Discrete Optimization

lecture notes, summer term 2025

Yann Disser

June 30, 2025

Contents

These lecture notes are based on the lecture notes of the german lecture “Diskrete Optimierung”, held by Marc Pfetsch at TU Darmstadt.

1 Introduction

This lecture is concerned with algorithmic problems of the following variety.

Definition 1.1. A *mixed-integer (linear) program (MIP or MILP)* is an optimization problem of the form

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \leq \boldsymbol{b}, \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{aligned}$$

with instances given by $\boldsymbol{c} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $n, m \in \mathbb{N}$ and $p \in \{1, \dots, n\}$.

Some special cases of a mixed-integer program are:

- linear program (LP): $p = 0$ (see *Introduction to Optimization*),
- integer (linear) program (IP or ILP): $p = n$,
- binary (linear) program (BP or BLP): $\boldsymbol{x} \in \{0,1\}^n$.

As we will see, MIPs encompass all problems with linear objective function and a finite set of feasible solutions, such as the minimum spanning tree, the shortest path, the matching, and even the 3SAT problem (see *Algorithmic Discrete Mathematics*). We formalize this class of problems.

Definition 1.2. A *(linear) combinatorial optimization* problem is a problem of the form

$$\max_{F \in \mathcal{F}} \sum_{e \in F} c(e),$$

with instances given by an *objective function* $c(F) := \sum_{e \in F} c(e)$ with $c \colon E \to \mathbb{R}$ over a finite *ground set* E and a set $\mathcal{F} \subseteq 2^E$ of *feasible solutions*.

In particular, binary programs have a finite set of feasible solutions.

Observation 1.3. Every binary program can be formulated as a combinatorial optimization problem by setting $E := \{1, \dots, n\}$ and $c(i) := c_i$ for all $i \in \{1, \dots, n\}$, and $\mathcal{F} := \{F \subseteq E : \sum_{j \in F} \boldsymbol{A}_{\cdot j} \leq \boldsymbol{b}\}$.

The following proof gives a preview of the typical reasoning we will employ in the lecture. It relies on a fundamental insight that will be shown in Chapter 2.

Proposition 1.4. Linear combinatorial optimization problems can be formulated as LPs.

Proof. Let a combinatorial optimization problem be given by $E = \{e_1, \dots, e_n\}$, $\mathcal{F} \subseteq 2^E$, and $c\colon E \to \mathbb{R}$. We define the characteristic vector $\boldsymbol{\chi}^F \in \{0,1\}^E$ of a set $F \subseteq E$ by

$$\chi_e^F := \begin{cases} 1, & \text{if } e \in F, \\ 0, & \text{otherwise.} \end{cases}$$

and we set $\mathcal{X} := \{\boldsymbol{\chi}^F \in \{0,1\}^E : F \in \mathcal{F}\}$ and $\bar{\boldsymbol{c}} \in \mathbb{R}^n$ with $\bar{c}_i = c(e_i)$. Then,

$$\begin{aligned} \max_{F \in \mathcal{F}} c(F) &= \max\{\bar{\boldsymbol{c}}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{X}\} \\ &\leq \max\{\bar{\boldsymbol{c}}^\top \boldsymbol{x} : \boldsymbol{x} \in \operatorname{conv}(\mathcal{X})\}. \end{aligned} \tag{1.1}$$

We will prove later that $\mathcal{X}$ being finite implies that $\operatorname{conv}(\mathcal{X})$ is a polytope (Corollary 2.9), which means that we can find $m \in \mathbb{N}$, $A \in \mathbb{R}^{m \times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$ such that

$$\operatorname{conv}(\mathcal{X}) = \mathcal{P}(A, \boldsymbol{b}) = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}. \tag{1.2}$$

We already know that, since $\bar{\boldsymbol{c}}^\top \boldsymbol{x}$ is linear, the maximum in (1.1) is attained in a vertex of this polytope, and its vertices are contained in $\mathcal{X}$, since they are extreme points and cannot be written as a convex combination of other points in $\mathcal{X} \subseteq \operatorname{conv}(\mathcal{X})$ (see *Introduction to Optimization*). Hence (1.1) holds with equality. With (1.2), we obtain that

$$\max_{F \in \mathcal{F}} c(F) = \max\{\bar{\boldsymbol{c}}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \mathbb{R}^n\}$$

can indeed be formulated as a linear program. □

Proposition 1.4 implies in particular that 3Sat (and thus every problem in NP) can be expressed as an LP. With Observation 1.3 it further follows that binary programs can be reduced to LPs. In Chapter 4, we will see that, under suitable assumptions, the same is true for MIPs in general.

On the other hand, we know that LPs can be solved in polynomial time with the ellipsoid method (see *Introduction to Optimization*). How does this fit together?

This apparent contradiction is resolved by the fact that, while it must exist, the LP representation of a combinatorial optimization problem may be difficult to find and it may be large. In particular, already the set of feasible solutions $\mathcal{F}$ of a combinatorial optimization problem is often given implicitly and is often large (such as the one of Observation 1.3). Many of the solution methods we will see are based on finding or approximating the LP representation of a MIP.

1.1 Examples

We dealt with many integer and combinatorial optimization problems in the past (see *Algorithmic Discrete Mathematics* and *Introduction to Optimization*). Let us briefly revisit some important examples.

Example 1.5 (assignment problem)**.** In a company, there are n employees and n jobs that need to be carried out. Each person can perform exactly one job. The cost incurred by the company when person i carries out job j is c_{ij}. How should the company deploy its employees as cost-efficiently as possible? In other words, we aim to find a perfect matching of minimum cost between employees and jobs (see *Algorithmic Discrete Mathematics*).

We use the variables

$$x_{ij} = \begin{cases} 1, & \text{if person } i \text{ is assigned to job } j, \\ 0, & \text{otherwise.} \end{cases}$$

This results in the following binary program:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} & \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1, & \text{for } i \in \{1,\dots,n\}, \\ & \sum_{i=1}^{n} x_{ij} = 1, & \text{for } j \in \{1,\dots,n\}, \\ & x_{ij} \in \{0,1\}, & \text{for } i,j \in \{1,\dots,n\}. \end{aligned}$$

The problem can be cast as a combinatorial optimization problem by setting $E := \{1,\dots,n\}^2$, $c((i,j)) := c_{ij}$, and

$$\begin{aligned} \mathcal{F} :=& \{F \subseteq E : |\{(i,j) \in F : j \in \{1,\dots,n\}\}| = 1 \ \forall\, i \in \{1,\dots,n\}\} \\ & \cap \{F \subseteq E : |\{(i,j) \in F : i \in \{1,\dots,n\}\}| = 1 \ \forall\, j \in \{1,\dots,n\}\} \end{aligned}$$ △

Example 1.6 (knapsack problem)**.** A thief wants to pack stolen goods of as large a value as possible in their knapsack. Each item $i \in \{1,\dots,n\}$ has weight $a_i > 0$ and value $c_i > 0$. The knapsack has capacity $\beta > 0$. Which items should the thief pack?

We use the variables

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

This results in the following binary program:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i\, x_i & \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i\, x_i \le \beta, & \\ & \boldsymbol{x} \in \{0,1\}^n. & \end{aligned}$$

The problem can be cast as a combinatorial optimization problem by setting $E := \{1,\dots,n\}$, $c(j) := c_j$, and

$$\mathcal{F} := \{F \subseteq E : \sum_{i \in F} a_i \le \beta\}.$$ △

Example 1.7 (set-packing/partitioning/covering problems)**.** We are given a finite ground set $U = \{1,\dots,m\}$ and a family $\{S_j\}_{j\in\{1,\dots,n\}}$ of subsets of U. Each subset S_j is associated with a cost/utility c_j. The problem of finding a minimum cost / maximum utility selection of subsets such that every $e \in U$ is contained in at most, exactly, resp. at least one subset is called *set-packing*, *set-partitioning*, resp. *set-covering* problem. Examples are placing wind turbines without overlap in their areas of effect (set-packing), dividing a country into electoral districts (set-partitioning), and positioning schools in a city (set-covering).

We use the variables

$$x_j = \begin{cases} 1, & \text{if } S_j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

We also define the binary matrix $A \in \{0,1\}^{m \times n}$, with $A_{ij} = 1$ if and only if $i \in S_j$. This yields the following binary programs:

$$\begin{aligned} \min / \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \left\{ \begin{array}{c} \leq \\ = \\ \geq \end{array} \right\} \mathbf{1}, \\ & \boldsymbol{x} \in \{0,1\}^n. \end{aligned}$$

The problem can be cast as a combinatorial optimization problem by setting $E := \{1, \ldots, n\}$, $c(j) := c_j$, and

$$\mathcal{F} = \{F \subseteq E : |\{j \in F : i \in S_j\}| \left\{ \begin{array}{c} \leq \\ = \\ \geq \end{array} \right\} 1 \; \forall i \in U\}. \qquad \triangle$$

Remark 1.8. The problem of Example 1.5 is polynomially time solvable, while the problems in Examples 1.6 and 1.7 are NP-hard (see *Algorithmic Discrete Mathematics*).

1.2 Outline

The main topic of this lecture are techniques for solving mixed-integer programs. We begin (Chapter 2) by extending our structural understanding of polyhedra from the lecture *Introduction to Optimization*. In particular, we will see that a set $P \subseteq \mathbb{R}^n$ is a polyhedron if and only if it can be written as the sum of a polytope and a polyhedral cone.

Subsequently, we present a general "Branch-and-Bound" approach, which reduces solving a MIP to repeatedly solving LPs (Chapter 3). The running time of this procedure is exponential in the worst case, and we will show that it is generally NP-complete just to decide decide whether a linear program has an integer solution. In contrast, we already know (see *Introduction to Optimization*) that *fractional* vertex solutions can be found in polynomial time. We can therefore efficiently solve integer programs when all vertex solutions of the LP relaxation are integral. We will show properties under which integer programs have this structure (Chapter 4).

We will then look at how we can use additional inequalities to exclude fractional solutions of the LP relaxation (Chapter 5), and how we can separately handle problematic inequalities/variables (Chapter 6).

Finally, we deal with alternative solution methods, such as heuristics (Chapter 7) and approximation algorithms (Chapter 8).

2 Basic Polyhedral Theory

We have already learned some simple facts about polyhedra (see *Introduction to Optimization*). In this chapter, we extend this understanding by techniques that we are going to need later.

Some of the arguments of his chapter hold for vector spaces over the field $\mathbb{Q}$ in addition to over $\mathbb{R}$. To avoid repeating arguments, we work over $\mathbb{K} \in \{\mathbb{Q}, \mathbb{R}\}$ for the time being. In particular, for $\mathbb{K} = \mathbb{Q}$, we consider polyhedra as subsets of $\mathbb{Q}^n$ for $n \in \mathbb{N}$.

2.1 Orthogonal projections of polyhedra

Projections are an important tool for establishing structural statements about polyhedra. We now discuss a a method to compute orthogonal projections of a polyhedron, which amounts to the elimination of variables.

Definition 2.1. The *orthogonal projection* of a set $\mathcal{S} \subseteq \mathbb{K}^n$ with respect to its k-th coordinate, $k \in \{1, \dots, n\}$ is

$$\mathrm{Proj}_k(\mathcal{S}) := \{\boldsymbol{x} \in \mathbb{K}^n : x_k = 0, \exists\, \lambda \in \mathbb{K} \text{ with } \boldsymbol{x} + \lambda \mathbf{e}^k \in \mathcal{S}\}.$$

We are particularly interested in the case where $\mathcal{S} = \mathcal{P}(A, \boldsymbol{b}) := \{\boldsymbol{x} \in \mathbb{K}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$ is a non-empty polyhedron with $A \in \mathbb{K}^{m \times n}$, $\boldsymbol{b} \in \mathbb{K}^m$ (see Figure 2.1). To construct the orthogonal projection with respect to the k-th coordinate of $\mathcal{P}(A, \boldsymbol{b})$, we divide the indices of the inequalities in $A\boldsymbol{x} \leq \boldsymbol{b}$ into the following three sets:

- $C_k^- := \{i \in \{1, \dots, m\} : A_{ik} < 0\}$ (negative coefficients),
- $C_k^0 := \{i \in \{1, \dots, m\} : A_{ik} = 0\}$ (zero coefficients),
- $C_k^+ := \{i \in \{1, \dots, m\} : A_{ik} > 0\}$ (positive coefficients).

Obviously, the inequalities $\boldsymbol{A_i}.\boldsymbol{x} \leq b_i$ for $i \in C_k^0$, do not contain the variable x_k and are therefore inherited by the projection.

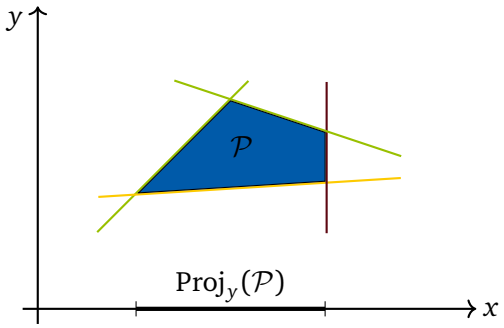

Figure 2.1: Example of an orthogonal projection with respect to the second coordinate. The colors of the hyperplanes correspond to the sets C_y^- (yellow), C_y^0 (red), and C_y^+ (green).

For $s \in C_k^-$ (i.e., $A_{sk} < 0$) we can isolate x_k in $\boldsymbol{A_s}.\boldsymbol{x} \leq b_s$ to obtain

$$x_k \geq \frac{b_s}{A_{sk}} - \sum_{\substack{j=1 \\ j \neq k}}^{n} \frac{A_{sj}}{A_{sk}} x_j. \tag{2.1}$$

Similarly, for $t \in C_k^+$ we obtain

$$x_k \leq \frac{b_t}{A_{tk}} - \sum_{\substack{j=1 \\ j \neq k}}^{n} \frac{A_{tj}}{A_{tk}} x_j. \tag{2.2}$$

Combining (2.1) and (2.2) yields

$$\frac{b_s}{A_{sk}} - \sum_{\substack{j=1 \\ j \neq k}}^{n} \frac{A_{sj}}{A_{sk}} x_j \leq \frac{b_t}{A_{tk}} - \sum_{\substack{j=1 \\ j \neq k}}^{n} \frac{A_{tj}}{A_{tk}} x_j \quad \overset{A_{sk}<0}{\Leftrightarrow} \quad \sum_{\substack{j=1 \\ j \neq k}}^{n} \big(A_{tk} A_{sj} - A_{sk} A_{tj}\big) x_j \leq A_{tk}\, b_s - A_{sk}\, b_t. \tag{2.3}$$

Observe that we obtained a new inequality which no longer contains the variable x_k. In particular, the k-th column of the resulting coefficient matrix is $\mathbf{0}$.

Lemma 2.2. The vector $\tilde{\boldsymbol{x}} \in \mathbb{K}^n$ lies in $\mathrm{Proj}_k(\mathcal{P}(A, \boldsymbol{b}))$ if and only if $\tilde{x}_k = 0$ and $\tilde{\boldsymbol{x}}$ satisfies the inequalities (2.3) for all $(s,t) \in C_k^- \times C_k^+$ and the inequalities $\boldsymbol{A_i}.\boldsymbol{x} \leq b_i$ for all $i \in C_k^0$.

Proof. Let $\tilde{\boldsymbol{x}} \in \mathrm{Proj}_k(\mathcal{P}(A, \boldsymbol{b}))$ and thus, in particular, $\tilde{x}_k = 0$. By assumption, there exists $\lambda \in \mathbb{K}$ such that $\boldsymbol{x} := \tilde{\boldsymbol{x}} + \lambda \mathbf{e^k} \in \mathcal{P}(A, \boldsymbol{b})$, i.e., $A\boldsymbol{x} \leq \boldsymbol{b}$. From the derivation above it follows that $\boldsymbol{x}$ then also fulfills the inequalities (2.3). Since these do not contain x_k, they are also fulfilled by $\tilde{\boldsymbol{x}}$. Of course, this also applies to the inequalities belonging to $i \in C_k^0$.

Now assume that $\tilde{x}_k = 0$ and $\tilde{\boldsymbol{x}}$ satisfies the inequalities (2.3) for all $(s,t) \in C_k^- \times C_k^+$ and the inequalities $\boldsymbol{A_i}.\boldsymbol{x} \leq b_i$ for all $i \in C_k^0$. Then, in particular, the maximum of the right-hand sides of (2.1) over all $s \in C_k^-$ cannot be greater than the minimum of the right-hand sides of (2.2) over all $t \in C_k^+$. We can therefore set $\lambda = x_k$ to an arbitrary value between these two bounds to satisfy all inequalities (2.1) and (2.2). The resulting vector $\boldsymbol{x} = \tilde{\boldsymbol{x}} + \lambda \mathbf{e^k}$ then lies in $\mathcal{P}(A, \boldsymbol{b})$. Hence, $\tilde{\boldsymbol{x}} \in \mathrm{Proj}_k(\mathcal{P}(A, \boldsymbol{b}))$. □

We can turn Lemma 2.2 into the following algorithm, the so-called *Fourier-Motzkin elimination*.

Algorithm: Fourier-Motzkin elimination

input: polyhedron $\mathcal{P}(A, \boldsymbol{b})$, $A \in \mathbb{K}^{m \times n}$, $\boldsymbol{b} \in \mathbb{K}^m$, index $k \in \{1, \ldots, n\}$
output: $\mathcal{P}(D, d)$ such that $\mathrm{Proj}_k(\mathcal{P}(A, \boldsymbol{b})) = \{\boldsymbol{x} \in \mathbb{K}^n : D\boldsymbol{x} \leq \boldsymbol{d},\ x_k = 0\}$

$C_k^- := \{i \in \{1, \ldots, m\} : A_{ik} < 0\}$, $C_k^0 := \{i \in \{1, \ldots, m\} : A_{ik} = 0\}$, $C_k^+ := \{i \in \{1, \ldots, m\} : A_{ik} > 0\}$
$i \leftarrow 1$
for $z \in C_k^0$ **:**
  $\boldsymbol{D_i}. \leftarrow \boldsymbol{A_z}.$; $d_i \leftarrow b_z$; $i \leftarrow i + 1$
for $(s,t) \in C_k^- \times C_k^+$ **:**
  $\boldsymbol{D_i}. \leftarrow A_{tk} \boldsymbol{A_s}. - A_{sk} \boldsymbol{A_t}.$ (it follows that $D_{ik} = 0$)
  $d_i \leftarrow A_{tk}\, b_s - A_{sk}\, b_t$
  $i \leftarrow i + 1$
return $\mathcal{P}(D, \boldsymbol{d})$

We note the following immediate consequences.

Corollary 2.3. Let $\mathcal{P}(D, \boldsymbol{d})$ be the result of the Fourier-Motzkin elimination. We have

(a) $\text{Proj}_k(\mathcal{P}(A, \boldsymbol{b})) = \{\boldsymbol{x} \in \mathbb{K}^n : x_k = 0,\ D\boldsymbol{x} \leq \boldsymbol{d}\}$;

(b) $D\mathbf{e^k} = \mathbf{0}$;

(c) The rows of $D\boldsymbol{x} \leq \boldsymbol{d}$ are non-negative linear combinations of the rows of $A\boldsymbol{x} \leq \boldsymbol{b}$, i.e., there exists a matrix $\bar{U} \geq 0$ with $\bar{U}A = D$ and $\bar{U}\boldsymbol{b} = \boldsymbol{d}$;

(d) $\mathcal{P}(A, \boldsymbol{b}) \neq \emptyset \iff \mathcal{P}(D, \boldsymbol{d}) \neq \emptyset \iff \text{Proj}_k(\mathcal{P}(A, \boldsymbol{b})) \neq \emptyset$.

Proof. Statement (a) holds because of Lemma 2.2 and the definition the Fourier-Motzkin elimination. Statement (b) follows from the fact that the algorithm ensures $D_{ik} = A_{tk}A_{sk} - A_{sk}A_{tk} = 0$. Statement (c) holds by definition of the algorithm with $A_{tk} > 0$ and $A_{sk} < 0$. It remains to prove (d).

If $\boldsymbol{x} \in \mathcal{P}(A, \boldsymbol{b})$ exists, then by (c) it holds that

$$D\boldsymbol{x} = \bar{U}A\boldsymbol{x} \overset{\bar{U}\geq 0}{\leq} \bar{U}\boldsymbol{b} = \boldsymbol{d}.$$

Thus, $\boldsymbol{x} \in \mathcal{P}(D, \boldsymbol{d})$ and therefore $\mathcal{P}(D, \boldsymbol{d}) \neq \emptyset$.

If $\boldsymbol{x} \in \mathcal{P}(D, \boldsymbol{d})$ exists, then, because of (b), also $\tilde{\boldsymbol{x}} := \sum_{i \neq k} x_i \mathbf{e^i} \in \mathcal{P}(D, \boldsymbol{d})$. Because of (a) and $\tilde{x}_k = 0$, it further follows that $\tilde{\boldsymbol{x}} \in \text{Proj}_k(\mathcal{P}(A, \boldsymbol{b}))$ and therefore $\text{Proj}_k(\mathcal{P}(A, \boldsymbol{b})) \neq \emptyset$.

If, in turn, $\boldsymbol{x} \in \text{Proj}_k(\mathcal{P}(A, \boldsymbol{b}))$ exists, then, by definition, there is $\lambda \in \mathbb{K}$ with $\boldsymbol{x} + \lambda\mathbf{e^k} \in \mathcal{P}(A, \boldsymbol{b})$ and therefore $\mathcal{P}(A, \boldsymbol{b}) \neq \emptyset$. □

Corollary 2.3 (d) provides a certificate for the infeasibility of the system $A\boldsymbol{x} \leq \boldsymbol{b}$ via the infeasibility of the lower dimensional system $D\boldsymbol{x} \leq \boldsymbol{d}$. By iteratively projecting out all variables, we eventually obtain a zero-dimensional certificate in the form of a single point. It turns out that this point is exactly the certificate provided by the Farkas lemma (see *Introduction to Optimization*).

To see this, we start by applying Fourier-Motzkin elimination to the first variable, yielding $\mathcal{P}(D^{(1)}, \boldsymbol{d^{(1)}})$. Corollary 2.3 implies

$$\begin{aligned} &\text{(i)} && D^{(1)}\mathbf{e^1} = \mathbf{0}. \\ &\text{(ii)} && \text{There is a matrix } \bar{U}^{(1)} \geq 0 \text{ with } \bar{U}^{(1)}A = D^{(1)},\ \bar{U}^{(1)}\boldsymbol{b} = \boldsymbol{d^{(1)}}. \\ &\text{(iii)} && \mathcal{P}(A, \boldsymbol{b}) = \emptyset \iff \mathcal{P}(D^{(1)}, \boldsymbol{d^{(1)}}) = \emptyset. \end{aligned} \tag{2.4}$$

In this manner, we can iteratively eliminate the first k variables. In the k-th step, we obtain a matrix $D^{(k)}$ and $\bar{U}^{(k)} \geq 0$ with $D^{(k)} = \bar{U}^{(k)}D^{(k-1)}$. We set $U^{(k)} = \bar{U}^{(k)}\bar{U}^{(k-1)} \cdots \bar{U}^{(1)}$ such that $U^{(k)}A = D^{(k)}$ and $U^{(k)}b = d^{(k)}$, and obtain

$$\begin{aligned} &\text{(i)} && D^{(k)}\mathbf{e^j} = \mathbf{0}, \text{ for all } j \in \{1, \ldots, k\}. \\ &\text{(ii)} && \text{There is a matrix } U^{(k)} \geq 0 \text{ with } U^{(k)}A = D^{(k)} \text{ and } U^{(k)}\boldsymbol{b} = \boldsymbol{d^{(k)}}. \\ &\text{(iii)} && \mathcal{P}(A, \boldsymbol{b}) = \emptyset \Leftrightarrow \mathcal{P}(D^{(k)}, \boldsymbol{d^{(k)}}) = \emptyset. \end{aligned} \tag{2.5}$$

Now let $\mathcal{P}(A, \boldsymbol{b}) = \emptyset$. According to (2.5) (iii), $\mathcal{P}(D^{(n)}, \boldsymbol{d^{(n)}}) = \emptyset$. Since $D^{(n)} = 0$ because of (2.5) (i), this implies that an index i must exist with $d_i^{(n)} < 0$. According to (2.5) (ii), it then holds for $\boldsymbol{u} := (\boldsymbol{U_{i\cdot}^{(n)}})^\top$ that $\boldsymbol{u} \geq \mathbf{0}$,

$\boldsymbol{u}^\top A = \boldsymbol{D}_{i\cdot}^{(n)} = \boldsymbol{0}^\top$ and $\boldsymbol{u}^\top \boldsymbol{b} = d_i^n < 0$. Conversely, it immediately follows from the existence of such a $\boldsymbol{u} \geq \boldsymbol{0}$ that $\mathcal{P}(A, \boldsymbol{b}) = \emptyset$, since then

$$\boldsymbol{u}^\top A\boldsymbol{x} = 0 > d_i^n = \boldsymbol{u}^\top \boldsymbol{b} \qquad \forall \boldsymbol{x} \in \mathbb{K}^n.$$

This means that exactly one of the following two systems has a solution (see *Introduction to Optimization*):

$$\exists\, \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b} \quad \dot{\bigvee} \quad \exists\, \boldsymbol{u} \geq \boldsymbol{0} \colon \boldsymbol{u}^\top A = \boldsymbol{0},\ \boldsymbol{u}^\top \boldsymbol{b} < 0.$$

Moreover, Corollary 2.3 (a) immediately implies the following.

Corollary 2.4. The orthogonal projection $\mathrm{Proj}_k(\mathcal{P}(A, \boldsymbol{b}))$ is a polyhedron.

This allows to prove an important fact that opens a new perspective on polyhedra. We will later show that every polyhedron can be written in the following form.[1]

Theorem 2.5. For every $A \in \mathbb{K}^{m\times n}$ and $B \in \mathbb{K}^{m\times n'}$, the set $\mathcal{P} = \mathrm{conv}(A) + \mathrm{cone}(B)$ is a polyhedron.

Proof. We have

$$\begin{aligned}\mathcal{P} &= \{\boldsymbol{x} \in \mathbb{K}^m \,:\, \exists \boldsymbol{\lambda}, \boldsymbol{\mu} \geq \boldsymbol{0} \colon \boldsymbol{1}^\top \boldsymbol{\lambda} = 1, \boldsymbol{x} = A\boldsymbol{\lambda} + B\boldsymbol{\mu}\} \\ &= \{\boldsymbol{x} \in \mathbb{K}^m \,:\, \exists \boldsymbol{\lambda}, \boldsymbol{\mu} \colon D\begin{pmatrix}\boldsymbol{\lambda}\\ \boldsymbol{\mu}\\ \boldsymbol{x}\end{pmatrix} \leq \boldsymbol{d}\},\end{aligned}$$

with

$$D = \begin{pmatrix} -\mathbb{I} & 0 & 0 \\ 0 & -\mathbb{I} & 0 \\ \boldsymbol{1}^\top & \boldsymbol{0}^\top & \boldsymbol{0}^\top \\ -\boldsymbol{1}^\top & \boldsymbol{0}^\top & \boldsymbol{0}^\top \\ A & B & -\mathbb{I} \\ -A & -B & \mathbb{I} \end{pmatrix} \quad \text{and} \quad \boldsymbol{d} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ 1 \\ -1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}.$$

Via iterative orthogonal projection of the variables in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ in $D\begin{pmatrix}\boldsymbol{\lambda}\\ \boldsymbol{\mu}\\ \boldsymbol{x}\end{pmatrix} \leq \boldsymbol{d}$, by Corollary 2.4, we obtain a polyhedron $\mathrm{Proj}_{\boldsymbol{\lambda},\boldsymbol{\mu}}(\mathcal{P}(D, \boldsymbol{d}))$. This polyhedron is exactly $\mathcal{P}$ embedded in $\mathbb{K}^{n+n'+m}$, i.e.,

$$\mathcal{P} = \{\boldsymbol{x} \in \mathbb{K}^m \,:\, \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{x}\end{pmatrix} \in \mathrm{Proj}_{\boldsymbol{\lambda},\boldsymbol{\mu}}(\mathcal{P}(D, \boldsymbol{d}))\},$$

or, conversely,

$$\mathrm{Proj}_{\boldsymbol{\lambda},\boldsymbol{\mu}}(\mathcal{P}(D, \boldsymbol{d})) = \{\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{x}\end{pmatrix} \in \mathbb{K}^{n+n'+m} \,:\, \boldsymbol{x} \in \mathcal{P}\}.$$ □

Observe that, in the proof of Theorem 2.5, the only non-zero entries of $\boldsymbol{d}$ are associated with $\boldsymbol{\lambda}$, i.e., with $\mathrm{conv}(A)$. It follows that every conic hull is a polyhedral cone.

Corollary 2.6. For every $B \in \mathbb{K}^{m\times n}$, there exists a matrix $D \in \mathbb{K}^{k\times m}$ with $\mathrm{cone}(B) = \mathcal{P}(D, \boldsymbol{0})$.

The argument of Theorem 2.5 essentially uses that any affine map of a polyhedron is a polyhedron. In particular, analogous arguments yield the following corollaries (exercises).

Corollary 2.7. For every $A \in \mathbb{K}^{m\times n}$, the sets $\mathrm{lin}(A)$, $\mathrm{aff}(A)$, $\mathrm{cone}(A)$, and $\mathrm{conv}(A)$ are polyhedra.

[1] Throughout the lecture, for a matrix $A \in \mathbb{K}^{m\times n}$, we denote $\mathrm{lin}(A) := \mathrm{lin}(\{\boldsymbol{A}_{\cdot 1}, \ldots, \boldsymbol{A}_{\cdot n}\}) = \{A\boldsymbol{\lambda} \,:\, \boldsymbol{\lambda} \in \mathbb{K}^n\}$ and analogously for $\mathrm{aff}(A) = \{A\boldsymbol{\lambda} \,:\, \boldsymbol{1}^\top \boldsymbol{\lambda} = 1\}$, $\mathrm{cone}(A) = \{A\boldsymbol{\lambda} \,:\, \boldsymbol{\lambda} \geq \boldsymbol{0}\}$ and $\mathrm{conv}(A) = \{A\boldsymbol{\lambda} \,:\, \boldsymbol{1}^\top \boldsymbol{\lambda} = 1, \boldsymbol{\lambda} \geq \boldsymbol{0}\}$.

Corollary 2.8. The Minkowski sum $\mathcal{P} + \mathcal{P}'$ of two polyhedra $\mathcal{P}, \mathcal{P}' \subseteq \mathbb{K}^n$ is a polyhedron.

We can already observe that not only does conv(A) induce a polytope, in fact *every* polytope can be expressed this way. In the next section, we will complement Theorem 2.5 in same way for polyhedra.

Corollary 2.9. A set $\mathcal{P} \subseteq \mathbb{K}^n$ is a polytope if and only if there exists a finite set $\mathcal{V} \subseteq \mathbb{K}^n$ with $\mathcal{P} = \text{conv}(\mathcal{V})$.

Proof. Let $\mathcal{V} \subseteq \mathbb{K}^n$ be finite and $\mathcal{P} = \text{conv}(\mathcal{V})$. Then, $\mathcal{P}$ is a polyhedron by Corollary 2.7, and $\mathcal{P}$ is bounded by

$$\|\boldsymbol{x}\| \leq \sum_{\boldsymbol{v} \in \mathcal{V}} \|\boldsymbol{v}\| \quad \forall \boldsymbol{x} \in \mathcal{P}.$$

Hence, $\mathcal{P}$ is a polytope.

Conversely, recall that every polytope $\mathcal{P} = \mathcal{P}^{=}(A, \boldsymbol{b})$ has a finite set of extreme points (i.e., vertices), each arising from a combination (i.e., a basis) of the columns of A – and as a compact and convex set, $\mathcal{P}$ is given by the convex hull of its extreme points (see *Introduction to Optimization*). □

2.2 Representations of polyhedra

Let's take another look at the Farkas lemma from a different point of view:

$$\exists \boldsymbol{x} \geq \boldsymbol{0} \colon A\boldsymbol{x} = \boldsymbol{b} \quad \dot{\bigvee} \quad \exists \boldsymbol{y} \colon \boldsymbol{y}^\top A \leq \boldsymbol{0}^\top,\ \boldsymbol{y}^\top \boldsymbol{b} > 0,$$

or, expressed differently,

$$\exists \boldsymbol{x} \geq \boldsymbol{0} \colon A\boldsymbol{x} = \boldsymbol{b} \quad \Longleftrightarrow \quad \forall \boldsymbol{y} \colon A^\top \boldsymbol{y} \leq \boldsymbol{0} \Rightarrow \boldsymbol{y}^\top \boldsymbol{b} \leq 0. \tag{2.6}$$

This characterizes all right-hand sides $\boldsymbol{b}$ for which the system $A\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$ is feasible. By definition,

$$\text{cone}(A) = \{\boldsymbol{b} \in \mathbb{K}^m \,:\, \exists \boldsymbol{x} \geq \boldsymbol{0} \text{ with } A\boldsymbol{x} = \boldsymbol{b}\}.$$

Together with the Farkas lemma (2.6), this yields the following observation.

Proposition 2.10. For all matrices $A \in \mathbb{K}^{m \times n}$ it holds that

$$\text{cone}(A) = \{\boldsymbol{b} \in \mathbb{K}^m \,:\, \boldsymbol{y}^\top \boldsymbol{b} \leq 0 \ \forall \boldsymbol{y} \in \mathcal{P}(A^\top, \boldsymbol{0})\}.$$

The geometric interpretation of Proposition 2.10 could be written as

$$\left\{\begin{matrix}\text{feasible right-hand sides } \boldsymbol{b} \\ \text{of } A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0}\end{matrix}\right\} \triangleq \left\{\begin{matrix}\text{vectors that form an obtuse } (\geq \pi/2) \text{ angle} \\ \text{with all vectors from } \mathcal{P}(A^\top, \boldsymbol{0})\end{matrix}\right\}.$$

The latter can be expressed more abstractly using the following definition.

Definition 2.11. The *polar cone* $\mathcal{S}^\circ$ of $\mathcal{S} \subseteq \mathbb{K}^n$ is the set of vectors that form an obtuse angle with all vectors in $\mathcal{S}$, i.e.,

$$\mathcal{S}^\circ := \{\boldsymbol{y} \in \mathbb{K}^n \,:\, \boldsymbol{y}^\top \boldsymbol{x} \leq 0 \ \forall \boldsymbol{x} \in \mathcal{S}\}.$$

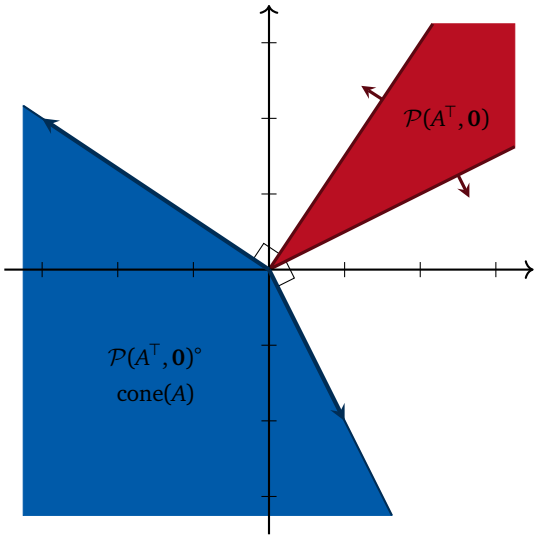

Figure 2.2: Illustration of Example 2.13 with $\mathcal{P}(A^\top, \mathbf{0})^\circ = \mathrm{cone}(\{\binom{-3}{2}, \binom{1}{-2}\})$.

With this, we can write Proposition 2.10 more concisely.[2]

Corollary 2.12. For all $A \in \mathbb{K}^{m\times n}$ it holds that $\mathrm{cone}(A) = \mathcal{P}(A^\top, \mathbf{0})^\circ$.

Our reasoning so far can be summarized as

$$A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \mathbf{0} \text{ is feasible} \overset{(2.6)}{\iff} \boldsymbol{b} \in \mathcal{P}(A^\top, \mathbf{0})^\circ \overset{\text{Cor. 2.12}}{\iff} \boldsymbol{b} \in \mathrm{cone}(A).$$

Example 2.13. For

$$A = \begin{pmatrix} -3 & 1 \\ 2 & -2 \end{pmatrix}$$

we have

$$\mathcal{P}(A^\top, \mathbf{0}) = \{\tbinom{x}{y} \in \mathbb{K}^2 : x/2 \leq y \leq 3x/2\},$$

and indeed $\mathcal{P}(A^\top, \mathbf{0})^\circ = \mathrm{cone}(A)$ (see Figure 2.2). Accordingly,

$$A\boldsymbol{x} = \tbinom{0}{1},\ \boldsymbol{x} \geq \mathbf{0} \text{ is infeasible, but } A\boldsymbol{x} = \tbinom{-1}{0},\ \boldsymbol{x} \geq \mathbf{0} \text{ is feasible.} \qquad \triangle$$

To reverse Corollary 2.12, we observe that the polar cone already is a conic hull and vice-versa.

Lemma 2.14. For $\mathcal{S} \subseteq \mathbb{K}^n$, it holds that

$$\mathcal{S}^\circ = \mathrm{cone}(\mathcal{S}^\circ) = \mathrm{cone}(\mathcal{S})^\circ.$$

Proof. exercise. □

We can now complement Corollary 2.12.

[2] For convenience, we write $\mathcal{P}(A, \boldsymbol{b})^\circ := (\mathcal{P}(A, \boldsymbol{b}))^\circ$, $\mathrm{cone}(\mathcal{S})^\circ := (\mathrm{cone}(\mathcal{S}))^\circ$, and $\mathcal{S}^{\circ\circ} := (\mathcal{S}^\circ)^\circ$.

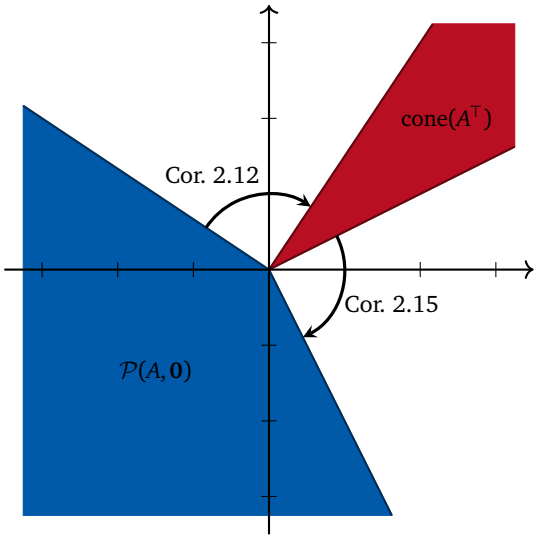

Figure 2.3: Correspondence between a polyhedral cone and a conic hull in "polar row space", and vice versa.

