $\mathcal{F}$  as

$$f(\{t(e) : e \in S\}) := r(S), \forall S \subseteq E. \quad (5.15)$$

It is easy to see that  $\mathcal{F}$  is a crossing family and the submodularity of  $r$  implies that  $f$  is submodular on  $\mathcal{F}$ . Finally, the LP (5.9) coincides with the LP above (5.13) by setting  $l = \mathbf{0}$  and  $u = \mathbf{1}$ . Thus, the integrality of the matroid polytope is a consequence of Theorem 5.7.

### Matroid Intersection, Directed Cut Coverings

Theorem 5.7 can be applied to many interesting cases beyond what we discussed earlier. We discuss two cases without describing details.

Given two matroids  $M$  and  $M'$  on the same ground set  $E$  with weights  $c_e, e \in E$ , finding a common independent set of largest total weight is polynomially solvable. The key observation is that the intersection of the matroid polytopes  $P_M$  and  $P_{M'}$  is again an integral polytope. This integrality is a special case of Theorem 5.7 by setting the family  $\mathcal{F}$  and the submodular function properly.

Let  $G = (V, E)$  be an acyclic digraph and let

$$D(G) := \{S \subseteq V : S \neq \emptyset, S \neq V, \delta^+(S) = \emptyset\}. \quad (5.16)$$

One can easily verify that  $D(G)$  is a crossing family. For each set  $S$  in  $D(G)$ ,  $\delta^-(S)$  is known as a *directed cut* of  $G$ . Now we set  $f(S) := -1$  for  $S \in D(G)$  (i.e. a constant function). Clearly  $f$  is a submodular function on  $D(G)$ . Consider the LP

$$\max \quad c^T x \quad (5.17)$$

$$\text{subject to} \quad x(\delta^+(S)) - x(\delta^-(S)) \leq f(S), \quad \forall S \in D(G) \quad (5.17a)$$

$$0 \leq x_e \leq 1, \quad \forall e \in E. \quad (5.17b)$$

It is clearly a special case of the LP (5.6). Observe that the constraint (5.17a) is equivalent to

$$x(\delta^-(S)) \geq 1, \quad \forall S \in D(G) \quad (5.18)$$

which means that every integer feasible solution is the incidence vector of a subset of edges that meets every directed cut. Such a subset of edges is known as *directed-cut 1-covering*. Then, by setting  $c = -\mathbf{1}$ , Theorem 5.7 implies the following well-known theorem:

**Theorem 5.8 (Lucchesi and Younger(1978))** *The minimum cardinality of a 1-covering of the directed cuts of  $G$  equals the maximum cardinality of a family of mutually disjoint directed cuts of  $G$ .*

## 5.6 Total Dual Integrality

A rational system of inequalities  $Ax \leq b$  in  $n$  variables is said to be *total dual integral* or *TDI* if for any integer vector  $c$ , the dual LP of the LP  $\max\{c^T x : Ax \leq b\}$  has an integral optimal solution whenever it has an optimal solution. The most important consequence of this notion is the following theorem due to Edmonds and Giles.

**Theorem 5.9** *Let  $Ax \leq b$  be a rational TDI system with  $b$  integer. Then the polyhedron  $P = \{x : Ax \leq b\}$  is integral, i.e., every nonempty face of  $P$  contains an integer point.*

**Proof.** Let  $Ax \leq b$  be a rational TDI system with  $b$  integer. Take any minimal nonempty face  $F$  of  $P = \{x : Ax \leq b\}$ . By Corollary 3.14,  $F = \{x : A^1 x = b^1\}$  for some subsystem  $A^1 x \leq b^1$  of  $Ax \leq b$ . Suppose  $F$  does not contain any integer point. Then, By Corollary 2.6, there exists  $\lambda$  such that  $\lambda^T A^1$  is integer and  $\lambda^T b^1$  is fractional.

For any  $\mu \geq \mathbf{0}$ , the LP  $\max\{c^T x : Ax \leq b\}$  is optimized by all points in  $F$  for  $c^T = \mu^T A^1$ . Select  $\mu \geq \mathbf{0}$  such that  $\mu + \lambda \geq \mathbf{0}$  and  $c_0^T := \mu^T A^1$  is integer. Let  $c_1^T := (\mu + \lambda)^T A^1$ , which is clearly integer. Then, by the assumption that  $Ax \leq b$  is TDI and  $b$  is integer,  $d_i := \max\{c_i^T x : Ax \leq b\}$  is integer for  $i = 0, 1$ . Moreover,  $d_i = c_i^T x$  for all  $x \in F$ . Observe that for any  $x \in F$ ,

$$d_1 - d_0 = (\mu + \lambda)^T A^1 x - \mu^T A^1 x = \lambda^T A^1 x = \lambda^T b.$$

This is a contradiction, because  $d_1 - d_0$  is integer and  $\lambda^T b^1$  is fractional. ■

## 6 Diophantine Approximation and Lattice Reduction

In Section 2, we presented a polynomial-time algorithm for solving a linear diophantine matrix equation. In this section, we discuss how we can find approximate solutions to linear diophantine equations with some prescribed accuracy. Our presentation is mainly based on the excellent book “Geometric Algorithms and Combinatorial Optimization” by Grötschel, Lovász and Schrijver [7, Chapter 5].

Two closely related optimization problems we discuss here are (1) the closest vector problem and (2) the shortest vector problem that were briefly presented in Section 1.

### 6.1 Diophantine Approximation

#### Dirichlet’s Theorems

For a given rational number  $a$ , a positive integer  $N$ , and a positive rational number  $\epsilon$ , the *diophantine approximation problem* is to decide whether there exists a rational number  $a'$  with denominator at most  $N$  such that  $|a - a'| < \epsilon$ .

**Theorem 6.1 (Dirichlet (1842))** *For a given real number  $a$  and  $0 < \epsilon < 1$ , there exist integers  $p$  and  $q$  such that  $1 \leq q \leq 1/\epsilon$  and  $|qa - p| \leq \epsilon$ .*

**Proof.** A short elegant non-polynomial proof omitted. See, for example, [7]. ■

For any rational  $a$ , there is a polynomial-time algorithm to find integers  $p$  and  $q$  required in Theorem 6.1. This uses the notion of continued fractions and the Euclidean algorithm. This will be discussed briefly in Section 6.1.

For given rational numbers  $a_1, \dots, a_n$ , a positive integer  $N$ , and a positive rational number  $\epsilon$ , the *simultaneous diophantine approximation problem* is to decide whether there exist integers  $p_1, \dots, p_n$ , and a positive integer  $q \leq N$  such that  $|qa_i - p_i| < \epsilon$  for all  $i$ .

The following classical theorem gives a sufficient condition for the existence of such an approximation.

**Theorem 6.2 (Dirichlet (1842))** *For given real numbers  $a_1, \dots, a_n$  and  $0 < \epsilon < 1$ , there exist integers  $p_1, \dots, p_n$ , and a positive integer  $q \leq \epsilon^{-n}$  such that  $|qa_i - p_i| < \epsilon$  for all  $i$ .*

**Proof.** A short elegant nonconstructive proof omitted. See, e.g., [7]. ■

Unfortunately, no polynomial algorithm is known for the simultaneous diophantine approximation problem nor to find an existing approximation in Theorem 6.2. We shall present a polynomial-time algorithm (Theorem 6.12) which finds an approximate solution, that satisfies substantially weakened conditions. Namely, we need to add an additional factor  $2^{n(n+1)/4}$  before  $\epsilon^{-n}$  for the upper bound of  $q$ .

### Continued Fractions

For a sequence of integers,  $a_0, a_1, \dots, a_i, \dots$  all positive except for possibly  $a_0$ , the expression

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_{i-1} + \frac{1}{a_i}}}}} \tag{6.1}$$

is called a *continued fraction*, and is denoted as  $\langle a_0, a_1, \dots, a_i \rangle$ .

For a given  $a \in \mathbb{R}$ , the sequence  $a_0, a_1, a_2, a_3, \dots$  defined by

$$a_0 := \lfloor a \rfloor, \quad b_0 := a - \lfloor a \rfloor; \tag{6.2}$$

$$a_i := \left\lfloor \frac{1}{b_{i-1}} \right\rfloor, \quad b_i := \frac{1}{b_{i-1}} - \left\lfloor \frac{1}{b_{i-1}} \right\rfloor, \quad \text{as long as } b_{i-1} \neq 0 \tag{6.3}$$

is called the *continued expansion* of the number  $a$ .

For any  $a \in \mathbb{Q}$ , its continued expansion is a finite sequence  $a_0, a_1, \dots, a_k$ . In fact, this sequence can be computed in polynomial time by using the Euclidean algorithm described in Section 2.2.

**Lemma 6.3** *Let  $a \in \mathbb{Q}$ , let  $a_0, a_1, a_1, a_2, \dots$  be the continued expansion of  $a$ , and let  $p_i/q_i$  denote the canonical (coprime) representation of  $\langle a_0, a_1, \dots, a_i \rangle$ . Let  $i$  be an odd index such that  $a_{i+1}$  exists. Then,*

- (a)  $\frac{p_i}{q_i} < a \leq \frac{p_{i+1}}{q_{i+1}}$ ,
- (b)  $p_{i+1} q_i - p_i q_{i+1} = 1$ .

Now we apply the Euclidean Algorithm to the pair of rational numbers  $a$  and 1, with a slight modification. In matrix form, the Euclidean algorithm generates a finite sequence of unimodular matrices  $T^1, T^2, \dots, T^k$ ,  $\alpha_i$ 's and  $\beta_i$ 's such that

$$\begin{aligned} [a, 1] T^1 &= [\alpha_1, \beta_0 = 1], \\ [a, 1] T^2 &= [\alpha_1, \beta_2], \\ &\vdots \\ [a, 1] T^i &= \begin{cases} [\alpha_i, \beta_{i-1}], & \text{if } i \text{ is odd,} \\ [\alpha_{i-1}, \beta_i], & \text{if } i \text{ is even,} \end{cases} \\ &\vdots \\ [a, 1] T^k &= [0, \beta_{k-1}] \text{ or } [\alpha_{k-1}, 0]. \end{aligned}$$

Here we assume that the algorithm starts by subtracting a multiple of the second (pivot) column with value 1 from the first and thus  $\beta_0 = 1$ , even if  $a < 1$  (in this case, no change takes place). Also, we do not use the swap operation and thus a pivot alternates between the first and the second columns.

**Theorem 6.4** *The unimodular matrices  $T^i$ 's generated by the Euclidean algorithm applied to  $[a, 1]$  satisfy*

$$T^i = \begin{cases} \begin{bmatrix} q_i & -q_{i-1} \\ -p_i & p_{i-1} \end{bmatrix}, & \text{if } i \text{ is odd,} \\ \begin{bmatrix} q_{i-1} & -q_i \\ -p_{i-1} & p_i \end{bmatrix}, & \text{if } i \text{ is even.} \end{cases} \quad (6.4)$$

Therefore, the algorithm generates the continued expansion of the number  $a$  in polynomial time, in particular, with  $q_k a - p_k = 0$  at the termination.

Now we show how Theorem 6.1 can be proved constructively by the polynomial algorithm for any rational  $a$  and  $0 < \epsilon < 1$ . First, compute the continued expansion of  $a$ , find the largest index  $i$  such that  $q_i \leq 1/\epsilon$  and set  $q := q_i$ . If  $i$  is the last index, then  $p_i/q_i$  coincides with  $a$ . Otherwise, by Lemma 6.3, we have

$$\left| a - \frac{p_i}{q_i} \right| < \left| \frac{p_{i+1}}{q_{i+1}} - \frac{p_i}{q_i} \right| = \frac{|p_{i+1} q_i - p_i q_{i+1}|}{q_i q_{i+1}} = \frac{1}{q_i q_{i+1}} < \frac{\epsilon}{q_i}.$$

## 6.2 Lattice Reduction

### Gram-Schmidt Orthogonalization

For an ordered basis  $B = (b_1, b_2, \dots, b_n)$  of  $\mathbb{R}^n$ , the *Gram-Schmidt orthogonalization* (abbreviated by GSO) is the following procedure to compute an orthogonal basis  $B^* = (b_1^*, b_2^*, \dots, b_n^*)$ :

$$\begin{aligned} b_1^* &:= b_1; \\ b_i^* &:= b_i - \sum_{j=1}^{i-1} \frac{b_i^T b_j^*}{\|b_j^*\|^2} b_j^*, \quad i = 1, \dots, n. \end{aligned} \quad (6.5)$$

When  $b_1, b_2, \dots, b_n$  are rational vectors, the running time of the GSO can be bounded by a polynomial function of the input size. The analysis is similar to that of the Gaussian elimination (Theorem 2.4), because the vectors  $b_i^*$ 's are unique solutions to certain linear equations that are defined by the input vectors  $b_1, b_2, \dots, b_n$ .

**Lemma 6.5** *The ordered vectors  $(b_1^*, b_2^*, \dots, b_n^*)$  computed by the GSO satisfy the following properties:*

- (a)  $(b_1^*, b_2^*, \dots, b_n^*)$  forms an orthogonal basis of  $\mathbb{R}^n$  with  $\|b_i^*\| \leq \|b_i\|$  for each  $i$ ;
- (b)  $\|b_1^*\| \|b_2^*\| \cdots \|b_n^*\| = |\det[b_1^*, b_2^*, \dots, b_n^*]| = |\det[b_1, b_2, \dots, b_n]|$ ;

(c)  $b_i = \sum_{j=1}^i \mu_{ij} b_j^*$  for some  $\mu_{ij}$  with  $\mu_{ii} = 1$  for  $i = 1, \dots, n$ .

Now we have an application of the GSO to lattices.

**Lemma 6.6** *Let  $L(B)$  be the lattice generated by an ordered basis  $B = (b_1, \dots, b_n)$  of  $\mathbb{R}^n$  and let  $(b_1^*, b_2^*, \dots, b_n^*)$  be the orthogonal basis computed by the GSO. Then, for any lattice point  $b \in L \setminus \{\mathbf{0}\}$ ,*

$$\|b\| \geq \min\{\|b_1^*\|, \|b_2^*\|, \dots, \|b_n^*\|\}. \quad (6.6)$$

**Proof.** Assume all the assumptions are met. Let  $b$  be any lattice point in  $L \setminus \{\mathbf{0}\}$ . Then,  $b = \sum_{i=1}^n \lambda_i b_i$  for some integer  $\lambda_i$ 's. Let  $k$  be the largest index  $i$  such that  $\lambda_i \neq 0$ . By Lemma 6.5 (c), there are  $\mu_i$ 's such that  $b = \sum_{i=1}^k \mu_i b_i^*$  and  $\mu_k = \lambda_k$  is a nonzero integer. Now,

$$\|b\|^2 = \sum_{i=1}^k \mu_i^2 \|b_i^*\|^2 \geq \mu_k^2 \|b_k^*\|^2 \geq \|b_k^*\|^2.$$

This completes the proof. ■

Recall the *shortest vector problem* which is to find a shortest nonzero vector in the lattice generated by  $B = (b_1, \dots, b_n)$ ,

$$\min\{\|Bx\| : x \in \mathbb{Z}^n \setminus \{\mathbf{0}\}\}. \quad (6.7)$$

Here we identify an ordered basis  $B$  with the  $n \times n$  matrix  $[b_1, \dots, b_n]$ . We will use this convention whenever no confusion may arise.

The following theorem provides a bounded region in which a shortest vector exists.

**Theorem 6.7** *Let  $L(B)$  be the lattice generated by a basis  $B = (b_1, \dots, b_n)$  of  $\mathbb{R}^n$  and let the value  $\alpha(B)$  be defined by*

$$\alpha(B) := \|b_1\| \|b_2\| \cdots \|b_n\|. \quad (6.8)$$

*Then for any solution  $x^0$  to*

$$\min\{\|Bx\| : |x_j| \leq \frac{\alpha(B)}{|\det(B)|} \text{ for all } j, x \in \mathbb{Z}^n \setminus \{\mathbf{0}\}\}, \quad (6.9)$$

*the vector  $Bx^0$  is a shortest vector in the lattice  $L(B)$ .*

**Proof.** Let  $v$  be a shortest vector in the lattice  $L(B)$ . Since  $v = Bx$  for some  $x \in \mathbb{Z}^n$ , by Cramer's rule,  $x_j = \det(B_j) / \det(B)$ , where  $B_j = [b_1, \dots, b_{j-1}, v, b_{j+1}, \dots, b_n]$ . By Hadamard's inequality,  $|\det(B_j)| \leq \|b_1\| \cdots \|b_{j-1}\| \|v\| \|b_{j+1}\| \cdots \|b_n\|$ . Since  $v$  is a shortest vector, we have  $\|v\| \leq \|b_j\|$  and thus  $|\det(B_j)| \leq \alpha(B)$ . It follows that  $|x_j| \leq \alpha(B) / |\det(B)|$ . This completes the proof. ■

### Reduced Bases and the LLL Algorithm

As we learned in Section 2.4,  $|\det(B)|$  has the same value for all bases of a full-dimensional lattice  $L$ , which we denote by  $\det L$ . Yet, there are bases that are “more orthogonal” than others in the sense that the value  $\alpha(B)/|\det(B)|$  is reasonably small. Such a basis will help us, for example, find a smaller region to search for a shortest vector, see Theorem 6.6. Note that the base of  $\mathbb{R}^n$  computed by the GSO from an ordered basis of a lattice is typically not a lattice basis.