Corollary 2.15. For all $A \in \mathbb{K}^{m\times n}$ it holds that $\mathcal{P}(A, \mathbf{0}) = \text{cone}(A^\top)^\circ$.

Proof. We have

$$(\text{cone}(A^\top))^\circ \overset{\text{Lem. 2.14}}{=} \{(\boldsymbol{A_1}.)^\top, \ldots, (\boldsymbol{A_m}.)^\top\}^\circ \overset{\text{Def. 2.11}}{=} \{\boldsymbol{y} : \boldsymbol{y}^\top A^\top \leq \mathbf{0}^\top\} = \{\boldsymbol{x} : A\boldsymbol{x} \leq \mathbf{0}\} = \mathcal{P}(A, \mathbf{0}). \qquad \square$$

We have all ingredients in place to prove that both polyhedral cones of the form $\mathcal{P}(A, \mathbf{0})$ and conic hulls of the form cone(A) induce themselves as polar cones (see Figure 2.3).

Theorem 2.16. For every $A \in \mathbb{K}^{m\times n}$, it holds that

$$\mathcal{P}(A, \mathbf{0})^{\circ\circ} = \mathcal{P}(A, \mathbf{0}),$$
$$\text{cone}(A)^{\circ\circ} = \text{cone}(A).$$

Proof. We use the relationship between polyhedral cones and conic hulls established in Corollaries 2.12 and 2.15:

$$\mathcal{P}(A, \mathbf{0}) \overset{\text{Cor. 2.15}}{=} \text{cone}(A^\top)^\circ \overset{\text{Cor. 2.12}}{=} \mathcal{P}(A, \mathbf{0})^{\circ\circ},$$
$$\text{cone}(A) \overset{\text{Cor. 2.12}}{=} \mathcal{P}(A^\top, \mathbf{0})^\circ \overset{\text{Cor. 2.15}}{=} \text{cone}(A)^{\circ\circ}. \qquad \square$$

We now show that the objects in Theorem 2.16 are not only intimately related but actually different representations of the same sets.

Theorem 2.17 (Minkowski 1896)**.** For every matrix $A \in \mathbb{K}^{m\times n}$ there is a matrix $B \in \mathbb{K}^{n\times k}$ with

$$\mathcal{P}(A, \mathbf{0}) = \text{cone}(B),$$

and vice versa.

Proof. For a given matrix B, the corresponding matrix A exists by Corollary 2.6. For given A, again by Corollary 2.6, there exists a matrix B with (see Figure 2.3)

$$\mathcal{P}(A, \mathbf{0}) \overset{\text{Cor. 2.15}}{=} \text{cone}(A^\top)^\circ \overset{\text{Cor. 2.6}}{=} \mathcal{P}(B^\top, \mathbf{0})^\circ \overset{\text{Cor. 2.12}}{=} \text{cone}(B). \qquad \square$$

By reducing the general case to the conic case, we now prove the dual statement to Theorem 2.5, i.e., every polyhedron can be decomposed into a bounded and a conic part.

Theorem 2.18. For every $A \in \mathbb{K}^{m\times n}$, $\boldsymbol{b} \in \mathbb{K}^m$ there exist finite sets $\mathcal{V}, \mathcal{E} \subseteq \mathbb{K}^n$ with

$$\mathcal{P}(A, \boldsymbol{b}) = \text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E}).$$

Proof. We embed $\mathcal{P}(A, \boldsymbol{b})$ into a polyhedral cone

$$\mathcal{H} = \mathcal{P}\left(\begin{pmatrix} A & -\boldsymbol{b} \\ \mathbf{0}^\top & -1 \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}\right).$$

In particular, we add an additional dimension and place our polyhedron onto the plane $\left\{\binom{\boldsymbol{\lambda}}{1}\right\}_{\boldsymbol{\lambda}\in\mathbb{K}^n}$, i.e., $\boldsymbol{x} \in \mathcal{P}(A, \boldsymbol{b}) \Leftrightarrow \binom{\boldsymbol{x}}{1} \in \mathcal{H}$.

According to Theorem 2.17, there is a matrix $B \in \mathbb{K}^{(n+1)\times d}$ with $\mathcal{H} = \text{cone}(B)$. Due to the last row in the description of $\mathcal{H}$, the last row of B has only non-negative entries. By scaling and swapping the columns of B we can transform B into a matrix $\bar{B}$ with $\text{cone}(\bar{B}) = \text{cone}(B) = \mathcal{H}$, so that

$$\bar{B} = \begin{pmatrix} \bar{V} & \bar{E} \\ \mathbf{1}^\top & \mathbf{0}^\top \end{pmatrix},$$

where $\bar{V}$, $\bar{E}$ may be empty. Let $\mathcal{V}$ and $\mathcal{E}$ be the sets of the columns of the matrices $\bar{V}$ and $\bar{E}$, respectively. This means that

$$\begin{aligned} \boldsymbol{x} \in \mathcal{P}(A, \boldsymbol{b}) &\iff \binom{\boldsymbol{x}}{1} \in \mathcal{H} \\ &\iff \binom{\boldsymbol{x}}{1} \in \text{cone}(\bar{B}) \\ &\iff \exists \boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0 \colon \begin{array}{l} \boldsymbol{x} = \bar{V}\boldsymbol{\lambda} + \bar{E}\boldsymbol{\mu} \\ 1 = \mathbf{1}^\top \boldsymbol{\lambda} \end{array} \\ &\iff \boldsymbol{x} \in \text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E}). \end{aligned} \qquad \square$$

Together, Theorems 2.5 and 2.18 yield the central structure theorem of polyhedral theory.

Theorem 2.19 (representation theorem)**.** A subset $\mathcal{P} \subseteq \mathbb{K}^n$ is a polyhedron if, and only if there are finite sets $\mathcal{V}, \mathcal{E} \subseteq \mathbb{K}^n$ with

$$\mathcal{P} = \text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E}).$$

Intuitively, the convex hull in the representation of a polyhedron corresponds to its bounded part, while the conic hull corresponds to the unbounded part. Accordingly, a polytope, i.e., a bounded polyhedron reduces to a convex hull. We can phrase Theorem 2.19 by saying that every polyhedron is the sum of a polytope (Corollary 2.9) and a polyhedral cone (Theorem 2.17).

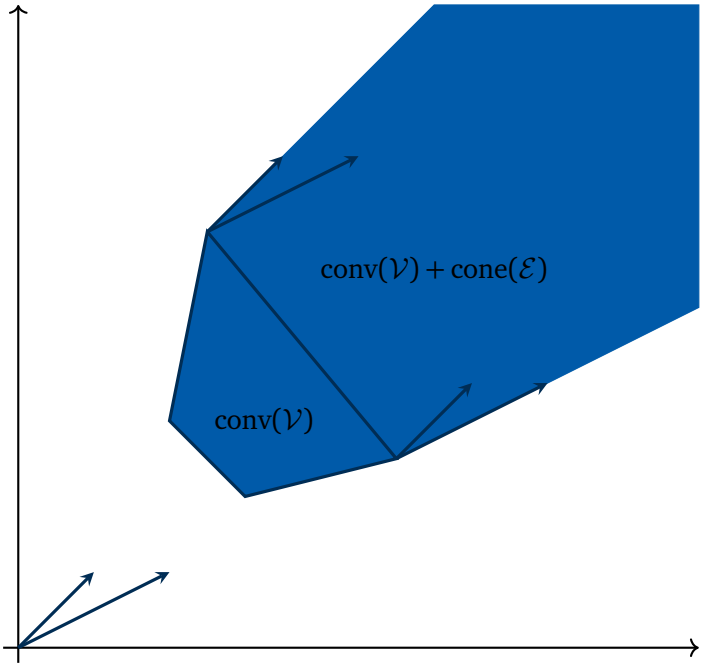

Figure 2.4: Internal structure of a polyhedron.

Remark 2.20. Note that $\text{conv}(\emptyset) = \emptyset$ ($\emptyset$ is convex) and $\text{cone}(\emptyset) = \{\mathbf{0}\}$. However, because $\emptyset + \mathcal{S} = \emptyset$ for every set $\mathcal{S}$, we have that $\mathcal{V}$ is non-empty if, and only if $\mathcal{P}$ is non-empty. Furthermore, the representation is not unique. For example, the vectors in $\mathcal{E}$ can be scaled arbitrarily, and if $\mathcal{P}$ is a linear subspace, any $\boldsymbol{v} \in \mathcal{P}$ can be chosen and any basis $\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(k)}$ (with $k = \dim(\mathcal{P})$) in $\mathcal{P} = \text{conv}(\{\boldsymbol{v}\}) + \text{cone}(\{\boldsymbol{b}^{(1)}, -\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(k)}, -\boldsymbol{b}^{(k)}\})$.

We now know two representations for polyhedra (see Figure 2.4):

(a) The *exterior representation* or *halfspace representation (H-representation)*:

$$\mathcal{P} = \mathcal{P}(A, \boldsymbol{b}) = \bigcap_{i=1}^{m} \{\boldsymbol{x} \in \mathbb{K}^n : \boldsymbol{A_i}.\boldsymbol{x} \leq b_i\},$$

i.e., $\mathcal{P}$ is regarded as the intersection of larger objects (halfspaces).

(b) The *interior representation* or *vertex representation (V-representation)*:

$$\mathcal{P} = \text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E}),$$

i.e., $\mathcal{P}$ is described by its vertices and extreme rays.

We will generally consider polyhedra as subsets of $\mathbb{R}^n$. The representation theorem still yields a correspondence between rational interior and exterior descriptions of polyhedra.

Corollary 2.21. Let $\mathcal{P} \subseteq \mathbb{R}^n$ and $\mathbb{K} = \mathbb{R}$. There exist $A \in \mathbb{Q}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Q}^m$, and $m \in \mathbb{N}$ with $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b})$ if and only if there exist finite sets $\mathcal{V}, \mathcal{E} \subseteq \mathbb{Q}^n$ with $\mathcal{P} = \text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E})$.

Proof. Theorem 2.19 for $\mathbb{K} = \mathbb{Q}$ yields the existence of A and $\boldsymbol{b}$, respectively of $\mathcal{V} = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)}\}$ and $\mathcal{E} = \{\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\ell)}\}$, such that

$$\{\boldsymbol{x} \in \mathbb{Q}^n : A\boldsymbol{x} \leq \boldsymbol{b}\} = \{\sum_{i=1}^{k} \lambda_i \boldsymbol{v}^{(i)} + \sum_{j=1}^{\ell} \mu_j \boldsymbol{r}^{(j)} : \boldsymbol{\lambda} \in \mathbb{Q}^k_{\geq 0}, \boldsymbol{\mu} \in \mathbb{Q}^\ell_{\geq 0}, \mathbf{1}^\top \boldsymbol{\lambda} = 1\}.$$

We claim that the set on the left-hand-side is dense in $\mathcal{P}(A, \boldsymbol{b}) = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$ and the set on the right-hand-side is dense in $\text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E})$. This completes the proof since the latter sets are closed in $\mathbb{R}^n$.

The right-hand-side is dense in $\text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E})$, since $\mathbb{Q}$ is dense in $\mathbb{R}$ and since, by linearity of the involved sums, every real point in $\text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E})$ can be approached by the sum of a rational convex and a rational conic combination of the (rational) points in $\mathcal{V}$ and $\mathcal{E}$. For the left-hand-side, we observe that every point $\boldsymbol{x} \in \mathcal{P}(A, \boldsymbol{b})$ lies in some polytope $\mathcal{P}' := \mathcal{P}(A, \boldsymbol{b}) \cap \{\boldsymbol{y} \in \mathbb{R}^n : \mathbb{I}\boldsymbol{y} \leq \beta\mathbf{1}, \mathbb{I}\boldsymbol{y} \geq -\beta\mathbf{1}\}$ with $\beta \in \mathbb{Q}$. Thus, $\boldsymbol{x}$ can again be approached by a rational convex combination of the extreme points of $\mathcal{P}'$. These extreme points are solutions of systems of equations with rational coefficients (see *Introduction to Optimization*) and thus lie in $\mathbb{Q}^n$. Hence, $\{\boldsymbol{x} \in \mathbb{Q}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$ is dense in $\mathcal{P}(A, \boldsymbol{b})$. □

Remark 2.22. In order to convert from the exterior to the interior representation, we can proceed as in Theorem 2.18 and obtain the matrix B as in the proofs of Theorems 2.17 and 2.5. Conversely, we can go from the interior to the exterior description directly as in Theorem 2.5. Observe that both directions use the Fourier-Motzkin elimination.

Remark 2.23. Unfortunately, when computing the vertices of a polytope with Fourier-Motzkin elimination, it can happen that in intermediate iterations, the number of generated vertices can be exponential in the dimension, even though the final result is small.

2.3 The integer hull

We now transition back to working with vector spaces over $\mathbb{R}$. In particular, we deal with the set of feasible solutions of a mixed-integer program of the form

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \leq \boldsymbol{b}, \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{aligned}$$

which is a subset of the polyhedron $\mathcal{P}(A, \boldsymbol{b})$.

Definition 2.24. The *integer hull* of a polyhedron $\mathcal{P}$ is the set $\mathcal{P}_{\mathrm{I},p} := \text{conv}(\{\boldsymbol{x} \in \mathcal{P} : x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}\})\}$ for $p \in \{0, 1, \ldots, n\}$. We write $\mathcal{P}_\mathrm{I} := \mathcal{P}_{\mathrm{I},n}$.

We are particularly interested in the fully integral case $p = n$.

Definition 2.25. A polyhedron P is *integral* if $\mathcal{P} = \mathcal{P}_\mathrm{I}$.

Intuitively, one might think that the integer hull, i.e., in particular, the convex hull of all feasible solutions of a MIP, is a polyhedron. However, this is not the case, as the following example shows.

Example 2.26. Consider the MIP

$$\sup \{x_1 - \sqrt{2}x_2 : \boldsymbol{x} \in \mathcal{P}_\mathrm{I}\}$$

with $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 - \sqrt{2}x_2 \leq 0, x_2 \geq 1\}$ (see Figure 2.5).

The proplem is feasible since, e.g., $\binom{1}{1} \in \mathcal{P} \cap \mathbb{Z}^2 \subset \text{conv}(\mathcal{P} \cap \mathbb{Z}^2) = \mathcal{P}_\mathrm{I}$, and bounded by $x_1 - \sqrt{2}x_2 \leq 0$, but does not have an optimum as we will see. By strong duality (see *Introduction to Optimization*), this implies that the MIP above cannot be an LP, i.e., that $\mathcal{P}_\mathrm{I}$ cannot be a polyhedron.

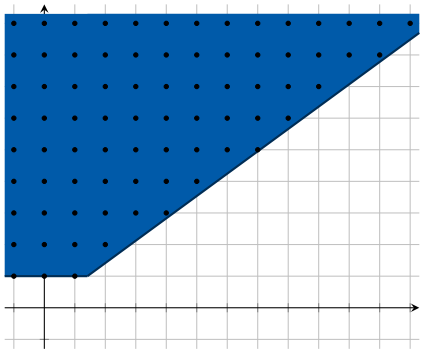

Figure 2.5: Illustration of Example 2.26.

To see that the supremum is not attained, first observe that, for points in $\mathcal{P}_I$, the inequality $x_1 - \sqrt{2}x_2 \leq 0$ is equivalent to $\frac{x_1}{x_2} \leq \sqrt{2}$, because $x_2 > 0$. Since $\sqrt{2}$ is irrational, $x_1 - \sqrt{2}x_2 < 0$ holds for all points in $\mathcal{P} \cap \mathbb{Z}^2$ and thus for all points in $\mathcal{P}_I = \text{conv}(\mathcal{P} \cap \mathbb{Z}^2)$. We construct a sequence $(\boldsymbol{x}^{(i)})_{i \in \mathbb{N}}$ with $\boldsymbol{x}^{(i)} \in \mathbb{N}^2$ and $0 < \sqrt{2}x_2^{(i)} - x_1^{(i)} < \frac{1}{i}$ to prove that the supremum is 0, which is not attained in $\mathcal{P}_I$.[3] Let $y_j := j\sqrt{2} - \lfloor j \cdot \sqrt{2} \rfloor \in [0,1)$ for $j \in \{0, \dots, i\}$, and let $\{\tilde{y}_0, \dots, \tilde{y}_i\} = \{y_0, \dots, y_i\}$ be sorted such that $\tilde{y}_0 \leq \tilde{y}_1 \leq \dots$. By pidgeonhole principle, there are $k > \ell$ with $0 \leq \tilde{y}_k - \tilde{y}_\ell < \frac{1}{i}$. We can now let $x_1^{(i)} := \lfloor k\sqrt{2} \rfloor - \lfloor \ell\sqrt{2} \rfloor$ and $x_2^{(i)} := k - \ell \leq 1$ to obtain $x_1^{(i)} - \sqrt{2}x_2^{(i)} = \tilde{y}_\ell - \tilde{y}_k \in (-\frac{1}{i}, 0]$, as desired. △

The root (literally) of the issue above lies in the irrationality of the data. We now establish that this is the only obstruction to $\mathcal{P}_I$ being a polyhedron.

Definition 2.27. A polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ is called *rational* if there are $A \in \mathbb{Q}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{Q}^m$ with $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b})$.

We begin with the following observation.

Proposition 2.28. If $\mathcal{P}_I \subseteq \mathbb{R}^n$ is a polyhedron, then $\mathcal{P}_I$ is rational.

Proof. If $\mathcal{P}_I$ is a polyhedron, then, according to Theorem 2.19, there are finite sets $\mathcal{V}, \mathcal{E}$ with $\mathcal{P}_I = \text{conv}(\mathcal{V}) + \text{cone}(\mathcal{E})$. Since $\mathcal{P}_I$ is the convex hull of integer points, all vertices of $\mathcal{P}_I$ are integral and every extreme ray contains integer points. So we can choose the sets $\mathcal{V}, \mathcal{E}$ to consist of integer vectors. We can use Corollary 2.21 to obtain a representation by rational inequalities. □

We collect sufficient conditions for $\mathcal{P}_I$ to be a polyhedron.

Proposition 2.29. Let $\mathcal{P} \subseteq \mathbb{R}^n$ be a polyhedron.

(a) If $\mathcal{P}$ is a polytope, then $\mathcal{P}_I$ is also a polytope.

(b) If $\mathcal{P}$ is a rational polyhedral cone, then $\mathcal{P} = \mathcal{P}_I$.

Proof.

(a) Since $\mathcal{P}$ is bounded, $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{Z}^n : \boldsymbol{x} \in \mathcal{P}\}$ is finite. By Corollary 2.9, $\text{conv}(\mathcal{X}) = \mathcal{P}_I$ is a polytope.

[3] It is insufficient to require $x_1^{(i)}/x_2^{(i)} \to \sqrt{2}$, since this does not imply $x_1^{(i)} - \sqrt{2}x_2^{(i)} \to 0$, e.g., for $x_1^{(i)} = \sqrt{i} + i\sqrt{2}$ and $x_2^{(i)} = i$.

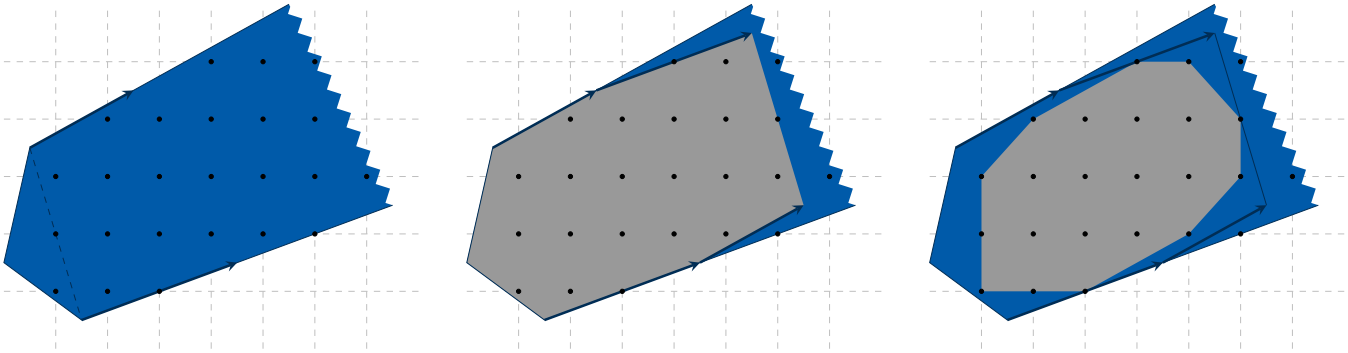

Figure 2.6: Illustration of the proof of Theorem 2.30: *left:* $\mathcal{P}$ with $\boldsymbol{y}^{(1)} = \binom{2}{1}$ and $\boldsymbol{y}^{(2)} = \binom{3}{1}$, *center:* $\mathcal{Q} + \mathcal{B}$ in gray, *right:* $(\mathcal{Q} + \mathcal{B})_I$ in gray.

(b) By Theorem 2.17 and analogously to Corollary 2.21, there exist rational $\boldsymbol{r}^{(i)}$ with $\mathcal{P} = \text{cone}(\{\boldsymbol{r}^{(1)}, \dots, \boldsymbol{r}^{(k)}\})$. By scaling, we can assume that $\boldsymbol{r}^{(i)}$ is integral for all i.

It is clear that $\mathcal{P} \supseteq \mathcal{P}_\text{I}$, since $\mathcal{P}$ is convex and $\mathcal{P}_\text{I}$ is the convex hull of points in $\mathcal{P}$. For the opposite inclusion, let $\boldsymbol{x} \in \mathcal{P}$. For $\boldsymbol{x} = \boldsymbol{0}$ we have $\boldsymbol{x} \in \mathcal{P}_\text{I}$. Otherwise, there exists a vector $\boldsymbol{\lambda} \geq \boldsymbol{0}$ with

$$\boldsymbol{x} = \sum_{i=1}^{k} \lambda_i \boldsymbol{r}^{(i)}.$$

To express $\boldsymbol{x}$ as a convex combination of integral vectors, we rescale each $\boldsymbol{r}^{(i)}$ by the same integral factor to decrease the sum of the coefficients to below 1 and add a suitable multiple of $\boldsymbol{0}$. Specifically, we set $\mu := \boldsymbol{1}^\top \boldsymbol{\lambda} > 0$ (since $\boldsymbol{x} \neq \boldsymbol{0}$) and define $\bar{\boldsymbol{\lambda}} := \frac{1}{\lceil \mu \rceil} \boldsymbol{\lambda}$ to write

$$\boldsymbol{x} = \sum_{i=1}^{k} \bar{\lambda}_i (\lceil \mu \rceil \boldsymbol{r}^{(i)}) + (1 - \boldsymbol{1}^\top \bar{\boldsymbol{\lambda}}) \boldsymbol{0}$$

with $\sum_{i=1}^{k} \bar{\lambda}_i + (1 - \boldsymbol{1}^\top \bar{\boldsymbol{\lambda}}) = 1$. Because the vectors $\boldsymbol{0}, \lceil \mu \rceil \boldsymbol{r}^{(1)}, \dots, \lceil \mu \rceil \boldsymbol{r}^{(k)}$ are integral and contained in $\mathcal{P}$, we have that $\boldsymbol{x}$ is a convex combination of integer points in $\mathcal{P}$, i.e., $\boldsymbol{x} \in \mathcal{P}_\text{I}$. □

We finally conclude that it is sufficient to require the system $A\boldsymbol{x} \leq \boldsymbol{b}$ to have rational coefficients in order to ensure that the convex hull of all integral solutions to form a polyhedron.

Theorem 2.30. If $\mathcal{P} \subseteq \mathbb{R}^n$ is a rational polyhedron, then $\mathcal{P}_\text{I}$ is also a (rational) polyhedron.

Proof. Once we establish that $\mathcal{P}_\text{I}$ is a polyhedron, rationality follows by Proposition 2.28.

By Corollary 2.21, we know that $\mathcal{P}$ has a representation of the form $\mathcal{P} = \mathcal{Q} + \mathcal{C}$, where $\mathcal{Q}$ is a convex hull of rational vectors and $\mathcal{C}$ is the conic hull of rational vectors. By scaling, we can thus determine $\boldsymbol{y}^{(i)} \in \mathbb{Z}^n$, $i \in \{1, \dots, k\}$, with $\mathcal{C} = \text{cone}(\{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(k)}\})$. We set

$$\mathcal{B} := \{\sum_{i=1}^{k} \mu_i \boldsymbol{y}^{(i)} \,:\, \boldsymbol{0} \leq \boldsymbol{\mu} \leq \boldsymbol{1}\}.$$

Note that the (bounded) set $\mathcal{B}$ is a polytope by Corollarly 2.8. By Corollaries 2.6 and 2.8, $\mathcal{Q}+\mathcal{B}$ is a polytope, since it is bounded. According to Proposition 2.29 (a), $(\mathcal{Q}+\mathcal{B})_I$ is therefore a polytope as well. Finally, by Theorem 2.17 and Corollary 2.8, it follows that $(\mathcal{Q}+\mathcal{B})_I+\mathcal{C}$ is a polyhedron.
With this, it suffices to show

$$\mathcal{P}_I = (\mathcal{Q}+\mathcal{B})_I + \mathcal{C}.$$

$\subseteq$: We argued that $(\mathcal{Q}+\mathcal{B})_I+\mathcal{C}$ is a polyhedron, i.e., in particular, it is convex. Since $\mathcal{P}_I$ is the convex hull of integer points, it suffices to show that all integer points in $\mathcal{P}_I$ also lie in $(\mathcal{Q}+\mathcal{B})_I+\mathcal{C}$. So let $\boldsymbol{p} \in \mathcal{P}_I$ be integral. Then there exist $\boldsymbol{q} \in \mathcal{Q}$ and $\boldsymbol{c} \in \mathcal{C}$ with $\boldsymbol{p} = \boldsymbol{q} + \boldsymbol{c}$. We can find $\boldsymbol{\mu} \geq \boldsymbol{0}$ with

$$\boldsymbol{c} = \sum_{i=1}^{k} \mu_i \boldsymbol{y}^{(i)}.$$

Now let

$$\boldsymbol{c'} := \sum_{i=1}^{k} \lfloor \mu_i \rfloor \boldsymbol{y}^{(i)}$$

and $\boldsymbol{b} := \boldsymbol{c} - \boldsymbol{c'}$. It follows that $\boldsymbol{c'} \in \mathcal{C} \cap \mathbb{Z}^n$ and $\boldsymbol{q} + \boldsymbol{b} = \boldsymbol{p} - \boldsymbol{c'} \in \mathbb{Z}^n$, since $\boldsymbol{p}$ and $\boldsymbol{c'}$ are integral. Furthermore, $\boldsymbol{b} \in \mathcal{B}$ by definition and therefore $\boldsymbol{q} + \boldsymbol{b} \in (\mathcal{Q}+\mathcal{B})_I$. Hence, $\boldsymbol{p} = \boldsymbol{q} + \boldsymbol{b} + \boldsymbol{c'} \in (\mathcal{Q}+\mathcal{B})_I + \mathcal{C}$.

$\supseteq$: Because of $\mathcal{B} \subseteq \mathcal{C}$ we have

$$(\mathcal{Q}+\mathcal{B})_I + \mathcal{C} \subseteq (\mathcal{Q}+\mathcal{C})_I + \mathcal{C} = \mathcal{P}_I + \mathcal{C},$$

and, by Proposition 2.29 (b),

$$\begin{aligned} \mathcal{P}_I + \mathcal{C} &= \mathcal{P}_I + \mathcal{C}_I \\ &\subseteq (\mathcal{P}+\mathcal{C})_I = \mathcal{P}_I, \end{aligned}$$

which establishes $(\mathcal{Q}+\mathcal{B})_I + \mathcal{C} \subseteq \mathcal{P}_I$. □

Corollary 2.31. Let $\mathcal{P} \subseteq \mathbb{R}^n$ be a rational polyhedron such that $\mathcal{P}_I \neq \emptyset$. Then, for all $\boldsymbol{c} \in \mathbb{R}^n$, $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\}$ is bounded if and only if $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}_I\}$ is bounded.

Proof. In the proof of Theorem 2.30, we have $\mathcal{P} = \mathcal{Q}+\mathcal{C}$ and $\mathcal{P}_I = (\mathcal{Q}+\mathcal{B})_I+\mathcal{C}$, where $\mathcal{C}$ is the only unbounded term in either equality. In addition, $\mathcal{P}_I \neq \emptyset$ (and thus $\mathcal{P} \neq \emptyset$) implies that $(\mathcal{B}+\mathcal{Q})_I \neq \emptyset$ and $\mathcal{Q} \neq \emptyset$. Hence, both $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\}$ and $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}_I\}$ are unbounded if and only if there exists $\boldsymbol{r} \in \mathcal{C}$ with $\boldsymbol{c}^\top \boldsymbol{r} > 0$. □

With analogous arguments, Theorem 2.30 can be extended to the mixed-integer case.

Theorem 2.32. If $\mathcal{P}$ is a rational polyhedron, then $\mathcal{P}_{I,p}$ is a rational polyhedron for all $p \in \{0, \ldots, n\}$.

We conclude the chapter with a useful characterization.

Lemma 2.33. Let $\mathcal{P}$ be a rational polyhedron. Then the following are equivalent:

(a) $\mathcal{P} = \mathcal{P}_I$.

(b) Every non-empty face of $\mathcal{P}$ contains an integer point.

(c) Every non-empty minimal face of $\mathcal{P}$, i.e., every face containing no other faces, contains an integer point.

(d) $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\}$ is assumed at an integer point, for all $\boldsymbol{c} \in \mathbb{R}^n$ (or $\boldsymbol{c} \in \mathbb{Z}^n$), for which the maximum is finite.

Proof. exercise. □

In particular, a MIP reduces to an LP if the underlying polyhedron is integral.

Corollary 2.34. If $\mathcal{P}(A, \boldsymbol{b})$ is integral, then the optimum vertex solutions of the following problems coincide

$$\begin{array}{ll} \max & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \leq \boldsymbol{b}, \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{array} \qquad \begin{array}{ll} \max & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \leq \boldsymbol{b}, \\ & \boldsymbol{x} \in \mathbb{R}^n. \end{array}$$

3 Branch-and-Bound Method

In this chapter we deal with the branch-and-bound method, an exact method for solving mixed-integer programs. The method exhaustively searches the solution space and thus guarantees that an optimum solution is found if one exists. In case the underlying problem is bounded, the method terminates, but with exponential running time. In practice, the branch-and-bound method is very often efficient and is the basis of all modern integer optimization solvers. We begin by showing that, from a theoretical perspective, solving mixed-integer programs is computationally difficult.

3.1 Complexity of integer programs

We already know (Proposition 2.29 (a) and Theorem 2.30) that every bounded or rational MIP can be reduced to a linear program, i.e., there exist $D \in \mathbb{Q}^{m\times n}$ and $\boldsymbol{d} \in \mathbb{Q}^m$, with $\mathcal{P}_I = \mathcal{P}(D, \boldsymbol{d})$. Here, m can become very large. We now show that, in general, we cannot hope to solve a rational MIP efficiently. More precisely, we show the following.

Theorem 3.1. The decision problem "Has a given rational system of inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ an integral solution?" is NP-complete (even if $\boldsymbol{x}$ is binary).

We begin by recalling central definitions (see *Algorithmic Discrete Mathematics*).

Definition 3.2. A *decision problem* is a tuple $(\mathcal{I}, (S_I)_{I\in\mathcal{I}})$ with $S_I \in \{\{\text{'yes'}\}, \{\text{'no'}\}\}$ for all $I \in \mathcal{I}$. The *input size* $|I|$ of an instance $I \in \mathcal{I}$ is defined to be equal to the number of bits in the binary representation of $\mathcal{I}$. An algorithm has polynomial running time if it computes $A(I) = S_I$ in time $|I|^{O(1)}$ for all $I \in \mathcal{I}$.

Roughly speaking, NP is the class of decision problems for which 'yes'-instances can be verified in polynomial time. For a rigorous definition we refer to *Algorithmic Discrete Mathematics*. Based on this class, we can capture the computational intractability as follows.

Definition 3.3. A decision problem $\Pi = (\mathcal{I}, (S_I)_{I\in\mathcal{I}})$ is *NP-hard* if, for all $\Pi' = (\mathcal{I}', (S'_{I'})_{I'\in\mathcal{I}'}) \in \text{NP}$, there is a *reduction* $R\colon \mathcal{I}' \to \mathcal{I}$ that satisfies $S'_{I'} = S_{R(I)}$ and can be computed in polynomial time. If, additionally, $\Pi \in \text{NP}$, then Π is *NP-complete*.

One of the most important NP-complete problems is the satisfiability problem (SAT).

Satisfiability (SAT) Problem
input: **CNF formula given by *clauses* $\mathcal{C} = \{C_i\}_{i=1,\dots,m}$ over *variables* $\mathcal{X} = \{x_i\}_{i=1,\dots,n}$**
problem: **Is there a *satisfying assignment* $\alpha\colon \mathcal{X} \to \{0,1\}$?**

An example of a CNF formula is $(x_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee \bar{x}_3) \wedge (x_3 \vee \bar{x}_4)$ with satisfying assignment, e.g., $\alpha(x_1) = \alpha(x_2) = 1$ and $\alpha(x_3) = \alpha(x_4) = 0$.

We need the following observation about the *coding length* of a solution, i.e., the length of a binary representation (for a proof see [39, Chapter 17]).

Lemma 3.4. Let $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b})$ be a rational polyhedron such that the coding length of each inequality is at most $\varphi \in \mathbb{N}$. If $\mathcal{P}_\mathrm{I} \neq \emptyset$, then there is an integer point in $\mathcal{P}$, whose coding length is at most $6n^3\varphi$.

Proof of Theorem 3.1. From Lemma 3.4 we obtain a polynomial certificate for $\mathcal{P} \neq \emptyset$. It follows that the decision problem is in NP.

We can model a given SAT instance with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$ as an integer system $A\hat{\boldsymbol{x}} \leq \boldsymbol{b}$ consisting of the (negated) inequalities

$$\sum_{j : x_j \in C_i} \hat{x}_j + \sum_{j : \bar{x}_j \in C_i} (1 - \hat{x}_j) \geq 1 \quad \forall\, i \in \{1, \ldots, m\},$$

with $\hat{\boldsymbol{x}} \in \{0,1\}^n$. An integral solution of $A\hat{\boldsymbol{x}} \leq \boldsymbol{b}$ can then be interpreted as

$$\hat{x}_j = \begin{cases} 1, & \text{if variable } x_j \text{ is set to “true”,} \\ 0, & \text{if variable } x_j \text{ is set to “false”.} \end{cases}$$

Now, $A\hat{\boldsymbol{x}} \leq \boldsymbol{b}$ has an integral solution (i.e. $\{\hat{x} \in \{0,1\}^n : A\hat{\boldsymbol{x}} \leq \boldsymbol{b}\} \neq \emptyset$) if and only if the given SAT instance has a solution. The transformation can be carried out in polynomial time and $\hat{\boldsymbol{x}}$ is obviously bounded. □

The following is an immediate consequence of Theorem 3.1.

Corollary 3.5. Every problem in NP can be modeled as a (binary) integer program.

Since finding a feasible solution to a MIP is NP-hard and thus solving a MIP is difficult, but LPs can be solved in polynomial time (see *Introduction to Optimization*), we cannot expect to efficiently find a complete description of $\mathcal{P}_\mathrm{I}$ or to solve the associated separation problem (unless P = NP). With this in mind, we now introduce the branch-and-bound method for solving general mixed-integer programs. In Chapter 4, we will deal with cases that allow to find solutions efficiently.

3.2 The branch-and-bound method

The *branch-and-bound* method revolves around the two operations of *branching* and *bounding*. We want to solve the mixed-integer optimization problem

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{aligned} \tag{3.1}$$

where $A \in \mathbb{Q}^{m \times n}$, $\boldsymbol{c} \in \mathbb{Q}^n$, $\boldsymbol{b} \in \mathbb{Q}^m$, and $p \in \{1, \ldots, n\}$. Note that we assume that all data are rational, because this is the case in practice and because the irrational case causes theoretical problems, as we saw in Section 2.3.

We use a "relaxation" that yields an upper bound on the optimum value of the problem (3.1) by forgoing the integer conditions. We obtain the *LP relaxation*

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{R}^n. \end{aligned} \tag{3.2}$$

Such linear programs can be solved efficiently (see *Introduction to Optimization*). If the computed optimum solution of the LP relaxation is integral, we have solved (3.1). Otherwise, we branch, i.e., we generate subproblems, which we solve recursively. In this way, we generate a collection of subproblems with shrinking feasible regions. Every optimum solution to our integer problem appears in one of the subproblems, so that no optimum solutions are lost on the way.

The LP relaxation can be further enhanced by inequalities (see Chapter 5) and other relaxations can be used (see Chapter 6). Furthermore, the procedure can be accelerated by heuristics (see Chapter 7).

3.2.1 Divide and conquer

As a first step, we solve the LP relaxation (3.2) to obtain a fractional optimum solution $\hat{\boldsymbol{x}}$. If we are lucky,

$$\hat{\boldsymbol{x}} \in \mathcal{X} := \{\boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : A\boldsymbol{x} \leq \boldsymbol{b}\},$$

i.e., $\hat{\boldsymbol{x}}$ is feasible for (3.1), and we have already found an optimum solution to the mixed-integer problem. Sufficient conditions that guarantee that this case occurs will be discussed in Chapter 4.

In general, the solution $\hat{\boldsymbol{x}}$ of the LP relaxation is not integral (i.e., $\hat{\boldsymbol{x}} \notin \mathbb{Z}^p \times \mathbb{R}^{n-p}$). In this case, there exists an index $i \in \{1, \dots, p\}$ such that $\hat{x}_i \notin \mathbb{Z}$. We now create two subproblems

$$\mathcal{X}^{\leq} := \{\boldsymbol{x} \in \mathcal{X} : x_i \leq \lfloor \hat{x}_i \rfloor\} \subset \mathcal{X} \qquad \text{and} \qquad \mathcal{X}^{\geq} := \{\boldsymbol{x} \in \mathcal{X} : x_i \geq \lceil \hat{x}_i \rceil\} \subset \mathcal{X}.$$

Evidently, $\mathcal{X} = \mathcal{X}^{\leq} \cup \mathcal{X}^{\geq}$ and $\mathcal{X}^{\leq} \cap \mathcal{X}^{\geq} = \emptyset$, so that every (optimum) solution of the MIP (3.1) must either lie in $\mathcal{X}^{\leq}$ or in $\mathcal{X}^{\geq}$. These sets are associated with the two polyhedra

$$\mathcal{P}^{\leq} := \{\boldsymbol{x} \in \mathcal{P} : x_i \leq \lfloor \hat{x}_i \rfloor\} \qquad \text{and} \qquad \mathcal{P}^{\geq} := \{\boldsymbol{x} \in \mathcal{P} : x_i \geq \lceil \hat{x}_i \rceil\},$$

where $\mathcal{P} := \mathcal{P}(A, \boldsymbol{b}) = \{\boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\}$. Because $\hat{\boldsymbol{x}} \notin \mathcal{P}^{\leq} \cup \mathcal{P}^{\geq}$, the LP relaxations of both subproblems yield optimum solutions different from $\hat{\boldsymbol{x}}$. We can now apply the procedure recursively to the two subproblems.

The recursive subdivision into subproblems creates a *branch-and-bound tree* whose nodes correspond to the subproblems, see Figure 3.1. In each step, a subproblem that has not yet been processed is selected and an optimum solution $\hat{\boldsymbol{x}}$ of the corresponding LP relaxation is computed; if it is infeasible, we continue with another subproblem. If $\hat{\boldsymbol{x}} \notin \mathcal{X}$, we create two new subproblems. If $\hat{\boldsymbol{x}} \in \mathcal{X}$, we have found a feasible solution for the MIP. In this case we keep track of the best solution $\boldsymbol{x}^\star$ we encountered so far and continue with another subproblem. The procedure terminates when all subproblems have been processed. The optimum solution then is $\boldsymbol{x}^\star$ if any solution was found, otherwise the problem is infeasible.

In practice, the procedure can be accelerated using the following simple insight. Let us assume that we already have found some feasible solution $\boldsymbol{x}^\star \in \mathcal{X}$. Then, obviously,

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{X}\} \geq \boldsymbol{c}^\top \boldsymbol{x}^\star.$$

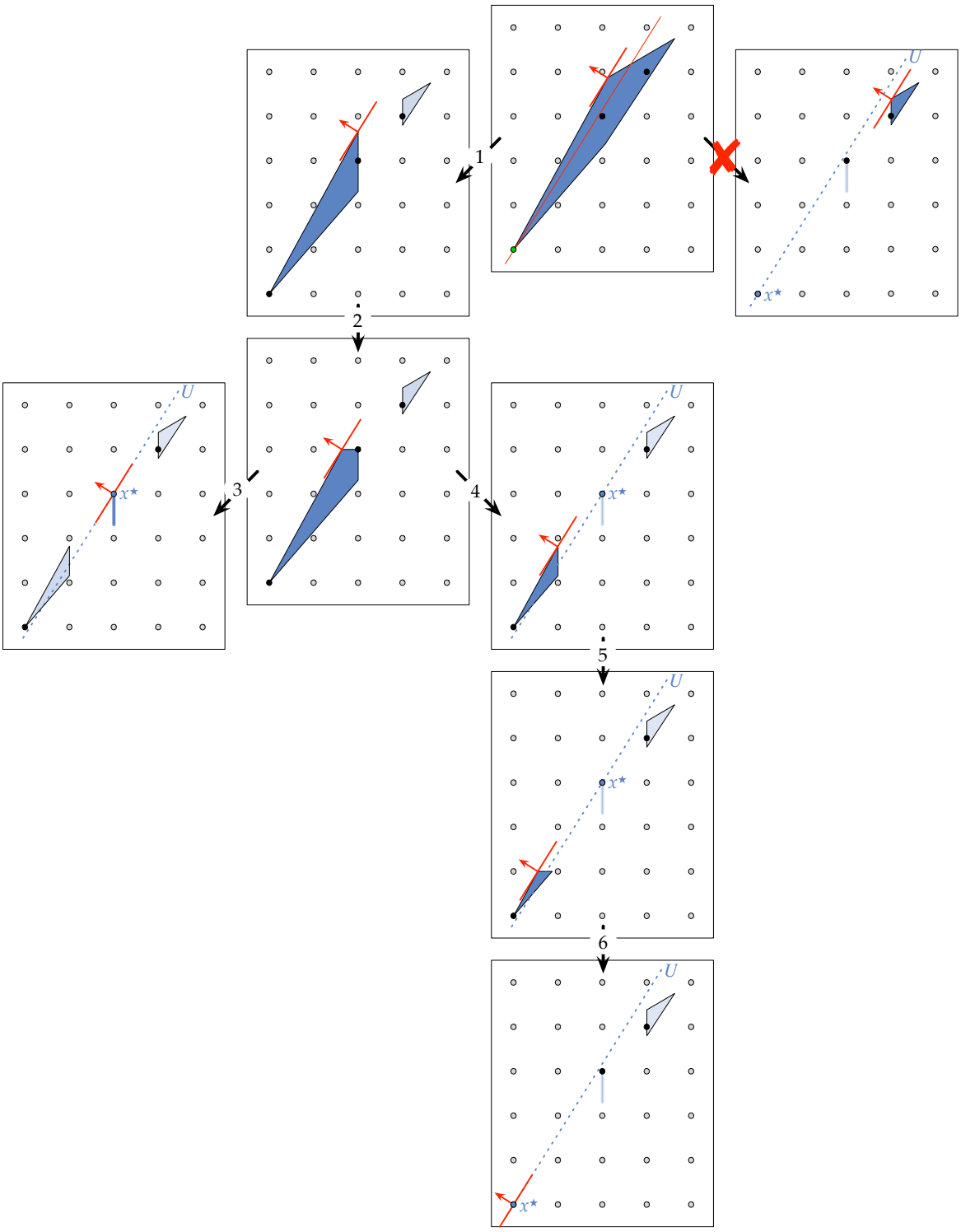

Figure 3.1: Example of an application of the branch-and-bound method: The tree is searched from left to right. In the lower parts, the respective other subproblems are infeasible and ommitted.

The solution $\boldsymbol{x}^\star$ therefore provides us with a lower bound on the optimum objective function value of our MIP. If we now solve a subproblem with associated polyhedron $\hat{\mathcal{P}}$ satisfying

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, \boldsymbol{x} \in \hat{\mathcal{P}}\} \leq \boldsymbol{c}^\top \boldsymbol{x}^\star,$$

then there cannot be any better feasible solution than $\boldsymbol{x}^\star$ in this subproblem, and this subproblem need not be further considered. Accordingly, we may prune (i.e., "cut off") the corresponding subtree.

3.2.2 Formal definition

The above procedure results in the *branch-and-bound method*, which is formally stated below.

Algorithm: branch-and-bound method

input: MIP $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}\}$ with $\mathcal{P}(A, \boldsymbol{b})$ bounded
output: optimum solution $\boldsymbol{x}^\star$ or "infeasible" if $\underline{z} = -\infty$

```
A ← {P(A, b)}                                              (active nodes)
z ← −∞, x̄ ← ∅                                              (lower bound, best solution)
while A ≠ ∅:
    choose P̂ ∈ A                                           (subproblem selection)
    if P̂ ≠ ∅:
        compute x̃ ∈ P̂ ∩ (Z^p × R^{n−p}) via primal heuristics   (see Chapter 7)
        if c^⊤ x̃ > z:
            z ← c^⊤ x̃, x̄ ← x̃
        x̂ ← arg max{c^⊤ x : x ∈ P̂}                         (or use Chapter 6)
        if c^⊤ x̂ > z:
            if x̂ ∈ Z^p × R^{n−p}:
                z ← c^⊤ x̂, x̄ ← x̂                           (see Chapter 4)
            else
                choose i ∈ {1, ..., p} with x̂_i ∉ Z        (variable selection)
                P̂^≤ ← {x ∈ P̂ : x_i ≤ ⌊x̂_i⌋}, P̂^≥ ← {x ∈ P̂ : x_i ≥ ⌈x̂_i⌉}   (or use Chapter 5)
                A ← A ∪ {P̂^≤, P̂^≥}
    A ← A \ {P̂}
return (x̄, z)
```

Remark 3.6. The grayed out part of the algorithm is optional and intended for acceleration – it is not necessary for correct execution. Techniques from Chapter 7 can be used here.

The key observation for correctness of branch-and-bound is that the algorithm maintains the following invariants.

Lemma 3.7. Branch-and-bound maintains the invariants that

(a) if $\underline{z} > -\infty$, then $\bar{\boldsymbol{x}}$ is feasible for (3.1) with $\boldsymbol{c}^\top \bar{\boldsymbol{x}} = \underline{z}$, and

(b) for every optimum solution $\boldsymbol{x}^\star$ of (3.1), either $\underline{z} = \boldsymbol{c}^\top \boldsymbol{x}^\star$, or there is an active node $\hat{\mathcal{P}} \in A$ with $\boldsymbol{x}^\star \in \hat{\mathcal{P}}$.

Proof. The first part of the invariant follows since $\underline{z}$ and $\bar{\boldsymbol{x}}$ are only updated in unison and only when $\bar{\boldsymbol{x}}$ is a feasible solution of value $\underline{z}$.

For the second part, observe that only fractional solutions are eliminated when branching. This means that every optimum solution $\boldsymbol{x}^\star$ remains in some active node $\hat{\mathcal{P}}$ until this node can be solved without branching, which only happens when the corresponding LP relaxation yields an integral optimum solution. This solution cannot be better than $\boldsymbol{x}^\star$ since the integral solutions in $\hat{\mathcal{P}}$ are a subset of $\mathcal{X}$, and it cannot be worse than $\boldsymbol{x}^\star$, since $\boldsymbol{x}^\star \in \hat{\mathcal{P}}$. By the first part of the invariant and optimality of $\boldsymbol{x}^\star$, we have $\underline{z} \leq \boldsymbol{c}^\top \boldsymbol{x}^\star$. Hence, after considering $\hat{\mathcal{P}}$, we have $\underline{z} = \boldsymbol{c}^\top \boldsymbol{x}^\star$ (with or without update). □

This immediately yields the following bounds.

Corollary 3.8. The branch-and-bound method maintains the invariant that $\underline{z}$ is a lower bound and

$$\bar{z} := \max\left\{\underline{z},\ \max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, \boldsymbol{x} \in \hat{\mathcal{P}},\ \hat{\mathcal{P}} \in A\}\right\}$$

is an upper bound on the optimum value of (3.1).

We show that the branch-and-bound method computes the correct result in finite time.

Theorem 3.9. If $\mathcal{P}$ is bounded, then the branch-and-bound method terminates after a finite number of steps and the final value of $\underline{z}$ is equal to the optimum value of the MIP (3.1).

Proof. Assume that branch-and-bound terminates. For the case that (3.1) is infeasible, Lemma 3.7 (a) implies that the correct value $\underline{z} = -\infty$ is maintained. If (3.1) is feasible, Lemma 3.7 (b) implies that branch-and-bound can only terminate once $\underline{z} = \boldsymbol{c}^\top \boldsymbol{x}^\star$ for every optimum solution $\boldsymbol{x}^\star$ and that, at this point, $\bar{\boldsymbol{x}}$ must be optimal.

We now show that the algorithm terminates. Since $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b})$ is bounded, there is a constant $\varphi \in \mathbb{N}$ so that

$$\mathcal{P} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n \,:\, -\varphi \mathbf{1} \leq \boldsymbol{x} \leq \varphi \mathbf{1}\}.$$

Every LP relaxation that occurs during branch-and-bound is defined by $A\boldsymbol{x} \leq \boldsymbol{b}$ and constraints on the variables of the form

$$x_i \geq \xi \quad \text{with } \xi \in \{-\varphi + 1, \dots, \varphi\} \quad \text{or} \tag{3.3}$$

$$x_i \leq \xi \quad \text{with } \xi \in \{-\varphi, \dots, \varphi - 1\} \tag{3.4}$$

for $i \in \{1, \dots, p\}$. Because $\hat{\boldsymbol{x}}$ is not feasible for either of the two subproblems generated when branching, and since all active subproblems have disjoint feasible regions, all LP relaxations encountered by the algorithm are different. This means that the LP relaxations use different subsets of inequalities of the type in (3.3) and (3.4). However, there are only $4p\varphi$ different inequalities, i.e., $2^{4p\varphi}$ different combinations, and thus subproblems, to consider. □

Remark 3.10. For binary programs, the height of the branch-and-bound tree is bounded by the dimension n, hence it has at most 2^n leaves and thus there are at most $2^{n+1} - 1$ subproblems to consider.

Remark 3.11. If $\mathcal{P}$ is unbounded, the method may not terminate (*exercise*). However, an estimation of the absolute values of the components of an optimum solution, as in Lemma 3.4, can be used to constrain the problem without losing all optimum solutions.

Thus, we can say that the branch-and-bound method terminates. However, it may encounter exponentially many subproblems, even for infeasible binary programs (*exercise*). This performance is not surprising, since it is NP-hard to find a feasible solution (Theorem 3.1).

3.2.3 Practical considerations

Dual simplex

A key insight for an efficient implementation is that the dual simplex algorithm is ideally suited to solve the LP relaxations. This is because when a subproblem is subdivided and an additional constraint on the variable x_i is imposed, the current basis remains dually feasible (not primally, of course). The same is true for the addition of further inequalities in the branch-and-cut method. The dual simplex can resume directly with the optimum basis of the parent node. This is particularly efficient when moving on to a child in subproblem selection, but is still useful otherwise if the information about the final basis every parent node is stored in memory.

Selection rules

So far, we have discussed a generic version of the branch-and-bound method. Two degrees of freedom remain: The selection of the next active subproblem and the selection of the variable $\hat{x}_i \notin \mathbb{Z}$ to branch on, i.e., the variable whose domain is subdivided.

Subproblem selection When selecting the next subproblem, there are, e.g., the following options.

- With *depth-first search*, the active node created most recently is selected next, i.e., in particular, one of the children of the previously considered node if possible. Depth-first search has the advantage that consecutive subproblems differ only slightly from one another and thus the required modification is small. Furthermore, the chance of finding an integral solution (and possibly improving the bound $\underline{z}$) increases with increasing depth, as long as the problems are feasible and are not pruned.
- With *breadth-first search*, the active node created least recently is processed next, i.e., a node of smallest depth in the tree. This method systematically works through the tree level by level. Under the assumption that the local upper bounds $\bar{z}(\hat{\mathcal{P}})$ of the nodes $\hat{\mathcal{P}}$ decrease with increasing depths, this method primarily reduces the upper bound $\bar{z}$ of Lemma 3.8.
- With *best-bound search,* the active node $\hat{P}$ of maximum local upper bound $\bar{z}(\hat{\mathcal{P}}) := \max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \hat{\mathcal{P}}\}$ is selected next. The idea is to attempt to reduce $\bar{z}$ as quickly as possible. The disadvantage of this variant (and of breadth-first search) lies in the fact that feasible integral solutions are found relatively late, i.e., that the lower bound $\underline{z}$ has no effect in the beginning.

In practice, best-bound search is typically used, but a few depth-first steps are carried out from time to time to find feasible integral solutions a bit sooner.

Variable selection – branching rule For the selection of the variable $\hat{x}_i \notin \mathbb{Z}$, there are various options, including the following.

- Choose $i \in \arg\max\left\{\min\{\hat{x}_j - \lfloor \hat{x}_j \rfloor, \lceil \hat{x}_j \rceil - \hat{x}_j\},\ j \in \{1,\dots,p\}\right\}$, i.e., the variable of largest distance to the next integer, or a maximally infeasible variable with respect to integrality. The hope is that the values of such variables must change considerably in both subproblems and therefore cause large changes in the objective function value.
- Choose $i = \arg\min\left\{\min\{\hat{x}_j - \lfloor \hat{x}_j \rfloor, \lceil \hat{x}_j \rceil - \hat{x}_j\},\ \hat{x}_j \notin \mathbb{Z},\ j \in \{1,\dots,p\}\right\}$, i.e., the variable of smallest distance to the next integer, or a minimally infeasible variable with respect to integrality. The hope is that these variables only need little change to become integral.

- *Pseudo-cost branching* uses historical information about the change in the objective function when a certain variable is selected as the branching variable. If, in the past, branches with variable j changed the objective function value by Δ_j^- or Δ_j^+ in the children whose subproblems were rounded down, respectively up, we set

$$\zeta_j^- := \frac{\Delta_j^-}{\hat{x}_j - \lfloor \hat{x}_j \rfloor} \quad \text{and} \quad \zeta_j^+ := \frac{\Delta_j^+}{\lceil \hat{x}_j \rceil - \hat{x}_j}.$$

  Then, we calculate the arithmetic means $\bar{\zeta}_j^-$ and $\bar{\zeta}_j^+$ of these values, which provide an estimator for the change of the objective function value per unit, similarly to reduced costs in the simplex method (see *Introduction to Optimization*). In the current node, we select a variable j of largest valuation $B\big((\hat{x}_j - \lfloor \hat{x}_j \rfloor) \cdot \bar{\zeta}_j^-, (\lceil \hat{x}_j \rceil - \hat{x}_j) \cdot \bar{\zeta}_j^+\big)$. The valuation function could, for example, be $B(q^-, q^+) = (1-\mu) \cdot \min\{q^-, q^+\} + \mu \cdot \max\{q^-, q^+\}$ (where $\mu \in [0,1]$). An alternative is $B(q^-, q^+) = \max\{q^-, \varepsilon\} \cdot \max\{q^+, \varepsilon\}$ (with $\varepsilon > 0$).

  This branching rule works particularly well if a lot of information about the variables is already available, especially if there has been a lot of branching. In the beginning, the selection is essentially random.
- *Strong branching* generates the two subproblems for all possible candidates $i \in \{1, \ldots, p\}$ and calculates the corresponding changes in objective function value. It then selects the variable that causes the largest change. This procedure is very time-consuming, but almost always yields small trees. In practice, it is often accelerated by limiting the number of simplex iterations and the number of candidates.
- *Reliability branching* combines pseudo-cost and strong branching. In the beginning, the pseudo-costs are initialized via strong-branching. Later, when the values are more reliable, the pseudo costs are used.

For details on the last three rules, we refer to the literature (see, e.g., [1]). It turns out that the first two rules – although intuitively good – perform very poorly in practice: They are about as good as a random selection rule. The last rule, on the other hand, yields significantly better results.

Presolving

Presolving is essential for the practical performance of the branch-and-bound method. It tries to simplify the MIP with the aim of making the resulting problem easier. Typically, presolving focuses on reducing the number of variables and constraints. We list examples of presolving steps:

- Eliminate trivial constraints, e.g., $\mathbf{0}^\top \boldsymbol{x} \leq 1$.
- Eliminate duplicate constraints.
- Eliminate redundant constraints, e.g., $2x_1 + x_2 \leq 3;\ \ x_1, x_2 \in \{0,1\}$.
- Simplify constraints, e.g., $2x_1 + 2x_2 = 2 \to x_1 + x_2 = 1$.
- Strengthen coefficients, e.g., $2x_1 + x_2 \geq 1;\ \ x_1, x_2 \in \{0,1\} \to x_1 + x_2 \geq 1$.
- Aggregate variables: Consider $\alpha_1 x_1 + \cdots + \alpha_k x_k = \beta$ with $\alpha_i \neq 0$. Then the equation can be solved for x_i and be eliminated together with x_i. The objective function must be adapted accordingly. Note that aggregation can make the coefficient matrix of a MIP denser, i.e., introduce more non-zero elements. This can have a negative impact on the the solving speed for the LP relaxations.
- Recognize special types of constraints, e.g., knapsack inequalities, set-packing constraints etc. For these, we can then, for example, generate the inequalities of Section 5.
- Presolve dual: E.g., if the only constraints involving x_2 are $2x_1 + x_2 \geq 3$ and $x_2 \leq 5$, the objective function coefficient of x_2 is non-negative, and $x_1 \geq 0$, we can fix $x_2 = 5$ and eliminate the first inequality.

- Use probing: Fix binary variables to 0 or 1 and test for consequences, e.g., for $x_1 + x_2 \leq 1$, $x_1 - x_2 \leq 0$, and $x_1, x_2 \in \{0,1\}$, trying $x_1 = 1$ yields $x_2 = 0$, which yields $x_1 = 0$ – a contradiction. So we can safely set $x_1 = 0$. Similarly, implications such as, e.g., $x_i = 1 \Rightarrow x_j = 0$ can be recognized. These implications then lead to set-packing conditions (e.g., $x_i + x_j \leq 1$).

Such presolving steps can dramatically decrease the size of MIPs, especially those generated by modeling languages.

Fixing reduced costs

During the branch-and-bound procedure, it is possible to use the reduced costs of the solution to the LP relaxation. To simplify notation, let us consider the problem

$$\max \{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0},\ \boldsymbol{x} \in \mathbb{Z}^n\}.$$

Let $\hat{\boldsymbol{x}}$ be an optimum basis solution of the relaxation $\max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0}\}$ with associated basis B and non-basis $N = \{1, \ldots, n\} \setminus B$, i.e., $\hat{\boldsymbol{x}}_{\boldsymbol{B}} = A_{\cdot B}^{-1}\boldsymbol{b}$, $\hat{\boldsymbol{x}}_{\boldsymbol{N}} = \boldsymbol{0}$. Let $\underline{z}$ be a lower bound on the optimum value. For every feasible integral solution $\boldsymbol{x}$ with objective function value at least $\underline{z}$, we have

$$\begin{aligned}
\underline{z} - \boldsymbol{c}^\top \hat{\boldsymbol{x}} \leq \boldsymbol{c}^\top \boldsymbol{x} - \boldsymbol{c}^\top \hat{\boldsymbol{x}} &= \boldsymbol{c}_{\boldsymbol{N}}^\top (\boldsymbol{x}_{\boldsymbol{N}} - \hat{\boldsymbol{x}}_{\boldsymbol{N}}) + \boldsymbol{c}_{\boldsymbol{B}}^\top (\boldsymbol{x}_{\boldsymbol{B}} - \hat{\boldsymbol{x}}_{\boldsymbol{B}}) \\
&= \boldsymbol{c}_{\boldsymbol{N}}^\top \boldsymbol{x}_{\boldsymbol{N}} + \boldsymbol{c}_{\boldsymbol{B}}^\top (A_{\cdot B}^{-1}\boldsymbol{b} - A_{\cdot B}^{-1} A_{\cdot N} \boldsymbol{x}_{\boldsymbol{N}} - A_{\cdot B}^{-1}\boldsymbol{b}) \\
&= \boldsymbol{c}_{\boldsymbol{N}}^\top \boldsymbol{x}_{\boldsymbol{N}} - \boldsymbol{c}_{\boldsymbol{B}}^\top A_{\cdot B}^{-1} A_{\cdot N} \boldsymbol{x}_{\boldsymbol{N}} \\
&= (\boldsymbol{c}_{\boldsymbol{N}}^\top - \boldsymbol{c}_{\boldsymbol{B}}^\top A_{\cdot B}^{-1} A_{\cdot N}) \boldsymbol{x}_{\boldsymbol{N}} = \boldsymbol{z}_{\boldsymbol{N}}^\top \boldsymbol{x}_{\boldsymbol{N}} \leq z_j x_j,
\end{aligned}$$

for all $j \in N$. For the reduced costs, we have $\boldsymbol{z}_{\boldsymbol{N}} \leq \boldsymbol{0}$, because the basis B is optimal. If now $z_j < 0$, then

$$x_j \leq \frac{\underline{z} - \boldsymbol{c}^\top \hat{\boldsymbol{x}}}{z_j} = \frac{\boldsymbol{c}^\top \hat{\boldsymbol{x}} - \underline{z}}{|z_j|}.$$

This results in an upper bound on the value of x_j. If, for example, $(\boldsymbol{c}^\top \hat{\boldsymbol{x}} - \underline{z})/|z_j| < 1$, the integer variable x_j can be fixed to 0. The same procedure can be used for lower bounds.

Branch-and-cut

The combination of the branch-and-bound method with the generation of cutting planes to eliminate half-spaces is called branch-and-cut (see [38]). Here, the inequalities from Chapter 5 can be used. There are many possibilities for variation: The exact configuration must be empirically evaluated for each application. We are still far away from a true theoretical understanding of the advantages and disadvantages of the different variants. In general, the branch-and-cut method works very well in practice, but requires careful tuning of the parameters.

4 Integral Polyhedra

In this chapter we deal with polyhedra whose faces always contain integer points. Vertex solutions of the corresponding linear programs are therefore integral. For these special cases, the integer optimization problem can be solved via its LP relaxation, and thus in polynomial time (see *Introduction to Optimization*).

4.1 Total unimodularity

We first examine sufficient conditions on the constraint matrix $A \in \mathbb{Z}^{m \times n}$ for the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ to be integral for every integral right-hand side $\boldsymbol{b} \in \mathbb{Z}^m$. In particular, this allows to solve the corresponding MIP in polynomial time, since it is sufficient to solve its LP relaxation.

Definition 4.1. A matrix $A \in \mathbb{Z}^{m \times n}$ is called

(a) *unimodular* if A has full row rank, and the determinant of every $(m \times m)$-submatrix of A is in $\{-1, 0, 1\}$.

(b) *totally unimodular (TU)* if the determinant of every square submatrix of A is in $\{-1, 0, 1\}$.

Note that all entries (the square submatrices of size size 1) of a totally unimodular matrix are either -1, 0, or 1. Furthermore, the property of a matrix being TU is closed under taking submatrices.

Remark 4.2. Every totally unimodular matrix of full row rank is also unimodular. The converse does not hold: The integer matrix

$$\begin{pmatrix} 2 & 3 & 2 \\ 4 & 2 & 3 \\ 9 & 6 & 7 \end{pmatrix}$$

has determinant 1 (and therefore also full rank), so it is unimodular, but not totally unimodular, since its entries are not in $\{-1, 0, 1\}$:

Total unimodularity can be characterized in various ways that will prove useful later on.

Proposition 4.3. The following are equivalent:

- A is totally unimodular
- $[A, \mathbb{I}]$ is unimodular
- $\begin{pmatrix} A \\ -A \\ \mathbb{I} \\ -\mathbb{I} \end{pmatrix}$ is totally unimodular.
- $\begin{pmatrix} A & 0 \\ \mathbb{I} & \mathbb{I} \end{pmatrix}$ is totally unimodular.
- $A^\top$ is totally unimodular.

Proof. exercise. □

Proposition 4.4. The vertex-arc incidence matrix of a directed graph $G = (V, E)$ (i.e., $A = (a_{v,e})_{v \in V, e \in E}$ with $a_{v,e} = 1$, if $e \in \delta^+(v)$, $a_{v,e} = -1$ if $e \in \delta^-(v)$, and $a_{v,e} = 0$ otherwise) is totally unimodular.

Proof. exercise. □

The following three results exhibit that a linear program with a (totally) unimodular constraint matrix always has an integral optimum solution, provided one exists at all.

Theorem 4.5. An integer matrix $A \in \mathbb{Z}^{m \times n}$ of full row rank is unimodular if and only if the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is integral for all $\boldsymbol{b} \in \mathbb{Z}^m$.

Proof. Assume that A is unimodular and let $\boldsymbol{b} \in \mathbb{Z}^m$ be an arbitrary integer vector. First observe that, since every minimal face $\mathcal{F}$ defines an affine space $\text{aff}(\mathcal{F}) = \mathcal{F}$ and since $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \geq \boldsymbol{0}\}$ does not contain any affine spaces that consist of more than a single point, every face of $\mathcal{P}^=(A, \boldsymbol{b}) = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ contains some vertex. By Lemma 2.33, it therefore suffices to show that every vertex $\bar{\boldsymbol{x}}$ of $\mathcal{P}^=(A, \boldsymbol{b})$ is integral. Since A has full row rank, there is a basis $B \subseteq \{1, \dots, n\}$, $|B| = m$, such that $\bar{\boldsymbol{x}}_{\boldsymbol{B}} = (A_{\cdot B})^{-1}\boldsymbol{b}$ and $\bar{\boldsymbol{x}}_{\boldsymbol{N}} = \boldsymbol{0}$, where $N := \{1, \dots, n\} \setminus B$ (see *Introduction to Optimization*). By Cramer's rule, for $i \in B$ we have

$$\bar{x}_i = \frac{\det(\boldsymbol{A}_{\cdot \boldsymbol{d}_1}, \dots, \boldsymbol{A}_{\cdot \boldsymbol{d}_{i-1}}, \boldsymbol{b}, \boldsymbol{A}_{\cdot \boldsymbol{d}_{i+1}}, \dots, \boldsymbol{A}_{\cdot \boldsymbol{d}_m})}{\det A_{\cdot B}},$$

where $\{d_1, \dots, d_m\} := B$. Since A is unimodular, $\bar{\boldsymbol{x}}_{\boldsymbol{B}}$ is integral (the numerator is an integer and the denominator is ± 1) and therefore also $\bar{\boldsymbol{x}}$.

Conversely, assume that $\mathcal{P}^=(A, \boldsymbol{b})$ is integral for all integer vectors $\boldsymbol{b} \in \mathbb{Z}^m$. Let $B \subseteq \{1, \dots, n\}$, $|B| = m$ be chosen arbitrarily such that $A_{\cdot B} \in \mathbb{Z}^{m \times m}$ is invertible, i.e., $\det A_{\cdot B} \neq 0$. We need to show that $\det A_{\cdot B} \in \{-1, 1\}$.

We claim that $(A_{\cdot B})^{-1}\boldsymbol{z} \in \mathbb{Z}^m$ for all $\boldsymbol{z} \in \mathbb{Z}^m$. Note that this immediately follows from integrality of $\mathcal{P}^=(A, \boldsymbol{z})$ in case $(A_{\cdot B})^{-1}\boldsymbol{z} \geq \boldsymbol{0}$.

Applying the claim for $\boldsymbol{z} = \boldsymbol{e}^j$, $j \in \{1, \dots, m\}$, yields that $(A_{\cdot B})^{-1}\boldsymbol{e}^j \in \mathbb{Z}^m$, i.e., the j-th column of $(A_{\cdot B})^{-1}$ is integral, and thus the entire matrix $(A_{\cdot B})^{-1}$ is integral. Because then $\det(A_{\cdot B})$ and $\det((A_{\cdot B})^{-1})$ are integers, $\det(A_{\cdot B}) \cdot \det((A_{\cdot B})^{-1}) = 1$ implies that $\det(A_{\cdot B}) \in \{-1, 1\}$. Since we have chosen $A_{\cdot B}$ as an arbitrary invertible $(m \times m)$-submatrix, it follows that A is unimodular.

To show the claim, take $\boldsymbol{y} \in \mathbb{Z}^m$ such that $\boldsymbol{t} := \boldsymbol{y} + (A_{\cdot B})^{-1}\boldsymbol{z} \geq \boldsymbol{0}$ and set $\boldsymbol{b} := A_{\cdot B}\boldsymbol{t} = A_{\cdot B}\boldsymbol{y} + \boldsymbol{z} \in \mathbb{Z}^m$. Then, $\boldsymbol{x}_{\boldsymbol{B}} := (A_{\cdot B})^{-1}\boldsymbol{b} = \boldsymbol{t} \geq \boldsymbol{0}$ and $\boldsymbol{x}_{\boldsymbol{N}} := \boldsymbol{0}$ for $N := \{1, \dots, n\} \setminus B$ defines a vertex $\boldsymbol{x}$ of $\mathcal{P}^=(A, \boldsymbol{b})$ (see *Introduction to Optimization*). Since $\mathcal{P}^=(A, \boldsymbol{b})$ is integral, $\boldsymbol{x}$ and thus $\boldsymbol{t}$ are also integral, implying $(A_{\cdot B})^{-1}\boldsymbol{z} \in \mathbb{Z}^m$. □

Corollary 4.6. For every square, unimodular matrix $U \in \mathbb{Z}^{m \times m}$, we have $U^{-1} \in \mathbb{Z}^{m \times m}$, and, in particular, U^{-1} is unimodular as well.

Proof. First note that U is invertible, since it has has full (row) rank. The proof that $U^{-1}\boldsymbol{b} \in \mathbb{Z}^m$ for every $\boldsymbol{b} \in \mathbb{Z}^m$ is analogous to the proof of Theorem 4.5 via Cramer's rule applied to U (instead of $A_{\cdot B}$). Since this also applies for $\boldsymbol{b} = \boldsymbol{e}^i$ and all $i \in \{1, \dots, m\}$, we have $U^{-1} \in \mathbb{Z}^{m \times m}$. From $\det(U) \cdot \det(U^{-1}) = 1$ and $\det(U) \in \{-1, 1\}$ (since U has full row rank) it follows that $\det(U^{-1}) \in \{-1, 1\}$, hence U^{-1} is unimodular. □

Example 4.7. The inverse of the matrix of Remark 4.2 is

$$\begin{pmatrix} -4 & -9 & 5 \\ -1 & -4 & 2 \\ 6 & 15 & -8 \end{pmatrix}$$

with determinant 1. △

Corollary 4.8 ([25])**.** An integer matrix $A \in \mathbb{Z}^{m\times n}$ is totally unimodular if and only if the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq 0\}$ is integral for all integer vectors $\boldsymbol{b} \in \mathbb{Z}^m$.

Proof. Proposition 4.3 (a) states that A is totally unimodular if and only if $[A, \mathbb{I}]$ is unimodular. Furthermore, for an integer vector $\boldsymbol{b}$ it holds that $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is integral if and only if $\{\boldsymbol{z} \in \mathbb{R}^{n+m} : [A, \mathbb{I}]\boldsymbol{z} = \boldsymbol{b}, \boldsymbol{z} \geq \boldsymbol{0}\}$ is integral, since the slack variables $\boldsymbol{s} = \boldsymbol{b} - A\boldsymbol{x} \geq \boldsymbol{0}$ can be chosen integral if $\boldsymbol{x}$, $\boldsymbol{b}$ and A are integral. The application of Theorem 4.5 for the matrix $[A, \mathbb{I}]$ completes the proof. □

By simple transformations that preserve integrality and total unimodularity (by Proposition 4.3), we can extend Corollary 4.8 as follows.

Corollary 4.9. An integer matrix $A \in \mathbb{Z}^{m\times n}$ is totally unimodular if and only if the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a} \leq A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$ is integral for all integer vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}^m$ and $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{Z}^n$.

We mention a similar extension of Theorem 4.5 without proof.

Corollary 4.10. An integer matrix $A \in \mathbb{Z}^{m\times n}$ of full row rank is unimodular, if and only if the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$ is integral for all integer vectors $\boldsymbol{b} \in \mathbb{Z}^m$ and $\boldsymbol{u} \in \mathbb{Z}^n$.

Note that we need bounds on the variables to obtain a characterization.

Proposition 4.11. If an integer matrix $A \in \mathbb{Z}^{m\times n}$ is totally unimodular, then the polyhedron $\mathcal{P}(A, \boldsymbol{b})$ is integral for all integer vectors $\boldsymbol{b} \in \mathbb{Z}^m$. The converse does not hold.

Proof. Consider any minimal face[1] $\mathcal{F} := \{\boldsymbol{x} \in \mathbb{R}^n : A_{R\cdot}\boldsymbol{x} = \boldsymbol{b_R}\}$ of $\mathcal{P}(A, \boldsymbol{b})$, where $R \subseteq \{1, \ldots, m\}$ can be chosen such that $A_{R\cdot}$ has full row rank. By Lemma 2.33, it is sufficient to show that $\mathcal{F}$ contains an integer point. By rearranging the variables, we can assume that $A_{R\cdot} = [U, V]$ with U invertible. Because A is totally unimodular, so is its submatrix U, and by Corollary 4.6, U^{-1} is integral as well. Thus $\begin{pmatrix} U^{-1}\boldsymbol{b_R} \\ \boldsymbol{0} \end{pmatrix}$ is an integer point of $\mathcal{F}$.
For the second part of the statement, consider the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

This matrix is unimodular, since it has determinant -1, but not totally unimodular, since the determinant of the upper left (2×2)-submatrix is 2. For every $\boldsymbol{b} \in \mathbb{Z}^3$, $\{\boldsymbol{x} \in \mathbb{R}^3 : A\boldsymbol{x} = \boldsymbol{b}\} = \{A^{-1}\boldsymbol{b}\} \subset \mathbb{Z}^3$ is contained in every face of $\mathcal{P}(A, \boldsymbol{b})$. Thus, by Lemma 2.33, the rational polyhedron $\mathcal{P}(A, \boldsymbol{b})$ is integral for every $\boldsymbol{b} \in \mathbb{Z}^3$, but A is not totally unimodular. □

[1] Note that a minimal face is a affine space (see exercise).

Importantly, total unimodularity allows to extend strong duality to integer optimization problems.

Corollary 4.12 (integral duality)**.** If $A \in \mathbb{Z}^{m\times n}$ is totally unimodular, then, for all integer vectors $\boldsymbol{b} \in \mathbb{Z}^m$ and $\boldsymbol{c} \in \mathbb{Z}^n$, both sides of

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} \le \boldsymbol{b},\, \boldsymbol{x} \ge \boldsymbol{0}\} = \min\{\boldsymbol{b}^\top \boldsymbol{y} \,:\, A^\top \boldsymbol{y} \ge \boldsymbol{c},\, \boldsymbol{y} \ge \boldsymbol{0}\}, \quad \text{and}$$
$$\max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} \le \boldsymbol{b}\} = \min\{\boldsymbol{b}^\top \boldsymbol{y} \,:\, A^\top \boldsymbol{y} = \boldsymbol{c},\, \boldsymbol{y} \ge \boldsymbol{0}\}$$

have integral optimum solutions, provided the optima exist.

Proof. For the first version, by Corollary 4.8 and Lemma 2.33, it follows that A is totally unimodular if, for all integer vectors $\boldsymbol{c} \in \mathbb{Z}^n$ and $\boldsymbol{b} \in \mathbb{Z}^m$, the optimum of the linear program $\max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} \le \boldsymbol{b},\, \boldsymbol{x} \ge \boldsymbol{0}\}$ is attained by an integer point, if at all. Since A is totally unimodular if and only if $-A^\top$ is totally unimodular (Proposition 4.3), the same applies for the dual linear program $\min\{\boldsymbol{b}^\top \boldsymbol{y} \,:\, A^\top \boldsymbol{y} \ge \boldsymbol{c},\, \boldsymbol{y} \ge \boldsymbol{0}\}$.

For the second version, by Proposition 4.11 and Lemma 2.33, if A is totally unimodular, then, for all integer vectors $\boldsymbol{c} \in \mathbb{Z}^n$ and $\boldsymbol{b} \in \mathbb{Z}^m$, the optimum of the linear program $\max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} \le \boldsymbol{b}\}$ is attained by an integer point, if at all. For the dual LP, integrality follows from Theorem 4.5 with Lemma 2.33, because $A^\top$ is totally unimodular (Proposition 4.3). Here, by possibly omitting redundant rows of $A^\top$, we may assume that the matrix has full row rank; the resulting matrix is then unimodular. □

A very useful characterization of total unimodularity is mentioned next.