An ordered basis  $B = (b_1, \dots, b_n)$  of a full-dimensional lattice  $L$  is called *reduced* if the following two conditions are satisfied:

$$|\mu_{ij}| \leq \frac{1}{2}, \quad \forall 0 \leq j < i \leq n; \tag{6.10}$$

$$\|b_{i+1}^* + \mu_{i+1,i}b_i^*\|^2 \geq \frac{3}{4} \|b_i^*\|^2, \quad \forall i = 1, \dots, n, \tag{6.11}$$

where  $B^* = (b_1^*, b_2^*, \dots, b_n^*)$  is the orthogonal basis computed by the GSO and  $\mu_{ij}$  are as defined in Lemma 6.5.

The first condition (6.10) says that the vectors  $b_1, \dots, b_n$  are nearly orthogonal. The second condition (6.11) is not very intuitive. One can interpret that when one exchanges the order of  $b_i$  and  $b_{i+1}$ , the vector  $(b_{i+1}^* + \mu_{i+1,i}b_i^*)$  represents the new  $b_i^*$  and this condition says the new  $b_i^*$  does not get shortened too much.

**Example 6.1** Let us take a small example. The lattice shown in Figure 6.1 is generated by (the columns of) a random  $2 \times 2$  matrix  $A = \begin{bmatrix} 46 & 48 \\ 48 & 23 \end{bmatrix}$ . A reduced basis computed by the function *LatticeReduce* of *Mathematica 5.2* is  $B = \begin{bmatrix} 2 & 50 \\ -25 & -2 \end{bmatrix}$ . The alpha value  $\alpha(A)$  is roughly 2.83 times larger than that  $\alpha(B)$  of the reduced basis. The critical value  $\frac{\alpha(B)}{|\det(B)|}$  in Lemma 6.7 is about 1.00722. From this, one can easily conclude (without watching the figure) that  $(2, -25)^T$  is a shortest vector.

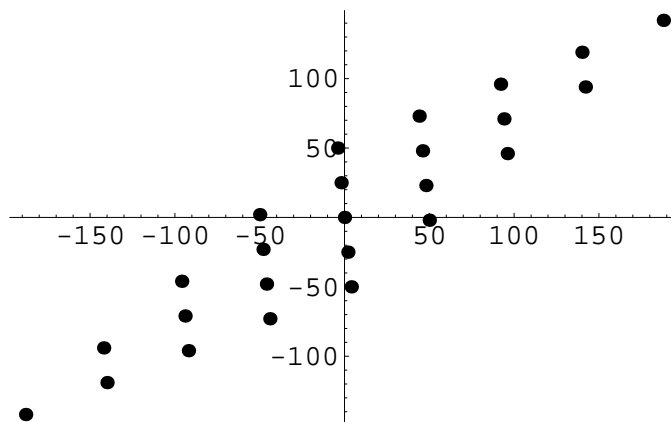


Figure 6.1: A Randomly Generated Lattice.

The following theorem provides some of the most important properties of reduced bases.

**Theorem 6.8** *Let  $B = (b_1, \dots, b_n)$  be a reduced basis of a full-dimensional lattice  $L$  and  $B^* = (b_1^*, b_2^*, \dots, b_n^*)$  the orthogonal basis by the GSO. The following statements hold.*

- (a)  $\|b_{i+1}^*\|^2 \geq 1/2 \|b_i^*\|^2$ ,
- (b)  $\|b_1\| \leq 2^{(n-1)/4} (\det L)^{1/n}$ ,
- (c)  $\|b_1\| \leq 2^{(n-1)/2} \min\{\|b\| : b \in L \setminus \{\mathbf{0}\}\}$ ,
- (d)  $\alpha(B) \leq 2^{n(n-1)/4} \det L$ .

**Proof.**

- (a) Because  $b_j^*$ 's are orthogonal, by (6.11), we have

$$\|b_{i+1}^* + \mu_{i+1,i} b_i^*\|^2 = \|b_{i+1}^*\|^2 + \mu_{i+1,i}^2 \|b_i^*\|^2 \geq 3/4 \|b_i^*\|^2.$$

By (6.10), we have  $\mu_{i+1,i}^2 \leq 1/4$ , and thus  $\|b_{i+1}^*\|^2 \geq 1/2 \|b_i^*\|^2$ .

- (b) By (a), we have  $\|b_i^*\|^2 \geq 2^{-(i-1)} \|b_1^*\|^2 = 2^{-(i-1)} \|b_1\|^2$ . By Lemma 6.5 (b),

$$\det L = \|b_1^*\| \|b_2^*\| \cdots \|b_n^*\| \geq \prod_{i=1}^n 2^{-(i-1)/2} \|b_1\| = 2^{-n(n-1)/4} \|b_1\|^n.$$

Hence, we have  $\|b_1\| \leq 2^{(n-1)/4} (\det L)^{1/n}$ .

- (c) By the proof of (b), we have  $\|b_i^*\| \geq 2^{-(n-1)/2} \|b_1\|$ . By Lemma 6.6, for any nonzero vector  $b$  in  $L$ ,

$$\|b\| \geq \min\{\|b_1^*\|, \|b_2^*\|, \dots, \|b_n^*\|\} \geq 2^{-(n-1)/2} \|b_1\|.$$

- (d) Because  $b_i = \sum_{j=1}^i \mu_{ij} b_j^*$  with  $(\mu_{ii} = 1)$ , by (6.10) and by (a), we have

$$\|b_i\|^2 = \sum_{j=1}^i \mu_{ij}^2 \|b_j^*\|^2 \leq \|b_i^*\|^2 + \frac{1}{4} \sum_{j=1}^{i-1} \|b_j^*\|^2 \quad (6.12)$$

$$\leq \|b_i^*\|^2 + \frac{1}{4} \sum_{j=1}^{i-1} 2^{i-j} \|b_i^*\|^2 \quad (6.13)$$

$$\leq 2^{i-1} \|b_i^*\|^2. \quad (6.14)$$

Multiplying all  $\|b_i^*\|^2$  together for  $i = 1, \dots, n$ , we obtain

$$\alpha(B)^2 = \|b_1\|^2 \cdots \|b_n\|^2 \leq 2^{n(n-1)/2} \|b_1^*\|^2 \cdots \|b_n^*\|^2 = 2^{n(n-1)/2} (\det L)^2. \quad (6.15)$$

This completes the proof.

**Theorem 6.9** *There is a polynomial-time algorithm to find a reduced basis  $B = (b_1, \dots, b_n)$  of the lattice  $L(A)$ , for a given rational nonsingular matrix  $A \in \mathbb{Q}^{n \times n}$ .*

We do not give a proof of Theorem 6.9, because it is quite technical. However, we briefly present the polynomial-time algorithm due to Lenstra, Lenstra and Lovász (1982).

Without loss of generality we assume that the input matrix  $A$  is integral. Set  $B := A$ . The algorithm consists of following two stages (I) and (II).

- (I) Compute the orthogonal basis  $B^*$  of  $B$  using the GSO. For each  $i = 1, \dots, n$  (in this order), replace  $b_i$  by  $b_i - \sum_{j=1}^{i-1} \lceil \mu_{ij} \rceil b_j$ , where  $\mu_{ij}$  is as defined in Lemma 6.5, and  $\lceil x \rceil$  denotes the (smaller) nearest integer to  $x$ . Just to clarify this notation,  $\lceil 2.5 \rceil = 2$  and  $\lceil 2.51 \rceil = 3$ . Note that the GSO basis  $B^*$  stays unchanged because of the fact that the subtraction of any linear combination of  $b_1, \dots, b_{i-1}$  from  $b_i$  has no effect on the GSO result. One can show that at the end of this stage, the new  $\mu_{ij}$ 's satisfy  $|\mu_{ij}| \leq 1/2$  and thus the first condition of reduced basis (6.10).
- (II) Now we only have to worry about the second condition (6.11). For that, find any index  $i$  such that the condition (6.11) is violated. If no such index exists, stop. Otherwise, swap  $b_i$  and  $b_{i+1}$ , and go back to the stage (I).

It can be shown that this algorithm terminates in polynomial-time.

### 6.3 Applications of Lattice Reduction

**Theorem 6.10 (Shortest Vector in Fixed Dimension)** *For fixed  $n$ , there is a polynomial-time algorithm to find a shortest vector in  $L(A)$  for a given rational nonsingular matrix  $A \in \mathbb{Q}^{n \times n}$ .*

**Proof.** This follows directly from Theorem 6.7 and Theorem 6.9. Namely, once a reduced basis  $B$  is computed, one only needs to enumerate all vectors  $x \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$  such that  $|x_j| \leq 2^{n(n-1)/4}$  for all  $j$ , and take a vector  $Bx$  with the smallest  $\|Bx\|$ . ■

**Theorem 6.11 (Approximate Closest Vector)** *There is a polynomial-time algorithm, for given rational nonsingular matrix  $A \in \mathbb{Q}^{n \times n}$  and a given rational vector  $b \in \mathbb{Q}^n$ , to compute a vector  $y \in L(A)$  such that*

$$\|b - y\| \leq 2^{(n/2)} \min\{\|b - v\| : v \in L(A)\}. \quad (6.16)$$

**Proof.** Let  $A \in \mathbb{Q}^{n \times n}$ , and let  $b \in \mathbb{Q}^n$ . First we find a reduced basis  $B = (b_1, \dots, b_n)$  of  $L(A)$  and compute the GSO basis  $B^* = (b_1^*, b_2^*, \dots, b_n^*)$ . Then, we find a lattice vector  $y \in L(A)$  such that

$$b - y = \sum_{i=1}^n \mu_i b_i^* \quad \text{with } |\mu_i| \leq \frac{1}{2} \quad \text{for all } i = 1, \dots, n. \quad (6.17)$$

To find such  $y$ , use the fact that  $B^*$  is a basis,  $b = \sum_{i=1}^n \lambda_i^0 b_i^*$  for some  $\lambda^0$ . Then,

$$b - \lceil \lambda_n^0 \rceil b_n = \sum_{i=1}^n \lambda_i^1 b_i^*, \quad (6.18)$$

for some  $\lambda^1$  with  $|\lambda_n^1| \leq 1/2$ . Continuing with the  $(n-1)$ st component by subtracting  $\lceil \lambda_{n-1}^1 \rceil b_{n-1}$  from both sides, we obtain  $\lambda^2$  with  $|\lambda_i^2| \leq 1/2$  for  $i = n-1, n$ , and do this all the way down to 1st component to obtain  $\lambda^n$ . By setting  $y := \sum_{i=1}^n \lceil \lambda_i^{n-i} \rceil b_i$  and  $\mu := \lambda^n$ , we have a lattice vector  $y$  we are looking for.

Now we claim that such a vector  $y$  satisfies the condition (6.16), which completes the proof. To prove the claim, let  $z$  be any vector in  $L(A) \setminus \{y\}$ . We can write  $b - z = \sum_{i=1}^n \zeta_i b_i^*$  for some  $\zeta_i$ 's. Let  $k$  be the largest index such that  $\mu_i \neq \zeta_i$ . Since both  $y$  and  $z$  are lattice points,  $z - y = \sum_{i=1}^k (\zeta_i - \mu_i) b_i^*$  is also a lattice point. This means that  $(\zeta_k - \mu_k)$  is integer by Lemma 6.5 (c). Because  $|\mu_k| \leq 1/2$ , we have  $|\zeta_k| \geq 1/2$ . Therefore,

$$\|b - z\|^2 \geq \sum_{i=k+1}^n \zeta_i^2 \|b_i^*\|^2 + \frac{1}{4} \|b_k^*\|^2 = \sum_{i=k+1}^n \mu_i^2 \|b_i^*\|^2 + \frac{1}{4} \|b_k^*\|^2. \quad (6.19)$$

On the other hand,

$$\begin{aligned} \|b - y\|^2 &\leq \sum_{i=k+1}^n \mu_i^2 \|b_i^*\|^2 + \frac{1}{4} \sum_{i=1}^k \|b_i^*\|^2 \\ &\leq \sum_{i=k+1}^n \mu_i^2 \|b_i^*\|^2 + \frac{1}{4} \|b_k^*\|^2 \left( \sum_{i=1}^k 2^{k-i} \right) \quad (\text{by Theorem 6.8 (a)}) \\ &< 2^k \|b - z\|^2 \leq 2^n \|b - z\|^2. \end{aligned}$$

■

**Theorem 6.12 (Simultaneous Diophantine Approximation)** *There is a polynomial-time algorithm, for given rational numbers  $a_1, \dots, a_n$  and  $0 < \epsilon < 1$ , to compute integers  $p_1, \dots, p_n$  and  $q$  such that*

$$1 \leq q \leq 2^{n(n+1)/4} \epsilon^{-n} \text{ and } |qa_i - p_i| < \epsilon, \quad \forall i = 1, \dots, n. \quad (6.20)$$

**Proof.** Let  $a_1, \dots, a_n$  be rational numbers and  $0 < \epsilon < 1$ . Let  $e_i$  be the  $i$ th unit vector in  $\mathbb{R}^{n+1}$ , and let  $a = (a_1, \dots, a_n, 2^{-n(n+1)/4} \epsilon^{n+1})^T$ . Let  $L$  be the lattice generated by  $[e_1, e_2, \dots, e_n, a]$ . It follows that  $\det L = 2^{-n(n+1)/4} \epsilon^{n+1}$ . We can compute a reduced basis  $B = (b_1, \dots, b_n)$  of  $L$  in polynomial-time. By Theorem 6.8 (b),

$$\|b_1\| \leq 2^{n/4} (\det L)^{1/(n+1)} = \epsilon. \quad (6.21)$$

The vector  $b_1$  can be represented as  $p_1 e_1 + \dots + p_n e_n - qa$  for some integer  $p_i$ 's and  $q$ . Since  $\epsilon < 1$ ,  $q \neq 0$  and we may assume  $q > 0$ . By (6.21), we have

$$|p_i - qa_i| < \epsilon, \quad (6.22)$$

$$2^{-n(n+1)/4} \epsilon^{n+1} q \leq \epsilon, \text{ equivalently, } q \leq 2^{n(n+1)/4} \epsilon^{-n}. \quad (6.23)$$

This completes the proof. ■

Compare this result with Dirichlet's non-constructive theorem, Theorem 6.2. In particular, there is a huge extra factor  $2^{n(n+1)/4}$  for the upper bound for  $q$ . Yet, this constructive theorem is strong enough for various applications.

## 7 Integer programming in fixed dimension

The exposition of this chapter follows Bertsimas and Weismantel's book [3, §§6.5–6].

### 7.1 Maximum volume inscribed ellipsoid

In this chapter, we describe Lenstra's (1983) algorithm for integer programming, which runs in polynomial time if the dimension (the number of variables) is fixed. Besides basis reduction and other lattice techniques we learned in the previous chapter, an important piece of the algorithm is to be able to squeeze a polytope between two ellipsoids whose sizes do not differ too much (to **approximate** the polytope by an ellipsoid). The existence of such ellipsoids goes back to John (1948).

An **ellipsoid**  $E(d, D)$  is the set  $\{x \in \mathbb{R}^n : (x - d)^T D^{-2}(x - d) \leq 1\}$ , where  $D \in \mathbb{R}^{n \times n}$  is a symmetric positive definite matrix and  $D^{-2}$  is the inverse of  $D^2 = DD$ . The ellipsoid  $E(d, D)$  is the image of the **unit ball**  $B(\mathbf{0}, 1) = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$  under the affine transformation  $x \mapsto Dx + d$ .

**Definition 7.1** The **maximum volume inscribed ellipsoid problem** is for an input matrix  $A \in \mathbb{Z}^{m \times n}$  and a vector  $b \in \mathbb{Z}^m$  such that  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  is full-dimensional and bounded, to find a matrix  $D$  and a vector  $d$  such that the ellipsoid  $E(d, D)$  is contained in  $P$  and has maximum volume among all ellipsoids contained in  $P$ .

**Theorem 7.2** *There is a polynomial-time algorithm for the maximum volume inscribed ellipsoid problem.*

**Proof.** The volume of  $E(d, D)$  is proportional to  $\det D$ . Hence maximizing the volume of an inscribed ellipsoid is equivalent to computing

$$\begin{aligned} M &:= \max\{\log \det D : E(d, D) \subseteq P\} \\ &= \max\left\{\log \det D : \max\{a_i^T x : (x - d)^T D^{-2}(x - d) \leq 1\} \leq b_i \text{ for } i = 1, \dots, m\right\}, \end{aligned} \quad (7.1)$$

where  $a_i$  is the transpose of the  $i$ th row of  $A$ .