Theorem 4.13 ([18])**.** A matrix $A \in \mathbb{Z}^{m\times n}$ is totally unimodular if and only if for every subset of rows $I \subseteq \{1,\dots,m\}$ a vector $\boldsymbol{r} \in \{-1,1\}^I$ exists, with $\sum_{i\in I} r_i \boldsymbol{A}_{i\cdot} \in \{-1,0,1\}^n$.

Proof. For a proof see [39]. □

4.1.1 Applications in combinatorial optimization

Total unimodularity yields an different perspective on the max-flow min-cut theorem (see *Algorithmic Discrete Mathematics*).

Theorem 4.14. Let $((V,E),\mu,s,t)$ be a flow network with integer capacities μ. Then, an integral maximum s-t-flow $f^\star$ exists and its value is equal to the minimum capacity over all s-t-cuts, i.e.,

$$|f^\star| = \min_{S\subseteq V\setminus\{s,t\}} \sum_{e\in\delta^+(S\cup\{s\})} \mu(e).$$

Proof. Let A be the vertex-arc incidence matrix of the directed graph $G = (V, E\cup\{(t,s)\})$ with $\mu((t,s)) := \infty$, and define $\boldsymbol{\mu} \in \mathbb{Z}^E_{\ge 0}$ by $\mu_e := \mu(e)$. Every s-t-flow in (V,E) corresponds to a circulation in G, i.e., a flow satisfying flow conservation also at s and t (see *Combinatorial Optimization*). Accordingly, a maximum s-t-flow is defined by the solution of the linear program

$$\max\{x_{(t,s)} \,:\, A\boldsymbol{x} = \boldsymbol{0},\, \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{\mu}\} = \max\{x_{(t,s)} \,:\, \begin{pmatrix} A \\ -A \\ \mathbb{I} \end{pmatrix}\boldsymbol{x} \le \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{\mu} \end{pmatrix}, \boldsymbol{x} \ge \boldsymbol{0}\}.$$

This LP is feasible (e.g., $\boldsymbol{x} = \mathbf{0}$) and bounded by $x_{(t,s)} = \sum_{e\in\delta^-(t)} x_e \leq \sum_{e\in\delta^-(t)} \mu_e$. By Propositions 4.4 and 4.3 and Corollary 4.12, the LP and its dual

$$\min\{\boldsymbol{\mu}^\top \boldsymbol{y} \,:\, A^\top \boldsymbol{z}^+ - A^\top \boldsymbol{z}^- + \boldsymbol{y} \geq \mathbf{e}^{(t,s)};\, \boldsymbol{y}, \boldsymbol{z}^+, \boldsymbol{z}^- \geq \mathbf{0}\} = \min\{\boldsymbol{\mu}^\top \boldsymbol{y} \,:\, A^\top \boldsymbol{z} + \boldsymbol{y} \geq \mathbf{e}^{(t,s)}, \boldsymbol{y} \geq \mathbf{0}\}$$

therefore have integral optimum solutions $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{y}}$ and $\bar{\boldsymbol{z}}$. In other words, the value $\bar{x}_{(t,s)}$ of a maximum s-t-flow is equal to the minimum value $\boldsymbol{\mu}^\top \bar{\boldsymbol{y}}$, where $\bar{\boldsymbol{y}}$ fulfills

$$\begin{aligned} \bar{z}_v - \bar{z}_w + \bar{y}_{(v,w)} &\geq 0, \quad \text{for } (v,w) \in E \setminus \{(t,s)\} \\ \bar{z}_t - \bar{z}_s + \bar{y}_{(t,s)} &\geq 1, \\ \bar{\boldsymbol{y}} &\geq \mathbf{0}. \end{aligned}$$

By weak complementary slackness (see *Introduction to Optimization*), and since $\mu_{(t,s)} = \infty$, we conclude that $\bar{y}_{(t,s)} = 0$, and therefore $\bar{z}_t \geq 1 + \bar{z}_s$. Since $s \notin W := \{w \in V \,:\, \bar{z}_w \geq \bar{z}_t\}$ and $t \in W$, it follows that $\delta^-(W)$ is an s-t-cut. Furthermore, $(v,w) \in \delta^-(W)$ implies $\bar{z}_w \geq \bar{z}_t > \bar{z}_v$, and thus

$$\bar{y}_{(v,w)} > \bar{z}_v - \bar{z}_w + \bar{y}_{(v,w)} \geq 0 \qquad \forall (v,w) \in \delta^-(W)$$

yields, together with integrality of $\bar{\boldsymbol{y}}$, that $\bar{\boldsymbol{y}}_{\delta^-(W)} \geq \mathbf{1}$. From this, and $\bar{\boldsymbol{y}}, \boldsymbol{\mu} \geq \mathbf{0}$, we conclude

$$\bar{x}_{(t,s)} = \boldsymbol{\mu}^\top \bar{\boldsymbol{y}} \geq \sum_{e\in\delta^-(W)} \mu_e.$$

In other words, there is an s-t-cut $\delta^-(W)$ whose capacity is not larger than the maximum flow value. Since there there cannot be a cut of capacity smaller than the maximum flow value (see *Algorithmic Discrete Mathematics*), the theorem follows. □

Totally unimodular matrices also naturally occur for undirected graphs. Let $G = (V, E)$ be an undirected graph and let $A \in \{0,1\}^{E\times V}$ be its edge-vertex incidence matrix. In the following, we investigate sufficient conditions for the polyhedron $\mathcal{P} = \{\boldsymbol{x} \,:\, A\boldsymbol{x} \leq \boldsymbol{b},\, \boldsymbol{x} \geq \mathbf{0}\}$ to be integral. This is interesting, e.g., because for the case $\boldsymbol{b} = \mathbf{1}$ the integral points of $\mathcal{P}$ correspond to stable sets in G (i.e., a subset S of the vertices, with $e \not\subseteq S$ for all $e \in E$) or, more generally, feasible solutions of the set packing problem from Example 1.7 (with ground set $U = E$ and subsets $S_v = \delta(v)$ for all $v \in V$).

Theorem 4.15. The edge-vertex incidence matrix A of an undirected graph G is totally unimodular if and only if G is bipartite.

Proof. First assume that G is not bipartite. Then, G contains a cycle $C \subseteq E$ of odd length. We may assume that C does not contain any chords, i.e., edges between vertices not adjacent along the cycle, since every odd cycle with a chord contains a smaller odd cycle. The submatrix of A that belongs to the edges $E(C)$ and vertices $V(C)$ of C is (after suitable permutation of rows and columns)

$$M := \begin{pmatrix} 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \in \{0,1\}^{|C|\times|C|}.$$

To see that this matrix is not totally unimodular, consider the Leibniz formula for the determinant

$$\det(M) = \sum_{\sigma \in S_{|C|}} \left(\operatorname{sgn}(\sigma) \cdot \prod_{i=1}^{|C|} M_{i\sigma(i)} \right),$$

where $S_{|C|}$ denotes the symmetric group of all permutations of $\{1, \ldots, |C|\}$. There are exactly two permutations with non-vanishing contributions to the sum, namely the identity (diagonal) and the cyclic map $\sigma(1) = 2$, $\sigma(2) = 3$, $\ldots$, $\sigma(|C|) = 1$. The number of inversions of the latter permutation is $|C| - 1$, and therefore it has a positive sign since $|C|$ is odd. Hence, the determinant of M is 2 and, consequently, A is not totally unimodular.

Now, conversely, assume that G is bipartite. We let $\{V_1, V_2\} = V$ denote the corresponding partition of the vertex set. We arbitrarily fix $\boldsymbol{b} \in \mathbb{Z}^m$ and $\boldsymbol{c} \in \mathbb{R}^n$ and show that the linear program $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ has an integral solution, provided it is bounded. This is sufficient because of Lemma 2.33 (d) and Corollary 4.8.

Let $\boldsymbol{x}^\star$ be an optimum solution of the linear problem $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$. Let $\tau \sim \mathcal{U}(0,1)$ be drawn uniformly at random from the interval $(0,1)$. For $v \in V$, define $\boldsymbol{z} \in \mathbb{Z}^V_{\geq 0}$ via

$$z_v := \begin{cases} \lfloor x^\star_v \rfloor + 1 & \text{if } v \in V_1 \text{ and } x^\star_v - \lfloor x^\star_v \rfloor \geq \tau, \\ \lfloor x^\star_v \rfloor + 1 & \text{if } v \in V_2 \text{ and } x^\star_v - \lfloor x^\star_v \rfloor > 1 - \tau, \\ \lfloor x^\star_v \rfloor & \text{otherwise.} \end{cases}$$

We claim that $\boldsymbol{z}$ is feasible, i.e., $A\boldsymbol{z} \leq \boldsymbol{b}$. To see this, consider $e = \{u, v\} \in E$ with $u \in V_1$ and $v \in V_2$ and the corresponding row $\boldsymbol{A_{e\cdot}} = (\mathbf{e^u} + \mathbf{e^v})^\top$ of A. We need to show that $\boldsymbol{z}$ satisfies $\boldsymbol{A_{e\cdot}}\boldsymbol{z} \leq b_e$, i.e., that $z_u + z_v \leq b_e$. We distinguish three cases.

- If $\lfloor x^\star_u \rfloor + \lfloor x^\star_v \rfloor = b_e$, then $x^\star_u, x^\star_v \in \mathbb{Z}$ and $z_u = \lfloor x^\star_u \rfloor$ and $z_v = \lfloor x^\star_v \rfloor$ (because $0 < \tau < 1$), hence $z_u + z_v = b_e$.
- If $\lfloor x^\star_u \rfloor + \lfloor x^\star_v \rfloor \leq b_e - 2$, then $\lfloor x^\star_u \rfloor + 1 + \lfloor x^\star_v \rfloor + 1 \leq b_e$ and therefore $z_u + z_v \leq b_e$.
- Otherwise, $\lfloor x^\star_u \rfloor + \lfloor x^\star_v \rfloor = b_e - 1$ holds. Then, $x^\star_u - \lfloor x^\star_u \rfloor + x^\star_v - \lfloor x^\star_v \rfloor = x^\star_u + x^\star_v - (b_e - 1) \leq 1$. If $x^\star_u - \lfloor x^\star_u \rfloor \geq \tau$, then $x^\star_v - \lfloor x^\star_v \rfloor \leq 1 - \tau$ and vice versa. So not both values $x^\star_u$ and $x^\star_v$ are rounded up, resulting in $z_u + z_v \leq b_e$.

Now, for $u \in V_1$ we obtain $\Pr_\tau[z_u = \lfloor x^\star_u \rfloor + 1] = \Pr_\tau[\tau \leq x^\star_u - \lfloor x^\star_u \rfloor] = x^\star_u - \lfloor x^\star_u \rfloor$ and for $v \in V_2$ that $\Pr_\tau[z_v = \lfloor x^\star_v \rfloor + 1] = \Pr_\tau[1 - \tau < x^\star_v - \lfloor x^\star_v \rfloor] = 1 - \Pr_\tau[1 - \tau \geq x^\star_v - \lfloor x^\star_v \rfloor] = 1 - \Pr_\tau[U \leq 1 - (x^\star_v - \lfloor x^\star_v \rfloor)] = 1 - (1 - (x^\star_v - \lfloor x^\star_v \rfloor)) = x^\star_v - \lfloor x^\star_v \rfloor$. Overall, we have shown

$$\Pr_\tau[z_v = \lfloor x^\star_v \rfloor + 1] = x^\star_v - \lfloor x^\star_v \rfloor \qquad \forall v \in V.$$

Let c^{IP} denote the largest objective function value of a feasible integer vector with respect to c. By feasibility of $\boldsymbol{z} \in \mathbb{Z}^V$ we obtain for the expectation that

$$\begin{aligned} \boldsymbol{c}^\top \boldsymbol{x}^\star \geq c^{\mathrm{IP}} \geq \mathbb{E}_\tau[\boldsymbol{c}^\top \boldsymbol{z}] &= \sum_{v \in V} c_v \lfloor x^\star_v \rfloor + \sum_{v \in V} c_v \Pr_\tau[z_v = \lfloor x^\star_v \rfloor + 1] \\ &= \sum_{v \in V} c_v \lfloor x^\star_v \rfloor + \sum_{v \in V} c_v (x^\star_v - \lfloor x^\star_v \rfloor) = \boldsymbol{c}^\top \boldsymbol{x}^\star. \end{aligned}$$

It follows that $\boldsymbol{c}^\top \boldsymbol{x}^\star = c^{\mathrm{IP}}$. Thus, by Lemma 2.33 (d), the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is integral. Corollary 4.8 implies that A is totally unimodular. □

Remark 4.16. The technique used in the second half of the proof of Theorem 4.15 is often called *randomized rounding*.

Corollary 4.17. Consider the assignment problem from Example 1.5. The LP-relaxation of the binary program (obtained by replacing $x_{ij} \in \{0,1\}$ by $x_{ij} \in [0,1]$) always has an integral optimum solution.

Proof. Let A be the edge-vertex incidence matrix of the bipartite graph in Example 1.5. The LP relaxation of the assignment problem can be formulated as

$$\min\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A^\top \boldsymbol{x} = \mathbf{1}, \boldsymbol{x} \geq \mathbf{0}\},$$

where the constraint $\boldsymbol{x} \leq \mathbf{1}$ is implied. According to Theorem 4.15, Proposition 4.3 (third and fifth item), and Corollary 4.8, this LP has an integral optimum solution. □

4.2 The Hermite normal form

In this section we show that every rational matrix can be brought into so-called Hermite normal form. This form will help us to prove an integer analogue of the Farkas lemma. We will later need this lemma to show the integrality of polyhedra.

Definition 4.18. A matrix with full row rank is in *Hermite normal form* if it has the form $[B, 0]$, where B is a non-negative lower triangular matrix and each row has a unique maximum entry that is on the main diagonal.

Note that $[B, 0]$ being in Hermite normal form implies that B is invertible, since it is lower triangular and its determinant thus is the product of its strictly positive diagonal entries. Crucially, we can bring matrices into this form without disturbing too much the structural properties of matrices that we aim to establish. It suffices to allow the following operations.

Definition 4.19. The *(elementary) unimodular column operations* on a matrix are

- swapping two columns,
- multiplication of a column by -1,
- addition of an integer multiple of a column to another column.

The name of these operations is justified because they can be encoded as multiplications with unimodular matrices.

Observation 4.20. The unimodular column operations of Definition 4.19 can be performed by multiplying the following unimodular matrices from the right:

$$\begin{pmatrix} 1 & & & \\ & 0 & 1 & \\ & 1 & 0 & \\ & & & 1 \end{pmatrix}, \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & & 1 \end{pmatrix}, \begin{pmatrix} 1 & & & \\ & 1 & 0 & \\ & k & 1 & \\ & & & 1 \end{pmatrix}.$$

With this, we can show how to bring matrices into the desired form.

Theorem 4.21. Every matrix $A \in \mathbb{Q}^{m \times n}$ of full row rank can be converted to Hermite normal form using unimodular column operations.

Proof. Let A be a rational matrix of full row rank. We may assume that $A \in \mathbb{Z}^{m\times n}$ is integral: Otherwise, we scale the matrix by a suitable positive factor before carrying out the following operations and by its reciprocal afterwards. This is possible, since all unimodular column operations commute with scalar multiplication.

We first show that A can be transformed into a matrix $[B,0]$, where B is a lower triangular matrix and $B_{ii} > 0$ for all $i \in \{1,\dots,m\}$. Suppose we have already transformed A into the form

$$\begin{pmatrix} B & 0 \\ C & D \end{pmatrix},$$

where B is a lower triangular matrix with a positive diagonal. Using unimodular column operations, we establish $D_{11} \geq D_{12} \geq \cdots \geq D_{1k} \geq 0$ for the first row of $D \in \mathbb{Z}^{\ell\times k}$ by multiplying columns by -1 and then reordering them. Iterative subtraction of the j-th column from the i-th column for $i < j$ and reordering can be used to bring the first row of D into non-decreasing order in such a way that, additionally, $\sum_{i=1}^{k} D_{1i}$ cannot further be reduced.

Since A has full row rank, we have $D_{11} > 0$. We claim that $D_{1i} = 0$ for $i \in \{2,\dots,k\}$. Otherwise, $D_{12} > 0$. If we subtract the second column from the first one and rearrange these columns if needed, we maintain that the first row of D is in non-increasing order, but we decreased $\sum_{i=1}^{k} D_{1i}$, in contradiction its minimality. Overall, we have brought one more row of the matrix into the desired form.

After we have performed this procedure m times, we have transformed A into a matrix $[B,0]$, where B is a lower triangular triangular matrix with a positive main diagonal.

In order to reach Hermite normal form, we need to establish $0 \leq B_{ij} < B_{ii}$ for every $j < i$. This can be achieved by considering for $i \leftarrow 2,\dots,m$ every $j \in \{1,\dots,i-1\}$ and adding an integer multiple (positive if $B_{ij} < 0$, negative if $B_{ij} \geq B_{ii}$) of the i-th column to the j-th column, such that $B_{ij} \in \{1,\dots,B_{ii}-1\}$. In this way, the entries above row i are not affected and we achieve Hermite normal form. □

Example 4.22. Consider the matrix

$$A = \begin{pmatrix} 4 & 8 & 0 & -4 & 4 \\ 3 & -6 & 6 & 9 & 0 \\ 2 & -2 & 0 & 4 & -4 \end{pmatrix}.$$

Multiplication of the first column by -2, 1, -1 and addition to the columns $2,4,5$ yields

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 3 & -12 & 6 & 12 & -3 \\ 2 & -6 & 0 & 6 & -6 \end{pmatrix}.$$

Negating and swapping the last column forward yields

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 3 & 3 & -12 & 6 & 12 \\ 2 & 6 & -6 & 0 & 6 \end{pmatrix}.$$

Addition of the third column to column 5 and multiplication of the second column by 4 and -2 and addition to columns 3 and 4 yields

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 \\ 2 & 6 & 18 & -12 & 0 \end{pmatrix}.$$

Addition of the fourth column to column 3 yields

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 \\ 2 & 6 & 6 & -12 & 0 \end{pmatrix}.$$

Addition of twice the third column to column 4 yields

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 \\ 2 & 6 & 6 & 0 & 0 \end{pmatrix}.$$

Subtraction of the second column from first column yields

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ -4 & 6 & 6 & 0 & 0 \end{pmatrix}.$$

Finally, subtraction of the third column from column 2 and addition to column 1 yields a matrix in Hermite normal form:

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 2 & 0 & 6 & 0 & 0 \end{pmatrix}.$$

△

The product of the unimodular matrices of Observation 4.20 involved in the proof of Theorem 4.21 results in a unimodular matrix U, because of the multiplicative property $\det(AB) = \det(A) \cdot \det(B)$ of the determinant.

Corollary 4.23. For every matrix $A \in \mathbb{Q}^{m \times n}$ of full row rank, there is a unimodular matrix $U \in \mathbb{Z}^{n \times n}$ such that

$$[B, 0] = AU,$$

is in Hermite normal form and is integral if A is.

Remark 4.24. For every rational matrix of full row rank, the Hermite normal form is unique. (U is unique if A is also regular). Furthermore, the Hermite normal form can be computed in polynomial time (see [39]).

We are now ready to prove an integer analogue of the Farkas lemma.

Theorem 4.25. For $A \in \mathbb{Q}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{Q}^m$ exactly one of the systems

$$\begin{array}{rcl} A\boldsymbol{x} = \boldsymbol{b}, & & \boldsymbol{y}^\top A \in \mathbb{Z}^n \\ \boldsymbol{x} \in \mathbb{Z}^n & \text{or} & \boldsymbol{y}^\top \boldsymbol{b} \notin \mathbb{Z} \\ & & \boldsymbol{y} \in \mathbb{Q}^m \end{array}$$

has a solution.

Proof. Both systems of equations cannot have a solution at the same time, because otherwise integrality of $\boldsymbol{x}$ and $\boldsymbol{y}^\top A$ would imply integrality of $\boldsymbol{y}^\top \boldsymbol{b} = \boldsymbol{y}^\top A\boldsymbol{x}$.

Assume that the second system has no solution, i.e., for all $\boldsymbol{y} \in \mathbb{Q}^m$ for which $\boldsymbol{y}^\top A$ is integral, $\boldsymbol{y}^\top \boldsymbol{b}$ is also integral. We have to show that $A\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \in \mathbb{Z}^n$ has a solution.

First, observe that $A\boldsymbol{x} = \boldsymbol{b}$ is (fractionally) feasible, since otherwise a rational vector $\boldsymbol{y} \in \mathbb{Q}^m$ with $\boldsymbol{y}^\top A = \boldsymbol{0}$ and $\boldsymbol{y}^\top \boldsymbol{b} \neq 0$ would exist by Gaussian elimination. After appropriate rescaling, there would then be a $\boldsymbol{y}$ with $\boldsymbol{y}^\top A = 0$ and $\boldsymbol{y}^\top \boldsymbol{b} = \frac{1}{2}$ – a contradiction to our assumption.

We may therefore assume that A has full row rank (after eliminating redundant rows from $A\boldsymbol{x} = \boldsymbol{b}$). According to Corollary 4.23, there is a unimodular matrix $U \in \mathbb{Z}^{n\times n}$ with $AU = [B, 0]$ is in Hermite normal form. Since $B^{-1}AU = B^{-1}[B, 0] = [\mathbb{I}, 0] \in \mathbb{Z}^{m\times n}$ and, by Corollary 4.6, $U^{-1} \in \mathbb{Z}^{n\times n}$, we have $B^{-1}A \in \mathbb{Z}^{m\times n}$. It follows from our assumption (with $\boldsymbol{y} = (B^{-1})_{i\cdot}^\top$ for $i \in \{1, \dots, m\}$) that $B^{-1}\boldsymbol{b} \in \mathbb{Z}^m$. Hence, $\hat{\boldsymbol{x}} := U\binom{B^{-1}\boldsymbol{b}}{\boldsymbol{0}} \in \mathbb{Z}^n$ is integral. Because of

$$A\hat{\boldsymbol{x}} = AU\binom{B^{-1}\boldsymbol{b}}{\boldsymbol{0}} = [B, 0]\binom{B^{-1}\boldsymbol{b}}{\boldsymbol{0}} = \boldsymbol{b},$$

we obtain that $\hat{\boldsymbol{x}}$ is an integral solution of $A\boldsymbol{x} = \boldsymbol{b}$. □

Corollary 4.26. We can in polynomial time find an integer solution of $A\boldsymbol{x} = \boldsymbol{b}$ with $A \in \mathbb{Q}^{m\times n}$, $\boldsymbol{b} \in \mathbb{Q}^m$ or a certificate that none exists.

Proof. We first solve $A\boldsymbol{x} = \boldsymbol{b}$ fractionally via Gaussian elimination. If no solution exists, we obtain a vector $\boldsymbol{y}$ with $\boldsymbol{y}^\top A = 0$ and $\boldsymbol{y}^\top \boldsymbol{b} = \frac{1}{2}$, that certifies that no integral solution exists according to Theorem 4.25. If a solution exists, we eliminate redundant rows and compute the Hermite normal form $[B, 0]$ in polynomial time via Remark 4.24, and invert B via Gaussian elimination. Now, either $\hat{\boldsymbol{x}} = U\binom{B^{-1}\boldsymbol{b}}{\boldsymbol{0}}$ is an integer solution of $A\boldsymbol{x} = \boldsymbol{b}$ or a certificate $\boldsymbol{y} = (B^{-1})_{i\cdot}^\top$ with $(B^{-1}\boldsymbol{b})_i \notin \mathbb{Z}$ for infeasibility has been found according to Theorem 4.25. □

Remark 4.27. Corollary 4.26 only holds for systems of equations of the form $A\boldsymbol{x} = \boldsymbol{b}$. If, additionally, $\boldsymbol{x}$ is constrained, i.e., via $\boldsymbol{x} \in \{0, 1\}^n$, the hardness result of Theorem 3.1 carries over.

4.3 Total dual integrality

In Section 4.1, we saw that total unimodularity of an integer matrix A ensures that, *for every* integral right-hand side $\boldsymbol{b}$, the polyhedron $\mathcal{P}_\mathrm{I} = \mathrm{conv}(\{\boldsymbol{x} \in \mathbb{Z}^n : A\boldsymbol{x} \leq \boldsymbol{b}\})$ is completely described by the original inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$. If, on the other hand, we decide on a particular right-hand side $\boldsymbol{b}$, then TDI (total dual integrality) is the right concept to guarantee integrality of a polyhedron.

Definition 4.28. The system of inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ with $A \in \mathbb{Q}^{m\times n}$, $\boldsymbol{b} \in \mathbb{Q}^m$ is called *totally dual integral (TDI)* if, for every integer vector $\boldsymbol{c} \in \mathbb{Z}^n$, there is an integral optimum solution of the LP $\min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\}$ if an optimum exists.

The TDI property of a system $A\boldsymbol{x} \leq \boldsymbol{b}$ has a geometric interpretation: Let $\boldsymbol{c}$ be an integer vector in the cone spanned by the rows of A, i.e., there is $\boldsymbol{y} \geq \boldsymbol{0}$ with $A^\top \boldsymbol{y} = \boldsymbol{c}$. Among all the possibilities to express $\boldsymbol{c}$ as a conic combination of the rows of A, let $\mathcal{S}$ be the set minimum conic combinations with respect to $\boldsymbol{b}$, i.e.,

$$\mathcal{S} := \arg\min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\}.$$

Then, $A\boldsymbol{x} \leq \boldsymbol{b}$ is TDI if and only if $\mathcal{S}$ contains an integer vector. In other words, this means that among all shortest ways to express $\boldsymbol{c}$ as a conic combination of the rows of A, there is one that is an integer. We will later express this interpretation in terms of Hilbert bases, which play an important role in integer programming.

Total dual integrality is directly connected to total unimodularity as in Corollary 4.12, but even for $\boldsymbol{b} \in \mathbb{Q}^m$.

Observation 4.29. If $A \in \mathbb{Z}^{m \times n}$ is totally unimodular, then $Ax \leq b$ is TDI for all $b \in Q^m$.

Proof. If A is totally unimodular, by Corollary 4.8, we have that

$$\{y \in \mathbb{R}^m : A^\top y = c, y \geq 0\} = \{y \in \mathbb{R}^m : \begin{pmatrix} A^\top \\ -A^\top \end{pmatrix} y \leq \begin{pmatrix} c \\ -c \end{pmatrix}, y \geq 0\}$$

is integral for all $c \in \mathbb{Z}^n$. Lemma 2.33 then implies that

$$\min\{b^\top y : A^\top y = c, y \geq 0\} = -\max\{-b^\top y : A^\top y = c, y \geq 0\}$$

has an integral optimum solution for all $b \in \mathbb{R}^m$ for which an optimum exists. By definition, $Ax \leq b$ is TDI. □

Note also that TDI is a property of a system of inequalities and not a property of the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leq b\}$. We will see that there are many ways to describe $\mathcal{P}$, but there are only a few that also possess the TDI property. To obtain such a description, it may be necessary to add many redundant inequalities to the initial formulation.

We now establish a connection between $Ax \leq b$ being TDI and the integrality of the corresponding polyhedron $\mathcal{P}(A, b)$. We first strengthen Lemma 2.33.

Lemma 4.30. If $\max\{c^\top x : Ax \leq b\}$ is an integer for all $c \in \mathbb{Z}^n$ for which it is finite, then $\mathcal{P}(A, b)$ is integral.

Proof. Let $\mathcal{F} \neq \emptyset$ be a minimal face of $\mathcal{P}(A, b)$. By Lemma 2.33, it suffices to show that $\mathcal{F}$ contains integral points. Since $\mathcal{F}$ is minimal, it follows that $\mathcal{F} = \{x \in \mathcal{P}(A, b) : A_{\mathrm{eq}(\mathcal{F})\cdot} x = b_{\mathbf{eq}(\mathcal{F})}\} = \{x \in \mathbb{R}^n : A_{\mathrm{eq}(\mathcal{F})\cdot} x = b_{\mathbf{eq}(\mathcal{F})}\}$.

If $\mathcal{F}$ does not contain any integral points, then, by Theorem 4.25 there exists $y \in \mathbb{Q}^{\mathrm{eq}(\mathcal{F})}$ with $c^\top := y^\top A_{\mathrm{eq}(\mathcal{F})\cdot} \in \mathbb{Z}^n$ and $\gamma := y^\top b_{\mathbf{eq}(\mathcal{F})} \notin \mathbb{Z}$. We may assume that y is non-negative, otherwise choose $s \in \mathbb{Z}^{\mathrm{eq}(\mathcal{F})}_{\geq 0}$ large enough so that $y + s \geq 0$, $(y + s)^\top A_{\mathrm{eq}(\mathcal{F})\cdot} \in \mathbb{Z}^n$ and $(y + s)^\top b_{\mathbf{eq}(\mathcal{F})} \notin \mathbb{Z}$, and use $y' := y + s$ instead of y. We claim that $\max\{c^\top x : Ax \leq b\}$ exists and is assumed by every $\hat{x} \in \mathcal{F}$. This is the case since $\mathcal{F} \neq \emptyset$ and, for all $x \in \mathcal{P}(A, b)$ it holds that

$$c^\top x = y^\top A_{\mathrm{eq}(\mathcal{F})\cdot} x \overset{y \geq 0}{\leq} y^\top b_{\mathbf{eq}(\mathcal{F})} = y^\top A_{\mathrm{eq}(\mathcal{F})\cdot} \hat{x} = c^\top \hat{x}.$$

It also follows that $\gamma = y^\top b_{\mathbf{eq}(\mathcal{F})} = \max\{c^\top x : Ax \leq b\}$. By assumption, the right-hand side is an integer, which contradicts our choice of γ. Therefore, $\mathcal{F}$ contains integral points. □

We can now weaken the condition of Prop 4.11 for the case where we have a specific right-hand side $b \in \mathbb{Z}^m$.

Theorem 4.31. If $Ax \leq b$ is TDI and $b \in \mathbb{Z}^m$, then $\mathcal{P}(A, b)$ is integral.

Proof. By definition of $Ax \leq b$ being TDI, and since b is an integer vector, the value $\min\{b^\top y : A^\top y = c, y \geq 0\}$ is an integer for all $c \in \mathbb{Z}^n$ for which it is finite. Due to strong duality (see *Introduction to Optimization*), this number coincides with $\max\{c^\top x : Ax \leq b\}$. The statement thus follows from Lemma 4.30. □

We have already hinted at the connection between the geometric interpretation of TDI and Hilbert bases. We now make this precise.

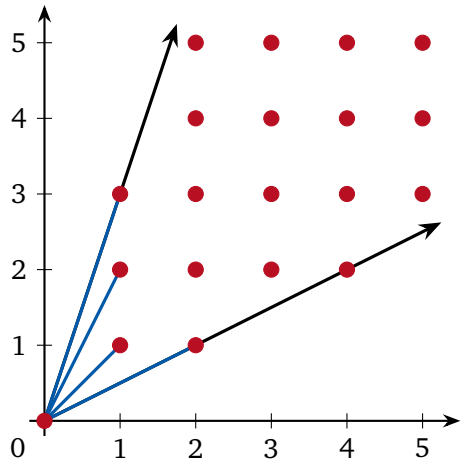

Figure 4.1: Hilbert basis of a polyhedral cone.

Definition 4.32. A *Hilbert* basis of a rational polyhedral cone $C = \mathcal{P}(A, \mathbf{0}) \subseteq \mathbb{R}^n$ is a finite subset $\mathcal{H} \subseteq C$ such that every $\boldsymbol{z} \in C \cap \mathbb{Z}^n$ is a non-negative integral linear combination of elements from $\mathcal{H}$, i.e.,

$$\boldsymbol{z} = \sum_{\boldsymbol{h} \in \mathcal{H}} \lambda(\boldsymbol{h})\, \boldsymbol{h} \quad \text{with } \lambda \colon \mathcal{H} \to \mathbb{Z}_{\geq 0}.$$

The Hilbert basis $\mathcal{H}$ is *integral* if $\mathcal{H} \subseteq \mathbb{Z}^n$.

Example 4.33. Consider $C = \operatorname{cone}(\{\binom{1}{3}, \binom{2}{1}\})$ (see Figure 4.1). An integer Hilbert basis of C is given by $\mathcal{H} = \{\binom{1}{3}, \binom{1}{2}, \binom{1}{1}, \binom{2}{1}\}$. △

For proofs of the following facts see [39, Chapter 16.4].

Remark 4.34.

(a) For every rational polyhedral cone there exists an integral Hilbert basis.

(b) If the cone has a vertex, the minimal integral Hilbert basis is uniquely determined.

We derive a characterization for $A\boldsymbol{x} \leq \boldsymbol{b}$ to be TDI.

Theorem 4.35. Let $A \in \mathbb{Q}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Q}^m$ and $\mathcal{P}(A, \boldsymbol{b}) \neq \emptyset$. Then, the system of inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ is TDI, if and only if the rows of $A_{\text{eq}(\mathcal{F})\cdot}$ form a Hilbert basis of $\operatorname{cone}((A_{\text{eq}(\mathcal{F})\cdot})^\top)$ for every (minimal) face $\mathcal{F}$ of $\mathcal{P}(A, \boldsymbol{b})$.

To understand the proof of Theorem 4.35, the following geometric interpretation may be helpful. Let $\boldsymbol{c} \in \operatorname{cone}(A^\top)$ be an integer vector such that $\min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \mathbf{0}\}$ is finite. Then, we know that $\boldsymbol{c}^\top \boldsymbol{x}$ assumes its optimum on a face of $\mathcal{P}$. By complementary slackness (see *Introduction to Optimization*), it follows that if $\boldsymbol{c}^\top \boldsymbol{x}$ assumes the optimum on face $\mathcal{F}$, then $\boldsymbol{c}$ lies in the cone spanned by the rows of $A_{\text{eq}(\mathcal{F})\cdot}$. The condition that the rows induced by $\mathcal{F}$ form a Hilbert basis is equivalent to the fact that $\boldsymbol{c}$ is a non-negative integer combination of the rows induced by $\mathcal{F}$. Formally, there is an integer vector $\boldsymbol{y}^\star \in \mathbb{Z}^m$, $A^\top \boldsymbol{y}^\star = \boldsymbol{c}$, $\boldsymbol{y}^\star \geq \mathbf{0}$ with $\boldsymbol{b}^\top \boldsymbol{y}^\star = \min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \mathbf{0}\}$. This corresponds exactly to the TDI property of the system $A\boldsymbol{x} \leq \boldsymbol{b}$.

Proof. Let $A\boldsymbol{x} \leq \boldsymbol{b}$ be TDI. Further, let $\mathcal{F} \neq \emptyset$ be a face of $\mathcal{P} := \mathcal{P}(A, \boldsymbol{b})$ and $\boldsymbol{c} \in \operatorname{cone}((A_{\text{eq}(\mathcal{F})\cdot})^\top) \cap \mathbb{Z}^n$. We need to show that $\boldsymbol{c}$ is a non-negative integer combination of the row vectors of $A_{\text{eq}(\mathcal{F})\cdot}$. By assumption, $\boldsymbol{c}$ is a

non-negative linear combination of the row vectors of $A_{\mathrm{eq}(\mathcal{F}).}$, i.e., there is $\hat{\boldsymbol{y}} \geq \boldsymbol{0}$ with $\hat{y}_i = 0$ for all $i \notin \mathrm{eq}(\mathcal{F})$ and $\boldsymbol{c}^\top = \hat{\boldsymbol{y}}^\top A$.

We now claim that $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\}$ exists and is assumed for every $\hat{\boldsymbol{x}} \in F$. This follows since $\mathcal{F} \neq \emptyset$ and, for all $\boldsymbol{x} \in \mathcal{P}$ and $\hat{\boldsymbol{x}} \in \mathcal{F}$, it holds that

$$\boldsymbol{c}^\top \boldsymbol{x} = \hat{\boldsymbol{y}}^\top A\boldsymbol{x} \leq \hat{\boldsymbol{y}}^\top \boldsymbol{b} = \hat{\boldsymbol{y}}_{\mathbf{eq}(\boldsymbol{\mathcal{F}})}{}^\top \boldsymbol{b}_{\mathbf{eq}(\boldsymbol{\mathcal{F}})} = \hat{\boldsymbol{y}}_{\mathbf{eq}(\boldsymbol{\mathcal{F}})}{}^\top A_{\mathrm{eq}(\mathcal{F}).}\hat{\boldsymbol{x}} = \hat{\boldsymbol{y}}^\top A\hat{\boldsymbol{x}} = \boldsymbol{c}^\top \hat{\boldsymbol{x}},$$

because $\hat{y}_i = 0$ for all $i \notin \mathrm{eq}(\mathcal{F})$. By strong duality (see *Introduction to Optimization*), we conclude that the value $\min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\}$ is finite. Using that $A\boldsymbol{x} \leq \boldsymbol{b}$ is TDI, we obtain

$$\min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \in \mathbb{Z}^m, \boldsymbol{y} \geq \boldsymbol{0}\} = \max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\}.$$

Let $\boldsymbol{y}^\star$ be an integer optimum solution of $\min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\}$. For every $\hat{\boldsymbol{x}} \in \mathcal{F}$ we have $\boldsymbol{A_i}.\hat{\boldsymbol{x}} = b_i$ for all $i \in \mathrm{eq}(\mathcal{F})$ and, by minimality of $\mathcal{F}$, $\boldsymbol{A_i}.\hat{\boldsymbol{x}} < b_i$ for all $i \notin \mathrm{eq}(\mathcal{F})$. Since $\hat{\boldsymbol{x}} \in \mathcal{F}$ is an optimal solution of $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\}$, it follows from weak complementary slackness that $y_i^\star = 0$ for all $i \notin \mathrm{eq}(\mathcal{F})$. This completes the first part of the proof, since $\boldsymbol{y}^\star$ is integral, $y_i^\star = 0$ for all $i \notin \mathrm{eq}(\mathcal{F})$, $y_i^\star \geq 0$ for all $i \in \mathrm{eq}(\mathcal{F})$, and $\boldsymbol{c}$ is a non-negative integer combination of the rows of $A_{\mathrm{eq}(\mathcal{F}).}$.

Now, let $\boldsymbol{c} \in \mathbb{Z}^n$ such that $\gamma := \min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\}$ exists. By strong duality, we know that $\gamma = \max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\}$ exists. Let $\mathcal{F}$ be a minimal face of $\mathcal{P}$ on which the optimum value of $\boldsymbol{c}^\top \boldsymbol{x}$ is attained.