**Exercise 7.1** Show that  $\max\{a_i^T x : (x - d)^T D^{-2}(x - d) \leq 1\}$  is attained by

$$x^* = d + \frac{D^2 a_i}{\sqrt{a_i^T D^2 a_i}}.$$

Therefore

$$\begin{aligned} M &= \max\left\{\log \det D : a_i^T d + \sqrt{a_i^T D^2 a_i} \leq b_i, \quad i = 1, \dots, m\right\} \\ &= \max\left\{\log \det D : a_i^T D^2 a_i - b_i^2 + 2b_i(a_i^T d) - (a_i^T d)^2 \leq 0, \quad i = 1, \dots, m\right\}. \end{aligned} \quad (7.2)$$

Problem (7.2) can be solved in polynomial time by interior-point methods. For details see, for instance, [5, §8.4]. ■

**Proposition 7.3** *If  $E = \{x : (x - d)^T D^{-2}(x - d) \leq 1\}$  is the maximum volume inscribed ellipsoid in  $P = \{x : Ax \leq b\}$ , found by the above algorithm, then*

$$P \subseteq E' = \{x : (x - d)^T D^{-2}(x - d) \leq n^2\}.$$

**Proof.** Let  $a_i$  be the transpose of the  $i$ th row of  $A$  ( $i = 1, \dots, m$ ). The Karush-Kuhn-Tucker (KKT) conditions for (7.2) are:

$$D^{-1} = \sum_{i=1}^m \lambda_i (Da_i a_i^T + a_i a_i^T D) \quad (7.3)$$

$$\sum_{i=1}^m \lambda_i (a_i b_i - a_i^T d a_i) = 0 \quad (7.4)$$

$$\|Da_i\| \leq b_i - a_i^T d \quad \text{for } i = 1, \dots, m \quad (7.5)$$

$$\lambda_i (a_i^T D^2 a_i - b_i^2 + 2b_i (a_i^T d) - (a_i^T d)^2) = 0 \quad \text{for } i = 1, \dots, m \quad (7.6)$$

$$\lambda_i \geq 0 \quad \text{for } i = 1, \dots, m \quad (7.7)$$

First let us assume that the optimum is the unit ball  $B(\mathbf{0}, 1)$ , that is, that  $D = I$  and  $d = \mathbf{0}$ . Then  $b > \mathbf{0}$ , so we may moreover assume that  $b = \mathbf{1}$ . The KKT conditions become:

$$I = \sum_{i=1}^m \lambda_i a_i a_i^T \quad (7.8)$$

$$\sum_{i=1}^m \lambda_i a_i = 0 \quad (7.9)$$

$$\lambda_i (\|a_i\| - 1) = 0 \quad \text{for } i = 1, \dots, m \quad (7.10)$$

$$\|a_i\| \leq 1 \quad \text{for } i = 1, \dots, m \quad (7.11)$$

$$\lambda_i \geq 0 \quad \text{for } i = 1, \dots, m \quad (7.12)$$

Comparing the traces of both sides in (7.8), we get that

$$\sum_{i=1}^m \lambda_i = n.$$

Now let  $x \in P$ ; we want to prove that  $x \in B(\mathbf{0}, n)$ , that is,  $\|x\| \leq n$ . Let  $\|x\| = r$ . Then  $-r \leq a_i^T x \leq 1$  for all  $i$  with  $\lambda_i \neq 0$ , and so

$$\begin{aligned} 0 &\leq \sum_{i=1}^m \lambda_i (1 - a_i^T x)(r + a_i^T x) \\ &= r \cdot \sum_{i=1}^m \lambda_i + (1 - r) \sum_{i=1}^m \lambda_i a_i^T x - \sum_{i=1}^m (a_i^T x)^2 \\ &= rn - r^2. \end{aligned}$$

Hence  $\|x\| = r \leq n$ .

If the optimum ellipsoid is not the unit ball, consider the (invertible) affine transformation  $\phi : x \mapsto D^{-1}x - d$  and let  $P' = \phi[P]$ . Then  $B(\mathbf{0}, 1) \subseteq P' \subseteq B(\mathbf{0}, n)$  and therefore

$$E = \phi^{-1}[B(\mathbf{0}, 1)] \subseteq P = \phi^{-1}[P'] \subseteq E' = \phi^{-1}[B(\mathbf{0}, n)].$$

■

## 7.2 Lenstra's algorithm

We want to solve the **feasibility problem** of integer programming, that is, given a rational polyhedron  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  to decide whether  $P \cap \mathbb{Z}^n = \emptyset$ .

### Algorithm 7.1 (Reduction of dimension)

**Input:**  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$  such that  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  is full-dimensional and bounded.

**Output:** A feasible  $y \in P \cap \mathbb{Z}^n$ , or a series of feasibility problems given by polytopes  $(P_k : k \in K)$  of dimension  $n - 1$ .

1. Apply the algorithm of Theorem 7.2 to find  $D$ ,  $d$  such that

$$E = \{x : (x - d)^T D^{-2}(x - d) \leq 1/n^2\} \subseteq P \subseteq E' = \{x : (x - d)^T D^{-2}(x - d) \leq 1\}.$$

2. Set  $P' := \{x \in \mathbb{R}^n : ADx \leq b\}$ ;  $p := D^{-1}d$ .

3. Let  $L := L(D^{-1})$  be the lattice generated by the columns of  $D^{-1}$ . Compute a reduced basis  $B = \{b_1, \dots, b_n\}$  of  $L$ ; rearrange so that  $\|b_n\|$  is maximum.

4. Apply the Gram-Schmidt orthogonalization to  $B$  to get  $B^* = \{b_1^*, \dots, b_n^*\}$ .

5. Apply the approximate closest vector algorithm (Theorem 6.11) to get  $y \in L$ , the approximate closest vector to  $p$ .

6. If  $y \in P'$ , then  $Dy \in P \cap \mathbb{Z}^n$ ; return  $Dy$  and stop.

Otherwise set

$$c^* := \frac{b_n^*}{\|b_n^*\|^2}, \tag{7.13}$$

$$B := (b_1 \quad b_2 \quad \dots \quad b_n), \tag{7.14}$$

$$K := \{ \lfloor c^{*T} p - \|c^*\| \rfloor, \dots, \lceil c^{*T} p + \|c^*\| \rceil \}, \tag{7.15}$$

$$P_k := \{z \in \mathbb{R}^{n-1} : ADB \begin{pmatrix} z \\ k \end{pmatrix} \leq b\} \quad \text{for } k \in K, \tag{7.16}$$

and return the polytopes  $(P_k : k \in K)$ .

**Lemma 7.4** *If  $y \notin P'$  in Step 6., then*

$$\max\{c^{*T}x : x \in P' \cap L\} - \min\{c^{*T}x : x \in P' \cap L\} \leq n^{3/2} \cdot 2^{n(n-1)/4}.$$

**Proof.** The algorithm of Theorem 6.11 finds a vector  $y \in L$  such that

$$p - y = \sum_{i=1}^n \mu_i b_i^*$$

with each  $|\mu_i| \leq \frac{1}{2}$  (see (6.17)).

If  $B(p, r) = \{x \in \mathbb{R}^n : \|x - p\| \leq r\}$ , then

$$B(p, 1/n) \subseteq P' \subseteq B(p, 1) \quad (7.17)$$

because  $E \subseteq P \subseteq E'$ . As  $y \notin P'$ , we have  $\|y - p\| > 1/n$ . Therefore

$$\begin{aligned} \frac{1}{n} < \|y - p\| &\leq \sqrt{\frac{1}{4} \sum_{i=1}^n \|b_i^*\|^2} \\ &\leq \frac{1}{2} \sqrt{\sum_{i=1}^n \|b_i\|^2} && \text{by Lemma 6.5(a)} \\ &\leq \frac{1}{2} \sqrt{n} \|b_n\| && \text{by maximality of } \|b_n\|. \end{aligned} \quad (7.18)$$

By Theorem 6.8(d),

$$\begin{aligned} \alpha(B) &= \|b_1\| \cdots \|b_n\| \leq 2^{n(n-1)/4} \det L \\ &= 2^{n(n-1)/4} \|b_1^*\| \cdots \|b_n^*\| \\ &\leq 2^{n(n-1)/4} \|b_1\| \cdots \|b_{n-1}\| \|b_n^*\|. \end{aligned}$$

Hence

$$\frac{\|b_n\|}{2^{n(n-1)/4}} \leq \|b_n^*\| \leq \|b_n\|.$$

By definition  $\|c^*\| = \frac{1}{\|b_n^*\|}$ , and from (7.17) we obtain:

$$\begin{aligned} &\max\{c^{*T}x : x \in P' \cap L\} - \min\{c^{*T}x : x \in P' \cap L\} \\ &\leq \max\{c^{*T}x : x \in B(p, 1)\} - \min\{c^{*T}x : x \in B(p, 1)\} \\ &\leq c^{*T} \left( p + \frac{c^*}{\|c^*\|} \right) - c^{*T} \left( p - \frac{c^*}{\|c^*\|} \right) \\ &= 2 \|c^*\| = \frac{2}{\|b_n^*\|} \leq \frac{2 \cdot 2^{n(n-1)/4}}{\|b_n\|} \leq n\sqrt{n} 2^{n(n-1)/4}, \end{aligned}$$

the last inequality follows from (7.18) and for finding the max and min we reuse Exercise 7.1.

■

**Lemma 7.5** *The polytopes  $P_k$  defined in (7.16) have the property that  $P \cap \mathbb{Z}^n = \emptyset$  if and only if  $P_k \cap \mathbb{Z}^{n-1} = \emptyset$  for all  $k \in K$ .*

**Proof.** Every  $x^* \in P' \cap L$  can be written as

$$x^* = \sum_{i=1}^{n-1} z_i b_i + k b_n \quad \text{for some } z_i, k \in \mathbb{Z}.$$

That is,  $x^* = B \begin{pmatrix} z \\ k \end{pmatrix}$ , where  $B$  has columns  $b_1, \dots, b_n$ . Now  $c^* = b_n^* / \|b_n^*\|^2$ , hence  $c^*$  is orthogonal to all  $b_i$  for  $1 \leq i \leq n-1$  and  $c^{*T} b_n = 1$ . Thus  $c^{*T} x^* = k$  is integral.

Moreover,

$$c^{*T} x^* \in [\min\{c^{*T} x : x \in P' \cap L\}, \max\{c^{*T} x : x \in P' \cap L\}] \subseteq [c^{*T} p - \|c^*\|, c^{*T} p + \|c^*\|],$$

hence  $c^{*T} x^* \in K$ .

Therefore

$$\begin{aligned} \text{some } x \in P \cap \mathbb{Z} &\iff x^* = D^{-1}x \in P' \cap L \\ &\iff ADx^* \leq b, \quad x^* = B \begin{pmatrix} z \\ k \end{pmatrix} \text{ for some } z \in \mathbb{Z}^{n-1}, k \in K \\ &\iff ADB \begin{pmatrix} z \\ k \end{pmatrix} \leq b \\ &\iff z \in P_k \cap \mathbb{Z}^{n-1} \text{ for some } k \in K. \end{aligned}$$

So  $P \cap \mathbb{Z} \neq \emptyset$  if and only if  $P_k \cap \mathbb{Z}^{n-1} \neq \emptyset$  for some  $k \in K$ . ■

**Theorem 7.6 (Lenstra (1983))** *Let  $n \in \mathbb{N}$  be fixed. Then there exists an algorithm that for given  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ , decides whether  $\{x \in \mathbb{Z}^n : Ax \leq b\} = \emptyset$  in time polynomial in the encoding size of  $A$  and  $b$ .*

**Proof.** Here we present a proof for the case when  $P$  is full-dimensional and bounded. If  $P$  is not full-dimensional, the ellipsoid method for LP can be used to reduce the dimension. If  $P$  is not bounded, one may consider the intersection of  $P$  with a suitable box whose size depends on the encoding size of the input (using ideas of Theorem 4.3).

Assuming that  $P$  is full-dimensional and bounded, we apply Algorithm 7.1. If  $y \in P \cap \mathbb{Z}^n$  is found, then  $P \cap \mathbb{Z}^n \neq \emptyset$ . Otherwise we run Algorithm 7.1 on the  $(n-1)$ -dimensional problems  $\{ADB \begin{pmatrix} z \\ k \end{pmatrix} \leq b\} \cap \mathbb{Z}^n$  for all  $k \in K$ .

All we need to check is that  $ADB$  is integral. Because  $L(D^{-1}) = L(B)$ , by Theorem 2.10 the matrices  $D^{-1}$  and  $B$  have the same Hermite normal form. So there exists a unimodular matrix  $U \in \mathbb{Z}^{n \times n}$  such that  $D^{-1}U = B$ . Therefore  $ADB = ADD^{-1}U = AU \in \mathbb{Z}^{m \times n}$  because  $A$  is integral.

By Lemma 7.4, the size of  $K$  is bounded by a constant (since  $n$  is fixed). The input for the  $(n-1)$ -dimensional subproblems is produced by a polynomial algorithm, hence of polynomial size with respect to the size of our input. Thus our algorithm consists in a constant number of applications of a polynomial algorithm, and therefore it is polynomial as well. ■

**Corollary 7.7** *Let  $n \in \mathbb{N}$  be fixed. Then there exists a polynomial algorithm to solve the IP  $\max\{c^T x : Ax \leq b, x \in \mathbb{Z}^n\}$ , where  $A \in \mathbb{Z}^{m \times n}$ .*

**Proof.** From Theorem 4.3 it follows that the size of the optimum is polynomially bounded in terms of the size of the input. Hence it can be found by binary search, using the feasibility algorithm. To work out the details is an exercise. ■

**Note.** Careful examination of the algorithms reveals that they work in polynomial time also for rational input.

## 8 Integral polyhedra and totally unimodular matrices

The first two sections of this chapter follow Schrijver's book [11, §§16.3,19.1]; the third is based on [4, §8.6].

### 8.1 Integral polyhedra

Recall that  $P_I$  is the integer hull of a polyhedron  $P$ , that is,  $P_I = \text{conv}(P \cap \mathbb{Z}^n)$ . A polyhedron  $P$  is **integral** if  $P = P_I$ . We saw in Chapter 5 that IP is often easy if the feasible region is an integral polyhedron. Now we investigate this topic once again.

Clearly, a rational polyhedron  $P$  is integral if and only if each face of  $P$  contains an integral vector. In order to prove that each face contains an integral vector, it suffices to prove the claim for *minimal* faces; every face contains a minimal face.

Also,  $P$  is an integral polyhedron if and only if  $\max\{c^T x : x \in P\}$  is attained by an integral vector  $x^*$  for each  $c$  for which the maximum is finite.

Solving an IP over an integral polyhedron can be done in polynomial time using Khachiyan's ellipsoid method for linear programming. This method finds the optimum  $\delta = \max\{c^T x : Ax \leq b\}$  as well as a system of linear equations  $A'x = b'$  determining a minimal face  $F$  of  $P = \{x : Ax \leq b\}$  such that if  $x \in F$ , then  $c^T x = \delta$ . By integrality of  $P$ , there exists an integral vector  $x \in F$ ; it can be found in polynomial time (see Section 2.3).

Thus we get the following:

**Theorem 8.1** *There is a polynomial algorithm which, given a rational system  $Ax \leq b$  defining an integral polyhedron, and a rational vector  $c$ , finds an optimal solution to the IP  $\max\{c^T x : Ax \leq b, x \in \mathbb{Z}^n\}$ .*

Next we look at a class of IPs that lead to integral polyhedra.

### 8.2 Total unimodularity

A matrix  $A$  is **totally unimodular** if each subdeterminant of  $A$  is 0, +1, or -1. Obviously, each entry of a unimodular matrix is 0, +1, or -1. Moreover,  $A$  is totally unimodular if and only if  $A^T$  is totally unimodular if and only if  $\begin{bmatrix} I & A \end{bmatrix}$  is totally unimodular.

**Theorem 8.2** *Let  $A$  be a totally unimodular matrix and  $b$  be an integral vector. Then the polyhedron  $P = \{x : Ax \leq b\}$  is integral.*

**Proof.** Let  $F = \{x : A'x = b'\}$  be a minimal face of  $P$ , where  $A'x \leq b'$  is a subsystem of  $Ax \leq b$  and  $A'$  has full row rank. Then we may permute the coordinates in such a way that  $A' = \begin{bmatrix} U & V \end{bmatrix}$  for some unimodular matrix  $U$ , and  $\begin{bmatrix} U^{-1}b' \\ \mathbf{0} \end{bmatrix}$  is an integral vector in  $F$ . ■

**Corollary 8.3** *Let  $A$  be a totally unimodular matrix and let  $b, c$  be integral vectors. Then both problems in the LP-duality equality*

$$\max\{c^T x : Ax \leq b\} = \min\{y^T b : y \geq 0 \text{ and } yA = c\}$$

*have integral optimal solutions.*

**Proof.** Also  $\begin{bmatrix} I \\ A^T \\ -A^T \end{bmatrix}$  is totally unimodular. ■

For  $A \in \mathbb{R}^{m \times n}$  of full row rank, a **basis** of  $A$  is a nonsingular submatrix of  $A$  of order  $m$ . The matrix  $A$  is **unimodular** if it is integral and each basis of  $A$  has determinant  $+1$  or  $-1$ . In case you still remember what a unimodular matrix was in Section 2.3, you can easily see that our current definition is equivalent to the one given there if  $A$  is a square matrix. Furthermore notice that  $A$  is totally unimodular if and only if  $[A \ I]$  is unimodular.

**Theorem 8.4 (Veinott & Danzig (1968))** *Let  $A$  be an integral matrix of full row rank. Then the polyhedron  $P(b) = \{x : x \geq \mathbf{0} \text{ and } Ax = b\}$  is integral for each integral vector  $b$  if and only if  $A$  is unimodular.*

**Proof.** Minimal faces of  $P(b)$  are vertices. Suppose  $A \in \mathbb{Z}^{m \times n}$  is unimodular,  $x'$  a vertex of  $P(b)$ . Then  $x'$  satisfies  $n$  linearly independent constraints with equality. The columns of  $A$  corresponding to non-zero components of  $x'$  are linearly independent. We can extend these columns to a basis  $B$  of  $A$ . Then  $x'$  restricted to the coordinates corresponding to  $B$  is  $B^{-1}b$ , which is integral. The rest of the entries are zeroes. Therefore  $x'$  is integral.

Now let  $P(b)$  be integral for each integral  $b$ . Let  $B$  be a basis of  $A$  and let  $t$  be an arbitrary integral vector. Then there exists integral  $y$  such that  $z = y + B^{-1}t \geq \mathbf{0}$ ;  $b = Bz = By + t$  is integral. Add zero components to  $z$  to get  $z'$  such that  $Az' = Bz = b$ ;  $z'$  satisfies  $n$  linearly independent constraints of  $P(b)$  with equality, hence it is a vertex of  $P(b)$ . Therefore  $z'$ ,  $z$  and  $B^{-1}t = z - y$  are integral. We have shown that  $B^{-1}t$  is integral for each integral  $t$ , and therefore  $\det B \in \{-1, +1\}$ . ■

**Theorem 8.5 (Hoffman & Kruskal (1956))** *Let  $A$  be an integral matrix. Then  $A$  is totally unimodular if and only if for every integral vector  $b$ , the polyhedron  $\{x : x \geq \mathbf{0} \text{ and } Ax \leq b\}$  is integral.*

**Proof.**  $A$  is totally unimodular if and only if  $[A \ I]$  is unimodular if and only if  $\{z : z \geq \mathbf{0} \text{ and } [A \ I]z = b\}$  is integral for every integral  $b$  if and only if  $\{x : x \geq \mathbf{0} \text{ and } Ax \leq b\}$  is integral for every integral  $b$ . ■

**Note.** There are plenty of other characterizations of totally unimodular matrices. Seymour (1980) proved a beautiful decomposition theorem for totally unimodular matrices, basically showing that every totally unimodular matrix arises in a well-described way from *network matrices* (matrices coming in a certain way from directed graphs and their spanning trees). Cunningham and Edmonds (1980) and Bixby (1982) then designed a polynomial algorithm for testing total unimodularity based on Seymour's decomposition theorem. Their work was done in the generalizing context of *matroids*.

### 8.3 Incidence matrices of graphs

First a sufficient condition for a matrix to be totally unimodular.

**Proposition 8.6 (Heller & Tompkins (1956))** *A matrix  $A$  is totally unimodular if*

- (i) *each entry is 0, 1, or  $-1$ ;*
- (ii) *each column contains at most two non-zeroes;*

(iii) the set  $N$  of row indices of  $A$  can be partitioned into  $N_1 \cup N_2$  so that in each column  $j$  with two non-zeroes we have  $\sum_{i \in N_1} a_{i,j} = \sum_{i \in N_2} a_{i,j}$ .

**Proof.** Assume that  $A$  is not totally unimodular and  $B$  is the smallest square submatrix with  $\det B \notin \{-1, 0, 1\}$ . By minimality,  $B$  contains no column with a single non-zero, so in every column there are two non-zeroes. By (iii) the rows of  $B$  are linearly dependent; hence  $\det B = 0$ , a contradiction. ■

**Corollary 8.7** *If  $M$  is the incidence matrix of a directed graph or the incidence matrix of a bipartite graph, then  $M$  is totally unimodular.*

In Sections 5.3–5.4 we saw an application of the integrality of polyhedra defined by incidence matrices of directed graphs. Now we will look at bipartite graphs and apply LP duality to get some interesting theoretical results.

**Definition 8.8** Let  $G = (V, E)$  be a graph. A subset  $I \subseteq V$  is an **independent set** if  $\binom{I}{2} \cap E = \emptyset$ . An **edge covering** of  $G$  is a subset  $F \subseteq E$  such that for every  $v \in V$  there is  $e \in F$  with  $v \in e$ . The **independence number**  $\alpha(G)$  of  $G$  is the cardinality of a maximum independent set. Let  $\beta'(G)$  be the minimum number of edges in an edge covering.

**Proposition 8.9** *Let  $M \in \{0, 1\}^{n \times m}$  be the incidence matrix of a graph  $G$ . If  $G$  has no isolated vertices, then*

$$\alpha(G) = \max\{\mathbf{1}^T x : M^T x \leq \mathbf{1}, x \geq \mathbf{0}, x \in \mathbb{Z}^n\}, \quad (\text{IP1})$$

$$\beta'(G) = \min\{y^T \mathbf{1} : My \geq \mathbf{1}, y \geq \mathbf{0}, y \in \mathbb{Z}^m\}. \quad (\text{IP2})$$

**Proof.** Because there are no isolated vertices, both LP relaxations are feasible and bounded. Integral feasible solutions to (IP1) are exactly incidence vectors of independent sets; integral feasible solutions to (IP2) with  $y \leq \mathbf{1}$  are exactly incidence vectors of edge coverings; an optimal solution obviously satisfies this inequality. ■

**Theorem 8.10 (The König–Rado Theorem (1933))** *In any bipartite graph without isolated vertices, the independence number is equal to the number of edges in a minimum edge covering.*

**Proof.** Let  $G$  be a bipartite graph without isolated vertices and with incidence matrix  $M$ . Then the LPs

$$\alpha_f := \max\{\mathbf{1}^T x : M^T x \leq \mathbf{1}, x \geq \mathbf{0}\}, \quad (\text{LP1})$$

$$\beta_f' := \min\{y^T \mathbf{1} : My \geq \mathbf{1}, y \geq \mathbf{0}\} \quad (\text{LP2})$$

both have integral optimal solutions, because  $M$  is totally unimodular. Hence  $\alpha(G) = \alpha_f$  and  $\beta'(G) = \beta_f$ . By LP duality,  $\alpha_f = \beta_f$ . ■

**Note.** Let  $\alpha'(G)$  denote the size of a maximum matching in a graph  $G$ . Gallai proved that in a graph  $G$  with no isolated vertices,  $\alpha'(G) + \beta'(G) = |V(G)|$ . Since there is a polynomial algorithm to compute  $\alpha'(G)$ , there is also one for  $\beta'(G)$ . This in turn gives another polynomial algorithm for computing  $\alpha(G)$  for bipartite graphs.

**Exercise 8.1** The optimal value of (LP1) is  $\alpha_f(G)$ , the **fractional independence number** of  $G$ . Note that computing  $\alpha(G)$  is NP-hard, whereas  $\alpha_f(G)$  can be computed in polynomial time. Show that for any graph  $G$  we have  $\alpha(G) \leq \alpha_f(G)$ . Show that the ratio  $\alpha_f(G)/\alpha(G)$  can be arbitrarily large. Hence solving (LP1) provides no constant-factor approximation for the independence number. (In fact, there is some constant  $\epsilon < 1$  such that it is impossible to approximate  $\alpha(G)$  within a factor of  $n^\epsilon$  in polynomial time unless  $P = NP$ .)

**Exercise 8.2** A **clique** in a graph is a set of mutually adjacent vertices. A **clique covering** is a set of cliques whose union is the entire vertex set of a graph. Formulate an IP that computes the minimum number of cliques in a clique covering. Show that the size of a maximum independent set in a graph is bounded from above by the minimum number of cliques in a clique covering. Give an example of a graph in which these two quantities are not equal.

**Definition 8.11** A **matching** in a graph is a set of pairwise disjoint edges. For a graph  $G$ , let  $\alpha'(G)$  be the size of a maximum matching in  $G$ . A **covering** (or vertex covering) of  $G$  is a set  $C$  of vertices such that each edge has at least one end in  $C$ ; the size of a minimum covering is denoted by  $\beta(G)$ .

**Proposition 8.12** Let  $M \in \{0, 1\}^{n \times m}$  be the incidence matrix of a graph  $G$ . Then

$$\alpha'(G) = \max\{\mathbf{1}^T x : Mx \leq \mathbf{1}, x \geq \mathbf{0}, x \in \mathbb{Z}^m\}, \quad (\text{IP3})$$

$$\beta(G) = \min\{y^T \mathbf{1} : M^T y \geq \mathbf{1}, y \geq \mathbf{0}, y \in \mathbb{Z}^n\}. \quad (\text{IP4})$$

**Proof.** Similar as before. ■

**Theorem 8.13 (The König–Egerváry Theorem (1931))** In any bipartite graph, the number of edges in a maximum matching is equal to the number of vertices in a minimum covering.

**Proof.** The LP relaxations corresponding to (IP3) and (IP4) have integral optimal solutions because  $M$  is totally unimodular. By duality of linear programming, the optimal values are equal. ■

**Exercise 8.3** Show that for any graph  $G = (V, E)$ , a subset  $S \subseteq V$  is a maximum independent set if and only if  $V \setminus S$  is a minimum covering.

**Note.** The exercise shows that  $\alpha(G) + \beta(G) = |V(G)|$ . Hence computing  $\beta(G)$  is NP-hard in general, even though it is polynomial for bipartite graphs.

## 9 Valid inequalities

Most of our exposition in this chapter is based on the book [9]; Section 9.5 follows [11, §23.1].

### 9.1 Valid inequalities for polyhedra

An inequality  $d^T x \leq d_0$  is **valid** for a set  $S \subseteq \mathbb{R}^n$  if  $d^T x \leq d_0$  for all  $x \in S$ . The inequalities  $d^T x \leq d_0$  and  $d'^T x \leq d'_0$  are **equivalent** if  $(d, d_0) = \lambda(d', d'_0)$  for some  $\lambda > 0$ . If they are not equivalent but there is  $\lambda > 0$  such that  $d' \geq \lambda d$  and  $d'_0 \leq \lambda d_0$ , then  $\{x : x \geq \mathbf{0}, d'^T x \leq d'_0\} \subset \{x : x \geq \mathbf{0}, d^T x \leq d_0\}$  and we say that  $d'^T x \leq d'_0$  **dominates**  $d^T x \leq d_0$ . If a valid inequality is not dominated by any other valid inequality, it is called a **maximal valid inequality**.

**Note.** Any maximal valid inequality for  $S$  defines a nonempty face of  $\text{conv}(S)$ . Any facet-defining inequality of  $\text{conv}(S)$  is a maximal valid inequality for  $S$ .

In the context of integer programming, we will be interested in valid inequalities for  $S = P \cap \mathbb{Z}^n$  for a polyhedron  $P$ ; since the valid inequalities for  $S$  and for  $P_I = \text{conv} S$  are the same, we use the two terms interchangeably. First we investigate valid inequalities for  $P$ .

**Theorem 9.1** *Let  $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq \mathbf{0}\} \neq \emptyset$ ,  $A \in \mathbb{R}^{m \times n}$ , and let  $d^T x \leq d_0$  be a valid inequality for  $P$ . Then there exists  $u \in \mathbb{R}^m$ ,  $u \geq \mathbf{0}$ , with at most  $\min\{m, n\}$  non-zero components such that  $d^T x \leq d_0$  is equivalent to or dominated by  $(u^T A)x \leq u^T b$ .*

**Proof.** The linear program  $D = \max\{d^T x : x \geq \mathbf{0}, Ax \leq b\}$  is feasible because  $P \neq \emptyset$  and bounded because  $D \leq d_0$ . Therefore the dual  $\min\{b^T u : u \geq \mathbf{0}, u^T A \geq d\}$  is also feasible and bounded. A basic optimal solution  $u$  proves the claim. ■

**Note.** Theorem 9.1 characterizes all valid inequalities if  $P \neq \emptyset$ , that is, if the *primal* LP  $\max\{d^T x : x \geq \mathbf{0}, Ax \leq b\}$  is feasible. A similar result applies if the *dual* LP  $\min\{b^T u : u \geq \mathbf{0}, u^T A \geq d\}$  is feasible. However, if both the primal and the dual are infeasible, then  $P = \emptyset$  and so *any* inequality is valid for  $P$ . They cannot, however, all be generated as linear combinations of the defining inequalities  $Ax \leq b$ . Therefore it is sometimes assumed that  $A = \begin{bmatrix} A' \\ I \end{bmatrix}$ , which provides an implicit bounding box for the primal problem and makes the dual feasible.

Next we look at some methods to generate valid inequalities for  $P_I$ .

### 9.2 Integer rounding

The main idea is that if  $a \leq b$  and  $a$  is an integer, then  $a \leq \lfloor b \rfloor$ .

Recall from Definition 8.11 that a matching in a graph is a set  $M$  of edges such that no vertex is contained in more than one edge from  $M$ . A maximum matching is a matching of largest cardinality. By Proposition 8.12, a maximum matching in a graph  $G = (V, E)$  can be computed by solving the integer program

$$\alpha'(G) = \max \left\{ \mathbf{1}^T x : \forall i \in V, \sum_{\substack{e \in E \\ i \in e}} x_e \leq 1; x \geq \mathbf{0}; x \in \mathbb{Z}^m \right\}. \quad (\text{IP3})$$

Just like we did in Section 5.2, we can argue that within any set  $S$  of vertices, there may be at most  $\lfloor |S|/2 \rfloor$  matching edges. Hence for any  $S \subseteq V$  the inequality

$$\sum_{\substack{e \in E \\ e \subseteq S}} x_e \leq \left\lfloor \frac{|S|}{2} \right\rfloor \quad (9.1)$$

is valid for

$$P_I = \text{conv} \left\{ x \in \mathbb{Z}^m : \forall i \in V, \sum_{\substack{e \in E \\ i \in e}} x_e \leq 1; x \geq \mathbf{0} \right\}.$$

In the RHS of (9.1) we may write  $\lfloor |S|/2 \rfloor$  instead of  $|S|/2$  because the LHS is an integer.

Is there a way to get (9.1) just from the inequalities of (IP3)? Let  $S \subseteq V$  and take a linear combination of the inequalities with coefficients  $u_i = 1/2$  for  $i \in S$  and  $u_i = 0$  for  $i \in V \setminus S$ . Thus we obtain

$$\sum_{\substack{e \in E \\ e \subseteq S}} x_e + \frac{1}{2} \sum_{\substack{e \in E \\ |e \cap S|=1}} x_e \leq \frac{|S|}{2}. \quad (9.2)$$

Since all  $x_e \geq 0$ , we have

$$-\frac{1}{2} \sum_{\substack{e \in E \\ |e \cap S|=1}} x_e \leq 0;$$

adding to (9.2) yields

$$\sum_{\substack{e \in E \\ e \subseteq S}} x_e \leq \frac{|S|}{2}. \quad (9.3)$$

Finally, as before, since the LHS of (9.3) is an integer, we get (9.1). In this case, taking inequalities 9.1 for all subsets  $S \subseteq V$  determines  $P_I$ ; this is not true for general IPs.

**Proposition 9.2** *Let  $P = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, Ax \leq b\}$ ,  $A = [a_1 \ a_2 \ \dots \ a_n]$ . Then for any  $u \geq \mathbf{0}$ , the inequality*

$$\sum_{j=1}^n \lfloor u^T a_j \rfloor x_j \leq \lfloor u^T b \rfloor \quad (9.4)$$

*is valid for  $P \cap \mathbb{Z}^n$ .*

**Proof.** First,

$$\sum_{j=1}^n u^T a_j x_j \leq u^T b$$

is valid because it is a non-negative linear combination of the valid inequalities  $Ax \leq b$ . Moreover,

$$-(u^T a_j - \lfloor u^T a_j \rfloor) x_j \leq 0$$

for every  $j$ . Hence

$$\sum_{j=1}^n \lfloor u^T a_j \rfloor x_j \leq u^T b$$

is valid. Finally,

$$\sum_{j=1}^n \lfloor u^T a_j \rfloor x_j \leq \lfloor u^T b \rfloor$$

is valid, because the LHS is an integer. ■

An inequality of the form (9.4) is called a **Chvátal–Gomory inequality (CGI)**.