Note that $\boldsymbol{c} \in \mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}).})^\top)$: Due to weak complementary slackness, there is a vector $\boldsymbol{y}^\star \geq \boldsymbol{0}$ with $y_i^\star = 0$ for all $i \notin \mathrm{eq}(\mathcal{F})$ and $\boldsymbol{c}^\top = \sum_{i \in \mathrm{eq}(\mathcal{F})} \boldsymbol{A_i}.y_i^\star$.

Since the rows of $A_{\mathrm{eq}(\mathcal{F}).}$ are a Hilbert basis of $\mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}).})^\top)$, there is a non-negative integer vector $\tilde{\boldsymbol{y}}$ with $\boldsymbol{c}^\top = \sum_{i \in \mathrm{eq}(\mathcal{F})} \boldsymbol{A_i}.\tilde{y}_i$. We can set $y_i := 0$ for all $i \notin \mathrm{eq}(\mathcal{F})$ and $y_i := \tilde{y}_i$ for all $i \in \mathrm{eq}(\mathcal{F})$ and obtain an integer vector $\boldsymbol{y}$ with $A^\top \boldsymbol{y} = \boldsymbol{c}$ and $\boldsymbol{y} \geq \boldsymbol{0}$. Furthermore, for $\hat{\boldsymbol{x}} \in \mathcal{F}$, we have

$$\boldsymbol{b}^\top \boldsymbol{y} = \sum_{i \in \mathrm{eq}(\mathcal{F})} y_i\, b_i = \sum_{i \in \mathrm{eq}(\mathcal{F})} y_i \boldsymbol{A_i}.\hat{\boldsymbol{x}} = \boldsymbol{c}^\top \hat{\boldsymbol{x}} = \gamma.$$

Hence, $A\boldsymbol{x} \leq \boldsymbol{b}$ is TDI. □

We apply Theorem 4.35 to show that the TDI property of a system $A\boldsymbol{x} \leq \boldsymbol{b}$ of inequalities is preserved if we restrict ourselves to subsystems corresponding to faces of the polyhedron $\mathcal{P}(A, \boldsymbol{b})$.

Corollary 4.36. If the system $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{c}^\top \boldsymbol{x} \leq d$ with $A \in \mathbb{Q}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Q}^m$, $\boldsymbol{c} \in \mathbb{Q}^n$ and $d \in \mathbb{Q}$ is TDI, then the system $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{c}^\top \boldsymbol{x} \leq d$, $-\boldsymbol{c}^\top \boldsymbol{x} \leq -d$ is TDI as well.

Proof. Let $\mathcal{F} \neq \emptyset$ be a face of the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{c}^\top \boldsymbol{x} \leq d, -\boldsymbol{c}^\top \boldsymbol{x} \leq -d\}$. Then, $\mathcal{F}$ is also a face of the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{c}^\top \boldsymbol{x} \leq d\}$. Since the system $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{c}^\top \boldsymbol{x} \leq d$ is TDI, by Theorem 4.35 that $((A_{\mathrm{eq}(\mathcal{F}).})^\top, \boldsymbol{c})$ is a Hilbert basis of $\mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}).})^\top, \boldsymbol{c})$. Let $\boldsymbol{z} \in \mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}).})^\top, \boldsymbol{c}, -\boldsymbol{c}) \cap \mathbb{Z}^n$. Then, $\boldsymbol{z}$ is of the form

$$\boldsymbol{z} = \sum_{i \in \mathrm{eq}(\mathcal{F})} \lambda_i \boldsymbol{A_i}.^\top + \mu\boldsymbol{c} - \sigma\boldsymbol{c}$$

with $\lambda_i \geq 0$, $i \in \mathrm{eq}(\mathcal{F})$, $\mu \geq 0$, $\sigma \geq 0$. This is equivalent to $\boldsymbol{z} + \sigma\boldsymbol{c} \in \mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}).})^\top, \boldsymbol{c})$. Let $\sigma' \in \mathbb{Z}_{\geq 0}$, $\sigma' \geq \sigma$ be chosen such that $\sigma'\boldsymbol{c} \in \mathbb{Z}^n$. It follows that $\boldsymbol{z} + \sigma'\boldsymbol{c} \in \mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}).})^\top, \boldsymbol{c}) \cap \mathbb{Z}^n$. Since $((A_{\mathrm{eq}(\mathcal{F}).})^\top, \boldsymbol{c})$ is a Hilbert basis of $\mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}).})^\top, \boldsymbol{c})$, we can write $\boldsymbol{z} + \sigma'\boldsymbol{c}$ as

$$\boldsymbol{z} + \sigma'\boldsymbol{c} = \sum_{i \in \mathrm{eq}(\mathcal{F})} \lambda_i' \boldsymbol{A_i}.^\top + \mu'\boldsymbol{c},$$

where $\lambda'_i \geq 0$ is an integer for all $i \in \text{eq}(\mathcal{F})$, and $\mu' \geq 0$ is also an integer. It follows that

$$\boldsymbol{z} = \sum_{i \in \text{eq}(\mathcal{F})} \lambda'_i \boldsymbol{A_i}.^\top + \mu' \boldsymbol{c} - \sigma' \boldsymbol{c}.$$

That is, $\boldsymbol{z}$ can be defined as a non-negative integer combination of the generators of $\text{cone}((A_{\text{eq}(\mathcal{F})\cdot})^\top, \boldsymbol{c}, -\boldsymbol{c})$. Hence, $((A_{\text{eq}(\mathcal{F})\cdot})^\top, \boldsymbol{c}, -\boldsymbol{c})$ is a Hilbert basis of $\text{cone}((A_{\text{eq}(\mathcal{F})\cdot})^\top, \boldsymbol{c}, -\boldsymbol{c})$. Applying Theorem 4.35 completes the proof. □

We mention without proof a characterization of TDI via Hilbert bases that is helpful for algorithmically checking whether an inequality system is TDI.

Theorem 4.37. Let $A \in \mathbb{Z}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{Q}^m$ with $\mathcal{P}(A, \boldsymbol{b}) \neq \emptyset$. The system $A\boldsymbol{x} \leq \boldsymbol{b}$ is TDI if and only if

- the rows of A form a Hilbert basis of the cone spanned by the rows of A, and
- for every $S \subseteq \{1, \ldots, m\}$ the linear program

$$\min \{\boldsymbol{b}^\top \boldsymbol{y} \,:\, A^\top \boldsymbol{y} = \sum_{i \in S} (\boldsymbol{A_i}.)^\top,\, \boldsymbol{y} \geq \boldsymbol{0}\}$$

  has an integral optimum solution.

The following theorem shows that every rational polyhedron $\mathcal{P}$ can be described by a TDI system. The key idea of the proof is to construct, for every minimal face $\mathcal{F}$ of $\mathcal{P}$, a Hilbert basis of the cone spanned by inequalities induced by $\mathcal{F}$.

Theorem 4.38. For every rational polyhedron $\mathcal{P}$ there is a TDI system $A\boldsymbol{x} \leq \boldsymbol{b}$ with integral matrix A such that $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b})$.

Proof. Let $D \in \mathbb{Q}^{m \times n}$, $\boldsymbol{d} \in \mathbb{Q}^m$ be such that $\mathcal{P} = \mathcal{P}(D, \boldsymbol{d})$ and the rows in $D\boldsymbol{x} \leq \boldsymbol{d}$ are not redundant. Let $\mathcal{F}_1, \ldots, \mathcal{F}_t$ be all (minimal) faces of $\mathcal{P}$ and let $\mathcal{H}_i \subseteq \mathbb{Z}^n$ be a minimal integer Hilbert basis of $\text{cone}((D_{\text{eq}(\mathcal{F}_i)\cdot})^\top)$ for every $i \in \{1, \ldots, t\}$ (exists by Remark 4.34). Define A as the matrix whose rows are exactly the vectors in $\bigcup_{i=1}^t \mathcal{H}_i$ and consider the k-th row vector $\boldsymbol{A_k}.$ of A. Then, $\boldsymbol{A_k}.$ is an element of some integer Hilbert basis $\mathcal{H}_i$. Consequently, A is an integral matrix. Let $\boldsymbol{b}$ be defined via $b_k := \max\{\boldsymbol{A_k}.\boldsymbol{x} \,:\, \boldsymbol{x} \in \mathcal{P}\}$. Note that this maximum exists because $\boldsymbol{A_k}.^\top = \sum_{j \in \text{eq}(\mathcal{F}_i)} \lambda_j (\boldsymbol{D_j}.)^\top \in \text{cone}((D_{\text{eq}(\mathcal{F}_i)\cdot})^\top)$, and thus

$$b_k = \max\{\boldsymbol{A_k}.\boldsymbol{x} \,:\, \boldsymbol{x} \in \mathcal{P}\} = \boldsymbol{A_k}.\hat{\boldsymbol{x}} \text{ for all } \hat{\boldsymbol{x}} \in \mathcal{F}_i. \tag{4.1}$$

We claim that $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b})$.

Clearly, $\mathcal{P} \subseteq \mathcal{P}(A, \boldsymbol{b})$, since, for every $\boldsymbol{y} \in \mathcal{P}$ and row $\boldsymbol{A_k}.$ it holds that $\boldsymbol{A_k}.\boldsymbol{y} \leq \max\{\boldsymbol{A_k}.\boldsymbol{x} \,:\, \boldsymbol{x} \in \mathcal{P}\} = b_k$.

Conversely, if $\boldsymbol{y} \notin \mathcal{P}$, then there is a row index $\ell \in \{1, \ldots, m\}$ such that $\boldsymbol{D_\ell}.\boldsymbol{y} > d_\ell$. Let $i \in \{1, \ldots, t\}$ such that $\ell \in \text{eq}(\mathcal{F}_i)$, which exists since $D\boldsymbol{x} \leq \boldsymbol{d}$ is not redundant.

Now, let $\delta \geq 0$ be such that $\delta(\boldsymbol{D_\ell}.)^\top \in \mathbb{Z}^n$. Since $\{\boldsymbol{a^{(1)}}, \ldots, \boldsymbol{a^{(s)}}\} := \mathcal{H}_i$ is a Hilbert basis of $\text{cone}((D_{\text{eq}(\mathcal{F}_i)\cdot})^\top)$, there are non-negative integer multiples $\delta_1, \ldots, \delta_s$ of δ with

$$\delta \boldsymbol{D_\ell}.^\top = \sum_{r=1}^{s} \delta_r \boldsymbol{a^{(r)}}.$$

Denote by k_r the row of A with $\boldsymbol{A_{k_r}}.^\top = \boldsymbol{a^{(r)}}$ for all $r \in \{1,\dots,s\}$. Taking into account (4.1), we obtain for $\hat{\boldsymbol{x}} \in \mathcal{F}_i$ that

$$\sum_{r=1}^{s} \delta_r \boldsymbol{A_{k_r}}.\boldsymbol{y} = \sum_{r=1}^{s} \delta_r (\boldsymbol{a^{(r)}})^\top \boldsymbol{y} = \delta \boldsymbol{D_\ell}.\boldsymbol{y} > \delta d_\ell = \delta \boldsymbol{D_\ell}.\hat{\boldsymbol{x}} = \sum_{r=1}^{s} \delta_r (\boldsymbol{a^{(r)}})^\top \hat{\boldsymbol{x}} = \sum_{r=1}^{s} \delta_r b_{k_r}.$$

So there is a $\hat{r} \in \{1,\dots,s\}$ with $\boldsymbol{A_{k_{\hat{r}}}}.\boldsymbol{y} > b_{k_{\hat{r}}}$. Thus $y \notin \mathcal{P}(A,\boldsymbol{b})$. This completes the proof that $\mathcal{P} = \mathcal{P}(A,\boldsymbol{b})$.
Finally, the system $A\boldsymbol{x} \leq \boldsymbol{b}$ is TDI as required: If $\mathcal{F}_i$ is a face of $\mathcal{P}$ and $\{\boldsymbol{a^{(1)}},\dots,\boldsymbol{a^{(s)}}\} := \mathcal{H}_i$, then $\boldsymbol{A_{k_r}}.\hat{\boldsymbol{x}} = (\boldsymbol{a^{(r)}})^\top \hat{\boldsymbol{x}} = b_{k_r}$ for all $\hat{\boldsymbol{x}} \in \mathcal{F}_i$ and $r \in \{1,\dots,s\}$, see (4.1). Thus $(\boldsymbol{a^{(r)}})^\top$ is a row of $A_{\mathrm{eq}(\mathcal{F}_i)}.$. In addition, $\{\boldsymbol{a^{(1)}},\dots,\boldsymbol{a^{(s)}}\}$ is an (integer) Hilbert basis of $\mathrm{cone}((D_{\mathrm{eq}(\mathcal{F}_i)}.)^\top) = \mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}_i)}.)^\top)$ because of $\mathcal{P}(D,\boldsymbol{d}) = \mathcal{P}(A,\boldsymbol{b})$.
Applying Theorem 4.35 completes the proof. □

In fact, it is even true that $\mathcal{P}$ is integral if and only if $\boldsymbol{b}$ can be chosen as an integer vector. One direction is obtained from the proof of Theorem 4.38, because $\mathcal{P}$ is integral if and only if every face contains an integer point (Lemma 2.33). For the TDI system in the proof of Theorem 4.38, every integer vector $\boldsymbol{z}$ on a face satisfies some of the inequalities of $A\boldsymbol{x} \leq \boldsymbol{b}$ with equality. Due to (4.1), the associated components of $\boldsymbol{b}$ must be integers. Conversely, Theorem 4.31 implies that integrality of $\boldsymbol{b}$ implies the integrality of $\mathcal{P}$. We obtain the following characterization.

Theorem 4.39. A rational polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ is integral if and only if there is a TDI system $A\boldsymbol{x} \leq \boldsymbol{b}$ with $A \in \mathbb{Q}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{Z}^m$ and $\mathcal{P} = \mathcal{P}(A,\boldsymbol{b})$.

4.3.1 Applications in combinatorial optimization

There are many examples where TDI systems naturally occur, two of which are presented below.

Example 4.40. A matrix $A \in \{0,1\}^{m\times n}$ is called *balanced* if it does not have a $(k \times k)$-submatrix for odd k and with exactly two ones per row and column. For example, the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

is balanced, but not totally unimodular (its determinant is -2). It can be shown (see [16]) that A being balanced implies that the system $A\boldsymbol{x} \leq \boldsymbol{1}, \boldsymbol{x} \geq \boldsymbol{0}$ is TDI. △

Often, TDI systems are used to prove discrete min-max results, such as Theorem 4.14.

Example 4.41. Let $G = (V,E)$ be a directed graph and $r \in V$. An *r-arborescence* is a subset $A \subseteq E$ with $|A| = |V| - 1$ such that every vertex is reachable from r in (V,A), i.e., in particular, $|\delta^-(v) \cap A| = 1$ for all $v \in V \setminus \{r\}$ and $|\delta^-(r) \cap A| = 0$. An *r-cut* is a directed cut $\delta^-(U)$ in G with $\emptyset \neq U \subseteq V \setminus \{r\}$.
Let M be the 0/1-matrix whose rows are the incidence vectors of all r-cuts in G. It can be shown (see [14]) that the system $\{M\boldsymbol{x} \geq \boldsymbol{1}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is TDI. The minimal $\boldsymbol{x} \in \{0,1\}^A$ that satisfy $M\boldsymbol{x} \geq \boldsymbol{1}$ are exactly the r-arborescences.
With this result, we can handily prove Fulkerson's optimal arborescence theorem (see [17]): Let $\boldsymbol{c} \in \mathbb{Z}^m_{\geq 0}$ be a non-negative integer cost vector. Because $\{\boldsymbol{x} \in \mathbb{R}^m : M\boldsymbol{x} \geq \boldsymbol{1}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is TDI and because of LP duality, we get that

$$\min\{\boldsymbol{c}^\top \boldsymbol{x} : M\boldsymbol{x} \geq \boldsymbol{1}, \boldsymbol{x} \geq \boldsymbol{0}\} = \max\{\boldsymbol{1}^\top \boldsymbol{y} : M^\top \boldsymbol{y} \leq \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\}$$

and both optima are assumed by integral points $\boldsymbol{x}^\star$ and $\boldsymbol{y}^\star$, see Theorem 4.31. This means that the minimum cost of an r-arborescence is equal to the maximum size of a set of r-cuts such that every $e \in A$ is contained in at most c_a r cuts. △

Remark 4.42. It can be checked in polynomial time whether a matrix is is totally unimodular (see [40]). The same applies to recognizing balanced matrices (see [11]). On the other hand, it is NP-complete to decide whether a system is TDI, even for $A\boldsymbol{x} \leq \mathbf{1}, A \in \{0,1\}^{m \times n}$ (see [13]).

5 Cutting Planes

We consider a general mixed-integer program as introduced in Definition 1.1, i.e.,

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \leq \boldsymbol{b}, \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{aligned} \tag{5.1}$$

with $A \in \mathbb{Q}^{m \times n}$, $\boldsymbol{c} \in \mathbb{Q}^n$, $\boldsymbol{b} \in \mathbb{Q}^m$ and $p \in \{0, \dots, n\}$. Note that, to ensure that the integer hull is a polyhedron, we restrict ourselves to rational data (see Chapter 2).

In Chapter 4 we investigated sufficient conditions for the LP-relaxation of (5.1) to have integral optima, and Chapter 3 we saw how to treat the general case via branch-and-bound. In this chapter, we investigate how to strengthen the LP-relaxation by adding additional constraints that shrink the feasible region of the LP without excluding feasible solutions of the MIP.

In particular, if a computed optimum vertex solution $\boldsymbol{x}^\star$ of the LP-relaxation does not fulfill the integer conditions (i.e., $\boldsymbol{x}^\star \notin \mathbb{Z}^p \times \mathbb{R}^{n-p}$), there must be an inequality (since $\mathcal{P}_\mathrm{I}$ is a polyhedron by Theorem 2.30), called *cutting plane*, that separates $\boldsymbol{x}^\star$ from $\mathcal{P}_\mathrm{I} = \operatorname{conv}(\{\boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : A\boldsymbol{x} \leq \boldsymbol{b}\})$ – finding such an inequality amounts to solving a separation problem (see *Introduction to Optimization*). If we find such a separating inequality, we can add it to strengthen the LP relaxation and repeat the process for as long as the LP relaxation yields infeasible optima. This procedure is often referred to as the *cutting plane method*.

In this chapter, we will devise strategies for generating cutting planes that, under suitable conditions, provably produce a feasible optimum solution. These approaches generally work for arbitrary IPs. Because (5.1) is NP-hard to solve (Theorem 3.1) and the separation problem is polynomially equivalent to the optimization problem (see *Introduction to Optimization*), we cannot hope for efficient procedures that are always successful (assuming P $\neq$ NP).

We will also examine the generation of cuts that can be used to accelerate the branch-and-bound method. Such cutting planes often exploit the underlying problem structure.

5.1 General cutting planes

In this section, we consider a class of inequalities that is valid for $\mathcal{P}_\mathrm{I}$ and which can be applied irrespective of the specific problem structure. We will see that this class is rich enough to provide a complete description of $\mathcal{P}_\mathrm{I}$. We begin by introducing these inequalities from a geometric perspective and later assume a more algorithmic approach.

5.1.1 Geometric approach: Chvátal-Gomory inequalities

We consider a rational polyhedron $\mathcal{P} := \mathcal{P}(A, \boldsymbol{b})$, with $A \in \mathbb{Q}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Q}^m$. Our goal is to describe $\mathcal{P}_\mathrm{I} := \operatorname{conv}(\mathcal{P} \cap \mathbb{Z}^n)$ by linear inequalities.

Let $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$ with integer vector $\boldsymbol{c} \in \mathbb{Z}^n$ be a *valid* inequality, i.e., $\mathcal{P} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \delta\}$ (and $\mathcal{P}_\mathrm{I} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \delta\}$). Since $\boldsymbol{c}^\top \boldsymbol{x} \in \mathbb{Z}$ for all $\boldsymbol{x} \in \mathcal{P} \cap \mathbb{Z}^n$, it follows that

$$\mathcal{P}_\mathrm{I} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor\}.$$

This observation suggests taking all valid inequalities with an integer normal vector and rounding down the right-hand sides in order to obtain a stronger formulation. Of course, it suffices to take supporting inequalities, i.e., those for which $\mathcal{P} \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} = \delta\} \neq \emptyset$. To do this, define

$$P^1 := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor \text{ for all valid supporting inequalities} \\ \boldsymbol{c}^\top \boldsymbol{x} \leq \delta \text{ of } \mathcal{P} \text{ with } \boldsymbol{c} \in \mathbb{Z}^n\}. \tag{5.2}$$

The resulting inequalities $\boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor$ are called *Chvátal-Gomory inequalities* and $\mathcal{P}^1$ is called *elementary closure* of $\mathcal{P}$.

At first glance, it is not clear whether $\mathcal{P}^1$ is a polyhedron again, because there are infinitely many supporting hyperplanes. However, we will prove that $\mathcal{P}^1$ is indeed a polyhedron. This allows us to iterate and define

$$\mathcal{P}^0 := \mathcal{P} \text{ and } \mathcal{P}^{t+1} := (\mathcal{P}^t)^1 \text{ for } t \in \mathbb{N}. \tag{5.3}$$

Then $\mathcal{P}^t$ is the *Chvátal-Gomory closure of rank* t. We immediately have

$$\mathcal{P} = \mathcal{P}^0 \supseteq \mathcal{P}^1 \supseteq \cdots \supseteq \mathcal{P}_I. \tag{5.4}$$

This raises the question of whether this method is finite. We will show that, indeed, a $t \in \mathbb{N}$ exists with $\mathcal{P}^t = \mathcal{P}_\mathrm{I}$. The resulting $\mathcal{P}^t$ provides the desired description of $\mathcal{P}_\mathrm{I}$ by linear inequalities.

Remark 5.1. By rounding the right-hand side, we move the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} = \delta\}$ for $\boldsymbol{c} \in \mathbb{Z}^n$ closer to $\mathcal{P}_\mathrm{I}$. If $\gcd(c_1, \ldots, c_n) = 1$, the resulting hyperplane then contains an integer point: Otherwise, according to Lemma 4.25, we would have $\lambda \in \mathbb{Q}$ with $\lambda \boldsymbol{c} \in \mathbb{Z}^n$ but $\lambda \lfloor \delta \rfloor \notin \mathbb{Z}$. Since $\gcd(c_1, \ldots, c_n) = 1$ and $\lambda \boldsymbol{c} \in \mathbb{Z}^n$, it follows that $\lambda \in \mathbb{Z}$, a contradiction.

If the resulting hyperplanes define supporting hyperplanes for $\mathcal{P}_\mathrm{I}$ we have reached $\mathcal{P} = \mathcal{P}_\mathrm{I}$, since, by Lemma 2.33, we have the following.

Observation 5.2. Every minimal face of a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ contains an integer point if and only if every supporting hyperplane of $\mathcal{P}$ contains an integer point of $\mathcal{P}$.

We start with the proof that $\mathcal{P}^1$ is a polyhedron. According to Theorem 4.38, every rational polyhedron can be represented by a TDI system of the form $A\boldsymbol{x} \leq \boldsymbol{b}$ with integer matrix A. This allows to calculate $\mathcal{P}^1$ directly.

Theorem 5.3. Let $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b}) \neq \emptyset$ with $A\boldsymbol{x} \leq \boldsymbol{b}$ TDI and $A \in \mathbb{Z}^{m \times n}$. Then, $\mathcal{P}^1 = \mathcal{P}(A, \lfloor \boldsymbol{b} \rfloor)$, i.e., $\mathcal{P}^1$ is a polyhedron.[a]

[a] $\lfloor \boldsymbol{b} \rfloor$ is to be understood component-wise.

Proof. We have $\mathcal{P}^1 \subseteq \mathcal{P}(A, \lfloor \boldsymbol{b} \rfloor)$, since $\boldsymbol{A}_i.\boldsymbol{x} \leq \lfloor b_i \rfloor$ is a Chvátal-Gomory inequality.

To prove the converse, let $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$ be a valid supporting inequality for $\mathcal{P}$ and $\boldsymbol{c}$ be integral. By strong duality, we have

$$\delta = \max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\} = \min\{\boldsymbol{y}^\top \boldsymbol{b} : A^\top \boldsymbol{y} = \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\}.$$

Since $A\boldsymbol{x} \leq \boldsymbol{b}$ is TDI, there is an integral optimum solution $\boldsymbol{y}^\star$ of the dual problem, i.e., $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$ can be expressed as integral linear combination of the rows of $A\boldsymbol{x} \leq \boldsymbol{b}$. For $\hat{\boldsymbol{x}} \in \mathcal{P}(A, \lfloor \boldsymbol{b} \rfloor)$, we have

$$\boldsymbol{c}^\top \hat{\boldsymbol{x}} = (A^\top \boldsymbol{y}^\star)^\top \hat{\boldsymbol{x}} = (\boldsymbol{y}^\star)^\top (A\hat{\boldsymbol{x}}) \overset{\boldsymbol{y}^\star \geq \boldsymbol{0}}{\leq} (\boldsymbol{y}^\star)^\top \lfloor \boldsymbol{b} \rfloor = \lfloor (\boldsymbol{y}^\star)^\top \lfloor \boldsymbol{b} \rfloor \rfloor \leq \lfloor (\boldsymbol{y}^\star)^\top \boldsymbol{b} \rfloor = \lfloor \delta \rfloor.$$

It follows that $\mathcal{P}(A, \lfloor \boldsymbol{b} \rfloor) \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor\}$. Let $S \subseteq \mathbb{Z}^n \times \mathbb{R}$ denote the set of all pairs $(\boldsymbol{c}, \delta)$ such that $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$ is a valid supporting inequality. We obtain

$$\mathcal{P}(A, \lfloor \boldsymbol{b} \rfloor) \subseteq \bigcap_{(c,\delta) \in S} \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor\} = \mathcal{P}^1. \qquad \square$$

We show the useful fact that the operations of taking a face and taking the elementary closure of a polyhedron commute.

Corollary 5.4. For every face $\mathcal{F}$ of $\mathcal{P}$ it holds that $\mathcal{F}^1 = \mathcal{P}^1 \cap \mathcal{F}$.

Proof. By Theorem 4.38, we may let $\mathcal{P} = \mathcal{P}(A, \boldsymbol{b})$ with $A\boldsymbol{x} \leq \boldsymbol{b}$ TDI and A integral. Let $\mathcal{F}$ be a face of $\mathcal{P}$ and let $\boldsymbol{c} \in \mathbb{Z}^n$, $\delta \in \mathbb{Z}$ such that $\mathcal{F} = \{x \in \mathcal{P} : \boldsymbol{c}^\top \boldsymbol{x} = \delta\}$ and $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$ is valid for $\mathcal{P}$. Theorem 4.35 implies that the system $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$ is also TDI, since the corresponding polyhedron has no additional faces. Therefore, by Corollary 4.36, the system $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$, $-\boldsymbol{c}^\top \boldsymbol{x} \leq -\delta$ is TDI as well. Since δ is an integer, Theorem 5.3 yields

$$\begin{aligned}
\mathcal{P}^1 \cap \mathcal{F} &= \mathcal{P}^1 \cap \mathcal{P} \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} = \delta\} \\
&\overset{\text{Thm 5.3}}{=} \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \lfloor \boldsymbol{b} \rfloor, \boldsymbol{c}^\top \boldsymbol{x} = \delta\} \\
&\overset{\delta \in \mathbb{Z}}{=} \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \lfloor \boldsymbol{b} \rfloor, \boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor, -\boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor -\delta \rfloor\} \\
&\overset{\text{Thm 5.3}}{=} \mathcal{F}^1. \qquad \square
\end{aligned}$$

Observation 5.5.

(a) $\mathcal{F}^1$ is a (possibly empty) face of $\mathcal{P}^1$, since $\mathcal{F}^1 = \mathcal{P}^1 \cap \mathcal{F} = \mathcal{P}^1 \cap \mathcal{P} \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} = \delta\} = \mathcal{P}^1 \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} = \delta\}$ (note, that $\mathcal{P}^1 \subseteq \mathcal{P} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \delta\}$).

(b) Therefore, $\mathcal{F}^2 = (\mathcal{F}^1)^1 = (\mathcal{P}^1)^1 \cap \mathcal{F}^1 = \mathcal{P}^2 \cap \mathcal{P}^1 \cap \mathcal{F} = \mathcal{P}^2 \cap \mathcal{F}$. Repeated application yields

$$\mathcal{F}^t = \mathcal{P}^t \cap \mathcal{P}^{t-1} \cap \cdots \cap \mathcal{P}^1 \cap \mathcal{F} = \mathcal{P}^t \cap \mathcal{F} \qquad \forall t \in \mathbb{N}.$$

We now have everything in place to show the finiteness of the rounding scheme.

Theorem 5.6 ([9])**.** For every rational polyhedron $\mathcal{P}$ there exists $t \in \mathbb{Z}_{\geq 0}$ with $\mathcal{P}^t = \mathcal{P}_\mathrm{I}$.

We first sketch the proof of Theorem 5.6. It uses induction on the dimension of the polyhedron. We show, that it suffices to consider the progress of supporting hyperplanes of a finite number of orientations – namely all normal directions needed for a complete description of the polyhedra $\mathcal{P}$ and $\mathcal{P}_\mathrm{I}$. For every fixed direction, we argue that we can make progress until the corresponding supporting hyperplane contains an integer point of $\mathcal{P}$: As long as this is not the case, we know by induction that, after a finite number of iterations, the Chvátal-Gomory closure no longer contains points of this supporting hyperplane. Eventually, all relevant supporting hyperplanes contain an integer point of $\mathcal{P}$ and thus we have found a description of $\mathcal{P}_\mathrm{I}$ (cf. Observation 5.2).

Proof of Theorem 5.6. If $\mathcal{P} = \emptyset$, then $\mathcal{P}^0 = \mathcal{P} = \mathcal{P}_\mathrm{I}$ follows. We may therefore assume $\mathcal{P} \neq \emptyset$ in the following.

We further prove that we may assume $\mathcal{P}$ to be full-dimensional. To that end, let $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}\} := \mathrm{aff}(\mathcal{P}) \neq \mathbb{R}^n$ be the affine hull of $\mathcal{P}$, i.e., $\mathcal{P} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}\}$. We can assume that A is an integer matrix of full row rank (otherwise we scale and eliminate redundant rows), i.e., $A \in \mathbb{Z}^{(n-d)\times n}$ with $d := \dim(\mathcal{P})$.

If $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}\}$ contains no integer point, then, according to Lemma 4.25, there is $\boldsymbol{y} \in \mathbb{Q}^{n-d}$ with $\boldsymbol{c} := A^\top \boldsymbol{y} \in \mathbb{Z}^n$ and $\delta := \boldsymbol{b}^\top \boldsymbol{y} \notin \mathbb{Z}$. Every $\hat{\boldsymbol{x}} \in \mathcal{P}$ satisfies $A\hat{\boldsymbol{x}} = \boldsymbol{b}$ and therefore $\boldsymbol{c}^\top \hat{\boldsymbol{x}} = (A^\top \boldsymbol{y})^\top \hat{\boldsymbol{x}} = \boldsymbol{y}^\top A\hat{\boldsymbol{x}} = \boldsymbol{y}^\top \boldsymbol{b} = \delta$. Thus $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} = \delta\}$ is a supporting hyperplane of $\mathcal{P}$. We conclude that

$$\begin{aligned}\mathcal{P}_\mathrm{I} \subseteq \mathcal{P}^1 &\subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor, -\boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor -\delta \rfloor\} \\ &= \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor, \boldsymbol{c}^\top \boldsymbol{x} \geq \lceil \delta \rceil\} \overset{\delta \notin \mathbb{Z}}{=} \emptyset,\end{aligned}$$

and thus $\mathcal{P}_\mathrm{I} = \mathcal{P}^1 = \emptyset$.

Now let $\hat{\boldsymbol{x}} \in \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{b}\}$ be integral. Theorem 5.6 is invariant under translation by the vector $\hat{\boldsymbol{x}}$, thus we can assume $\mathrm{aff}(\mathcal{P}) = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} = \boldsymbol{0}\}$. By Theorem 4.21 and Remark 4.24 (b), as well as the fact that the row rank of A is $n-d$, a square unimodular matrix U with $AU = [B, 0]$ in Hermite normal form exists. By Corollary 4.6, U^{-1} is unimodular as well and $U\mathbb{Z}^n = \mathbb{Z}^n$. Via the (bijective) variable transformation $\boldsymbol{x} = U\boldsymbol{z}$, we can therefore assume that

$$\mathrm{aff}(\mathcal{P}) = \{\boldsymbol{z} \in \mathbb{R}^n : [B, 0]\boldsymbol{z} = \boldsymbol{0}\} = \{0\}^{n-d} \times \mathbb{R}^d.$$

Every supporting hyperplane $\mathcal{H} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^\top \boldsymbol{x} = \delta\}$ of $\mathcal{P}$ can be transformed into the form

$$\mathcal{H}' = \{\boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^{n-d} 0\, x_i + \sum_{i=n-d+1}^{n} c_i x_i = \delta\}$$

by adding suitable multiples of the rows of $[B, 0]$ to $\boldsymbol{c}$ (this is possible, because $[B, 0]\boldsymbol{x} = \boldsymbol{0}$ for $\boldsymbol{x} \in \mathcal{P} \subseteq \mathrm{aff}(\mathcal{P})$; δ and $c_{n-d+1}, \ldots, c_n$ are unaffected). In the construction of $\mathcal{P}^1$ we can restrict ourselves to support hyperplanes of the form $\mathcal{H}'$. We may therefore assume $n - d = 0$, i.e., $\mathcal{P}$ is full dimensional.

Towards the proof of the theorem, we employ induction over the dimension d of $\mathcal{P}$. If $d = 0$, then $\mathcal{P} = \{\bar{\boldsymbol{x}}\}$ for some $\bar{\boldsymbol{x}} \in \mathbb{R}^n$. If $\bar{\boldsymbol{x}} \in \mathbb{Z}^n$, we obtain $\mathcal{P}^0 = \mathcal{P} = \mathcal{P}_\mathrm{I}$. Otherwise, $\mathcal{P}^1 = \emptyset = \mathcal{P}_\mathrm{I}$. Now, consider $d > 0$ and let the theorem hold for all polyhedra of smaller dimensions.

By Theorem 2.30, by Corollary 2.21, and by scaling, we can find $W \in \mathbb{Z}^{m\times n}$ and $\boldsymbol{w} \in \mathbb{Z}^m$ with $\mathcal{P}_\mathrm{I} = \mathcal{P}(W, \boldsymbol{w})$. We may further assume that $\boldsymbol{W_i}. \neq \boldsymbol{0}^\top$ and $w'_i := \max\{\boldsymbol{W_i}.\boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\}$ is bounded for all $i \in \{1, \ldots, m\}$: If $\mathcal{P}_\mathrm{I} \neq \emptyset$, this holds since zero-rows are redundant and by Corollary 2.31. Now assume $\mathcal{P}_\mathrm{I} = \emptyset$. Since $\mathcal{P} \neq \emptyset$ is rational, Corollary 2.21 yields a representation $\mathcal{P} = \mathcal{V} + \mathcal{E}$. We have $\dim(\mathcal{E}) < n$, otherwise $\mathcal{P}$ would contain integral points (exercise). Hence, there is $\boldsymbol{d} \in \mathbb{Z}^n$ with $\boldsymbol{d}^\top \boldsymbol{r} = 0$ for all $\boldsymbol{r} \in \mathcal{E}$, and both $\max\{\boldsymbol{d}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\} = \max\{\boldsymbol{d}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{V}\}$ and $\max\{-\boldsymbol{d}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\} = -\min\{\boldsymbol{d}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{V}\}$ are bounded. We can thus set $W := \binom{\boldsymbol{d}^\top}{-\boldsymbol{d}^\top}$ and $\boldsymbol{w} = \binom{-1}{1}$ with $\mathcal{P}_\mathrm{I} = \mathcal{P}(W, \boldsymbol{w}) = \emptyset$. In either case, we can find $\boldsymbol{w'}$ with $\mathcal{P} \subseteq \mathcal{P}(W, \boldsymbol{w'})$.

Now, consider an inequality $\boldsymbol{a}^\top \boldsymbol{x} \leq \beta$ of the system $W\boldsymbol{x} \leq \boldsymbol{w}$ with $\boldsymbol{a} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$ and $\beta \in \mathbb{Z}$ and let $\boldsymbol{a}^\top \boldsymbol{x} \leq \beta'$ be the corresponding inequality of $W\boldsymbol{x} \leq \boldsymbol{w'}$ (feasible for $\mathcal{P}$). We claim that we can find $s \in \mathbb{N}$ with $\mathcal{P}^s \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta\}$. Suppose this was not the case, i.e., we cannot find any Chvátal-Gomory inequality that implies $\boldsymbol{a}^\top \boldsymbol{x} \leq \beta$. Then, $\mathcal{P}^1 \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \lfloor \beta' \rfloor\}$, but, by assumption, $\mathcal{P}^1 \nsubseteq \{x \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta\}$ holds. It follows that $\beta < \lfloor \beta' \rfloor$. Since $\boldsymbol{w}$ and therefore β is integral, there is $\beta'' \in \mathbb{Z}$ and $r \in \mathbb{Z}_{\geq 0}$ with

$$\begin{aligned}&\beta < \beta'' \leq \lfloor \beta' \rfloor, \\ &\mathcal{P}^t \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta''\} \text{ for all } t \geq r, \text{ and} \\ &\mathcal{P}^t \nsubseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta'' - 1\} \text{ for all } t \geq r.\end{aligned}$$

Due to the choice of r, we have that $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} = \beta''\}$ is a supporting hyperplane for $\mathcal{P}^r$. Since $\boldsymbol{a} \neq \boldsymbol{0}$, we have that $\mathcal{F} := \mathcal{P}^r \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} = \beta''\}$ has lower dimension than n. We further conclude from $\mathcal{P}_\mathrm{I} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta\}$ and $\beta'' > \beta$ that $\mathcal{F} \cap \mathbb{Z}^n = \emptyset$. Therefore, by the induction hypothesis, there is $\tilde{r} \in \mathbb{Z}_{\geq 0}$ with $\mathcal{F}^{\tilde{r}} = \mathcal{F}_\mathrm{I} = \emptyset$. By Remark 5.5 (b),

$$\emptyset = \mathcal{F}^{r+\tilde{r}} = \mathcal{P}^{r+\tilde{r}} \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} = \beta''\}.$$

Therefore, $\mathcal{P}^{r+\tilde{r}} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} < \beta''\}$, which means that $\mathcal{P}^{r+\tilde{r}+1} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta'' - 1\}$, which is a contradiction to the choice of β'' and r.