### 9.3 Gomory cutting planes

Let  $P = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, a^T x = b\}$  and let  $S = \{x \in \mathbb{Z}^n : x \geq \mathbf{0}, a^T x - b \in \mathbb{Z}\}$ . Clearly,

$$S = \left\{ x \in \mathbb{Z}^n : x \geq \mathbf{0}, \sum_{i=1}^n (a_i - \lfloor a_i \rfloor) x_i - (b - \lfloor b \rfloor) \in \mathbb{Z} \right\}.$$

Since  $\sum_{i=1}^n (a_i - \lfloor a_i \rfloor) x_i \geq 0$  and  $b - \lfloor b \rfloor < 1$ , we get that

$$\sum_{i=1}^n (a_i - \lfloor a_i \rfloor) x_i \geq b - \lfloor b \rfloor \tag{9.5}$$

is valid for  $S$ . Because  $P \cap \mathbb{Z}^n \subseteq S$ , (9.5) is also valid for  $P \cap \mathbb{Z}^n$ . An inequality of the form (9.5) is called a **Gomory cutting plane (GCP)**.

**Example 9.1** 1. If

$$\frac{7}{5}x_1 - \frac{3}{4}x_2 + \frac{1}{4}x_3 - \frac{2}{5}x_4 = \frac{19}{5}$$

and the variables are required to be non-negative integers, then

$$\frac{2}{5}x_1 + \frac{1}{4}x_2 + \frac{1}{4}x_3 + \frac{3}{5}x_4 \geq \frac{4}{5}.$$

2. If

$$13x_1 - 5x_2 + 19x_3 = 35$$

and the variables are non-negative integers, then

$$\frac{13}{6}x_1 - \frac{5}{6}x_2 + \frac{19}{6}x_3 = \frac{35}{6}$$

and thus

$$\begin{aligned} \frac{1}{6}x_1 + \frac{1}{6}x_2 + \frac{1}{6}x_3 &\geq \frac{5}{6}, \\ x_1 + x_2 + x_3 &\geq 5. \end{aligned}$$

## 9.4 Disjunctive constraints

**Example 9.2** Let  $P = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, -2x_1 + 2x_2 \leq 3, 2x_1 + x_2 \leq 3\}$ . The constraints defining  $P$  are equivalent to

$$x_1 + 2x_2 - 3x_1 \leq 3, \quad (9.6a)$$

$$x_1 + 2x_2 + 3(x_1 - 1) \leq 3. \quad (9.6b)$$

If  $x_1 \in \mathbb{Z}$ , then  $x_1 \leq 0$  or  $x_1 \geq 1$ . In the former case, (9.6a) implies that  $x_1 + 2x_2 \leq 3$ ; in the latter case (9.6b) implies that  $x_1 + 2x_2 \leq 3$ . Hence  $x_1 + 2x_2 \leq 3$  is valid for  $P \cap \mathbb{Z}^2$ .

**Proposition 9.3** Let  $S \subseteq \mathbb{Z}^n$ . If  $r, s$  are positive numbers such that  $d^T x + r(x_i - \ell) \leq d_0$  is valid for  $S$  and  $d^T x - s(x_i - \ell + 1) \leq d_0$  is valid for  $S$ , then  $d^T x \leq d_0$  is valid for  $S$ .

**Proof.** Exercise. ■

**Definition 9.4** Let  $S = \{x \in \mathbb{Z}^n : Ax \leq b\}$ . **Disjunctive constraints (DCs)** for  $S$  are defined recursively as follows:

1. All the inequalities of  $Ax \leq b$  are DCs.
2. A non-negative linear combination of DCs is a DC.
3. An inequality obtained from DCs using Proposition 9.3 is a DC.
4. An inequality equivalent to or dominated by a DC is a DC.

Next we consider 0, 1-programs. Let  $S = \{x \in \{0, 1\}^n : Ax \leq b\}$ ,  $P_I = \text{conv } S$ ,  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ . Our goal is the following theorem.

**Theorem 9.5** Let  $S = \{x \in \{0, 1\}^n : Ax \leq b\}$ . Then every valid inequality for  $S$  is a DC.

In fact,  $P_I$  is determined by inequalities in a special form. For an inequality  $d^T x \leq d_0$  valid for  $S$ ,  $t \in \{0, 1, \dots, n\}$ ,  $r \geq 0$  and  $N_0, N_1$  such that  $N_0 \cap N_1 = \emptyset$ ,  $N_0 \cup N_1 = \{1, 2, \dots, t\}$ , let  $\text{DC}(d, d_0, t, r, N_0, N_1)$  be the inequality

$$\sum_{j=1}^n d_j x_j - r \sum_{j \in N_0} x_j - r \sum_{j \in N_1} (1 - x_j) \leq d_0. \quad (\text{DC}(d, d_0, t, r, N_0, N_1))$$

**Lemma 9.6** If  $d^T x \leq d_0$  is valid for  $S$ , then there exists  $r \geq 0$  such that all the inequalities  $\text{DC}(d, d_0, n, r, N_0, N_1)$  for all partitions  $(N_0, N_1)$  of  $\{1, \dots, n\}$  are valid for  $P$ .

**Proof.** If  $P = \emptyset$ , then any inequality is valid for  $P$ . Otherwise  $P$  is nonempty and bounded, and so it suffices to show that each such inequality is valid for the vertices of  $P$ . Determining the right value of  $r$  is left as an exercise. ■

Let  $P_n = P$  and let  $P_t = \text{conv}\left(\left(P_{t+1} \cap \{x : x_{t+1} = 0\}\right) \cup \left(P_{t+1} \cap \{x : x_{t+1} = 1\}\right)\right)$  for  $t = 0, 1, \dots, n-1$ .

**Lemma 9.7** If  $d^T x \leq d_0$  is valid for  $S$ , then there is some  $r \geq 0$  such that every inequality  $\text{DC}(d, d_0, t, r, N_0, N_1)$  is a DC for  $P_t$ .

**Proof.** For  $t = n$  the assertion follows from Lemma 9.6 and Theorem 9.1. Then we proceed by downward induction. By induction hypothesis, the inequalities

$$\begin{aligned} d^T x - r \sum_{j \in N_0 \cup \{t+1\}} x_j - r \sum_{j \in N_1} (1 - x_j) &\leq d_0, \\ d^T x - r \sum_{j \in N_0} x_j - r \sum_{j \in N_1 \cup \{t+1\}} (1 - x_j) &\leq d_0 \end{aligned}$$

are DCs for  $P_{t+1}$ , which is exactly the setup of Proposition 9.3. It follows that

$$d^T x - r \sum_{j \in N_0} x_j - r \sum_{j \in N_1} (1 - x_j) \leq d_0$$

is a DC for  $P_t$ . ■

**Proof.** (of Theorem 9.5) Lemmas 9.6 and 9.7 imply that every valid inequality for  $P_I$  is valid for  $P_0$ . Conversely, every point of  $S$  is contained in  $P_0$ . Hence  $P_0 = P_I$ . Finally, let  $d^T x \leq d_0$  be valid for  $S$ . But  $\text{DC}(d, d_0, 0, r, \emptyset, \emptyset)$  is exactly  $d^T x \leq d_0$ , and it is a DC. ■

## 9.5 All valid inequalities for IPs

All polyhedra we consider in this section will be subsets of the positive orthant, i.e., the set  $\{x : x \geq \mathbf{0}\}$ , even if the theory can be developed for general polyhedra. Our assumption is convenient mainly because of our definition of dominated inequalities.

**Definition 9.8** Let  $P = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, Ax \leq b\}$ . An inequality  $dx \leq d_0$  valid for the integer hull  $P_I$  is

- ▷ a **Chvátal rank 0 inequality** if it is equivalent to or dominated by a non-negative linear combination of the inequalities  $Ax \leq b$  defining  $P$ ;
- ▷ a **Chvátal rank  $t$  inequality** if it is not a Chvátal rank  $t'$  inequality for any  $t' < t$  but it is equivalent to or dominated by a non-negative linear combination of inequalities of the form

$$\sum_{j=1}^n \lfloor u^T g_j \rfloor x_j \leq \lfloor u^T h \rfloor,$$

where  $Gx \leq h$  is a system of inequalities of Chvátal rank at most  $t - 1$ ,  $g_j$  is the  $j$ th column of  $G$  and  $u \geq \mathbf{0}$  (cf. Proposition 9.2);

- ▷ a **Chvátal–Gomory inequality (CGI)** if it is a Chvátal rank  $t$  inequality for some  $t$ .

Let  $P^t = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, x \text{ satisfies all inequalities of Chvátal rank at most } t \text{ for } P\}$ . Clearly  $P^0 = P$ ,  $P_I \subseteq P^t$  and  $P^{t+1} \subseteq P^t$  for all  $t$ . Our goal is to prove:

**Theorem 9.9 (Schrijver (1980))** *For every rational polyhedron  $P = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, Ax \leq b\}$  there exists a non-negative integer  $t$  such that  $P_I = P^t$ .*

**Corollary 9.10** *Every valid inequality for the integer hull of a rational polyhedron is a CGI.*

Recall from Section 5.6 that a rational system  $Ax \leq b$  is **totally dual integral** if for any integral vector  $c$ , if the LP  $\min\{b^T y : y^T A = c, y \geq \mathbf{0}\}$  has an optimal solution, it has an integral optimal solution.

**Lemma 9.11** *Suppose  $P = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, Ax \leq b\}$  is a polyhedron with  $A$  integral and  $x \geq \mathbf{0}, Ax \leq b$  totally dual integral. Then  $P^1 = P^* := \{x \in \mathbb{R}^n : x \geq \mathbf{0}, Ax \leq \lfloor b \rfloor\}$ , where  $\lfloor b \rfloor$  is the vector whose  $i$ th component is  $\lfloor b_i \rfloor$ .*

**Proof.** Each inequality among  $Ax \leq \lfloor b \rfloor$  has Chvátal rank at most 1; thus  $P^1 \subseteq P^*$ . Now consider  $\bar{x} \in P^*$  and  $u \geq \mathbf{0}$ . We want to show that

$$\sum_{j=1}^n \lfloor u^T a_j \rfloor \bar{x}_j \leq \lfloor u^T b \rfloor.$$

Observe that

$$\sum_{j=1}^n \lfloor u^T a_j \rfloor x_j \leq u^T Ax \leq u^T b$$

for any  $x \in P$ . Hence

$$\begin{aligned} u^T b &\geq \max \left\{ \sum_{j=1}^n \lfloor u^T a_j \rfloor x_j : x \geq \mathbf{0}, Ax \leq b \right\} \\ &= \min \{ b^T y : y \geq \mathbf{0}, y^T a_j \geq \lfloor u^T a_j \rfloor \text{ for all } j \}. \end{aligned} \quad (9.7)$$

Because of total dual integrality, the minimization problem has an integral optimal solution  $y^*$ . Therefore

$$\begin{aligned} \sum_{j=1}^n \lfloor u^T a_j \rfloor \bar{x}_j &\leq y^{*T} A \bar{x} && \text{because } y^{*T} a_j \geq \lfloor u^T a_j \rfloor \text{ for all } j \\ &\leq y^{*T} \lfloor b \rfloor && \text{because } \bar{x} \in P^* \\ &\leq \lfloor y^{*T} b \rfloor && \text{because } y^* \geq \mathbf{0} \\ &\leq \lfloor u^T b \rfloor && \text{by (9.7).} \end{aligned}$$

■

**Lemma 9.12** *For any rational polyhedron  $P$ ,  $P^1$  is a polyhedron.*

**Proof.** By a theorem of Giles and Pulleyblank (1979), for every rational polyhedron  $P$  there is a totally dual integral system  $Ax \leq b$  with  $A$  integral such that  $P = \{x : Ax \leq b\}$ . Then the claim follows from Lemma 9.11. ■

**Lemma 9.13** *If  $F$  is a face of a rational polyhedron  $P$ , then  $F^t = F \cap P^t$ .*

**Proof.** We prove that  $F^1 = F \cap P^1$ ; the rest follows by induction. By the same theorem of Giles and Pulleyblank (1979),  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  with  $A$  integral and  $Ax \leq b$  totally dual integral. A face  $F = \{x \in \mathbb{R}^n : Ax \leq b, d^T x = d_0\}$ , where  $d^T x \leq d_0$  is some valid

inequality for  $P$ ,  $d \in \mathbb{Z}^n$ ,  $d_0 \in \mathbb{Z}$ . It can be shown that the system  $Ax \leq b$ ,  $d^T x = d_0$  is totally dual integral; thus by Lemma 9.11,

$$F^1 = \{x \in \mathbb{R}^n : Ax \leq [b], d^T x \leq [d_0], d^T x \geq [d_0]\} = \{x \in \mathbb{R}^n : Ax \leq [b], d^T x = d_0\},$$

whereas  $P^1 = \{x \in \mathbb{R}^n : Ax \leq [b]\}$ , and so  $F \cap P^1 = \{x \in \mathbb{R}^n : Ax \leq [b], d^T x = d_0\}$ . ■

**Proof.** (of Schrijver's Theorem 9.9) By induction on the dimension  $d$  of  $P$ . If  $d = -1$  ( $P = \emptyset$ ) or  $d = 0$  ( $P = \{x\}$ ), the claim is trivial.

For higher dimensions, we have already observed that  $P_I \subseteq P^t$  for all  $t$ ; thus it remains to show that  $P^t \subseteq P_I$  for some  $t$ . We distinguish three cases:

1. *The affine hull  $\text{aff}(P)$  contains no integral vectors.* Then  $\text{aff}(P)$  is contained in a hyperplane  $H = \{x \in \mathbb{R}^n : a^T x = b\}$  with  $H \cap \mathbb{Z}^n = \emptyset$  and  $a$  integral; the inequalities  $a^T x \leq b$  and  $a^T x \geq b$  are valid for  $P$ , hence (using Lemma 9.1)

$$P^1 \subseteq \{x \in \mathbb{R}^n : a^T x \geq [b], a^T x \leq [b]\} = \emptyset = P_I.$$

2.  *$\text{aff}(P)$  contains integral vectors but  $P$  is not full-dimensional.*

By Hermite normal form theory and because the theorem is invariant under integral translations, we may assume that  $\text{aff}(P) = \{x \in \mathbb{R}^n : [B \ \mathbf{0}] x = \mathbf{0}\}$  for some invertible  $B$ , hence  $\text{aff}(P) = \{0\}^{n-d} \times \mathbb{R}^d$  and  $P = \{0\}^{n-d} \times P'$  for some full-dimensional polyhedron  $P' \subseteq \mathbb{R}^d$ . Since  $P_I = \{0\}^{n-d} \times P'_I$  and  $P^t = \{0\}^{n-d} \times (P')^t$ , we may assume that:

3.  *$P$  is full-dimensional.* It is not difficult to show that there exists an integral  $m \times n$  matrix  $A$  and  $b, b' \in \mathbb{Q}^m$  such that  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  and  $P' = \{x \in \mathbb{R}^n : Ax \leq b'\}$ .

**Claim.** *Every inequality  $a_i^T x \leq b'_i$  from the system  $Ax \leq b'$  is valid for  $P^{t_i}$  for some non-negative integer  $t_i$ .*

If we can establish the claim, let  $t = \max\{t_i : i = 1, \dots, m\}$ ; then  $P^t \subseteq P_I$ .

*Proof of Claim.* By contradiction. Assume that  $a_i^T x \leq b'_i$  is not valid for any  $P^s$ . The inequality  $a_i^T x \leq b'_i$  has Chvátal rank 0 for  $P$ , thus  $a_i^T x \leq [b'_i]$  has Chvátal rank at most 1. Hence there exists  $b''_i \in \mathbb{Z}$  such that  $b'_i < b''_i \leq [b'_i]$  and for all sufficiently large  $s$ , the inequality  $a_i^T x \leq b''_i$  is valid for  $P^s$  but  $a_i^T x \leq b''_i - 1$  is *not* valid for  $P^s$ .

In particular, the set  $\{x \in \mathbb{R}^n : a_i^T x = b''_i\}$  contains no integral vectors; nor does  $F = P^s \cap \{x \in \mathbb{R}^n : a_i^T x = b''_i\}$ , whose dimension is at most  $n - 1$ .

If  $F = \emptyset$ , then  $P^s \subseteq \{x \in \mathbb{R}^n : a_i^T x < b''_i\}$ , and so  $P^{s+1} \subseteq \{x \in \mathbb{R}^n : a_i^T x \leq b''_i - 1\}$ , a contradiction. Hence  $F$  is a nonempty face of  $P^s$ . By induction,  $F^r = \emptyset$  for some non-negative integer  $r$ . Therefore

$$P^{s+r} \cap \{x \in \mathbb{R}^n : a_i^T x = b''_i\} = P^{s+r} \cap F = F^r = \emptyset,$$

and so  $P^{s+r} \subseteq \{x \in \mathbb{R}^n : a_i^T x < b''_i\}$  and  $P^{s+r+1} \subseteq \{x \in \mathbb{R}^n : a_i^T x \leq b''_i - 1\}$ , again a contradiction. ■

Another consequence of Theorem 9.9 is the following earlier result:

**Theorem 9.14 (Chvátal (1973))** *For each polytope  $P$  there exists a non-negative integer  $t$  such that  $P_I = P^t$ .*

Here, the assumption of rationality is replaced with boundedness.