Since $\boldsymbol{a}^\top \boldsymbol{x} \leq \beta$ was chosen arbitrarily and the system $W\boldsymbol{x} \leq \boldsymbol{w}$ is finite, we can thus find $s \in \mathbb{Z}_{\geq 0}$ with $\mathcal{P}^s \subseteq \mathcal{P}(W, \boldsymbol{w}) = \mathcal{P}_\mathrm{I}$. It follows that $\mathcal{P}^s = \mathcal{P}_\mathrm{I}$ as desired (see (5.4)). □

Let us review what we have shown so far. The procedure to obtain a linear description of the integral polyhedron $\mathcal{P}_\mathrm{I} = \mathrm{conv}(\{\boldsymbol{x} \in \mathbb{Z}^n : A\boldsymbol{x} \leq \boldsymbol{b}\})$ proceeds as follows. We start with the linear relaxation $\mathcal{P}^0 = \mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$ of $\mathcal{P}_\mathrm{I}$. Next, we consider every supporting hyperplane of $\mathcal{P}$ whose left-hand side has integer coefficients, and round the right-hand side down to the next integer. This operation, for all such supporting hyperplanes results in the polyhedron $\mathcal{P}^1$. Theorem 5.3 shows that there is no need to consider all supporting hyperplanes – all we need is a TDI system $D\boldsymbol{x} \leq \boldsymbol{d}$ that describes $\mathcal{P}$. For this TDI system, we need to round down the vector $\boldsymbol{d}$ of right-hand sides. In the proof of Theorem 4.38, we explicitly constructed the TDI system for a rational polyhedron $\mathcal{P}$ by explicitly constructing, for every face $\mathcal{F}$ of $\mathcal{P}$ a generating Hilbert basis of $\mathrm{cone}((A_{\mathrm{eq}(\mathcal{F}),\cdot})^\top)$. Overall, the method for constructing a Hilbert basis for a polyhedral cone yields an algorithm for computing $\mathcal{P}^1$. According to Theorem 5.6, this algorithm only needs to be performed a finite number of times times in order to obtain a linear description of $\mathcal{P}_\mathrm{I}$.

The entire procedure is hardly practicable. First of all, the number $t \in \mathbb{N}$ in Theorem 5.6 is possibly exponential in the coding length of the input variables $A, \boldsymbol{b}$ (see exercise). Secondly, in every iteration, we have to determine Hilbert bases for all cones generated by the faces of $\mathcal{P}^{i-1}$. In general, not only is the number of faces exponential, we cannot even compute a single Hilbert basis in polynomial time.

On the other hand, a complete description of $\mathcal{P}_\mathrm{I} = \mathrm{conv}(\{\boldsymbol{x} \in \mathbb{Z}^n : A\boldsymbol{x} \leq \boldsymbol{b}\})$ is not necessary for solving (5.1). We "merely" need to find an optimum solution. Put another way, we are only interested in one face, namely the one that contains the optimum solutions. Even for this face, it is not necessary to find a Hilbert basis for the corresponding cone. We only need to be able to find integral vectors in this cone.

More formally, assume we have the solution of the LP relaxation $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\}$. Let $\boldsymbol{x}^\star$ be an optimum solution that is not integral. The challenge is to find an integral vector in $\mathrm{cone}((A_{\mathrm{eq}(\boldsymbol{x}^\star),\cdot})^\top)$ that separates the current fractional optimum solution $\boldsymbol{x}^\star$, i.e., to find $\boldsymbol{d} \in \mathbb{Z}^n \cap \mathrm{cone}((A_{\mathrm{eq}(\boldsymbol{x}^\star),\cdot})^\top)$ with $\boldsymbol{d}^\top \boldsymbol{x}^\star \notin \mathbb{Z}$.

5.1.2 Algorithmic approach: Gomory inequalities

We can systematically find cutting planes by evaluating information gained during the course of the simplex algorithm. We first consider the purely integral case ($p = n$) and assume that all coefficients are integers. We can transform (5.1) into standard form by splitting $\boldsymbol{x}$ into $\boldsymbol{x}^+, \boldsymbol{x}^- \geq \boldsymbol{0}$ with $\boldsymbol{x} = \boldsymbol{x}^+ - \boldsymbol{x}^-$ and introducing (integral) slack variables (see *Introduction to Optimization*). We obtain a problem of the form

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{x} \in \mathbb{Z}^n. \end{aligned} \tag{5.5}$$

with $A \in \mathbb{Z}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{Z}^m$ and the LP relaxation

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

Solving the latter via the simplex method, we obtain an optimum vertex solution $\boldsymbol{x}^\star$ and an optimum basis B with $B \subseteq \{1,\dots,n\}$, $|B| = m$, and $A_{\cdot B}$ regular. As usual, we denote the non-basis by $N = \{1,\dots,n\} \setminus B$. This means that $\boldsymbol{x}^\star_{\boldsymbol{N}} = \boldsymbol{0}$ and $\boldsymbol{x}^\star_{\boldsymbol{B}} = \bar{\boldsymbol{b}} - \bar{A}_{\cdot N}\boldsymbol{x}^\star_{\boldsymbol{N}} = A^{-1}_{\cdot B}\boldsymbol{b}$ and $\boldsymbol{x}^\star_{\boldsymbol{N}} = \boldsymbol{0}$ with $\bar{\boldsymbol{b}} := A^{-1}_{\cdot B}\boldsymbol{b}$ and $\bar{A} := A^{-1}_{\cdot B}A$. If $\boldsymbol{x}^\star_{\boldsymbol{B}}$ is integral, we have found an optimum solution of (5.5). Otherwise, there exists $i \in B$ with $x^\star_i \notin \mathbb{Z}$.

For every feasible solution $\boldsymbol{x}$ of (5.5) it holds that $\boldsymbol{x}_{\boldsymbol{B}} = \bar{\boldsymbol{b}} - \bar{A}_{\cdot N}\boldsymbol{x}_{\boldsymbol{N}}$ or

$$x_i = \bar{b}_i - \sum_{j\in N} \bar{A}_{ij}\, x_j. \tag{5.6}$$

With $\boldsymbol{x}_{\boldsymbol{N}} \geq \boldsymbol{0}$ we get

$$x_i \leq \bar{b}_i - \sum_{j\in N} \lfloor \bar{A}_{ij} \rfloor\, x_j.$$

Because $x_i \in \mathbb{Z}$ and $\lfloor \bar{A}_{ij} \rfloor\, x_j \in \mathbb{Z}$ for $j \in N$, we can round the right-hand side to obtain

$$x_i \leq \lfloor \bar{b}_i \rfloor - \sum_{j\in N} \lfloor \bar{A}_{ij} \rfloor\, x_j. \tag{5.7}$$

This valid inequality is called *(fractional) Gomory cut*. It cuts off $\boldsymbol{x}^\star$ from the current LP relaxation because $\boldsymbol{x}^\star_{\boldsymbol{N}} = \boldsymbol{0}$ and $x^\star_i = \bar{b}_i \notin \mathbb{Z}$. Furthermore, (5.7) is an inequality of the form $\boldsymbol{c}^\top \boldsymbol{x} \leq \lfloor \delta \rfloor$, $c \in \mathbb{Z}^n$ for a supporting hyperplane $\boldsymbol{c}^\top \boldsymbol{x} \leq \delta$, thus it appears in the approach of the previous section.

The inequality (5.7) is converted by means of a slack variable into an equation and added to the system $A\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$, maintaining the property that all coefficients are integers. Therefore, the slack variable introduced for the new inequality can also be regarded as an integer and thus the procedure can be iterated. It can be shown that, with a certain choice of optimum solution and choice of fractional basis element, we obtain a finite algorithm (see [20]), i.e., after a finite number of inequalities have been added, an integral solution is found.

The inequality (5.7) can be rewritten as follows. Let $f_i \in (0,1)$ be the fractional part of $\bar{b}_i$, i.e. $\bar{b}_i = \lfloor \bar{b}_i \rfloor + f_i$. Analogously, let $f_{ij} \in [0,1)$ be the fractional parts of $\bar{A}_{ij}$, i.e., $\bar{A}_{ij} = \lfloor \bar{A}_{ij} \rfloor + f_{ij}$. From (5.6) we thus obtain

$$x_i = f_i - \sum_{j\in N} f_{ij}\, x_j + \Big(\lfloor \bar{b}_i \rfloor - \sum_{j\in N} \lfloor \bar{A}_{ij} \rfloor\, x_j \Big).$$

If we subtract this equation from (5.7), we obtain the equivalent inequality

$$\sum_{j\in N} f_{ij}\, x_j \geq f_i. \tag{5.8}$$

Example 5.7. Consider the IP

$$\begin{aligned} \max \quad & & x_2 \\ \text{s.t.} \quad & 4x_1 & +x_2 \leq 4 \\ & -4x_1 & +x_2 \leq 0 \\ & & x_1, x_2 \geq 0 \\ & & x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

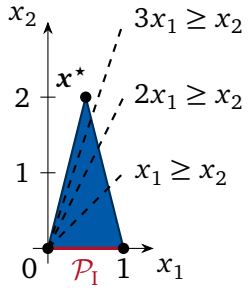

Figure 5.1: Illustration of Examples 5.7 and 5.9.

See Figure 5.1 for an illustration. Adding slack variables results in

$$\begin{array}{lrrrrl} \max & & x_2 & & & \\ \text{s.t.} & 4x_1 & +x_2 & +x_3 & & = 4 \\ & -4x_1 & +x_2 & & +x_4 & = 0 \\ & & & x_1, x_2, x_3, x_4 & & \geq 0 \\ & & & x_1, x_2, x_3, x_4 & & \in \mathbb{Z}. \end{array}$$

The optimal basis for the LP relaxation is $B = \{1, 2\}$ with

$$A_{\cdot B} = \begin{pmatrix} 4 & 1 \\ -4 & 1 \end{pmatrix}, \quad A_{\cdot N} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_{\cdot B}^{-1} = \bar{A}_{\cdot N} = \begin{pmatrix} 1/8 & -1/8 \\ 1/2 & 1/2 \end{pmatrix}.$$

Hence, $\boldsymbol{x}_{\boldsymbol{B}}^\star = \bar{\boldsymbol{b}} = (A_{\cdot B})^{-1}\boldsymbol{b} = \binom{1/2}{2}$. The only fractional component is x_1 and for $i = 1$ (5.6) becomes

$$x_1 = \tfrac{1}{2} - \tfrac{1}{8}x_3 + \tfrac{1}{8}x_4 \in \mathbb{Z},$$

which is true for all integral solutions $\boldsymbol{x}$. Inequality (5.7) then becomes

$$x_1 \leq \lfloor \tfrac{1}{2} \rfloor - \lfloor \tfrac{1}{8} \rfloor x_3 - \lfloor -\tfrac{1}{8} \rfloor x_4 = 0 - 0x_3 + x_4,$$

which reduces to $x_1 - x_4 \leq 0$. With respect to the original variables x_1, x_2, by replacing the slack variable $x_4 = 4x_1 - x_2$, we obtain

$$-3x_1 + x_2 \leq 0.$$

This is a valid inequality for

$$\mathcal{P}_I = \operatorname{conv}(\{\boldsymbol{x} \in \mathbb{Z}^2_{\geq 0} : 4x_1 + x_2 \leq 4,\ -4x_1 + x_2 \leq 0\}).$$

The alternative representation (5.8) is, with $x_3 = 4 - 4x_1 - x_2$,

$$\tfrac{1}{8}x_3 + \tfrac{7}{8}x_4 \geq \tfrac{1}{2} \quad \leftrightarrow \quad \tfrac{1}{2} - \tfrac{1}{2}x_1 - \tfrac{1}{8}x_2 + \tfrac{7}{2}x_1 - \tfrac{7}{8}x_2 \geq \tfrac{1}{2} \quad \leftrightarrow \quad 3x_1 - x_2 \geq 0. \qquad \triangle$$

5.1.3 Gomory's mixed-integer cuts

The fractional Gomory cuts presented in the last section are dominated by the so-called mixed-integer Gomory cuts, which we now derive. We need the following observation.

Observation 5.8. Let $(\boldsymbol{a}^{(k)})^\top \boldsymbol{x} \geq \beta^{(k)}$ be a valid inequality for the polyhedron $\mathcal{P}^{(k)} \subseteq \mathbb{R}^n_{\geq 0}$ for $k \in \{1,2\}$. Then,

$$\sum_{i=1}^{n} \max(a_i^{(1)}, a_i^{(2)})\, x_i \geq \min(\beta^{(1)}, \beta^{(2)})$$

is valid for $\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}$ and $\operatorname{conv}(\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)})$.

As in the previous section, we start with (5.6) and split the sum on the right to obtain

$$x_i = \bar{b}_i - \sum_{j \in N : f_{ij} \leq f_i} \bar{A}_{ij}\, x_j - \sum_{j \in N : f_{ij} > f_i} \bar{A}_{ij}\, x_j, \tag{5.9}$$

where we again use the fractional values $f_i \in (0,1)$ of $\bar{b}_i$ (since $x_i^\star \notin \mathbb{Z}$) and $f_{ij} \in [0,1)$ of $\bar{A}_{ij}$. For j with $f_{ij} \leq f_i$ we use $\bar{A}_{ij} = \lfloor \bar{A}_{ij} \rfloor + f_{ij}$. For j with $f_{ij} > f_i$, and thus $f_{ij} > 0$, we use $\bar{A}_{ij} = \lceil \bar{A}_{ij} \rceil - 1 + f_{ij}$. Inserted in (5.9) these yield that all feasible $\boldsymbol{x}$ satisfy

$$\sum_{j \in N : f_{ij} \leq f_i} f_{ij}\, x_j + \sum_{j \in N : f_{ij} > f_i} (f_{ij} - 1)\, x_j = r + f_i,$$

where

$$r = \lfloor \bar{b}_i \rfloor - \sum_{f_{ij} \leq f_i} \lfloor \bar{A}_{ij} \rfloor\, x_j - \sum_{f_{ij} > f_i} \lceil \bar{A}_{ij} \rceil\, x_j - x_i \in \mathbb{Z}.$$

Now, either $r \geq 0$ or $r \leq -1$ must hold. Therefore, every feasible point $\boldsymbol{x} \in \mathbb{Z}^n_{\geq 0}$ with $A\boldsymbol{x} = \boldsymbol{b}$ either satisfies the inequality

$$\sum_{j \in N : f_{ij} \leq f_i} \frac{f_{ij}}{f_i}\, x_j - \sum_{j \in N : f_{ij} > f_i} \frac{1 - f_{ij}}{f_i}\, x_j \geq 1$$

or

$$-\sum_{j \in N : f_{ij} \leq f_i} \frac{f_{ij}}{1 - f_i}\, x_j + \sum_{j \in N : f_{ij} > f_i} \frac{1 - f_{ij}}{1 - f_i}\, x_j \geq 1.$$

According to Observation 5.8, we can combine both inequalities to obtain the *Gomory mixed-integer cut*

$$\sum_{j \in N : f_{ij} \leq f_i} \frac{f_{ij}}{f_i}\, x_j + \sum_{j \in N : f_{ij} > f_i} \frac{1 - f_{ij}}{1 - f_i}\, x_j \geq 1. \tag{5.10}$$

We have just shown that this cutting plane is valid for all feasible $\boldsymbol{x}$.

Because

$$\frac{1 - f_{ij}}{1 - f_i} < 1 < \frac{f_{ij}}{f_i} \quad \text{for} \quad f_{ij} > f_i,$$

the cut (5.10) dominates the fractional Gomory cuts (5.8). In particular, (5.10) cuts off the current solution of the LP relaxation.

Example 5.9. We consider the same IP as in Example 5.7 and again get

$$x_1 = \tfrac{1}{2} - \tfrac{1}{8} x_3 + \tfrac{1}{8} x_4.$$

We have $f_{13} = \frac{1}{8}$, $f_{14} = \frac{7}{8}$, $f_1 = \frac{1}{2}$, i.e., the two cases $f_{13} \leq f_1$ and $f_{14} > f_1$ occur. We obtain the mixed-integer Gomory cut (5.10) as

$$\frac{\frac{1}{8}}{\frac{1}{2}} x_3 + \frac{1 - \frac{7}{8}}{1 - \frac{1}{2}} x_4 \geq 1 \qquad \Leftrightarrow \qquad \frac{1}{4} x_3 + \frac{1}{4} x_4 \geq 1 \qquad \Leftrightarrow \qquad x_3 + x_4 \geq 4.$$

Elimination of $x_3 = 4 - 4x_1 - x_2$ and $x_4 = 4x_1 - x_2$ results in

$$x_3 + x_4 \geq 4 \quad \leftrightarrow \quad 4 - 4x_1 - x_2 + 4x_1 - x_2 \geq 4 \quad \Leftrightarrow \quad -2x_2 \geq 0 \quad \Leftrightarrow \quad x_2 \leq 0.$$

In this case, we directly reach the integer hull $\mathcal{P}_I$ (see Figure 5.1).

For the two inequalities of the disjunction $r \geq 0$ or $r \leq -1$ we get $x_3 - x_4 \geq 4$ or $-x_3 + x_4 \geq 4$ respectively. With regard to the original variables, this corresponds to $x_1 \leq 0$ and $x_1 \geq 1$. The mixed-integer Gomory cut is exactly valid for this disjunction: In the left part, the intersection of the corresponding inequality with the polyhedron is $\{\binom{0}{0}\}$, in the right part it is $\{\binom{1}{0}\}$. △

The name "mixed-integer Gomory cut" indicates that these inequalities were developed for the case in which, in addition to integer variables, continuous variables may occur (i.e., $p < n$). If $I = \{1, \dots, p\} \cap N$ is the set of indices of integer variables in the non-basis N, the general form of the cut is

$$\sum_{j \in I : f_{ij} \leq f_i} \frac{f_{ij}}{f_i} z_j + \sum_{j \in I : f_{ij} > f_i} \frac{1 - f_{ij}}{1 - f_i} z_j + \sum_{j \in N \setminus I : \bar{A}_{ij} > 0} \frac{\bar{A}_{ij}}{f_i} z_j - \sum_{j \in N \setminus I : \bar{A}_{ij} < 0} \frac{\bar{A}_{ij}}{1 - f_i} z_j \geq 1. \tag{5.11}$$

Remark 5.10. In the mixed-integer case, the ideas of the previous two sections do not work. Chvátal's approach does not work because the right-hand side in (5.3) cannot be rounded down. Gomory's approach fails because the argument for rounding (5.7) no longer holds.

It can be shown (see [19]) that an algorithm based on iterative addition of the inequalities (5.11) solves $\min\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathcal{X}\}$ with $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{Z}^p_{\geq 0} \times \mathbb{R}^{n-p}_{\geq 0} : A\boldsymbol{x} = \boldsymbol{b}\}$ in a finite number of steps if $\boldsymbol{c}^\top \boldsymbol{x} \in \mathbb{Z}$ holds for all $\boldsymbol{x} \in \mathcal{X}$. In general, it is currently not known whether a finite cutting plane method exists.

Various other inequalities have been considered independently of a specific problem structure. Among these are for example, mixed-integer rounding cuts (MIR) (see [34]) and so-called lift-and-project cuts (see [3]).

5.2 Specialized cuts

After dealing with valid inequalities for general IPs and MIP's let us focus our attention on a single or a small subset of such constraints, where a problem might exhibit some local structure. For example, all variables in a constraint might be binary, or a MIP may contain a network flow problem as a part of its constraints. We are looking for ways to obtain stronger inequalities by exploiting such local structures.

5.2.1 Inequalities for knapsack problems

Every row of $A\boldsymbol{x} \leq \boldsymbol{b}$ defines a knapsack problem (see Example 1.6). We first attempt to derive inequalities for this problem that can then be used for the entire problem. To simplify the representation, we restrict ourselves to binary problems, i.e., problems of the form

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n, \end{aligned}$$

with $A \in \mathbb{Q}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Q}^m$, $\boldsymbol{c} \in \mathbb{Q}^n$.

Now, we select a row $\boldsymbol{a}^\top \boldsymbol{x} \leq \beta$ from $A\boldsymbol{x} \leq \boldsymbol{b}$ and consider the polytope

$$\mathcal{R} := \{\boldsymbol{x} \in [0,1]^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta\}.$$

We want to find valid inequalities for $\mathcal{R}_I$. Because

$$\mathcal{R}_I = \text{conv}(\{\boldsymbol{x} \in \{0,1\}^n : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta\}) \supseteq \text{conv}(\{\boldsymbol{x} \in \{0,1\}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}),$$

these inequalities are also valid for the original problem.

To simplify presentation, we bring the knapsack problem into a specific format. Recall that the knapsack problem is NP-hard (see *Algorithmic Discrete Mathematics*).

Observation 5.11. We may assume that

(a) $\boldsymbol{a}$ and β are integral, which can be achieved by multiplication by the least common multiple of the denominators,

(b) $\boldsymbol{a} \geq \boldsymbol{0}$, otherwise we replace x_j by $1 - x_j$ for all j with $a_j < 0$,

(c) $0 \leq \beta < \boldsymbol{1}^\top \boldsymbol{a}$, otherwise the problem is not interesting, because either $\beta \geq a_1 + \cdots + a_n$ and all $\boldsymbol{x} \in \{0,1\}^n$ are feasible, or $\beta < 0$ and no $\boldsymbol{x}$ is feasible because we ensured $\boldsymbol{a} \geq \boldsymbol{0}$ above.

(d) $a_j \leq \beta$ for all $j \in \{1,\ldots,n\}$, otherwise $x_j = 0$ in every feasible solution and x_j can be ignored,

(e) $\boldsymbol{c} \geq \boldsymbol{0}$ when determining $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta\}$, otherwise $x_j = 0$ for every optimum solution (because of $a \geq 0$) for all $j \in \{1,\ldots,n\}$ with $c_j < 0$. Furthermore, we can assume by scaling that $\boldsymbol{c}$ is integral.

Definition 5.12. An index set $C \subseteq \{1,\ldots,n\}$ is a *cover*, if

$$a(C) := \sum_{j \in C} a_j > \beta.$$

The cover C is *minimal* if no $C' \subsetneq C$ is a cover. The corresponding inequality

$$x(C) := \sum_{j \in C} x_j \leq |C| - 1$$

is called *cover inequality*.

Note that the cover inequality and is valid for $\mathcal{R}_I$, because for integral values it demands that $x_j = 0$ for at least one $j \in C$.

Two questions arise: How strong are these inequalities? How to find a (minimal) cover? We first focus on the former question.

The potential of an inequality for accelerating the solution process is difficult to assess. Presumably, however, facets of $\mathcal{R}_I$ are particularly important, because once we know them, we can solve the integer optimization problem easily. So we could hope that cover inequalities define facets of $\mathcal{R}_I$. It will turn out that this is not always the case. We can show the following, however.

Theorem 5.13 ([2, 22, 36, 43])**.** A minimal cover inequality $x(C) \leq |C| - 1$ defines a facet of the restricted knapsack polytope

$$\mathcal{R}_I(C) := \text{conv}(\{\boldsymbol{x} \in \{0,1\}^C : \boldsymbol{a}_C^\top \boldsymbol{x} \leq \beta\}).$$

Proof. We first need to determine $\dim \mathcal{R}_I(C)$. According to Observation 5.11, we may assume $a_j \leq \beta$ for all $j \in C$ and thus the vectors $\mathbf{e}^j$ are feasible for all $j \in C$. Together with 0 they form $|C|+1$ affine independent vectors. Thus $\dim \mathcal{R}_I(C) = |C|$ follows.

For the cover inequality to define a facet, there must be a number of $\dim \mathcal{R}_I(C) = |C|$ affinely independent vectors of $\mathcal{R}_I(C)$ that satisfy it with equality. Vectors with this property are $\mathbf{1} - \mathbf{e}^j \in \mathbb{Z}^C$ for $j \in C$, which are in $\mathcal{R}_I(C)$, because C is a minimal cover. □

As mentioned above, cover inequalities do not always define facets of the general knapsack polytope $\mathcal{R}_I$. However, this can be addressed by *lifting* the inequalities. For this, we, in general, calculate the largest possible coefficient d_k for a variable x_k with coefficient 0 in the inequality, i.e., in our case $k \notin C$, and add it to the inequality to obtain

$$\sum_{j \in C} x_j + d_k x_k \leq |C| - 1. \tag{5.12}$$

The new coefficient d_k must be chosen such that the inequality is valid. However, it should also be as large as possible to strengthen the inequality as much as possible. The largest possible coefficient can be found by solving

$$d_k = |C| - 1 - \max\{\mathbf{1}^\top \boldsymbol{x} \,:\, \boldsymbol{a}_C^\top \boldsymbol{x} \leq \beta - a_k,\ \boldsymbol{x} \in \{0,1\}^C\}. \tag{5.13}$$

If $x_k = 0$ holds for some point $\boldsymbol{x} \in \mathcal{R}_I$, then it obviously satisfies (5.12). If $x_k = 1$, then $\boldsymbol{a}_C^\top \boldsymbol{x} \leq \beta - a_k$ must hold. We optimize exactly over these constraints while demanding that (5.12) remains satisfied. Note that (5.13) again defines a knapsack problem.

Example 5.14. Let the inequality $\boldsymbol{a}^\top \boldsymbol{x} \leq \beta$ be

$$6x_1 + 5x_2 + 5x_3 + 5x_4 + 8x_5 \leq 16.$$

The set $C = \{1,2,3,4\}$ forms a (minimal) cover because $a(C) = 21 > 16$. The corresponding cover inequality is $x(C) \leq 3$. We now consider the only candidate $k = 5$ and calculate

$$d_5 = 3 - \max\{\{x_1 + x_2 + x_3 + x_4 \,:\, 6x_1 + 5x_2 + 5x_3 + 5x_4 \leq 8,\ x_1, \ldots, x_4 \in \{0,1\}\}.$$

The maximum is 1 and therefore $d_5 = 2$. We obtain the lifted inequality

$$x_1 + x_2 + x_3 + x_4 + 2x_5 \leq 3. \qquad \triangle$$

If we iterate the procedure described above, we obtain a set $C' \subseteq \{1, \ldots, n\} \setminus C$ and the valid inequality

$$\sum_{j \in C} x_j + \sum_{j \in C'} d_j x_j \leq |C| - 1. \tag{5.14}$$

The lifting problem (5.13) changes for $k \notin C \cup C'$ to

$$d_k = |C| - 1 - \max\{\sum_{j \in C} x_j + \sum_{j \in C'} d_j x_j \,:\, \sum_{j \in C \cup C'} a_j x_j \leq \beta - a_k,\ \boldsymbol{x} \in \{0,1\}^{C \cup C'}\}. \tag{5.15}$$

By construction, $d_k \in \{0, \ldots, |C| - 1\}$ holds. We obtain the following theorem.

Theorem 5.15. The lifted inequality (5.14) defines a facet of $\mathcal{R}_I(C \cup C')$. If $C \cup C' = \{1, \ldots, n\}$, then (5.14) defines a facet of $\mathcal{R}_I$.

Proof. We proceed by induction over $|C'|$. The result follows for $|C'| = 0$ from Theorem 5.13. Now let k be the index that is to be added to C'. Because, by induction, (5.14) defines a facet of $\mathcal{R}_\mathrm{I}(C \cup C')$, there are affinely independent vectors $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^t \in \{0,1\}^{C \cup C'}$ of $\mathcal{R}_\mathrm{I}$ satisfying it with equality, where $t = |C| + |C'|$ since $\mathcal{R}_\mathrm{I}(C \cup C')$ is full-dimensional as in Theorem 5.13. Now we define vectors $\hat{\boldsymbol{v}}^1, \ldots, \hat{\boldsymbol{v}}^t \in \{0,1\}^{C \cup C' \cup \{k\}}$ via

$$\hat{v}_i^j := \begin{cases} 0 & \text{if } i = k, \\ v_i^j, & \text{otherwise,} \end{cases}$$

for $j' \in \{1, \ldots, t\}$ and $i \in C \cup C' \cup \{k\}$. Then, $\hat{\boldsymbol{v}}^1, \ldots, \hat{\boldsymbol{v}}^t$ each fulfill the lifted inequality

$$\sum_{j \in C} x_j + \sum_{j \in C'} d_j x_j + d_k x_k \leq |C| - 1 \tag{5.16}$$

with equality.

Now let $\boldsymbol{x}^\star$ be an optimum solution for the maximum in (5.15) when calculating d_k. We define the vector $\hat{\boldsymbol{x}}^\star$ via

$$\hat{x}_i^\star := \begin{cases} 1 & \text{if } i = k, \\ x_i^\star & \text{otherwise,} \end{cases}$$

for $i \in C \cup C' \cup \{k\}$. By construction, $\hat{\boldsymbol{x}}^\star$ satisfies the inequality (5.16) with equality. Therefore $\hat{\boldsymbol{v}}^1, \ldots, \hat{\boldsymbol{v}}^t, \hat{\boldsymbol{x}}^\star$ together form $t+1$ affinely independent vectors, which proves the theorem because of $\dim \mathcal{R}_\mathrm{I}(C \cup C' \cup \{k\}) = |C| + |C'| + 1$. □

To summarize, from minimal covers we can, step-by-step, determine facets of $\mathcal{R}_\mathrm{I}$. However, we have not yet discussed how to find minimal covers or how to solve the lifting problems. Unfortunately, solving knapsack problems is NP-hard (see *Algorithmic Discrete Mathematics*). However, we can use the dynamic program (DP) given below. Note that, by Observation 5.11, we may assume to $\boldsymbol{a}, \boldsymbol{c} \in \mathbb{N}^n$, $\beta \in \mathbb{N}$.

Algorithm: Dynamic program for the knapsack problem.

input: $\boldsymbol{a}, \boldsymbol{c} \in \mathbb{N}^n$; $\beta \in \mathbb{N}$, upper bound $\bar{c} \in \mathbb{N}$ on optimal value
output: optimum value of $\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta, \boldsymbol{x} \in \{0,1\}^n\}$

$z(0,0) \leftarrow 0$, $z(0,k) \leftarrow \infty \; \forall k \in \{1, \ldots, \bar{c}\}$
for $j \leftarrow 1, \ldots, n$**:**
  for $k \leftarrow 0, \ldots, c_j - 1$**:**
    $z(j,k) \leftarrow z(j-1,k)$
  for $k \leftarrow c_j, \ldots, \bar{c}$**:**
    $z(j,k) \leftarrow \min\{z(j-1, k-c_j) + a_j, z(j-1,k)\}$
return $\max\{k \in \{0, \ldots, \bar{c}\} : z(n,k) \leq \beta\}$

We prove correctness of the DP.

Theorem 5.16. The above DP solves the knapsack problem in time $\mathcal{O}(n \cdot \bar{c})$ when given an upper bound $\bar{c}$ on optimum value, e.g., $\bar{c} = \mathbf{1}^\top \boldsymbol{c}$.

Proof. The algorithm runs in pseudo polynomial time $\mathcal{O}(n \cdot \bar{c})$. It uses a DP table where entry $z(j,k)$ is intended to hold the smallest weight of a partial solution only using the first j variables and reaching objective function value $k \in \{0,\dots,\bar{c}\}$, and ∞ if no such solution exists, i.e.,

$$z(j,k) = \min\{a(S) : a(S) \leq \beta,\ c(S) = k,\ S \subseteq \{1,\dots,j\}\}.$$

We have $z(j,k) \leq z(j-1,k)$, because it is always possible to discard item j. A solution of smaller weight is available when using the j-th item if and only if $k \geq c_j$, $z(j-1,k-c_j)+a_j \leq \beta$ and $z(j-1,k-c_j)+a_j < z(j-1,k)$. Hence, the DP computes the entries of the table correctly and therefore obtains the correct result overall.

An optimum solution can be found, as usual, via backtracking (see *Algorithmic Discrete Mathematics*). □

The smallest minimal cover can be found by solving the knapsack problem

$$\begin{aligned}&\min\{\mathbf{1}^\top \boldsymbol{x} : \boldsymbol{a}^\top \boldsymbol{x} \geq \beta + 1, \boldsymbol{x} \in \{0,1\}^n\}\\ &= n - \max\{\mathbf{1}^\top \boldsymbol{y} : \boldsymbol{a}^\top \boldsymbol{y} \leq \mathbf{1}^\top \boldsymbol{a} - \beta - 1, \boldsymbol{y} \in \{0,1\}^n\},\end{aligned}$$

where we substituted $\boldsymbol{y} := \mathbf{1} - \boldsymbol{x}$. Because in this case $\bar{c} = n$ is polynomial in the input size, the DP runs in polynomial time. Note that, we can also obtain minimal (but not always the smallest) cover by iteratively removing elements as long as possible.

The lifting problem (5.15) again involves a knapsack problem, which can again be solved by our dynamic program. The optimum objective function value is polynomially bounded by $\bar{c} := |C| - 1$. Overall, using Theorem 5.16 we can lift cover inequalities in polynomial time.

Corollary 5.17. The computation of a fully lifted cover inequality is possible in time $\mathcal{O}(n^2\,|C|) \subseteq \mathcal{O}(n^3)$.

However, it is difficult to guarantee that the corresponding inequality truncates a given fractional point, in fact is NP-complete to decide whether a lifted cover inequality exists that excludes a given point (see [31]). Note that this does not follow from the equivalence of optimization and separation for the NP-complete knapsack problem (see *Introduction to Optimization*), since we restrict ourselves to inequalities of a specific type. The hardness of the specific separation problem means that, in practice, we have to resort to heuristics for lifting and have to terminate the lifting procedure after some number of lifting operations.

Remark 5.18. There are many other classes of inequalities for the knapsack polytope, e.g., the so-called $(1,k)$-configuration or extended-weight inequalities (see [37, 42, 33]).

5.2.2 Inequalities for set-packing problems

Mixed-integer problems often contain constraints that only involve binary variables and coefficients. In particular, many binary programs contain logical inequalities of the form $x_i + x_j \leq 1$ or $x_i \leq x_j$. Furthermore, preprocessing routines (see Section 3.2.3) for integer problems can automatically extract implicit logical conditions. Such conditions occur, for example, when the binary variables x_i model whether a certain combination of resources is used. Conflicts between two combinations x_i and x_j are then expressed by $x_i + x_j \leq 1$.

This leads to binary programs with binary coefficient matrices. The study of such problems and, in particular, set-packing problems plays an important role in combinatorial optimization. A deep theory has been developed for these problems, which deals with notions such as perfect, ideal, or balanced matrices, perfect graphs, blocking and anti-blocking polyhedra, independence systems and semidefinite optimization.

The focus of this section is on the (partial) description of the associated polyhedra by means of inequalities. Since relaxations of many integer problems lead to set-packing problems, understanding these polyhedra can often lead to improved formulations of the initial problem.

Definition 5.19. The *set-packing problem* is given by

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} \leq \mathbf{1},\, \boldsymbol{x} \in \{0,1\}^n\}$$

with $A \in \{0,1\}^{m\times n}$ and $\boldsymbol{c} \in \mathbb{R}^n$, cf. Example 1.7.

Remark 5.20. Every column $\boldsymbol{A}_{\cdot j}$ of A can be viewed as the incidence vector of a subset $F_j \subseteq \{1,\dots,m\}$, i.e., $F_j := \{i \in \{1,\dots,m\} \,:\, A_{ij} = 1\}$. With this interpretation, the set-packing problem consists in finding a maximum (wrt. $\boldsymbol{c}$) selection of sets from $F_1,\dots,F_n$ that are pairwise disjoint.

Feasible solutions of the set-packing problem have a nice graph-theoretic interpretation. We introduce a vertex for every column of A and an edge $\{j,\ell\}$ between two vertices j and ℓ if their corresponding columns have a 1 in the same row.

Definition 5.21. The *(column) conflict graph* $G(A) = (V,E)$ of $A \in \{0,1\}^{m\times n}$ is defined by $V := \{1,\dots,n\}$ and

$$E := \{\{j,\ell\} \,:\, \exists\, i \in \{1,\dots,m\} \colon A_{ij} = A_{i\ell} = 1\}.$$

Obviously, every feasible $\boldsymbol{x} \in \{0,1\}^n$ that satisfies the inequality $A\boldsymbol{x} \leq \mathbf{1}$ is the incidence vector of a stable set in the graph $G(A)$. Conversely, the incidence vector of a stable set in $G(A)$ is a feasible solution of the set-packing problem $A\boldsymbol{x} \leq \mathbf{1}$. In other words, the study of stable sets in graphs is equivalent to the study of the set-packing problem.

Now consider $A \in \{0,1\}^{m\times n}$ and denote the *set-packing polytope* by

$$\mathcal{P}(A) = \operatorname{conv}(\{\boldsymbol{x} \in \{0,1\}^n \,:\, A\boldsymbol{x} \leq \mathbf{1}\}).$$

Let $G(A) = (V,E)$ be the conflict graph of A. From our previous considerations it follows that

$$\mathcal{P}(A) = \operatorname{conv}(\{\boldsymbol{x} \in \{0,1\}^n \,:\, x_j + x_\ell \leq 1 \;\forall \{j,\ell\} \in E\}) =: \mathcal{P}(G),$$

where the set in the convex hull consists of the stable sets in G, and $\mathcal{P}(G)$ is the *stable-set polytope*. Expressed differently, two matrices A and A' yield the same set-packing problem if and only if their corresponding conflict graphs coincide. We can therefore consider $\mathcal{P}(A)$ via the graph G and denote both the set-packing and the stable set polytope by $\mathcal{P}(G)$.

First, we make a few simple observations regarding $\mathcal{P}(G)$.

Proposition 5.22.

(a) $\mathcal{P}(G)$ is full-dimensional, i.e., $\dim(\mathcal{P}(G)) = n$.

(b) $\mathcal{P}(G)$ is downwards monotone, i.e., $\boldsymbol{x} \in \mathcal{P}(G)$ implies $\boldsymbol{y} \in \mathcal{P}(G)$ for all $\mathbf{0} \leq \boldsymbol{y} \leq \boldsymbol{x}$.

(c) All facets of $\mathcal{P}(G)$ that are not facets of $[0,1]^n$ have non-negative coefficients, i.e., if $\boldsymbol{a}^\top \boldsymbol{x} \leq \beta$ defines such a facet, then $\boldsymbol{a} \geq \mathbf{0}$ and $\beta \geq 0$.

(d) The non-negativity conditions $x_j \geq 0$ induce facets of $\mathcal{P}(G)$.