The **Chvátal rank of a polyhedron**  $P$  is the maximum Chvátal rank of an inequality valid for  $P_I$ . Theorem 9.9 implies that every polyhedron has finite Chvátal rank.

There is an interesting connection between the complexity of a (combinatorial) optimization problem and the Chvátal ranks of arising polyhedra. For instance, in Section 9.2, we considered the polyhedra arising from (IP3), the integer program to compute a maximum matching in a graph. Edmonds (1965) showed that all these so-called *matching polyhedra* have Chvátal rank at most 1. On the other hand, Boyd and Pulleyblank (1984) proved that unless  $\text{NP} = \text{co-NP}$ , if the optimization problem over the polyhedra arising from an NP-hard problem can be solved in polynomial time, then their Chvátal rank is not bounded by any constant.

## 10 General IP algorithms

A general algorithm to solve  $\max\{c^T x : x \in S\}$  for some  $S \subseteq \mathbb{R}^n$  may be described as follows:

1. **Initialization:** Input:  $c$ , some representation of  $S$ , possibly an upper bound  $w^*$  and a feasible solution  $x^*$ . Set  $t := 0$ ;  $z^* := c^T x^*$  if  $x^*$  is given, or  $z^* := -\infty$  otherwise.
2. **Optimality test:** If  $w^* = z^*$ , return  $x^*$  and stop.
3. **Dual step:** Find an upper bound  $w^t$ . If  $w^t < w^*$ , set  $w^* := w^t$ .
4. **Primal step:** Find a feasible solution  $x^t \in S$ . If  $c^T x^t > z^*$ , set  $x^* := x^t$  and  $z^* := c^T x^t$ .
5. Set  $t := t + 1$  and go to 2.

In each iteration (or in all iterations, depending on the algorithm), the primal step or the dual step may be omitted. Some algorithms actually do not find a feasible solution until at the very end. One such class of algorithms is fractional cutting plane algorithms, which we study next.

### 10.1 A fractional cutting plane algorithm

These algorithms are **relaxation algorithms**, which means that they obtain an upper bound in the  $t$ th dual step by solving  $\max\{c^T x : x \in P^t\}$  for some set  $P^t \supset S$ . If the found optimal solution to the relaxation belongs to the set  $S$ , the algorithm stops; otherwise it continues with a stronger relaxation ( $P^{t+1} \subset P^t$ ).

To be concrete, we assume that the set  $S$  is specified as  $S = P \cap \mathbb{Z}^n$ , where  $P = \{x \in \mathbb{R}^n : x \geq \mathbf{0}, Ax = b\}$  is a polyhedron represented by a matrix  $A \in \mathbb{Z}^{m \times n}$  and a vector  $b \in \mathbb{Z}^m$  and the objective function is given by an integral vector  $c \in \mathbb{Z}^n$ .

Then  $P^0 = P$  and  $P^t$  will be obtained by adding an inequality valid for  $S$  in each iteration of the algorithm, i.e.,  $P^{t+1} = P^t \cap \{x \in \mathbb{R}^n : d^{tT} x \leq d_0^t\}$ . A basic requirement is that the found optimal solution  $x^t$  to the relaxation  $\max\{c^T x : x \in P^t\}$  violates the added constraint, that is,  $d^{tT} x^t > d_0^t$ .

Let us get more concrete.

#### Gomory fractional cuts

The problem to solve is

$$\max\{x_0 : x_0 \in \mathbb{Z}, x \in \mathbb{Z}^n, x \geq \mathbf{0}, x_0 - c^T x = 0, Ax = b\}. \quad (10.1)$$

We assume that  $A$ ,  $b$  and  $c$  are all integral. Suppose we have obtained an optimal basic solution to the LP relaxation

$$\max\{x_0 : x_0 \in \mathbb{R}, x \in \mathbb{R}^n, x \geq \mathbf{0}, x_0 - c^T x = 0, Ax = b\}, \quad (10.2)$$

with basis  $B$  and non-basis  $N$ ;  $B, N \subseteq \{1, 2, \dots, n\}$ . Then our problem can be rewritten as:

$$\begin{aligned} \max \quad & x_0 \\ \text{s.t.} \quad & x_i + \sum_{j \in N} a_{ij}^0 x_j = b_i^0 \quad \text{for } i = 0, 1, \dots, n \\ & x_i \geq 0 \quad \text{for } i = 1, \dots, n \\ & x_i \in \mathbb{Z} \quad \text{for } i = 0, 1, \dots, n \end{aligned} \quad (10.3)$$

For  $j \in N$  we get the trivial identities  $x_j - x_j = 0$ , that is,  $a_{jj}^0 = 1$ ,  $a_{jk}^0 = 0$  for  $k \neq j$  and  $b_j^0 = 0$ . Assuming primal and dual feasibility, we have  $b_j^0 \geq 0$  for  $j = 1, \dots, n$  and  $a_{0j}^0 \geq 0$  for  $j \in N$ .

If some  $b_i^0 \notin \mathbb{Z}$ , then we get the **Gomory fractional cut**

$$\sum_{j \in N} (a_{ij}^0 - \lfloor a_{ij}^0 \rfloor) x_j = b_i^0 - \lfloor b_i^0 \rfloor + x_{n+1}, \quad x_{n+1} \geq 0. \quad (10.4)$$

**Proposition 10.1** *If some  $b_i^0 \notin \mathbb{Z}$  in (10.3), then (10.4),  $x_{n+1} \in \mathbb{Z}$  are valid for all feasible solutions of the IP (10.1).*

**Proof.** Follows directly from the discussion in Section 9.3. ■

To solve the IP (10.1), we repeatedly solve the corresponding LP relaxation, add a Gomory cut to strengthen the relaxation, and if we are lucky, after a finite number of iterations the relaxation will have an integral solution.

**Example 10.1** Solve the IP:

$$\begin{array}{llll} \max & -x_1 & +2x_2 & \\ \text{s.t.} & -4x_1 & -2x_2 & \leq -7 \\ & -7x_1 & +6x_2 & \leq 2 \\ & 16x_1 & +15x_2 & \leq 72 \\ & & x_1, x_2 & \geq 0 \\ & & x_1, x_2 & \in \mathbb{Z} \end{array}$$

We rewrite the IP in equational form with slack variables  $x_3, x_4, x_5$ :

$$\begin{array}{llllll} \max & x_0 & & & & \\ \text{s.t.} & x_0 & +x_1 & -2x_2 & & = 0 \\ & & -4x_1 & -2x_2 & +x_3 & = -7 \\ & & -7x_1 & +6x_2 & & +x_4 = 2 \\ & & 16x_1 & +15x_2 & & +x_5 = 72 \\ & & & & x_0, x_1, x_2, x_3, x_4, x_5 & \geq 0 \\ & & & & x_0, x_1, x_2, x_3, x_4, x_5 & \in \mathbb{Z} \end{array}$$

Solving the LP relaxation yields the tableau

$$\begin{array}{llll} x_0 & +\frac{47}{201}x_4 & +\frac{8}{201}x_5 & = \frac{10}{3} \\ x_1 & -\frac{5}{67}x_4 & -\frac{2}{67}x_5 & = 2 \\ x_2 & +\frac{16}{201}x_4 & +\frac{7}{201}x_5 & = \frac{8}{3} \\ x_3 & -\frac{28}{201}x_4 & +\frac{38}{201}x_5 & = \frac{19}{3} \end{array}$$

From the first row, we obtain the Gomory fractional cut

$$\frac{47}{201}x_4 + \frac{8}{201}x_5 = \frac{1}{3} + x_6,$$

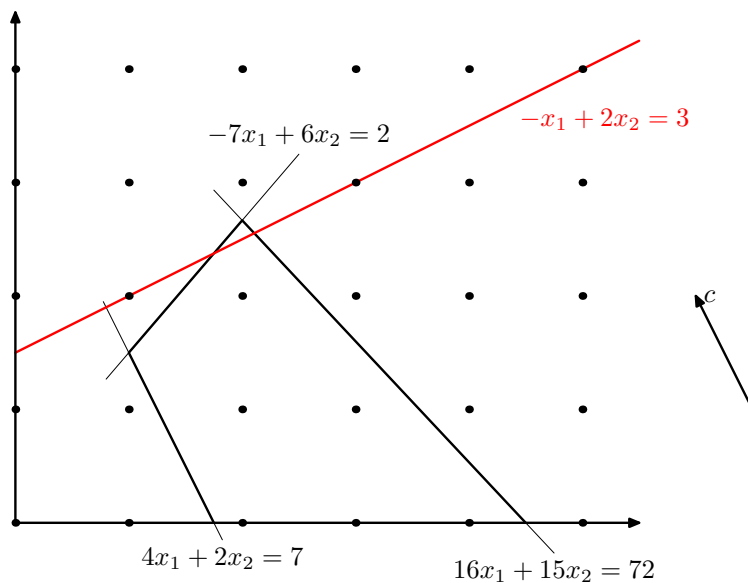


Figure 10.1: A Gomory cut

with a new slack variable  $x_6 \geq 0$ ,  $x_6 \in \mathbb{Z}$ . The cut corresponds to the inequality  $-x_1 + 2x_2 \leq 3$  in the original variables; see Figure 10.1.

Solving the tightened relaxation, we notice that the optimum is attained on a 1-dimensional face of the polytope; a basic solution is given by:

$$\begin{array}{rcl}
 x_0 & & +x_6 = 3 \\
 x_1 & +\frac{2}{47}x_5 & -\frac{15}{47}x_6 = \frac{99}{47} \\
 x_2 & +\frac{1}{47}x_5 & +\frac{16}{47}x_6 = \frac{120}{47} \\
 x_3 & +\frac{10}{47}x_5 & -\frac{28}{47}x_6 = \frac{307}{47} \\
 x_4 & +\frac{8}{47}x_5 & -\frac{201}{47}x_6 = \frac{67}{47}
 \end{array}$$

Later we will see that this is the *lexicographically largest* basic solution. Now we use the third row to get the Gomory cut

$$\frac{1}{47}x_5 + \frac{16}{47}x_6 = \frac{26}{47} + x_7,$$

corresponding to the inequality  $x_2 \leq 2$ .

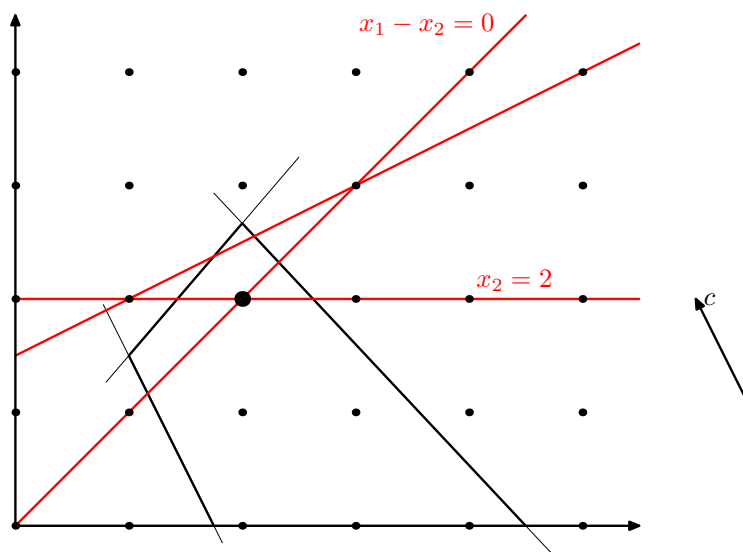


Figure 10.2: After three Gomory cuts

This time, we get the optimal solution from the tableau:

$$\begin{array}{rclcl}
 x_0 & +\frac{1}{7}x_4 & +\frac{8}{7}x_7 & = & \frac{18}{7} \\
 x_1 & -\frac{1}{7}x_4 & +\frac{6}{7}x_7 & = & \frac{10}{7} \\
 x_2 & & +x_7 & = & 2 \\
 x_3 & -\frac{4}{7}x_4 & +\frac{38}{7}x_7 & = & \frac{19}{7} \\
 x_5 & +\frac{16}{7}x_4 & -\frac{201}{7}x_7 & = & \frac{134}{7} \\
 x_6 & -\frac{1}{7}x_4 & -\frac{8}{7}x_7 & = & \frac{3}{7}
 \end{array}$$

The first row gives the cut

$$\frac{1}{7}x_4 + \frac{1}{7}x_7 = \frac{4}{7} + x_8,$$

corresponding to  $x_1 - x_2 \geq 0$ .

This relaxation has an integral solution  $(x_0, x_1, \dots, x_8) = (2, 2, 2, 5, 4, 10, 1, 0, 0)$  (see Figure 10.2). We conclude that the optimal solution to the original IP is  $(x_1, x_2) = (2, 2)$  with optimal objective value 2.

Next we will examine a version of the simplex algorithm for solving LPs. Using this algorithm for solving the LP relaxations, together with a clever selection of the Gomory cuts, will guarantee finding the optimum in a finite number of iterations.

### The lexicographic dual simplex algorithm

Given a dual-feasible basis  $B$ , we want to solve the LP given by (10.3) (without integrality constraints). Dual-feasibility means that  $a_{0j}^0 \geq 0$  for all  $j \in N$ . If  $b^0 \geq \mathbf{0}$ , then the

corresponding basic solution is also primal-feasible and it is an optimal solution. Otherwise the algorithm replaces  $B$  with a new basis obtained by the following pivot rule:

- ▷ to *leave the basis*, it chooses any basic variable  $x_i$  with  $b_i^0 < 0$  ( $i \neq 0$ );
- ▷ if there is no  $j \in N$  such that  $a_{ij}^0 < 0$ , then the LP is *primal-infeasible*;
- ▷ otherwise choose the variable  $x_k$ ,  $k \in N$  to *enter the basis*, for which  $\frac{1}{a_{ik}^0} a_k^0$  is lexicographically largest. Note that  $k$  is uniquely determined.

Here and in the following,  $a_k^t = (a_{0k}^t, a_{1k}^t, \dots, a_{nk}^t)$ .

**Lexicographically largest** means maximal in the **lexicographic ordering**:

$$(a_0, a_1, \dots, a_m) <_{\text{lex}} (b_0, b_1, \dots, b_n)$$

if there exists  $j \geq 0$  such that  $a_i = b_i$  for all  $i < j$  and  $a_j < b_j$ , or if  $m < n$  and  $a_i = b_i$  for all  $i \leq m$ . It reminds of the alphabetical ordering of a phone directory.

Let us consider the set of all feasible solutions to the LP (10.2):

$$P = \{(x_0, x) : x_0 \in \mathbb{R}, x \in \mathbb{R}^n, x \geq \mathbf{0}, x_0 - c^T x = 0, Ax = b\}. \quad (10.5)$$

Obviously, the lexicographically largest element of  $P$  is an optimal solution to the LP

$$\max\{x_0 : (x_0, x) \in P\}. \quad (10.6)$$

The next lemma tells us how to recognize the lexicographically largest feasible solution from the tableau (10.3).

**Lemma 10.2** *Let  $B$  be a basis of the LP (10.6) and  $N$  its complement. Consider this LP written as in (10.3). If  $b^0 \geq \mathbf{0}$  and  $a_j^0 >_{\text{lex}} \mathbf{0}$  for all  $j \in N$ , then  $b^0$  is the lexicographically largest feasible solution to (10.6).*

**Proof.** Obviously,  $b^0$  is feasible. Any other feasible solution has the form  $b^0 - \sum_{j \in N} a_j^0 x_j$  with  $x \geq \mathbf{0}$  and  $a_j^0 >_{\text{lex}} \mathbf{0}$  for all  $j \in N$ ; hence it is lexicographically smaller than  $b^0$ . ■

**Proposition 10.3** *Starting with a basic solution satisfying  $a_j^0 >_{\text{lex}} \mathbf{0}$  for all  $j \in N^0$ , the lexicographic dual simplex algorithm in a finite number of steps either shows that (10.6) has no feasible solution or finds the lexicographically largest feasible solution.*

**Proof.** The basis, non-basis, the coefficients of the tableau (10.3) in the  $t$ th step of the algorithm are denoted with an exponent  $t$ .

Suppose  $(x_0, x) = b^t - \sum_{j \in N^t} a_j^t x_j$  is a basic solution and  $a_j^t >_{\text{lex}} \mathbf{0}$  for all  $j \in N^t$ . If  $b_i^t < 0$ ,  $x_i$  is made non-basic and  $x_k$  is made basic in the  $t$ th step, then  $a_{ik}^t < 0$  and

$$b^{t+1} = b^t - \frac{b_i^t}{a_{ik}^t} a_k^t <_{\text{lex}} b^t. \quad (10.7)$$

Thus the sequence of  $b^t$ 's is lexicographically decreasing, and hence no basis can be repeated. Therefore the algorithm stops in a finite number of steps.

Now let  $k \in N^t$  be the element lexicographically maximizing  $\frac{1}{a_{ik}^t} a_k^t$ . As  $a_{ik}^t < 0$  and  $a_k^t >_{\text{lex}} \mathbf{0}$ ,

$$a_i^{t+1} = -\frac{1}{a_{ik}^t} a_k^t >_{\text{lex}} \mathbf{0}.$$

For the remaining elements  $j \in N^{t+1}$ , we have

$$a_j^{t+1} = a_j^t - \frac{a_{ij}^t}{a_{ik}^t} a_k^t.$$

If  $a_{ij}^t \geq 0$ , then  $a_{ij}^t/a_{ik}^t \leq 0$ , and by assumption  $a_j^t$  and  $a_k^t$  are lexicographically positive, hence so is  $a_j^{t+1}$ . If on the other hand  $a_{ij}^t < 0$ , then  $a_j^{t+1}$  is lexicographically positive because  $\frac{1}{a_{ik}^t} a_k^t > \frac{1}{a_{ij}^t} a_j^t$  by the choice of  $k$ .

Therefore if the algorithm finds a primal-feasible basic solution, by Lemma 10.2 it will be the lexicographically largest basic solution.  $\blacksquare$

### Gomory's cutting plane algorithm

If the LP relaxation (10.2) is unbounded, then the IP (10.1) is unbounded or infeasible. It follows from the results of Section 4.2 that there exists a bound  $d$  (whose encoding size is polynomially bounded in the size of the input) such that the IP is feasible if and only if there exists an integral point in  $P \cap \{x : x_i \leq d \text{ for all } i\}$  with  $P$  as in (10.5). Thus we may assume that the bounding inequalities are part of the input and hence the dual to the LP relaxation is always feasible.

**Algorithm 10.1 (Gomory (1958–63))** to solve (10.1).

1. **Initialization:** Input:  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ ,  $c \in \mathbb{Z}^n$ . Set  $t := 0$ ;  $P^0 := \{(x_0, x) : x_0 \in \mathbb{R}, x \in \mathbb{R}^n, x \geq \mathbf{0}, x_0 - c^T x = 0, Ax = b\}$ .
2. **Solve the LP relaxation**  $LP^0$ :  $\max\{x_0 : (x_0, x) \in P^0\}$ . Let the optimal basis be  $B^0$ , the optimal non-basis  $N^0$ , the coefficients in the tableau (10.3) be  $a_{ij}^0, b_i^0$ , and the optimal solution  $(x_0^0, x^0)$ . We may assume that  $a_j^0 >_{\text{lex}} \mathbf{0}$  for all  $j \in N^0$ .
3. **Optimality test:** If  $(x_0^t, x^t)$  is integral, it is the solution to the IP. Return it and stop.
4. **Gomory cut:** Otherwise let  $s \geq 0$  be the minimum index such that  $x_s^t$  is not an integer. Let

$$P^{t+1} := P \cap \left\{ (x_0, x) : \sum_{j \in N^t} (a_{sj}^t - \lfloor a_{sj}^t \rfloor) x_j = b_s^t - \lfloor b_s^t \rfloor + x_{n+t+1} \right\}$$

be the strengthened relaxation with a Gomory cut on the  $s$ th row.

5. Use the **lexicographic dual simplex algorithm** starting with the previous optimal tableau extended by the Gomory cut to find the lexicographically largest feasible solution of the relaxation  $LP^{t+1}$ :  $\max\{x_0 : (x_0, x) \in P^{t+1}\}$ .
6. Set  $t := t + 1$ . Go to 3.

**Theorem 10.4** *After finitely many cuts, Algorithm 10.1 finds an optimal solution or shows that the IP is infeasible.*

For the proof, we need a lemma, whose proof is left as an exercise.

**Lemma 10.5** *Let  $(x_0^t, x^t)$  be the lexicographically largest solution to  $LP^t$  and let  $s$  be the smallest index such that  $x_s^t$  is not an integer. Then the lexicographically largest feasible solution  $(x_0^{t+1}, x^{t+1})$  to the relaxation  $LP^{t+1}$  defined in Step 4. of the algorithm obtained by the lexicographic dual simplex method in Step 5. of the algorithm satisfies*

$$(x_0^{t+1}, x^{t+1}) \leq_{\text{lex}} (x_0^t, x_1^t, \dots, x_{s-1}^t, \lfloor x_s^t \rfloor, d, \dots, d).$$

**Proof.** (of Theorem 10.4)

Let  $D = d \sum_{j=1}^n \max\{c_j, 0\}$  and  $D' = d \sum_{j=1}^n \min\{c_j, 0\}$ . It follows from the previous lemma that the number of cuts is bounded by the number of vectors  $y \in \mathbb{Z}^{n+1}$  such that  $(D', 0, \dots, 0) \leq_{\text{lex}} y \leq_{\text{lex}} (D, d, \dots, d)$ . ■

## 10.2 Mixed integer programs

In **mixed integer programs**, only some of the variables are required to be integral. The feasible set is then

$$S = \{(x, y) \in \mathbb{Z}^n \times \mathbb{R}^p : Ax + Gy \leq b\}$$

and the mixed integer program (MIP) is

$$\max\{c^T x + h^T y : (x, y) \in S\}.$$

There are several ways to generate valid inequalities for MIPs. However, it is unknown how to generate all valid inequalities. We consider **Gomory mixed integer inequalities** introduced by Gomory in 1963.

**Proposition 10.6** *Let  $S = \{(x, y) \in \mathbb{Z}^n \times \mathbb{R}^p : a^T x + g^T y = b, x \geq \mathbf{0}, y \geq \mathbf{0}\}$  and let  $\bar{a}_j = a_j - \lfloor a_j \rfloor$  for  $j = 1, \dots, n$ ,  $\bar{b} = b - \lfloor b \rfloor$ . Then the inequality*

$$\sum_{\bar{a}_j \leq \bar{b}} \frac{\bar{a}_j}{\bar{b}} x_j + \sum_{\bar{a}_j > \bar{b}} \frac{1 - \bar{a}_j}{1 - \bar{b}} x_j + \sum_{g_j > 0} \frac{g_j}{\bar{b}} y_j - \sum_{g_j < 0} \frac{g_j}{1 - \bar{b}} y_j \geq 1 \quad (10.8)$$

*is valid for  $S$ .*

**Proof.** Let  $(x, y) \in S$ . Then

$$k = \sum_{\bar{a}_j \leq \bar{b}} \bar{a}_j x_j + \sum_{\bar{a}_j > \bar{b}} (\bar{a}_j - 1) x_j + \sum_{j=1}^p g_j y_j - \bar{b}$$

is an integer. If  $k \geq 0$ , then

$$\sum_{\bar{a}_j \leq \bar{b}} \frac{\bar{a}_j}{\bar{b}} x_j - \sum_{\bar{a}_j > \bar{b}} \frac{1 - \bar{a}_j}{\bar{b}} x_j + \sum_{j=1}^p \frac{g_j}{\bar{b}} y_j \geq 1; \quad (10.9)$$

if  $k \leq -1$ , then

$$- \sum_{\bar{a}_j \leq \bar{b}} \frac{\bar{a}_j}{1 - \bar{b}} x_j + \sum_{\bar{a}_j > \bar{b}} \frac{1 - \bar{a}_j}{1 - \bar{b}} x_j - \sum_{j=1}^p \frac{g_j}{1 - \bar{b}} y_j \geq 1. \quad (10.10)$$

Hence the sum of all non-negative summands in (10.9) and (10.10) is also greater than or equal to 1, which proves (10.8). ■

If all variables are integral, then (10.8) dominates (is stronger than) the Gomory cutting plane (9.5).

Gomory's cutting plane algorithm from the previous section extends naturally to MIPs. However, our proof of finite convergence (Theorem 10.4) strongly depended on the fact that the objective function was integer-valued. Indeed, if an MIP has an integer-valued objective function, Gomory's algorithm solves it with a finite number of Gomory mixed integer cuts. A finite cutting plane method for general MIPs is currently unknown.

### 10.3 Branch and bound

Another class of algorithms is branch and bound algorithms. They are based on a very simple idea: If the problem to solve is

$$z = \max\{c^T x : x \in S\} \quad (\text{IP})$$

and  $S = S_1 \cup S_2 \cup \dots \cup S_p$ , then  $z = \max\{z^j : j = 1, 2, \dots, p\}$ , where  $z^j = \max\{c^T x : x \in S_j\}$ .

Typically, the  $p$  subproblems are solved by a similar decomposition too. Then the algorithm can be described as traversing a decision tree. Thus it “branches” in every node of the tree.

The algorithm, as described above, will eventually go through all feasible solutions (that is, all elements of  $S$ ) and find some with maximum objective value. This approach is unpractical even if the set  $S$  is finite; and outright impossible, if it is infinite.

Hence the “bounding” part of the algorithm. An upper bound is established before branching each subproblem. If a lower bound  $z_l \leq z$  is known on the optimum of the whole problem (IP), and the upper bound  $z_u^j$  for some subproblem  $S_j$  is smaller than the lower bound (that is,  $z_u^j < z_l$ ), then obviously  $S_j$  does not contain an optimal solution to (IP). The algorithm will then no longer consider  $S_j$ ; this is called **pruning by bound**.

How to establish an upper bound for  $z^j$ ? An obvious option is to solve the *LP relaxation*, if the set  $S_j$  is given as the set of integral points within a polyhedron (which is our typical setting). If the LP relaxation leads to a solution  $x^j$  in  $S_j$ , then, firstly, we may update the lower bound to  $c^T x^j$ , and secondly, the problem  $S_j$  need not be branched anymore; it is **pruned by optimality**. Another way of establishing potentially good upper bounds is by means of *Lagrangian relaxation*, described in the next chapter.

There is actually a third type of pruning: **pruning by infeasibility**. Pruning by infeasibility occurs when the algorithm recognizes that  $S_j = \emptyset$ .

Now we may describe a generic branch-and-bound algorithm:

#### Algorithm 10.2 (Branch and Bound)

**Input:** a vector  $c \in \mathbb{Z}^n$  and a set  $S \subseteq \mathbb{Z}^n$ . The algorithm will keep a list  $\mathcal{L}$  of subproblems.

1. Set  $t := 0$ ,  $z_l := -\infty$ ,  $x^* := \text{void}$ ,  $\mathcal{L} := \{S\}$ .
2. If  $\mathcal{L} = \emptyset$ , stop and output  $x^*$  as optimal solution.
3. Set  $t := t + 1$ . Choose  $S^t$  from  $\mathcal{L}$ , remove it from  $\mathcal{L}$ .
4. If  $S^t = \emptyset$ , prune by infeasibility and go to 2.
5. Compute an upper bound  $z_u^t$  on  $\max\{c^T x : x \in S^t\}$ .
6. If  $z_u^t < z_l$ , prune by bound and go to 2.
7. If a feasible solution  $x^t \in S^t$  was found during the computation of the upper bound, update  $x^* := x^t$ ,  $z_l := c^T x^t$ .
8. Generate subproblems  $S_1^t, \dots, S_p^t$  with  $S^t = \bigcup_{i=1}^p S_i^t$  and set  $\mathcal{L} := \mathcal{L} \cup \{S_1^t, \dots, S_p^t\}$ . Go to 2.

**Example 10.2** Solve the IP:

$$\begin{aligned}
 \max \quad & x_1 - 3x_2 \\
 \text{s.t.} \quad & 2x_1 + 5x_2 \leq 18 & (1) \\
 & 2x_1 - 4x_2 \leq -3 & (2) \\
 & x_1 + 7x_2 \leq 21 & (3) \\
 & x_1, x_2 \geq 0 & (N) \\
 & x_1, x_2 \in \mathbb{Z}
 \end{aligned}$$

The first feasible set  $S^1 = \{x \in \mathbb{Z}^2 : x \text{ satisfies } (1), (2), (3), (N)\}$ ; let us introduce the notation  $S^1 = ((1), (2), (3), (N))$ . Solving the LP relaxation gives an optimal solution  $(x_1^1, x_2^1) = (0, 3/4)$ , which yields an upper bound  $z_u^1 = -9/4$ . Now we branch: we consider the two inequalities

$$x_2 \leq 0 \quad (4)$$

$$x_2 \geq 1 \quad (5)$$

and insert  $((1), (2), (3), (N), (4))$  and  $((1), (2), (3), (N), (5))$  into  $\mathcal{L}$ .

Choose  $S^2 = ((1), (2), (3), (N), (4))$ . The LP relaxation is infeasible, so we prune  $S^2$ .

Next is  $S^3 = ((1), (2), (3), (N), (5))$ . An optimal solution to the LP relaxation is  $(x_1^3, x_2^3) = (1/2, 1)$  with an upper bound  $z_u^3 = -5/2$ . For the branching, consider

$$x_1 \leq 0 \quad (6)$$

$$x_1 \geq 1 \quad (7)$$

and insert  $((1), (2), (3), (N), (5), (6))$  and  $((1), (2), (3), (N), (5), (7))$  into  $\mathcal{L}$ .

Now choose  $S^4 = ((1), (2), (3), (N), (5), (6))$ , whose LP relaxation has optimal solution  $(x_1^4, x_2^4) = (0, 1)$  and  $z_u^4 = -3$ ; this solution is integral, so we may consider it a candidate for the optimal solution and it provides the first lower bound on the optimum ( $x^* := (0, 1)$ ,  $z_l := -3$ ). We prune by optimality.

Next is  $S^5 = ((1), (2), (3), (N), (5), (7))$ . Its LP relaxation has optimal solution  $(x_1^5, x_2^5) = (1, 5/4)$  with  $z_u^5 = -11/4$ . This is potentially better than our current lower bound, so we branch:

$$x_2 \leq 1 \quad (8)$$

$$x_2 \geq 2 \quad (9)$$

For  $S^6 = ((1), (2), (3), (N), (5), (7), (8))$ , the LP relaxation is infeasible. For  $S^7 = ((1), (2), (3), (N), (5), (7), (9))$ , we get  $(x_1^7, x_2^7) = (5/2, 2)$  with upper bound  $z_u^7 = -7/2 < z_l$ , hence we prune by bound.

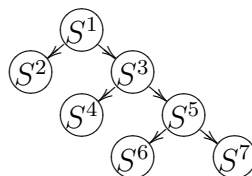


Figure 10.3: Decision tree for Example 10.2

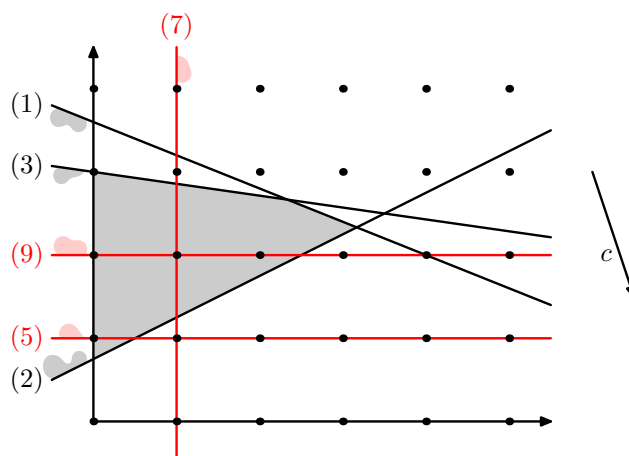


Figure 10.4: Cuts for Example 10.2

Now  $\mathcal{L} = \emptyset$  and we conclude that  $x^* = (0, 1)$  is an optimal solution to the IP.

The branching can be followed in the decision tree in Figure 10.3. The geometric meaning of some of the conditions (1)–(9) is illustrated in Figure 10.4.

The example illustrates the run of a branch-and-bound algorithm based on LP relaxations. This specifies Step **5.** of the algorithm: upper bounds are obtained by solving LP relaxations.

What this simple example does not illustrate very well are two other issues: How to branch (Step **8.**)? And: How to choose  $S^t$  from  $\mathcal{L}$  (Step **3.**)?

A common branching practice is to select a variable with a fractional value in the optimal LP solution: Let  $x_j^t$  be fractional. Then one branch will add the constraint  $x_j \leq \lfloor x_j^t \rfloor$  and a second branch will add  $x_j \geq \lceil x_j^t \rceil$ . If several variables are fractional, we must choose one of them; sometimes the “most fractional” variable is selected (that is, the variable with maximum distance from an integer value).

Choosing the right subproblem  $S^t$  can be tricky. We might hope to find a good lower bound early enough, so we can prune by bound often; then a depth-first search through the decision tree would be appropriate. On the other hand, we may try to evaluate as few subproblems as possible by always selecting the subproblem with the best upper bound coming from its predecessor. With this rule we actually never evaluate a subproblem whose upper bound is worse than the optimum  $z$  of the IP. In practice, some compromise is usually applied: perhaps one starts with a depth-first search to quickly find a feasible solution, and then switches to the best upper bound strategy.

Branch-and-bound algorithms have proved very efficient in practical applications and all commercial solvers contain such an algorithm. Most of them also apply some preprocessing, several LP algorithms, priorities for selecting the branching variable, heuristics, etc.