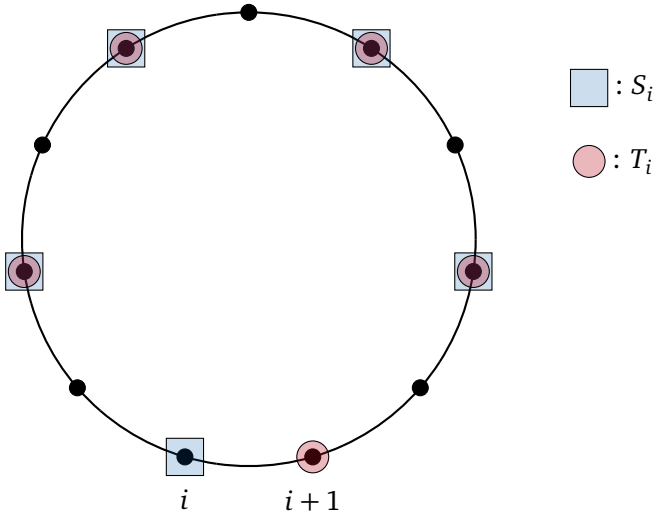

Figure 5.2: Illustration of the sets S_i and T_i in the proof of Theorem 5.23. The sets differ only by the elements i and $i+1$ (modulo k).

Proof. exercise. □

From Theorems 4.15 and 4.8 we know that if G is bipartite, then the edge and the non-negativity conditions are sufficient to fully describe $\mathcal{P}(G)$. Conversely, non-bipartite graphs contain odd cycles generating new valid inequalities that are not induced by linear combinations of edge inequalities. By lifting, similarly to Theorem 5.15, we obtain facets of $\mathcal{P}(G)$.

Theorem 5.23 ([35])**.** Let $K = (V_K, E_K) \subseteq G$ be a cycle of odd length in an undirected graph G. Then, the *odd-cycle inequality*

$$\sum_{v \in V_K} x_v \leq \frac{|E_K| - 1}{2}, \tag{5.17}$$

is valid for $\mathcal{P}(G)$. It defines a facet of $\mathcal{P}(G[V_K])$ for the induced subgraph $G[V_K]$ if and only if K is a *hole*[a].

[a] A *hole* is an induced subgraph that is a cycle.

Proof. The validity of the inequality is clear.

Consider an odd hole $K = (V_K, E_K)$ with, without loss of generality, $V_K = \{0, \dots, k-1\}$ and $E_K = \big\{\{0,1\}, \{1,2\}, \dots, \{k-2, k-1\}, \{k-1, 0\}\big\}$.

We first observe that

$$\mathcal{F} := \{\boldsymbol{y} \in \mathcal{P}(G[V_K]) : \mathbf{1}^\top \boldsymbol{y} = \frac{k-1}{2}\}$$

is a nontrivial face of $P(G[V_K])$. If it is not a facet, $\mathcal{F}$ is part of a facet $\mathcal{F}' := \{\boldsymbol{y} \in \mathcal{P}(G[V_K]) : \boldsymbol{b}^\top \boldsymbol{y} = \beta\} \supseteq \mathcal{F}$ of $\mathcal{P}(G[V_K])$. We consider the following sets for $i \in V_K$ (see Figure 5.2):

$$S_0 = \{0, 3, 5, \dots, k-2\}, \qquad S_i = (S_0 + i) \mod k,$$
$$T_0 = \{1, 3, 5, \dots, k-2\}, \qquad T_i = (T_0 + i) \mod k.$$

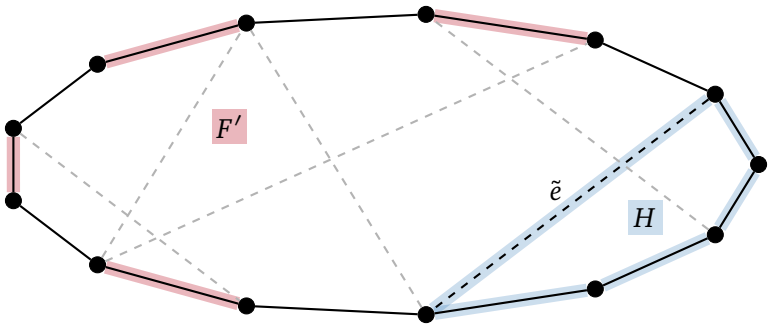

Figure 5.3: Example for the sets H and F' in the proof of Theorem 5.23. In the example, the cycle K has length 13.

(*Example:* For $k = 7$, $S_0 = \{0,3,5\}$, $S_1 = \{1,4,6\}$, $S_2 = \{0,2,5\}$, …, $S_6 = \{2,4,6\}$, $T_0 = \{1,3,5\}$, $T_1 = \{2,4,6\}$, $T_2 = \{0,3,5\}$, …, $T_6 = \{1,2,4\}$).

Since K has no chords, all sets are stable with respect to G and $\mathbf{1}^\top \boldsymbol{\chi}^{S_i} = \mathbf{1}^\top \boldsymbol{\chi}^{T_i} = \frac{k-1}{2}$, where $\boldsymbol{\chi}^S := \sum_{i \in S} \mathbf{e}^i \in \{0,1\}^{V_K}$ denotes the characteristic vector of the set S . From this it follows that $\boldsymbol{\chi}^{S_i}, \boldsymbol{\chi}^{T_i} \in \mathcal{F} \subseteq \mathcal{F}'$ and thus, for $i < k-1$,

$$0 = \boldsymbol{b}^\top \boldsymbol{\chi}^{S_i} - \boldsymbol{b}^\top \boldsymbol{\chi}^{T_i} = \boldsymbol{b}^\top \boldsymbol{\chi}^{S_i \setminus T_i} - \boldsymbol{b}^\top \boldsymbol{\chi}^{T_i \setminus S_i} = \boldsymbol{b}^\top \mathbf{e}^i - \boldsymbol{b}^\top \mathbf{e}^{i+1} = b_i - b_{i+1},$$

and $0 = b_i - b_0$ for $i = k-1$. Since i was chosen arbitrarily, it follows that $b_i = b_j$ for all $i, j \in V_K$. This means that $\boldsymbol{b}^\top \boldsymbol{x} \leq \beta$ is the odd-cycle inequality up to a positive factor, since both have the same left-hand side up to a positive (by Proposition 5.22 (c)) factor and $\mathcal{F} \subseteq \mathcal{F}'$. It follows that $\mathcal{F} = \mathcal{F}'$, hence $\mathcal{F}$ is a facet of $\mathcal{P}(G')$.

Now assume that the cycle K contains a chord $\tilde{e} \in G[V_K] \setminus E_K$ in $G[V_K]$ (see Figure 5.3). Then, $E_K \cup \{\tilde{e}\}$ consists of two cycles that have $\tilde{e}$ in common, where exactly one of the cycles has odd length. Let $H = (V_H, E_H)$ be this odd cycle and let $E_F = \{e \in K : e \cap V_H = \emptyset\}$. Then, $|E_F| = (|E_K| + 1) - |E_H| - 2 = |E_K| - |E_H| - 1$ is odd. Hence, there are pairwise disjoint edges, i.e., a matching, $M \subseteq E_F$ with $|M| = \frac{|E_F|+1}{2} = \frac{|E_K|-|E_H|}{2}$. Now, the odd-cycle inequality (5.17) is the the sum of the valid inequalities $\sum_{v \in V(H)} x_v \leq \frac{|E_H|-1}{2}$ and $x_u + x_v \leq 1$ for all $\{u,v\} \in M$. The corresponding faces therefore each contain the face belonging to (5.17) (if (5.17) is satisfied with equality, each of the other inequalities must also be satisfied with equality). Thus, (5.17) cannot induce a facet of $\mathcal{P}(G[V_K])$. □

Remark 5.24. Odd-cycle inequalities can be separated in polynomial time as follows (see [21, Lemma 9.1.11]). We create from the conflict graph $G(A) = (V, E)$ a bipartite auxiliary graph $G' = (V \cup V', E')$, where V' contains a vertex v' for every vertex $v \in V$ and $E' := \{\{u, v'\} : \{u, v\} \in E\}$, i.e, we replace every edge $\{u, v\} \in E$ by the two edges $\{u, v'\}$ and $\{v, u'\}$. Let $\boldsymbol{x}^\star \in [0,1]^V$ be an optimum solution of the LP relaxation of the set-packing problem. We define edge weights via $w_{\{u,v'\}} = \frac{1}{2}(1 - x^\star_u - x^\star_v) \geq 0$ for all $\{u, v'\} \in E'$ with $u \in V$ and $v' \in V'$.

Odd cycles $K = (V_K, E_K)$ in $G(A)$ correspond to minimal paths K' from a vertex $s \in V$ to the corresponding vertex $s' \in V'$ in G', in particular, K' thus contains at most one of the vertices v, v' for all $v \in V \setminus \{s\}$. The weight of K' is

$$w(K') = \sum_{e \in K'} w_e = \sum_{\{u,v'\} \in K'} \tfrac{1}{2}(1 - x^\star_u - x^\star_v) = \frac{|E_K|}{2} - \sum_{v \in V_K} x^\star_v,$$

because every $x^\star_v$ occurs twice in the sum. This means that

$$w(K') < \tfrac{1}{2} \quad \Leftrightarrow \quad \frac{|E_K| - 1}{2} < \sum_{v \in V_K} x^\star_v.$$

We can therefore find a violated odd-cycle inequality by looking for a v-v'-path in G' of weight less than $\frac{1}{2}$, e.g., using Dijkstra's algorithm (see *Algorithmic Discrete Mathematics*).

Graphs $G = (V, E)$, for which $\mathcal{P}(G)$ is completely characterized by the non-negativity constraints, the edge inequalities $x_i + x_j \leq 1$ for $\{i, j\} \in E$ and the odd-cycle inequalities are called *t-perfect* (see [10]). The class of t-perfect graphs contains a number of graphs, such as, e.g., series-parallel and bipartite graphs. In practice, odd-cycle inequalities, however, are usually of little help in finding integral optima.

Another important class of valid inequalities for the stable set polytope are clique inequalities. Recall that a clique is a complete subgraph.

Theorem 5.25. Let C be a clique in an undirected graph G. Then, the *clique inequality*

$$\sum_{j \in C} x_j \leq 1$$

is valid for $\mathcal{P}(G)$ and it defines a facet of $\mathcal{P}(G)$ if and only if C is maximal with respect to vertex inclusion.

Proof. exercise. □

Graphs G, for which $\mathcal{P}(G)$ is completely described by the non-negativity conditions and the clique inequalities, are called *perfect* (see [5]).

Unfortunately, the separation problem for the class of clique inequalities is NP-hard (see [21]). Surprisingly, there is a larger class of inequalities, the *orthonormal-representation inequalities*, that generalize the clique inequalities and can be separated in polynomial time. In addition to the cycle, clique and orthonormal-representation inequalities, a number of other inequalities are known for the stable set polytope, e.g., the blossom, odd antihole, wheel, antiweb and web, wedge, and many more inequalities (see [6]).

6 Decomposition Methods

The idea of decomposition methods is to remove a part (constraints and/or variables) from the problem and consider them separately in a so-called *master problem*. The remaining subproblem can often be solved more efficiently. Decomposition methods now work alternatingly on the master problem and the subproblem and iteratively exchange information in order to solve the initial problem optimally. In this section we look at three well-known examples of this approach: the Lagrangian relaxation/decomposition, Dantzig-Wolfe decomposition and Benders' decomposition. In Lagrangian relaxation/decomposition, parts of the constraint matrix are omitted and instead treated via the objective function. Dantzig-Wolfe decomposition and Benders' decomposition also omit part of the constraint matrix, but instead of treating it via the objective function, it is reformulated and reinserted into the constraints.

6.1 Lagrangian relaxation

In the previous chapter, we solved the mixed-integer problem by relaxing the integrality constraints and trying to increase integrality of the solution by adding cutting planes. The method considered in this section employs a different relaxation that relies on the so-called *Lagrangian function*.

The system of inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ is divided into two parts $A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}}$ and $A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}$ such that

$$A = \begin{pmatrix} A^{(1)} \\ A^{(2)} \end{pmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{b^{(1)}} \\ \boldsymbol{b^{(2)}} \end{pmatrix},$$

with $A \in \mathbb{Q}^{m \times n}$, $A^{(1)} \in \mathbb{Q}^{m_1 \times n}$, $A^{(2)} \in \mathbb{Q}^{m_2 \times n}$, $\boldsymbol{b^{(1)}} \in \mathbb{Q}^{m_1}$, $\boldsymbol{b^{(2)}} \in \mathbb{Q}^{m_2}$ and $m_1 + m_2 = m$. The general mixed-integer program (1.1) then becomes

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}}, \\ & A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}, \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}. \end{aligned} \tag{6.1}$$

The subdivision is chosen so that the part $A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}}$ contains the inequalities that are difficult to treat and should therefore be relaxed. More specifically, the relaxed subsystem will be handled via a penalty term in the objective rather than as constraints. The penalty will be set such that it ensures that the relaxed constraints will be implicitly fulfilled by every optimum solution.

For a fixed $\boldsymbol{\lambda} \in \mathbb{R}^{m_1}$, $\boldsymbol{\lambda} \geq \boldsymbol{0}$, consider the *Lagrangian function*

$$L(\boldsymbol{\lambda}) = \max\{\boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top (\boldsymbol{b^{(1)}} - A^{(1)}\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X}^{(2)}\}, \tag{6.2}$$

where $\mathcal{X}^{(2)} := \{\boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} : A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}\}$. The value $L(\boldsymbol{\lambda})$ is an upper bound for (6.1), since every feasible solution $\bar{\boldsymbol{x}}$ of (6.1) satisfies

$$\boldsymbol{c}^\top \bar{\boldsymbol{x}} \leq \boldsymbol{c}^\top \bar{\boldsymbol{x}} + \boldsymbol{\lambda}^\top (\boldsymbol{b^{(1)}} - A^{(1)}\bar{\boldsymbol{x}}) \leq \max_{\boldsymbol{x} \in \mathcal{X}^{(2)}} \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top (\boldsymbol{b^{(1)}} - A^{(1)}\boldsymbol{x}) = L(\boldsymbol{\lambda}),$$

where the first inequality holds because $A^{(1)}\bar{x} \leq b^{(1)}$ and $\lambda \geq 0$ imply $\lambda^\top(b^{(1)} - A^{(1)}\bar{x}) \geq 0$, i.e., a violation of $A^{(1)}x \leq b^{(1)}$ would decrease the objective function value and thus penalize it.

Since this applies for all $\lambda \geq 0$, the *Lagrangian relaxation*

$$\min_{\lambda \geq 0} L(\lambda) \tag{6.3}$$

yields the smallest upper bound for (6.1) of this kind. Sometimes, (6.3) is referred to as the *Lagrangian dual* (see Observation 6.4 for a motivation of the term).

But how can we compute an optimum solution $\lambda^\star$ of (6.3)? And how close is $L(\lambda^\star)$ to the optimum value of (6.1)? We first answer the latter question. In the following, observe that $\text{conv}(\mathcal{X}^{(2)}) = \mathcal{P}(A^{(2)}, b^{(2)})_{\text{I},p}$ is the integer hull of $\mathcal{P}(A^{(2)}, b^{(2)})$.

Theorem 6.1. It holds that

$$\min_{\lambda \geq 0} L(\lambda) = \max\{c^\top x : A^{(1)}x \leq b^{(1)},\ x \in \text{conv}(\mathcal{X}^{(2)})\}.$$

Proof. We obtain

$$L(\lambda) = \max\{c^\top x + \lambda^\top(b^{(1)} - A^{(1)}x) : x \in \mathcal{X}^{(2)}\} = \max\{c^\top x + \lambda^\top(b^{(1)} - A^{(1)}x) : x \in \text{conv}(\mathcal{X}^{(2)})\},$$

because the objective function is linear in x and $\text{conv}(\mathcal{X}^{(2)}) = \mathcal{P}(A^{(2)}, b^{(2)})_{\text{I},p}$ is a polyhedron (Theorem 2.32), i.e., the second maximum is attained at an extreme point, which cannot be a convex combination of other points in $\mathcal{X}^{(2)}$ (see *Introduction to Optimization*). It follows that

$$\begin{aligned}\min_{\lambda \geq 0} L(\lambda) &= \min_{\lambda \geq 0} \max\{c^\top x + \lambda^\top(b^{(1)} - A^{(1)}x) : x \in \text{conv}(\mathcal{X}^{(2)})\} \\ &= \min_{\lambda \geq 0} \left\{\lambda^\top b^{(1)} + \max\{(c^\top - \lambda^\top A^{(1)})x : x \in \text{conv}(\mathcal{X}^{(2)})\}\right\}.\end{aligned}$$

If $\mathcal{X}^{(2)} = \emptyset$, the inner maximum is defined as $-\infty$ for all λ and thus $\min_{\lambda \geq 0} L(\lambda) = -\infty$. Otherwise, by Theorems 2.32 and 2.19, vectors $v^{(1)}, \dots, v^{(k)} \in \mathbb{R}^n$ and $r^{(1)}, \dots, r^{(\ell)} \in \mathbb{R}^n$ exist such that $\text{conv}(\mathcal{X}^{(2)}) = \text{conv}(\{v^{(1)}, \dots, v^{(k)}\}) + \text{cone}(\{r^{(1)}, \dots, r^{(\ell)}\})$. This means that

$$\max\{(c^\top - \lambda^\top A^{(1)})x : x \in \text{conv}(\mathcal{X}^{(2)})\} = \begin{cases} +\infty & \text{if } \exists\, i \in \{1, \dots, \ell\}\colon (c^\top - \lambda^\top A^{(1)})r^{(i)} > 0, \\ \max\limits_{j \in \{1,\dots,k\}} (c^\top - \lambda^\top A^{(1)})v^{(j)} & \text{otherwise.} \end{cases} \tag{6.4}$$

It follows that

$$\min_{\lambda \geq 0} L(\lambda) = \min\{\lambda^\top b^{(1)} + \max_{j \in \{1,\dots,k\}} (c^\top - \lambda^\top A^{(1)})v^{(j)} : \lambda \geq 0, (c^\top - \lambda^\top A^{(1)})r^{(i)} \leq 0\ \forall i \in \{1, \dots, \ell\}\}. \tag{6.5}$$

Note that if the problem is infeasible, i.e. no $\lambda \geq 0$ with $(c^\top - \lambda^\top A^{(1)})r^{(i)} \leq 0$ for all $i \in \{1, \dots, \ell\}$ exists, then the value of the minimum is ∞ by convention, in accordance with (6.4). Otherwise, the value is finite and attained in a vertex.

Using an additional variable $\eta \in \mathbb{R}$ with $\eta \geq c^\top v^{(j)} + \lambda^\top(b^{(1)} - A^{(1)}v^{(j)})$ for all $j \in \{1, \dots, k\}$, we can reformulate (6.5) as

$$\begin{aligned}\min_{\lambda \geq 0} L(\lambda) = \min_{\lambda \geq 0,\, \eta \in \mathbb{R}}\ & \eta & & \\ \text{s.t.}\ & \eta - \lambda^\top(b^{(1)} - A^{(1)}v^{(j)}) \geq c^\top v^{(j)} & & \forall j \in \{1, \dots, k\}, \\ & \lambda^\top A^{(1)} r^{(i)} \geq c^\top r^{(i)} & & \forall i \in \{1, \dots, \ell\}.\end{aligned} \tag{6.6}$$

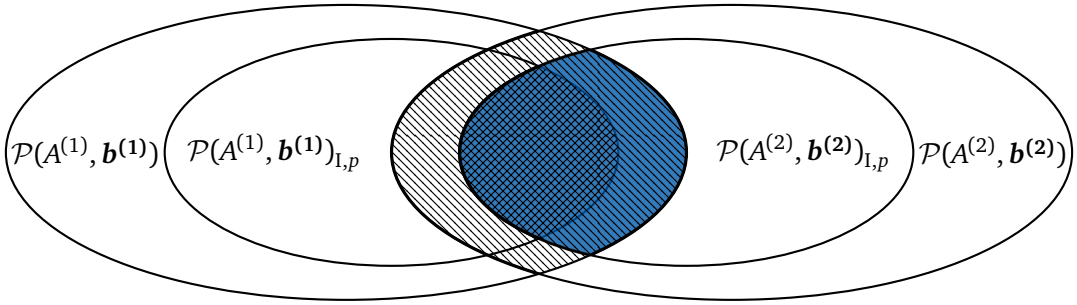

Figure 6.1: Illustration of the feasible region of the Lagrangian relaxation (blue). It contains the intersection of the integer hulls of $\mathcal{P}(A^{(1)}, \boldsymbol{b^{(1)}})$ and $\mathcal{P}(A^{(2)}, \boldsymbol{b^{(2)}})$ (crosshatch pattern) which contains the feasible region of the MIP. Conversely, the feasible region of the Lagrangian relaxation is contained in the feasible region of the LP relaxation (line pattern).

By strong duality, we finally obtain

$$\begin{aligned}
\min_{\boldsymbol{\lambda}\geq \mathbf{0}} L(\boldsymbol{\lambda}) = \max \quad & \boldsymbol{c}^\top \Big(\sum_{j=1}^{k} \boldsymbol{v^{(j)}} \alpha_j + \sum_{i=1}^{\ell} \boldsymbol{r^{(i)}} \beta_i \Big) \\
\text{s.t.} \quad & \sum_{j=1}^{k} \alpha_j = 1, \\
& A^{(1)} \Big(\sum_{j=1}^{k} \boldsymbol{v^{(j)}} \alpha_j + \sum_{i=1}^{\ell} \boldsymbol{r^{(i)}} \beta_i \Big) \leq \boldsymbol{b^{(1)}} \sum_{j=1}^{k} \alpha_j, \\
& \alpha_j \geq 0 \qquad \forall j \in \{1, \dots, k\}, \\
& \beta_i \geq 0 \qquad \forall i \in \{1, \dots, \ell\}.
\end{aligned}$$

Since $\sum_{j=1}^{k} \alpha_j = 1$, this is equivalent to

$$\begin{aligned}
\min_{\boldsymbol{\lambda}\geq \mathbf{0}} L(\boldsymbol{\lambda}) &= \max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}},\ \boldsymbol{x} \in \operatorname{conv}(\{\boldsymbol{v^{(1)}}, \dots, \boldsymbol{v^{(k)}}\}) + \operatorname{cone}(\{\boldsymbol{r^{(1)}}, \dots, \boldsymbol{r^{(\ell)}}\})\} \\
&= \max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}},\ \boldsymbol{x} \in \operatorname{conv}(\mathcal{X}^{(2)})\}. \qquad \square
\end{aligned}$$

Because of

$$\{\boldsymbol{x} \in \mathbb{R}^n \,:\, A\boldsymbol{x} \leq \boldsymbol{b}\} \supseteq \{\boldsymbol{x} \in \mathbb{R}^n \,:\, A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}},\ x \in \operatorname{conv}(\mathcal{X}^{(2)})\} \supseteq \operatorname{conv}(\{\boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \,:\, A\boldsymbol{x} \leq \boldsymbol{b}\}),$$

Theorem 6.1 implies the following, see Figure 6.1.

Corollary 6.2. It holds that $z_{\mathrm{MIP}} \leq \min_{\boldsymbol{\lambda}\geq \mathbf{0}} L(\boldsymbol{\lambda}) \leq z_{\mathrm{LP}}$, where z_{MIP} and z_{LP} denote the optimum solution values of the MIP (6.1) and its LP relaxation, respectively.

In particular, when $p = 0$, we have $z_{\mathrm{MIP}} = z_{\mathrm{LP}}$ and the Lagrangian relaxation coincides with the LP-relaxation. The latter is also true if the non-relaxed part of the problem induces an integral feasible region.

Observation 6.3. If $\mathcal{P}(A^{(2)}, \boldsymbol{b^{(2)}})$ is integral, i.e., $\operatorname{conv}(\mathcal{X}^{(2)}) = \mathcal{P}(A^{(2)}, \boldsymbol{b^{(2)}})$, Theorem 6.1 yields that the Lagrangian relaxation coincides with the LP relaxation, i.e.,

$$\min_{\boldsymbol{\lambda}\geq \mathbf{0}} L(\boldsymbol{\lambda}) = \max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, \boldsymbol{x} \in \mathcal{P}(A^{(1)}, \boldsymbol{b^{(1)}}) \cap \mathcal{P}(A^{(2)}, \boldsymbol{b^{(2)}})\} = \max\{\boldsymbol{c}^\top \boldsymbol{x} \,:\, A\boldsymbol{x} \leq \boldsymbol{b}\}.$$

We justify why $\min_{\boldsymbol{\lambda}\geq \mathbf{0}} L(\boldsymbol{\lambda})$ is called the Lagrangian *dual*.

Observation 6.4. If $p = 0$ and all inequalities are relaxed, i.e., $\mathcal{P}(A^{(1)}, \boldsymbol{b^{(1)}}) = \mathcal{P}(A, \boldsymbol{b})$, then

$$L(\boldsymbol{\lambda}) = \max\{\boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top(\boldsymbol{b} - A\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^n\} = \max\{(\boldsymbol{c}^\top - \boldsymbol{\lambda}^\top A)\boldsymbol{x} + \boldsymbol{\lambda}^\top \boldsymbol{b} : \boldsymbol{x} \in \mathbb{R}^n\}.$$

This problem is only bounded if $\boldsymbol{c} = A^\top \boldsymbol{\lambda}$, since $\boldsymbol{x}$ is unconstrained. The Lagrangian relaxation thus can be rewritten as

$$\min_{\boldsymbol{\lambda}\geq \mathbf{0}} L(\boldsymbol{\lambda}) = \min\{\boldsymbol{\lambda}^\top \boldsymbol{b} : A^\top \boldsymbol{\lambda} = \boldsymbol{c},\ \boldsymbol{\lambda} \geq \mathbf{0}\},$$

which corresponds exactly to the LP dual of $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}\}$.

Remark 6.5. Of course, the utility of the Lagrangian relaxation very much depends on the set of constraints being relaxed. On the one hand, we need to be able to compute $L(\boldsymbol{\lambda})$ reasonably quickly, which makes it desirable to relax as many difficult constraints as possible. On the other hand, the more constraints are relaxed, the worse the bound $L(\boldsymbol{\lambda}^\star)$ becomes.

We still need to address the computation of $\min L(\boldsymbol{\lambda})$. From a theoretical point of view, the equivalence of separation and optimization can be used to compute $\min L(\boldsymbol{\lambda})$ in polynomial time, provided that $\min\{\tilde{\boldsymbol{c}}^\top \boldsymbol{x} : \boldsymbol{x} \in \mathrm{conv}(\mathcal{X}^{(2)})\}$ can be computed in polynomial time for every $\tilde{\boldsymbol{c}}$ (see [39]). In practice, the *subgradient method* is often employed to compute $\min L(\boldsymbol{\lambda})$.

6.1.1 Subgradient method

The *subgradient method* solves a general minimization problem $\min\{f(\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \mathcal{S}\}$ for a convex function $f : \mathcal{S} \to \mathbb{R}$ and a convex and closed set $\mathcal{S} \subseteq \mathbb{R}^r$. It is particularly suited for the case when f is not differentiable. The subgradient method is based on the following generalization of a gradient for convex functions.

Definition 6.6. A vector $\boldsymbol{h} \in \mathbb{R}^r$ is a *subgradient* of a convex function $f : \mathbb{R}^r \to \mathbb{R}$ at point $\hat{\boldsymbol{\lambda}}$ if, for all $\boldsymbol{\lambda} \in \mathbb{R}^r$,

$$f(\boldsymbol{\lambda}) \geq f(\hat{\boldsymbol{\lambda}}) + \boldsymbol{h}^\top(\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}).$$

In other words, subgradients are linear underestimators. Importantly, subgradients still allow to characterize global minima of convex functions.

Lemma 6.7. The point $\boldsymbol{\lambda}^\star \in \mathbb{R}^r$ minimizes a convex function $f : \mathbb{R}^r \to \mathbb{R}$ if and only if $\mathbf{0}$ is a subgradient of f at $\boldsymbol{\lambda}^\star$.

Proof. We have that $f(\boldsymbol{\lambda}^\star)$ is a (global) minimum if and only if, for all $\boldsymbol{\lambda} \in \mathbb{R}^r$, it holds that $f(\boldsymbol{\lambda}) - f(\boldsymbol{\lambda}^\star) \geq 0 = \mathbf{0}^\top(\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star)$, i.e., if and only if $\mathbf{0}$ is a subgradient of f at $\boldsymbol{\lambda}^\star$. □

Much like simple gradient descent, the subgradient method relies on the fact that if $\boldsymbol{\lambda}^\star \in \mathcal{S}$ is a global minimum of f, then, for every subgradient $\boldsymbol{h} \in \mathbb{R}^r$ at every point $\boldsymbol{\lambda} \in \mathcal{S}$,

$$0 \geq f(\boldsymbol{\lambda}^\star) - f(\boldsymbol{\lambda}) \geq \boldsymbol{h}^\top(\boldsymbol{\lambda}^\star - \boldsymbol{\lambda}),$$

i.e., $\boldsymbol{h}$ points in opposing direction of the vector $(\boldsymbol{\lambda}^\star - \boldsymbol{\lambda})$, which means that $-\boldsymbol{h}$ points from $\boldsymbol{\lambda}$ (vaguely) towards $\boldsymbol{\lambda}^\star$.

With this in mind, the subgradient method starts with a point $\boldsymbol{\lambda}^{(0)} \in \mathbb{R}^r$ and then iteratively computes $\boldsymbol{\lambda}^{(k+1)}$ for $k \leftarrow 1, 2, \ldots$ via $\boldsymbol{\lambda}^{(k+1)} \leftarrow \Pi_{\mathcal{S}}(\boldsymbol{\lambda}^{(k)} - \mu_k \boldsymbol{h}^{(k)}/\|\boldsymbol{h}^{(k)}\|)$. Here, $\boldsymbol{h}^{(k)}$ is a subgradient of f at point $\boldsymbol{\lambda}^{(k)}$, the *step size* $\mu_k > 0$ is chosen such that it limits the overshoot in each step and thus improves convergence of the method, and $\Pi_{\mathcal{S}}$ is a projection onto the closed set $\mathcal{S}$ defined by

$$\Pi_{\mathcal{S}}(\hat{\boldsymbol{\lambda}}) := \arg\min_{\boldsymbol{\lambda} \in \mathcal{S}} \|\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}\|.$$

Overall, we obtain the following general method.

Algorithm: subgradient method

input: convex function $f : \mathcal{S} \to \mathbb{R}$ with $\mathcal{S} \subseteq \mathbb{R}^r$ convex and closed
output: $\arg\min\{f(\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \mathcal{S}\}$

choose $\boldsymbol{\lambda}^{(0)} \in \mathcal{S}$; set $k \leftarrow 0$
compute subgradient $\boldsymbol{h}^{(0)}$ of f at $\boldsymbol{\lambda}^{(0)}$
while $\boldsymbol{h}^{(k)} \neq \mathbf{0}$**:**
  choose step size μ_k
  $\boldsymbol{\lambda}^{(k+1)} \leftarrow \Pi_{\mathcal{S}}(\boldsymbol{\lambda}^{(k)} - \mu_k \frac{\boldsymbol{h}^{(k)}}{\|\boldsymbol{h}^{(k)}\|})$
  $k \leftarrow k + 1$
  compute subgradient $\boldsymbol{h}^{(k)}$ of f at $\boldsymbol{\lambda}^{(k)}$
return $\boldsymbol{\lambda}^{(k)}$

We have already seen (*Introduction to Optimization*) that convex functions are continuous in the interior of their domains. We need the following strengthening (see lecture *Nonsmooth Optimization*).[1]

Lemma 6.8. Every convex function $f : \mathcal{U} \to \mathbb{R}$ on a convex and open set $\mathcal{U} \subseteq \mathbb{R}^r$ is locally Lipschitz-continuous.

Note that it is crucial to require an open domain $\mathcal{U}$, since a convex functions may be discontinuous at the boundary. With this, we obtain the following convergence theorem.

Theorem 6.9. Let f be convex on an open set $\mathcal{U} \supseteq \mathcal{S}$, let $\sum_{k=0}^{\infty} \mu_k = \infty$ and let $\sum_{k=0}^{\infty} \mu_k^2 < \infty$. Then, the subgradient method either returns a minimum point of f or converges to one, provided any exists.

Proof. The statement for the case that the subgradient method terminates follows from Lemma 6.7.

We use (without proof) that projections onto convex sets are non-expansive, i.e., $\|\Pi_{\mathcal{S}}(\boldsymbol{x}) - \Pi_{\mathcal{S}}(\boldsymbol{y})\| \leq \|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^r$. For every minimum point $\boldsymbol{\lambda}^\star \in \mathcal{S}$ of f, we have $\Pi_{\mathcal{S}}(\boldsymbol{\lambda}^\star) = \boldsymbol{\lambda}^\star$ and thus

$$\begin{aligned}
\|\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^\star\|^2 &= \|\Pi_{\mathcal{S}}(\boldsymbol{\lambda}^{(k)} - \mu_k \tfrac{\boldsymbol{h}^{(k)}}{\|\boldsymbol{h}^{(k)}\|}) - \Pi_{\mathcal{S}}(\boldsymbol{\lambda}^\star)\|^2 \\
&\leq \|\boldsymbol{\lambda}^{(k)} - \mu_k \tfrac{\boldsymbol{h}^{(k)}}{\|\boldsymbol{h}^{(k)}\|} - \boldsymbol{\lambda}^\star\|^2 \\
&= \|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^\star\|^2 - 2\,\tfrac{\mu_k}{\|\boldsymbol{h}^{(k)}\|}(\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^\star)^\top \boldsymbol{h}^{(k)} + \mu_k^2 \\
&\leq \|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^\star\|^2 - 2\,\tfrac{\mu_k}{\|\boldsymbol{h}^{(k)}\|}(f(\boldsymbol{\lambda}^{(k)}) - f(\boldsymbol{\lambda}^\star)) + \mu_k^2, \qquad (6.7)
\end{aligned}$$

[1] Recall that a function is locally Lipschitz-continuous if it is Lipschitz-continuous on every compact subset of its domain.

where the last inequality follows by definition of $\boldsymbol{h}^{(k)}$ being a subgradient of f at $\boldsymbol{\lambda}^{(k)}$. By applying this inequality iteratively, and because $f(\boldsymbol{\lambda}^{(i)}) \geq f(\boldsymbol{\lambda}^\star)$ and $\mu_i > 0$, we obtain, for all $k \in \mathbb{N}$,

$$\|\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^\star\|^2 \leq \|\boldsymbol{\lambda}^{(0)} - \boldsymbol{\lambda}^\star\|^2 - 2\sum_{i=0}^{k} \tfrac{\mu_i}{\|\boldsymbol{h}^{(i)}\|}(f(\boldsymbol{\lambda}^{(i)}) - f(\boldsymbol{\lambda}^\star)) + \sum_{i=0}^{k} \mu_i^2 \leq \|\boldsymbol{\lambda}^{(0)} - \boldsymbol{\lambda}^\star\|^2 + \sum_{i=0}^{k} \mu_i^2 < \infty. \tag{6.8}$$

Thus, the sequence $(\boldsymbol{\lambda}^{(k)})_{k\in\mathbb{N}}$ is bounded, and we can find a compact set $\mathcal{L} \subseteq \mathcal{U}$ with $\boldsymbol{\lambda}^{(k)} \in \mathcal{L}$ in its interior for all $k \in \mathbb{N}$. Lemma 6.8 now implies Lipschitz-continuity on $\mathcal{L}$ with some Lipschitz-constant $H \geq 0$. This means that the subgradients are bounded via $\varepsilon \left\|\boldsymbol{h}^{(k)}\right\|^2 = ((\boldsymbol{\lambda}^{(k)} + \varepsilon \boldsymbol{h}^{(k)}) - \boldsymbol{\lambda}^{(k)})^\top \boldsymbol{h}^{(k)} \leq f(\boldsymbol{\lambda}^{(k)} + \varepsilon \boldsymbol{h}^{(k)}) - f(\boldsymbol{\lambda}^{(k)}) \leq H \left\|\varepsilon \boldsymbol{h}^{(k)}\right\|$, i.e., $\|\boldsymbol{h}^{(k)}\| \leq H$, for $\varepsilon > 0$ sufficiently small to stay within $\mathcal{U}$.

Now, let $f^\star := f(\boldsymbol{\lambda}^\star)$ and $\bar{f}_k := \min\{f(\boldsymbol{\lambda}^{(0)}), \ldots, f(\boldsymbol{\lambda}^{(k)})\}$. Because of $\|\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^\star\|^2 \geq 0$, rearranging (6.8) yields

$$2\Big(\sum_{i=0}^{k} \tfrac{\mu_i}{\|\boldsymbol{h}^{(i)}\|}\Big)(\bar{f}_k - f^\star) \leq 2\sum_{i=0}^{k} \tfrac{\mu_i}{\|\boldsymbol{h}^{(i)}\|}(f(\boldsymbol{\lambda}^{(i)}) - f(\boldsymbol{\lambda}^\star)) \leq \|\boldsymbol{\lambda}^{(0)} - \boldsymbol{\lambda}^\star\|^2 + \sum_{i=0}^{k} \mu_i^2,$$

which implies that, for every minimum point $\boldsymbol{\lambda}^\star \in \mathcal{S}$ of f,

$$\bar{f}_k - f^\star \leq \frac{\|\boldsymbol{\lambda}^{(0)} - \boldsymbol{\lambda}^\star\|^2 + \sum_{i=0}^{k} \mu_i^2}{2\Big(\sum_{i=0}^{k} \frac{\mu_i}{\|\boldsymbol{h}^{(i)}\|}\Big)} \leq \frac{H}{2} \cdot \frac{\|\boldsymbol{\lambda}^{(0)} - \boldsymbol{\lambda}^\star\|^2 + \sum_{i=0}^{k} \mu_i^2}{\sum_{i=0}^{k} \mu_i}.$$

Because of $\sum_{i=0}^{\infty} \mu_i^2 < \infty$ and $\sum_{i=0}^{\infty} \mu_i = \infty$ it follows that $\bar{f}_k \to f^\star$ for $k \to \infty$.