An interesting family of branch-and-bound algorithms is **branch-and-cut** algorithms. In this case, fractional cuts are added when solving the LP relaxations. Good cuts can significantly improve the quality of upper bounds on the subproblems, which in turn leads to a smaller number of iterations. The trade-off is however, that the algorithm might spend too much time looking for a good cut.

## 11 Lagrangian duality

Our exposition follows [13] and [9].

### 11.1 Lagrangian relaxation and Lagrangian dual

We want to solve the problem

$$z = \max\{c^T x : x \geq \mathbf{0}, Ax \leq b, Dx \leq d, x \in \mathbb{Z}^n\}. \quad (\text{IP})$$

Sometimes it may be the case that some constraints are “easier” to handle (in our case these are the  $Ax \leq b$ ) and some are “complicating” ( $Dx \leq d$ ). Then we may like to get rid of the  $Dx \leq d$  and instead make them part of the objective function:

$$z(u) = \max\{c^T x + u^T(d - Dx) : x \geq \mathbf{0}, Ax \leq b, x \in \mathbb{Z}^n\}, \quad (\text{LR}(u))$$

for  $u \in \mathbb{R}^m$ ,  $u \geq \mathbf{0}$ , where  $m$  is the number of rows of  $D$ .

Each problem  $(\text{LR}(u))$  is called a **Lagrangian relaxation** of (IP); the coefficients  $u_i$  are called **Lagrange multipliers**. Indeed,  $(\text{LR}(u))$  is a relaxation of (IP):

**Proposition 11.1** *Let  $u \in \mathbb{R}^m$ ,  $u \geq \mathbf{0}$ . Then the feasible region of (IP) is a subset of the feasible region of  $(\text{LR}(u))$ . Moreover, if  $z$  is the optimal value of (IP) and  $z(u)$  is the optimal value of  $(\text{LR}(u))$ , then  $z(u) \geq z$ .*

**Proof.** Inclusion of feasible regions is obvious. If  $x$  is a feasible solution of (IP), then  $c^T x + u^T(d - Dx) \geq c^T x$ ; this implies the second claim. ■

In this way, for every  $u$  the Lagrangian relaxation  $(\text{LR}(u))$  provides an upper bound on  $z$ . To find the best Lagrangian upper bound, we would like to compute

$$z_{\text{LD}} = \min\{z(u) : u \geq \mathbf{0}, u \in \mathbb{R}^m\}. \quad (\text{LD})$$

Problem (LD) is called the **Lagrangian dual**.

Now we are left with three issues:

- (1) How do we compute  $z(u)$ ?
- (2) How do we compute  $z_{\text{LD}}$ ?
- (3) How good is the upper bound provided by  $z_{\text{LD}}$ ?

### 11.2 How to compute $z(u)$ ?

**Example 11.1** The uncapacitated facility location problem of Assignment 1 leads to the IP

$$\min \quad \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n o_i y_i \quad (11.1a)$$

$$\text{s.t.} \quad \sum_{i=1}^n x_{ij} = d_j, \quad \forall j = 1, \dots, m \quad (11.1b)$$

$$x_{ij} \leq y_i d_j, \quad \forall i = 1, \dots, n, \forall j = 1, \dots, m \quad (11.1c)$$

$$x_{ij} \geq 0, \quad \forall i = 1, \dots, n, \forall j = 1, \dots, m \quad (11.1d)$$

$$y_j \in \{0, 1\}. \quad \forall j = 1, \dots, m \quad (11.1e)$$

We dualize the demand constraints (11.1b):

$$z(u) = \min \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n o_i y_i + \sum_{j=1}^m u_j \left( d_j - \sum_{i=1}^n x_{ij} \right) \quad (11.2a)$$

$$\text{s.t.} \quad x_{ij} \leq y_i d_j, \quad (11.2b)$$

$$x_{ij} \geq 0, \quad (11.2c)$$

$$y_j \in \{0, 1\}, \quad (11.2d)$$

which in turn yields

$$z(u) = u^T d + \sum_{i=1}^n z_i(u),$$

where

$$z_i(u) = \min \left\{ \sum_{j=1}^m (c_{ij} - u_j) x_{ij} + o_i y_i : x_{ij} \leq y_i d_j, x_{ij} \geq 0, y_i \in \{0, 1\} \text{ for all } i, j \right\}.$$

Now it is easy to compute  $z_i(u)$ . If  $y_i = 0$ , then all  $x_{ij} = 0$  and the objective value is 0. If on the other hand  $y_i = 1$ , then to minimize the sum, set  $x_{ij} = d_j$  if  $c_{ij} < u_j$  and set  $x_{ij} = 0$  if  $c_{ij} \geq u_j$ . Hence  $z_i(u) = \min \{0, o_i + \sum_{j=1}^m \min\{0, (c_{ij} - u_j) d_j\}\}$ .

**Example 11.2** Recall the symmetric traveling salesman problem from Chapter 1. It led to the IP

$$\min \sum_{ij \in E} w_{ij} x_{ij} \quad (11.3a)$$

$$\text{s.t.} \quad \sum_{ij \in E} x_{ij} = 2, \quad \forall i \in V \quad (11.3b)$$

$$\sum_{\substack{ij \in E \\ i, j \in S}} x_{ij} \leq |S| - 1, \quad \forall S \subset V : 2 \leq |S| \leq |V| - 1 \quad (11.3c)$$

$$x_{ij} \in \{0, 1\}. \quad \forall ij \in E \quad (11.3d)$$

In fact, as

$$|S| - \sum_{\substack{ij \in E \\ i, j \in S}} x_{ij} = \frac{1}{2} \sum_{i \in S} \sum_{ij \in E} x_{ij} - \sum_{\substack{ij \in E \\ i, j \in S}} x_{ij} = \frac{1}{2} \sum_{\substack{ij \in E \\ i \in S, j \notin S}} x_{ij},$$

we have

$$|S| - \sum_{\substack{ij \in E \\ i, j \in S}} x_{ij} = |V \setminus S| - \sum_{\substack{ij \in V \setminus S \\ ij \in E}} x_{ij},$$

and so half of the constraints (11.3c) are redundant. Thus we may replace (11.3c) with

$$\sum_{\substack{ij \in E \\ i, j \in S}} x_{ij} \leq |S| - 1, \quad \forall S \subset V : 2 \leq |S| \leq |V| - 1, 1 \notin S. \quad (11.3e)$$

Now we dualize the degree constraints (11.3b), leaving the degree constraint on the vertex 0 in place:

$$z(u) = \min \sum_{ij \in E} (w_{ij} - u_i - u_j)x_{ij} + 2 \sum_{i \in V} u_i \quad (11.4a)$$

$$\text{s.t.} \quad \sum_{1j \in E} x_{1j} = 2, \quad (11.4b)$$

$$\sum_{\substack{ij \in E \\ i,j \in S}} x_{ij} \leq |S| - 1, \quad \forall S \subset V : 2 \leq |S| \leq |V| - 1, 1 \notin S \quad (11.4c)$$

$$\sum_{ij \in E} x_{ij} = n, \quad (11.4d)$$

$$x_{ij} \in \{0, 1\}. \quad \forall ij \in E \quad (11.4e)$$

A feasible solution to (11.4), interpreted as a set of edges or a subgraph  $T$  of  $G$ , satisfies: 1. The degree of vertex 1 in  $T$  is 2. 2. The subgraph of  $T$  induced by  $\{2, 3, \dots, n\}$  is a tree. Such a subgraph  $T$  of  $G$  is called a **1-tree**. Finding a minimum-weight 1-tree (with weight  $w_{ij} - u_i - u_j$  on edge  $ij$ ) is easy. Solving the Lagrangian dual  $z_{\text{LD}} = \max\{z(u) : u \in \mathbb{R}^n\}$  is less easy; it was first approached by Held & Karp (1970–71), who were able to solve TSP instances too large for all other approaches of the time.

### 11.3 How good is the upper bound?

Sometimes it is the best possible:

**Theorem 11.2** *If  $u \geq \mathbf{0}$ ,  $x(u)$  is an optimal solution to (LR( $u$ )),  $Dx(u) \leq d$  and  $u^T \cdot (d - Dx(u)) = 0$ , then  $x(u)$  is an optimal solution to (IP).*

**Proof.** If  $z$  is the optimal value of (IP), then  $c^T x(u) \leq z$ , since by assumption  $x(u)$  is feasible for (IP). But

$$c^T x(u) = c^T x(u) + u^T (d - Dx(u)) = z(u) \geq z_{\text{LD}} \geq z.$$

Therefore  $z = z_{\text{LD}} = z(u)$ . ■

In general, though, we have:

**Theorem 11.3** *Let  $Q = \{x \in \mathbb{Z}^n : x \geq \mathbf{0}, Ax \leq b\}$ . Then  $z_{\text{LD}} = \max\{c^T x : Dx \leq d, x \in \text{conv } Q\}$ .*

**Proof.**

By definition,

$$z(u) = \max\{c^T x + u^T (d - Dx) : x \in Q\} = \max\{c^T x + u^T (d - Dx) : x \in \text{conv } Q\}.$$

Hence if  $Q = \emptyset$ , then  $z_{\text{LD}} = -\infty = \max\{c^T x : Dx \leq d, x \in \text{conv } Q\}$ . Otherwise, by Theorem 3.16,  $\text{conv } Q = \text{conv}\{v_1, \dots, v_s\} + \text{cone}\{r_1, \dots, r_t\}$ . In other words,  $\text{conv } Q$  has vertices  $v_1, \dots, v_s$  and extremal rays  $r_1, \dots, r_t$ .

If  $(c^T - u^T D)r_j > 0$  for some  $j$ , then  $\max\{c^T x + u^T(d - Dx) : x \in \text{conv } Q\} = +\infty$ . Otherwise  $\max\{c^T x + u^T(d - Dx) : x \in \text{conv } Q\} = c^T v_k + u^T(d - Dv_k)$  for some  $k$ . Thus

$$\begin{aligned} z_{\text{LD}} &= \min\left\{\max\{c^T v_k + u^T(d - Dv_k) : k = 1, \dots, s\} : u \geq \mathbf{0}, \right. \\ &\quad \left. (c^T - u^T D)r_j \leq 0 \text{ for all } j = 1, \dots, t\right\} \\ &= \min\left\{w : w + u^T(Dv_k - d) \geq c^T v_k \text{ for } k = 1, \dots, s; \right. \\ &\quad \left. u^T D r_j \geq c^T r_j \text{ for } j = 1, \dots, t; u \geq \mathbf{0}\right\} \end{aligned} \quad (11.5)$$

$$\begin{aligned} &= \max\left\{c^T \left(\sum_{k=1}^s y_k v_k + \sum_{j=1}^t z_j r_j\right) : \sum_{k=1}^s y_k = 0; D \left(\sum_{k=1}^s y_k v_k + \sum_{j=1}^t z_j r_j\right) \leq d \sum_{k=1}^s y_k; \right. \\ &\quad \left. y_k, z_j \geq 0 \text{ for all } k, j\right\} \end{aligned} \quad (11.6)$$

$$= \max\{c^T x : x \in \text{conv } Q, Dx \leq d\}.$$

■

**Note.** So  $z_{\text{LD}}$  can be computed by solving one of the dual LPs (11.5) and (11.6).

**Corollary 11.4** *If  $\{x \in \mathbb{R}^n : x \geq \mathbf{0}, Ax \leq b\}$  is integral, then  $z_{\text{LD}}$  is equal to the value of the LP relaxation of (IP).*

**Note.** In the STSP, the corollary implies that we can potentially solve the LP relaxation with exponentially many constraints by setting weights so as to maximize the weight of a minimum-weight 1-tree, without explicitly treating the many constraints.

**Corollary 11.5** *The value  $z(u)$  is finite if and only if  $u^T D r_j \geq c^T r_j$  for all  $j = 1, \dots, t$ . The vectors  $u$  satisfying these conditions form a polyhedron, over which  $z(u)$  is convex and piecewise linear.*

**Proof.** On this polyhedron,  $z(u) = \max\{c^T v_k + u^T(d - Dv_k) : k = 1, \dots, s\}$ ; the maximum of finitely many affine functions is always piecewise linear and convex. ■

## 11.4 How to compute $z_{\text{LD}}$ ?

The function  $z(u)$  is piecewise linear and convex, but in general not differentiable. As a generalization of the gradient method for differentiable convex functions, Held & Karp (1970) proposed the **subgradient method**.

For a convex function  $z : \mathbb{R}^m \rightarrow \mathbb{R}$ , a **subgradient** at  $u$  is a vector  $s \in \mathbb{R}^m$  such that  $z(v) \geq s^T(v - u)$  for all  $v \in \mathbb{R}^m$ .

**Lemma 11.6** *Let  $u_0 \in \mathbb{R}^m$ ; let  $x^*(u_0)$  be an optimal solution to*

$$z(u_0) = \max\{c^T x + u_0^T(d - Dx) : x \in Q\}$$

*with  $Q$  defined as in Theorem 11.3. Then  $d - Dx^*(u_0)$  is a subgradient of  $z(u)$  at  $u_0$ .*

**Algorithm 11.1 (Subgradient method for the Lagrangian dual)****Given** a starting point  $u$  and step lengths  $\{\mu_t : t = 0, 1, \dots\}$ .

1. Set  $u^0 := u, t := 0$ .
2. Find an optimal solution  $x^*(u^t)$  to  $\max\{c^T x + u_0^T(d - Dx) : x \in Q\}$ .
3. Set  $u^{t+1} := \max\{u^t - \mu_t(d - Dx^*(u^t)), 0\}$ .
4. If stopping criterion fulfilled, stop. Otherwise set  $t := t + 1$  and go to **2**.

There are still several degrees of freedom. How should the starting point be selected? (One might go with  $u = \mathbf{0}$ .) What are good step lengths? What is a good stopping criterion?

As for the step lengths, it may be proved that if  $\mu_k \rightarrow \infty$  and  $\sum_{i=0}^k \mu_t \rightarrow \infty$  as  $k \rightarrow \infty$ , then  $z(u^t) \rightarrow z_{\text{LD}}$  as  $t \rightarrow \infty$ . In practice, the convergence may be too slow. Similarly  $z(u^t) \rightarrow z_{\text{LD}}$  as  $t \rightarrow \infty$  if  $\mu_t = \mu_0 \lambda^t$  for some  $\lambda < 1$ , provided  $\mu_0$  and  $\lambda$  are sufficiently large. This may lead to faster convergence, but sometimes the geometric series tends to zero too rapidly and  $u^t$  converges before reaching an optimal point.

The method is often terminated before the solution  $z_{\text{LD}}$  is reached. The result may be used as an upper bound in some branch-and-bound algorithm; as we saw in the STSP case, it might be easier to compute than directly solving the LP relaxation.

**Note.** In fact, if  $(\text{LR}(u))$  can be solved in polynomial time, then so can be (LD).

## Earlier Mathematicians Contributed to LP and ILP

Mathematician	Birth	Doctoral Advisor(s)	Students	Friends
Joseph Fourier	1768	Lagrange	Dirichlet	
Charles Hermite	1822	Catalan	Poincaré	
David Hilbert	1862	Lindemann	Kneser Takagi, Weyl	Minkowski
(Albert Einstein)	1879	Kleiner		Marić (wife)
Hermann Weyl	1885 (~1955)	Hilbert (Göttingen)		Einstein Schrödinger
Hermann Minkowski	1864 (~1909)	Lindemann (1885)	Carathéodory Einstein in class	Hilbert
John von Neumann	1903	Fejér	Halmos	

## 12 References

For the theory of integer programming, excellent sources are [11, 3, 9, 13]. Our lecture uses many of the materials given in these books.

Combinatorial optimization problems yields important models of integer programming. Some of the excellent books on combinatorial optimization are [6, 7, 12]. There are a few classical books, e.g. [8, 10] that are still very useful.

There is an open source software for mixed integer programming [2], while CPLEX is a very efficient commercial MIP solver. Another useful software related to integer programming is 4TI2 [1] which can be used to, for example, Hilbert bases of polyhedral cones.

## References

- [1] 4ti2 team. 4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces. <http://www.4ti2.de>.
- [2] M. Berkelaar and J. Dirks. LP\_SOLVE, a mixed integer linear programming (MILP) solver. <http://lpsolve.sourceforge.net/>.
- [3] D. Bertsimas and R. Weismantel. *Optimization over integers*. Dynamic Ideas, Belmont, 2005.
- [4] J. A. Bondy and U. S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, 2008.
- [5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [6] W. Cook, W. Cunningham, W. Pullyblank, and A. Schrijver. *Combinatorial optimization*. Series in Discrete Mathematics and Optimization. John Wiley & Sons, 1998.
- [7] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, Berlin, 1988.
- [8] E. L. Lawler. *Combinatorial optimization: networks and matroids*. Holt, Rinehart and Winston, New York, 1976.
- [9] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- [10] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Printice-Hall, 1982.
- [11] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, New York, 1986.
- [12] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A, B, C*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003.
- [13] L. Wolsey. *Integer programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1998.