It remains to show that $(\boldsymbol{\lambda}^{(k)})_{k\in\mathbb{N}}$ also converges. Since the subgradient method does not terminate and since $(\bar{f}_k)_{k\in\mathbb{N}}$ converges to $f^\star$, there is an infinite subset $K \subseteq \mathbb{N}$ such that $(f(\boldsymbol{\lambda}^{(k)}))_{k\in K}$ converges to $f^\star$. Because $(\boldsymbol{\lambda}^{(k)})_{k\in\mathbb{N}}$ and thus $(\boldsymbol{\lambda}^{(k)})_{k\in K}$ is bounded, there is a convergent subsequence $(\boldsymbol{\lambda}^{(k)})_{k\in K'}$. Since $(f(\boldsymbol{\lambda}^{(k)}))_{k\in K'}$ converges to $f^\star$ and f is continuous on $\mathcal{U} \supseteq \mathcal{S}$ (Lemma 6.8), $(\boldsymbol{\lambda}^{(k)})_{k\in K'}$ converges to a minimum point $\boldsymbol{\lambda}^\star$. For every $\varepsilon > 0$ there exists $k_0 \in K'$ such that $\|\boldsymbol{\lambda}^{(k_0)} - \boldsymbol{\lambda}^\star\| < \varepsilon/2$ and $\sum_{k=k_0}^{\infty} \mu_k^2 < \varepsilon/2$. We obtain that, for all $k \geq k_0$,

$$\|\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^\star\|^2 \overset{(6.7)}{\leq} \|\boldsymbol{\lambda}^{(k_0)} - \boldsymbol{\lambda}^\star\|^2 + \sum_{i=k_0}^{k} \mu_i^2 < \varepsilon.$$

Thus, $\lim_{k\to\infty} \boldsymbol{\lambda}^{(k)} = \boldsymbol{\lambda}^\star$. □

Remark 6.10. We give some additional information pertaining to the subgradient method.

(a) The function values $f(\boldsymbol{\lambda}^{(k)})$ are typically not monotone.

(b) A possible choice for μ_k is $\frac{1}{k}$.

(c) Requiring $\sum_{k=0}^{\infty} \mu_k = \infty$ and $\mu_k \to 0$ is, in general, insufficient for $(\boldsymbol{\lambda}^{(k)})_{k\in\mathbb{N}}$ or $(f(\boldsymbol{\lambda}^{(k)}))_{k\in\mathbb{N}}$ to converge.

(d) The method is often aborted early when no more significant progress is made.

The subgradient method is easy to implement and well suited when subgradients can be efficiently calculated. However, the step size must be chosen carefully. A disadvantage of the method is that it does not provide an indication as to the progress of convergence towards the optimum value. More advanced methods are discussed in the lecture *Nonsmooth Optimization*.

6.1.2 Solving the Lagrangian relaxation

We now describe how the subgradient method can be used to compute the Lagrangian relaxation $\min_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\boldsymbol{\lambda})$. To apply the method, we need to establish that L is a convex function that can be extended convexly to an open set.

Lemma 6.11. The function $L(\boldsymbol{\lambda})$ is piecewise (affine) linear and convex on the domain where it is bounded.

Proof. First, we recall that $L(\hat{\boldsymbol{\lambda}}) = \max\{\boldsymbol{c}^\top \boldsymbol{x} + \hat{\boldsymbol{\lambda}}^\top (\boldsymbol{b^{(1)}} - A^{(1)}\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X}^{(2)}\}$ and again consider (6.4) with $\operatorname{conv}(\mathcal{X}^{(2)}) = \operatorname{conv}(\{\boldsymbol{v^{(1)}}, \ldots, \boldsymbol{v^{(k)}}\}) + \operatorname{cone}(\{\boldsymbol{r^{(1)}}, \ldots, \boldsymbol{r^{(\ell)}}\})$. We can see that $L(\hat{\boldsymbol{\lambda}})$ is finite if and only if $\hat{\boldsymbol{\lambda}}$ falls in the polyhedron $\{\boldsymbol{\lambda} \in \mathbb{R}^{m_1}_{\geq 0} : \boldsymbol{\lambda}^\top A^{(1)} \boldsymbol{r^{(i)}} \geq \boldsymbol{c}^\top \boldsymbol{r^{(i)}} \ \forall i \in \{1, \ldots, \ell\}\}$, and that, in this polyhedron, we have $L(\boldsymbol{\lambda}) = \max_{j \in \{1,\ldots,k\}} \{\boldsymbol{c}^\top \boldsymbol{v^{(j)}} + \boldsymbol{\lambda}^\top (\boldsymbol{b^{(1)}} - A^{(1)} \boldsymbol{v^{(j)}})\}$, i.e., $L(\boldsymbol{\lambda})$ is the maximum over a finite number of affine linear functions. In particular, $L(\boldsymbol{\lambda})$ is convex. □

The key ingredient of the subgradient method is the computation of subgradients. For the Langrangian relaxation, we can obtain subgradients by finding optimum solutions $\boldsymbol{x}^\star$ of $L(\boldsymbol{\lambda})$ for fixed $\boldsymbol{\lambda}$.

Lemma 6.12. Consider $\hat{\boldsymbol{\lambda}} \in \mathbb{R}^{m_1}_{\geq 0}$ and an optimum solution $\hat{\boldsymbol{x}}$ of $L(\hat{\boldsymbol{\lambda}})$. Then, $\boldsymbol{h} = \boldsymbol{b^{(1)}} - A^{(1)}\hat{\boldsymbol{x}}$ is a subgradient of L at $\hat{\boldsymbol{\lambda}}$.

Proof. Let $\boldsymbol{\lambda} \in \mathbb{R}^{m_1}_{\geq 0}$ and $\boldsymbol{x^{(\lambda)}}$ be an optimum solution of $L(\boldsymbol{\lambda})$. Then,

$$\begin{aligned} L(\boldsymbol{\lambda}) - L(\hat{\boldsymbol{\lambda}}) &= \boldsymbol{c}^\top \boldsymbol{x^{(\lambda)}} + \boldsymbol{\lambda}(\boldsymbol{b^{(1)}} - A^{(1)}\boldsymbol{x^{(\lambda)}}) - (\boldsymbol{c}^\top \hat{\boldsymbol{x}} + \hat{\boldsymbol{\lambda}}^\top (\boldsymbol{b^{(1)}} - A^{(1)}\hat{\boldsymbol{x}})) \\ &\geq \boldsymbol{c}^\top \hat{\boldsymbol{x}} + \boldsymbol{\lambda}(\boldsymbol{b^{(1)}} - A^{(1)}\hat{\boldsymbol{x}}) - (\boldsymbol{c}^\top \hat{\boldsymbol{x}} + \hat{\boldsymbol{\lambda}}^\top (\boldsymbol{b^{(1)}} - A^{(1)}\hat{\boldsymbol{x}})) = \boldsymbol{h}^\top (\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}), \end{aligned}$$

where the inequality follows because $\boldsymbol{x^{(\lambda)}}$ is an optimum solution with respect to $\boldsymbol{\lambda}$. □

Finally, we have $\mathcal{S} = \mathbb{R}^{m_1}_{\geq 0}$ in the subgradient method and the projection $\Pi_{\mathcal{S}}$ is given by $\Pi(\boldsymbol{\lambda})_i = \max\{\lambda_i, 0\}$ for all $i \in \{1, \ldots, m_1\}$.

In the following, we describe applications where the Lagrangian relaxation has successfully been used in combination with the subgradient method. In these applications a good balance between efficiency of the subgradient method and strength of the relaxation can be achieved.

Example 6.13. If the constraint matrix has (almost) block diagonal form, see Figure 6.2, we can choose $A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}}$ as the the coupling conditions. Relaxing them in a Lagrangian relaxation decomposes the remaining matrix into k independent blocks. In this case, $L(\boldsymbol{\lambda})$ is the sum of k independent terms that can separately be determined.

Often, each individual block $M^{(i)}$ represents a network flow problem, a knapsack problem, or the like, and can therefore be solved using specialized combinatorial algorithms. Other examples are multicommodity flow problems, which arise, for example, in vehicle routing or in decompositions of stochastic mixed-integer problems. △

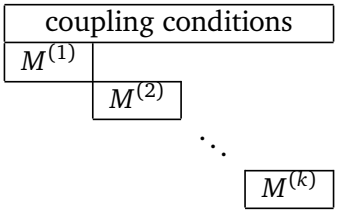

Figure 6.2: Matrix mostly in block diagonal form, except for a coupling part.

6.1.3 Application in stochastic optimization*

Two-stage stochastic optimization problems give rise to a constraint matrix with a structure as in Figure 6.2. In these problems, we consider a finite set Ω of mutually disjoint random events called *scenarios*. Each scenario $\omega \in \Omega$ occurs with some probability p_ω. Furthermore, mixed-integer first-stage decisions $\boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$ are to be made, that are unaffected by stochasticity. Once a certain scenario ω occurs, then second-stage decisions $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}, \omega)$ have to be made based on the predetermined first-stage decisions and the event at hand. The goal is to minimize the total cost, consisting of the costs for $\boldsymbol{x}$ and the *expected* cost for $\boldsymbol{y}$. If the constraints on the second-stage decisions are linear, the problem can be formulated as its so-called *deterministic equivalent*

$$\begin{aligned} \min \quad & \boldsymbol{c}^\top \boldsymbol{x} + \sum_{\omega \in \Omega} p_\omega (\boldsymbol{d}^{(\omega)})^\top \boldsymbol{y}^{(\omega)} \\ \text{s.t.} \quad & W^{(\omega)} \boldsymbol{y}^{(\omega)} \leq \boldsymbol{h}^{(\omega)} - T^{(\omega)} \boldsymbol{x} && \forall \omega \in \Omega, \\ & A\boldsymbol{x} \leq \boldsymbol{b}, \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}. \end{aligned}$$

Here, the second term in the objective function specifies the expected costs for the second-stage decisions, where $\boldsymbol{d}^{(\omega)}$ denotes the costs for scenario ω. The coupling between the first and second stage for a $\omega \in \Omega$ is done by means of the *recourse matrix* $W^{(\omega)}$ and *technology matrix* $T^{(\omega)}$.

For fixed first-stage decision $\boldsymbol{x}$, the problem decomposes into $|\Omega|$ independent subproblems. In order to use a Lagrangian relaxation we introduce a copy $\boldsymbol{x}^{(\omega)}$ of $\boldsymbol{x}$ for every scenario ω and constrain them to be all of equal value through coupling conditions $\boldsymbol{x}^{(\omega)} = \boldsymbol{x}^{(\omega')}$. We formulate these coupling conditions as $\sum_{\omega \in \Omega} H^{(\omega)} \boldsymbol{x}^{(\omega)} = \boldsymbol{0}$ and obtain the equivalent problem

$$\begin{aligned} \min \quad & \sum_{\omega \in \Omega} \left(\boldsymbol{c}^\top \boldsymbol{x} + p_\omega (\boldsymbol{d}^{(\omega)})^\top \boldsymbol{y}^{(\omega)} \right) \\ \text{s.t.} \quad & \sum_{\omega \in \Omega} H^{(\omega)} \boldsymbol{x}^{(\omega)} = \boldsymbol{0}, \\ & A\boldsymbol{x}^{(\omega)} \leq \boldsymbol{b}, && \forall \omega \in \Omega, \\ & \boldsymbol{x}^{(\omega)} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, && \forall \omega \in \Omega, \\ & W^{(\omega)} \boldsymbol{y}^{(\omega)} \leq \boldsymbol{h}^{(\omega)} - T^{(\omega)} \boldsymbol{x}^{(\omega)}, && \forall \omega \in \Omega. \end{aligned}$$

The constraint matrix of this problem has the structure of Figure 6.2. A Lagrangian relaxation of the coupling conditions then leads to a decomposition. The equality of the different $\boldsymbol{x}^{\boldsymbol{\omega}}$ is established step by step (see [7]).

6.1.4 Application in combinatorial optimization

Another application of Lagrangian relaxation is the famous traveling salesperson problem (TSP) (see *Algorithmic Discrete Mathematics*). Recall that an instance of TSP is given by an undirected complete graph $G = (V, E)$

with edge weights $\boldsymbol{c} \in \mathbb{Q}^E_{\geq 0}$, and the problem consists in finding a cycle (called a *tour*) R that visits all vertices (exactly once) and minimizes $c(R) := \sum_{e \in R} c_e$. The problem is NP-hard via reduction from Hamiltonian cycle. We consider a particular relaxation for this problem that relates to graphs containing exactly one cycle.

Definition 6.14. A *1-tree with respect to vertex* $v_1 \in V$ in graph $G = (V, E)$ is a subset of edges $B \subseteq E$ such that

- $B \setminus \delta(v_1)$ is a spanning tree of $G \setminus \{v_1\}$, and
- the degree of v_1 in (V, B) is 2.

In the following, we fix a vertex $v_1 \in V$. Because every tour is a 1-tree, we have

$$\min_{B \text{ 1-tree}} c(B) \leq \min_{R \text{ tour}} c(R). \tag{6.9}$$

This lower bound for TSP can be improved as follows. We consider a *vertex potential* $\pi \in \mathbb{R}^V$ and define the *reduced costs* with respect to this potential as (see *Combinatorial Optimization*)

$$c^\pi_e := c_e - \pi_v - \pi_u \quad \forall\, e = \{u, v\} \in E.$$

Let $\pi(V) := \sum_{v \in V} \pi_v$ and $c^\pi(B) := \sum_{e \in B} c^\pi_e$. We obtain a stronger inequality.

Theorem 6.15 ([24])**.** For $f(\pi) := 2\,\pi(V) + \min_{B \text{ 1-tree}} c^\pi(B)$ it holds that

$$\max_{\pi \in \mathbb{R}^V} f(\pi) \leq \min_{R \text{ tour}} c(R).$$

Proof. Let $\pi \in \mathbb{R}^V$ be arbitrary and R be a TSP tour. Then,

$$\begin{aligned} f(\pi) &= 2\,\pi(V) + \min_{B \text{ 1-tree}} c^\pi(B) \\ &\overset{(6.9)}{\leq} 2\,\pi(V) + c^\pi(R) \\ &= 2\,\pi(V) + c(R) - 2\,\pi(V) = c(R). \end{aligned}$$

Here, the first inequality uses that R is a 1-tree, and the next equality holds because every vertex occurs exactly twice in the edges of R. □

The function f is the sum of an affine part and the minimum of affine (i.e., concave) functions, hence it is concave. This means that the subgradient method can be applied to the convex function $-f$. We can determine a subgradient as follows.

Lemma 6.16. Let $\tilde{\pi} \in \mathbb{R}^V$ and $\tilde{B} \in \arg\min_{B \text{ 1-tree}} c^{\tilde{\pi}}(B)$. Then, a subgradient $\boldsymbol{h}$ for $-f$ at point $\tilde{\pi}$ is defined by

$$h_v = \deg_{(V,\tilde{B})}(v) - 2 \qquad \forall\, v \in V.$$

Proof. We need to show that $-f(\pi) \geq -f(\tilde{\pi}) + \boldsymbol{h}^\top(\pi - \tilde{\pi})$ holds for all $\pi \in \mathbb{R}^V$. We have

$$\begin{aligned} f(\tilde{\pi}) &= 2\,\tilde{\pi}(V) + \min_{B \text{ 1-tree}} c^{\tilde{\pi}}(B) \\ &= 2\,\tilde{\pi}(V) + c(\tilde{B}) - \sum_{v \in V} \deg_{(V,\tilde{B})}(v) \cdot \tilde{\pi}_v \\ &= c(\tilde{B}) + \sum_{v \in V} (2 - \deg_{(V,\tilde{B})}(v)) \cdot \tilde{\pi}_v \\ &= c(\tilde{B}) - \boldsymbol{h}^\top \tilde{\pi}. \end{aligned}$$

Moreover,

$$f(\pi) = 2\,\pi(V) + \min_{B \text{ 1-tree}} c^\pi(B) \leq 2\,\pi(V) + c^\pi(\tilde{B}) = c(\tilde{B}) - \boldsymbol{h}^\top \pi.$$

Overall, we obtain

$$-f(\pi) + f(\tilde{\pi}) \geq \left(\boldsymbol{h}^\top \pi - c(\tilde{B})\right) + \left(c(\tilde{B}) - \boldsymbol{h}^\top \tilde{\pi}\right) = \boldsymbol{h}^\top(\pi - \tilde{\pi}).$$ □

This means that we can apply the subgradient method, provided that we can efficiently compute minimum cost 1-trees. This can easily be accomplished by computing a minimum spanning tree of $G \setminus \{v_1\}$ (see *Algorithmic Discrete Mathematics*) and then adding the two edges of smallest weight incident to vertex v_1 to the result.

To relate this approach to a Lagrangian relaxation, we need the following result (without proof). Here $\boldsymbol{\chi^B} \in \{0,1\}^E$ denotes the incidence vector of a set $B \subseteq E$, i.e., $\chi_e^B = 1 \Leftrightarrow e \in B$.

Lemma 6.17 ([23])**.** It holds that

$$\begin{aligned}\operatorname{conv}(\{\boldsymbol{\chi^B} : B \subseteq E \text{ is a 1-tree wrt. } v_1\}) = \Big\{\boldsymbol{x} \in [0,1]^E : & \sum_{e \in \delta(v_1)} x_e = 2 \\ & \sum_{e \in G[S]} x_e \leq |S| - 1 \qquad \forall\, S \subseteq V \setminus \{v_1\},\ S \neq \emptyset, \\ & \sum_{e \in E} x_e = |V|\Big\}.\end{aligned}$$

In particular, the polyhedron on the right-hand side is integral.

Together with Theorem 6.1 we can show a characterization of the strength of the lower bound of Theorem 6.15.

Theorem 6.18 ([23])**.** It holds that

$$\begin{aligned}\max_{\pi \in \mathbb{R}^V} f(\pi) = \min \quad & \sum_{e \in E} c_e\, x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} x_e = 2 && \forall\, v \in V && \text{(degree constraints)} \\ & \sum_{e \in G[S]} x_e \leq |S| - 1 && \forall\, S \subset V,\ S \neq \emptyset && \text{(subtour-elimination)} \\ & \boldsymbol{x} \in [0,1]^E.\end{aligned}$$

Proof. Let $A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}$ be the system of Lemma 6.17 and let $A^{(1)}\boldsymbol{x} = \boldsymbol{b^{(1)}}$ describe the degree conditions

$$\sum_{e \in \delta(v)} x_e = 2 \quad \forall\, v \in V \setminus \{v_1\}.$$

We observe that, together, $A^{(1)}\boldsymbol{x} = \boldsymbol{b^{(1)}}$ and $A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}$ yield the system in the statement of the theorem: The degree conditions are all present, the equation $\sum_{e \in E} x_e = |V|$ is redundant (sum of all degree conditions

divided by 2), and the missing subtour elimination constraints for $S \subseteq V$ with $v_1 \in S$ are implied via

$$\begin{aligned}
\sum_{e\in G[S]} x_e &= \sum_{e\in G[S]} x_e + \sum_{e\in E} x_e - |V| \\
&= 2\sum_{e\in G[S]} x_e + \sum_{e\in\delta(S)} x_e + \sum_{e\in G[V\setminus S]} x_e - |V| \\
&= \sum_{v\in S}\sum_{e\in\delta(v)} x_e + \sum_{e\in G[V\setminus S]} x_e - |V| \\
&\overset{v_1\in S}{\leq} 2|S| + (|V\setminus S| - 1) - |V| \\
&= |S| - 1,
\end{aligned}$$

where the inequality uses the degree constraints and the subtour elimination constraints for $v_1 \notin S$.
In order to inspect the Lagrangian relaxation, we consider the negated integer program

$$\begin{aligned}
\max \quad & -\boldsymbol{c}^\top \boldsymbol{x} \\
\text{s.t.} \quad & A^{(1)}\boldsymbol{x} = \boldsymbol{b^{(1)}} \\
& A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}} \\
& \boldsymbol{x} \in \{0,1\}^E,
\end{aligned}$$

where $A^{(1)}$ is the vertex-edge incidence matrix of G and $\boldsymbol{b^{(1)}} = 2\cdot\mathbf{1}$. Note that the system is rational.
Since we relax a system of equations $A^{(1)}\boldsymbol{x} = \boldsymbol{b^{(1)}}$, the Lagrangian function (6.2) may be considered without requiring non-negativity and may be written as

$$\begin{aligned}
L(\boldsymbol{\pi}) &= \max\{-\boldsymbol{c}^\top\boldsymbol{x} + \boldsymbol{\pi}^\top(A^{(1)}\boldsymbol{x} - \boldsymbol{b^{(1)}}) : \boldsymbol{x} \in \mathcal{P}(A^{(2)}, \boldsymbol{b^{(2)}}) \cap \{0,1\}^E\} \\
&\overset{6.17}{=} \max\{-\boldsymbol{c}^\top\boldsymbol{x} + \boldsymbol{\pi}^\top(A^{(1)}\boldsymbol{x} - \boldsymbol{b^{(1)}}) : \boldsymbol{x} \in \operatorname{conv}(\{\boldsymbol{\chi^B} : B \subseteq E \text{ is a 1-tree wrt. } v_1\})\} \\
&\overset{2.7}{=} -2\pi(V) - \min\{(\boldsymbol{c}^\top - \boldsymbol{\pi}^\top A^{(1)})\,\boldsymbol{\chi^B} : B \subseteq E \text{ is a 1-tree wrt. } v_1\} \\
&= -2\pi(V) - \min_{1\text{-tree}\,B} \sum_{e=\{u,v\}\in E} (c_e - \pi_u - \pi_v)\,\chi_e^B \\
&= -f(\boldsymbol{\pi}).
\end{aligned}$$

Lemma 6.17 implies that $\mathcal{P}(A^{(2)}, \boldsymbol{b^{(2)}})$ is integral, and Observation 6.3 thus yields

$$\begin{aligned}
\max_{\boldsymbol{\pi}\in\mathbb{R}^V} f(\boldsymbol{\pi}) &= -\min_{\boldsymbol{\pi}\in\mathbb{R}^V} L(\boldsymbol{\pi}) \\
&\overset{6.3}{=} -\max\{-\boldsymbol{c}^\top\boldsymbol{x} : A^{(1)}\boldsymbol{x} = \boldsymbol{b^{(1)}}, A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}, \boldsymbol{x} \in [0,1]^E\} \\
&= \min\{\boldsymbol{c}^\top\boldsymbol{x} : A^{(1)}\boldsymbol{x} = \boldsymbol{b^{(1)}}, A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}, \boldsymbol{x} \in [0,1]^E\}. \quad \square
\end{aligned}$$

The proof of Theorem 6.18 shows that the 1-tree relaxation is a Lagrangian relaxation of the system in the theorem, where the degree constraints are relaxed for all vertices except v_1. Note that $L(\boldsymbol{\pi}) = -f(\boldsymbol{\pi})$ also implies that Lemma 6.16 follows immediately from Lemma 6.12.

Remark 6.19. If we add to the LP in Theorem 6.18 the condition $\boldsymbol{x} \in \{0,1\}^E$, we obtain an IP formulation of the TSP problem. The degree conditions guarantee that every vertex has degree 2 in the solution. The subtour elimination constraints exclude partial tours that are disconnected from the remaining graph. Theorem 6.18 shows that the Lagrangian relaxation of this formulation is as good as the corresponding LP relaxation.

6.2 Dantzig-Wolfe decomposition

The Dantzig-Wolfe decomposition (see [12]) is a different method for treating MIPs with two parts, i.e., of the form

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}} \\ & A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}} \\ & \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, \end{aligned} \tag{6.10}$$

with $A^{(1)} \in \mathbb{Q}^{m_1 \times n}$, $A^{(2)} \in \mathbb{Q}^{m_2 \times n}$, $\boldsymbol{b^{(1)}} \in \mathbb{Q}^{m_1}$, $\boldsymbol{b^{(2)}} \in \mathbb{Q}^{m_2}$ and $m_1 + m_2 = m$.

We first assume that $p = 0$, i.e., we have an LP. Consider the polyhedron $\mathcal{P}^{(2)} = \mathcal{P}(A^{(2)}, \boldsymbol{b^{(2)}})$. By Theorem 2.19, there exist vectors $\boldsymbol{v^{(1)}}, \dots, \boldsymbol{v^{(k)}}$ and $\boldsymbol{r^{(1)}}, \dots, \boldsymbol{r^{(\ell)}}$ with $\mathcal{P}^{(2)} = \text{conv}(\{\boldsymbol{v^{(1)}}, \dots, \boldsymbol{v^{(k)}}\}) + \text{cone}(\{\boldsymbol{r^{(1)}}, \dots, \boldsymbol{r^{(\ell)}}\})$. In other words, $\boldsymbol{x} \in \mathcal{P}^{(2)}$ can be expressed in the form

$$\boldsymbol{x} = \sum_{i=1}^{k} \lambda_i \boldsymbol{v^{(i)}} + \sum_{j=1}^{\ell} \mu_j \boldsymbol{r^{(j)}} \tag{6.11}$$

with $\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \boldsymbol{0}$ and $\boldsymbol{1}^\top \boldsymbol{\lambda} = 1$. With (6.11), we can rewrite (6.10) as

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \Big(\sum_{i=1}^{k} \lambda_i \boldsymbol{v^{(i)}} + \sum_{j=1}^{\ell} \mu_j \boldsymbol{r^{(j)}}\Big) \\ \text{s.t.} \quad & A^{(1)}\Big(\sum_{i=1}^{k} \lambda_i \boldsymbol{v^{(i)}} + \sum_{j=1}^{\ell} \mu_j \boldsymbol{r^{(j)}}\Big) \leq \boldsymbol{b^{(1)}}, \\ & \boldsymbol{1}^\top \boldsymbol{\lambda} = 1, \\ & \boldsymbol{\lambda}, \boldsymbol{\mu} \geq \boldsymbol{0}, \end{aligned}$$

which is equivalent to

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} (\boldsymbol{c}^\top \boldsymbol{v^{(i)}})\, \lambda_i + \sum_{j=1}^{\ell} (\boldsymbol{c}^\top \boldsymbol{r^{(j)}})\, \mu_j \\ \text{s.t.} \quad & \sum_{i=1}^{k} (A^{(1)} \boldsymbol{v^{(i)}})\, \lambda_i + \sum_{j=1}^{\ell} (A^{(1)} \boldsymbol{r^{(j)}})\, \mu_j \leq \boldsymbol{b^{(1)}}, \\ & \boldsymbol{1}^\top \boldsymbol{\lambda} = 1, \\ & \boldsymbol{\lambda} \geq \boldsymbol{0}, \boldsymbol{\mu} \geq \boldsymbol{0}. \end{aligned} \tag{6.12}$$

The formulation (6.12) is called the *master problem* of (6.10). When comparing the two formulations, we realize that we reduced the number of constraints from m to $m_1 + 1$, but we now have $k + \ell$ variables instead of n. The number $k + \ell$ can be very large compared to n, even exponential, e.g., for the unit cube in $\mathbb{R}^n$ with $2n$ constraints and 2^n vertices. At first glace, the advantage of using (6.12) is not apparent.

The key insight is that we can use the simplex method to solve (6.12) without having to create all variables explicitly. We abbreviate (6.12) by

$$\max \{\tilde{\boldsymbol{c}}^\top \boldsymbol{z} : D\boldsymbol{z} = \boldsymbol{d}, \boldsymbol{z} \geq \boldsymbol{0}\},$$

where (including additional slack variables)

$$D = \left(\begin{array}{c|c|c} A^{(1)}\boldsymbol{v}^{(i)} & A^{(1)}\boldsymbol{r}^{(j)} & \mathbb{I} \\ \hline \mathbf{1}^\top & \mathbf{0}^\top & \mathbf{0}^\top \end{array}\right) \in \mathbb{Q}^{(m_1+1)\times(k+\ell+m_1)}, \quad \tilde{\boldsymbol{c}} = \left(\begin{array}{c} \boldsymbol{c}^\top\boldsymbol{v}^{(i)} \\ \hline \boldsymbol{c}^\top\boldsymbol{r}^{(j)} \\ \hline \mathbf{0} \end{array}\right) \in \mathbb{Q}^{k+\ell+m_1}, \quad \boldsymbol{d} = \left(\begin{array}{c} \boldsymbol{b}^{(1)} \\ \hline 1 \end{array}\right) \in \mathbb{Q}^{m_1+1}.$$

Recall that the simplex method (for LPs in standard form) works with a feasible basis $B \subseteq \{1,\dots,k+\ell+m_1\}$, $|B| = m_1+1$, where $D_{\cdot B}$ is regular and yields a feasible solution $\boldsymbol{z}^\star_{\boldsymbol{B}} = D_{\cdot B}^{-1}\boldsymbol{d}$ and $\boldsymbol{z}^\star_{\boldsymbol{N}} = \mathbf{0}$ with $N = \{1,\dots,k+\ell+m_1\}\setminus B$. Note that $D_{\cdot B} \in \mathbb{R}^{(m_1+1)\times(m_1+1)}$ is smaller than a basis for the original system (6.10) would have to be, and that only a part of the variables (m_1+1 of $k+\ell+m_1$) are non-zero. In addition, the only operation in the simplex method that uses all columns is the *pricing* step, which checks whether the reduced costs satisfy $\tilde{\boldsymbol{c}}_{\boldsymbol{N}}{}^\top - \tilde{\boldsymbol{y}}^\top D_{\cdot N} \leq \mathbf{0}$, where $\tilde{\boldsymbol{y}} = (D_{\cdot B})^{-1}\tilde{\boldsymbol{c}}_{\boldsymbol{B}}$ is the solution of $\boldsymbol{y}^\top D_{\cdot B} = \tilde{\boldsymbol{c}}_{\boldsymbol{B}}{}^\top$. Recall that $\tilde{\boldsymbol{y}}$ is the dual solution, and the condition on the reduced costs asks whether $\tilde{\boldsymbol{y}}$ is dually feasible. We need to find a variable of positive reduced cost that can enter the basis, or, equivalently, we are looking for a violated inequality of the dual. This can be accomplished via the following *pricing problem*:

$$\begin{array}{ll} \max & (\boldsymbol{c}^\top - \bar{\boldsymbol{y}}^\top A^{(1)})\boldsymbol{x} \\ \text{s.t.} & A^{(2)}\boldsymbol{x} \leq \boldsymbol{b}^{(2)}, \\ & \boldsymbol{x} \in \mathbb{R}^n, \end{array} \tag{6.13}$$

where $\bar{\boldsymbol{y}}$ represents the first m_1 components of the solution $\tilde{\boldsymbol{y}} = \left(\begin{smallmatrix} \bar{y} \\ \tilde{y}_{m_1+1} \end{smallmatrix}\right)$. The following cases can occur.

(a) A slack variable has positive reduced cost. In that case, we can let the variable enter the basis. This can be checked directly, because there are only m_1+1 slack variables.

(b) (6.13) has a vertex solution $\bar{\boldsymbol{x}}$ with $(\boldsymbol{c}^\top - \bar{\boldsymbol{y}}^\top A^{(1)})\bar{\boldsymbol{x}} > \tilde{y}_{m_1+1}$.

In this case, $\bar{\boldsymbol{x}} = \boldsymbol{v}^{(i)}$ for some $i \in \{1,\dots,k\}$, corresponding to $\boldsymbol{\lambda} = \mathbf{e}^i$ and $\boldsymbol{\mu} = \mathbf{0}$ in (6.12). The associated column $\boldsymbol{D_{\cdot i}}$ has reduced costs

$$\tilde{c}_i - \tilde{\boldsymbol{y}}^\top \boldsymbol{D_{\cdot i}} = \boldsymbol{c}^\top\boldsymbol{v}^{(i)} - \tilde{\boldsymbol{y}}^\top\begin{pmatrix} A^{(1)}\boldsymbol{v}^{(i)} \\ 1 \end{pmatrix} = \boldsymbol{c}^\top\boldsymbol{v}^{(i)} - \bar{\boldsymbol{y}}^\top A^{(1)}\boldsymbol{v}^{(i)} - \tilde{y}_{m_1+1} > 0.$$

In other words, $\boldsymbol{D_{\cdot i}} = \left(\begin{smallmatrix} A^{(1)}\boldsymbol{v}^{(i)} \\ \mathbf{1} \end{smallmatrix}\right)$ can be used as the entering column in the simplex algorithm (in particular, $i \in N$).

(c) (6.13) is unbounded.

Here we obtain a feasible extreme ray $\bar{\boldsymbol{r}} = \boldsymbol{r}^{(j)}$ with $(\boldsymbol{c}^\top - \bar{\boldsymbol{y}}^\top A^{(1)})\bar{\boldsymbol{r}} > 0$ for some $j \in \{1,\dots,\ell\}$, corresponding to $\boldsymbol{\lambda} = \mathbf{0}$ and $\boldsymbol{\mu} = \mathbf{e}^j$ in (6.12). The corresponding column $\boldsymbol{D_{\cdot(k+j)}}$ has reduced costs

$$\tilde{c}_{k+j} - \tilde{\boldsymbol{y}}^\top \boldsymbol{D_{\cdot(k+j)}} = \boldsymbol{c}^\top\boldsymbol{r}^{(j)} - \tilde{\boldsymbol{y}}^\top\begin{pmatrix} A^{(1)}\boldsymbol{r}^{(j)} \\ 0 \end{pmatrix} = \boldsymbol{c}^\top\boldsymbol{r}^{(j)} - \bar{\boldsymbol{y}}^\top(A^{(1)}\boldsymbol{r}^{(j)}) > 0.$$

Thus, $\boldsymbol{D_{\cdot(k+j)}} = \left(\begin{smallmatrix} A^{(1)}\boldsymbol{r}^{(j)} \\ \mathbf{0} \end{smallmatrix}\right)$ can be chosen as the entering column.

(d) (6.13) has an optimum solution $\bar{\boldsymbol{x}}$ with $(\boldsymbol{c}^\top - \bar{\boldsymbol{y}}^\top A^{(1)})^\top\bar{\boldsymbol{x}} \leq \tilde{y}_{m_1+1}$.

In this case, using the same arguments as in (b) and (c), we obtain that $\tilde{c}_i - \tilde{\boldsymbol{y}}^\top\boldsymbol{D_{\cdot i}} \leq 0$ for all $i \in \{1,\dots,k+\ell\}$. Since no slack variable has positive reduced cost either, $\tilde{\boldsymbol{c}}_{\boldsymbol{N}}{}^\top - \tilde{\boldsymbol{y}}^\top D_{\cdot N} \leq \mathbf{0}$ holds, and thus $\boldsymbol{z}^\star$ is an optimum solution of the master problem (6.12).

With these arguments, a variable of positive reduced costs can be computed, if any exists, without having to evaluate the reduced costs of all (non-basis) variables.

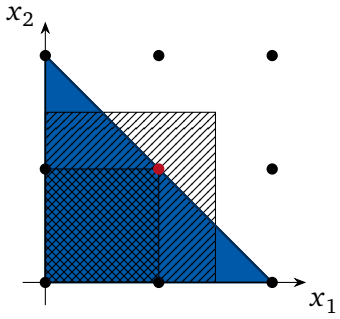

Figure 6.3: Example for the extension of the Dantzig-Wolfe decomposition to ILPs in in Remark 6.20. The integral optimum solution $\binom{1}{1}$ cannot be written as an integer linear combination of the vertices of $\mathcal{P}^{(2)}$ (blue).

Note that the whole problem (6.1) is split into two subproblems, i.e., into (6.12) and (6.13) and the approach alternates between working at the higher level (6.12) and at the lower level (6.13). The procedure starts with a feasible solution for (6.12) and generates new promising columns on demand by solving (6.13). Such methods are called *(delayed) column-generation algorithms*.

The approach can also be applied to ILPs with some caution. In this case, the problem (6.13) changes from an LP to an ILP. In addition we have to make sure that in (6.11) all feasible integral solutions $\boldsymbol{x}$ of (6.1) can be generated by linear combinations of the vectors $\boldsymbol{v^{(1)}}, \ldots, \boldsymbol{v^{(k)}}$ and $\boldsymbol{r^{(1)}}, \ldots, \boldsymbol{r^{(\ell)}}$ with

$$\operatorname{conv}(\{\boldsymbol{x} \in \mathbb{Z}^n : A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}\}) = \operatorname{conv}(\{\boldsymbol{v^{(1)}}, \ldots, \boldsymbol{v^{(k)}}\}) + \operatorname{cone}(\{\boldsymbol{r^{(1)}}, \ldots, \boldsymbol{r^{(\ell)}}\}).$$

On the other hand, we need to restrict to linear combinations that yield integer vectors $\boldsymbol{x}$.

Remark 6.20. It is not sufficient to require that $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ be integral. As a counterexample, consider

$$A^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{b^{(1)}} = \begin{pmatrix} \frac{3}{2} \\ \frac{3}{2} \end{pmatrix} \qquad A^{(2)} = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \boldsymbol{b^{(2)}} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix},$$

and the problem

$$\max\{x_1 + x_2 : A^{(1)}\boldsymbol{x} \leq \boldsymbol{b^{(1)}}, A^{(2)}\boldsymbol{x} \leq \boldsymbol{b^{(2)}}, \boldsymbol{x} \in \mathbb{Z}^2\}.$$

Then, $\mathcal{P}^{(2)} = \operatorname{conv}(\{\binom{0}{0}, \binom{2}{0}, \binom{0}{2}\})$, see Figure 6.3. But the optimum solution $\binom{1}{1}$ of the integer problem is not an integer linear combination of the vertices of $\mathcal{P}^{(2)}$. Thus, we loose optimum solutions if we do not handle the integrality of the variables in the master problem in some other way.

However, if all variables are binary, this difficulty does not arise, since every binary solution of the LP relaxation of a binary MIP is a vertex of the corresponding polyhedron (as it is an extreme point even for the hypercube and can thus not be written as nontrivial convex combination). In this case, it suffices to require $\boldsymbol{\lambda} \in \{0,1\}^k$ (no variables $\boldsymbol{\mu}$ occur, since binary programs have a bounded feasible region). In fact, column-generation algorithms are not only used for solving large linear problems, but especially for large binary problems.

Of course, the Dantzig-Wolfe decomposition for linear or binary integer problems is only one possibility for a column-generation algorithm. Other algorithms do not solve the lower-order problem using general techniques for LPs or ILPs, but by combinatorial algorithms or exhaustive enumeration. Furthermore, the problems are often not modeled via (6.1), but rather directly as in (6.12). This is the case, for example, when the feasible region allows for a concise interior representation.

6.2.1 Application in combinatorial optimization*

As an example for the application of the Dantzig-Wolfe decomposition, we consider the the *bin-packing problem*: How many bins of fixed size $W \geq 0$ do we need to fit n items of sizes $w_i > 0$, $i \in \{1, \dots, n\}$? For clarity, let $m := n$ denote an upper bound on the number of bins required.

We can model this problem as a binary problem using variables $\boldsymbol{z} \in \{0,1\}^m$ with $z_j = 1$ if and only if the j-th potential bin is used, and $\boldsymbol{x} \in \{0,1\}^{\{1,\dots,n\}\times\{1,\dots,m\}}$ with $x_{ij} = 1$ if and only if item i is packed in bin j. We obtain the following assignment problem (see Example 1.5):

$$\begin{aligned}
\min \quad & \mathbf{1}^\top \boldsymbol{z} \\
\text{s.t.} \quad & \sum_{j=1}^{m} x_{ij} = 1 && \forall i \in \{1,\dots,n\}, && (6.14)\\
& \sum_{i=1}^{n} w_i x_{ij} \leq W z_j && \forall j \in \{1,\dots,m\}, && (6.15)\\
& \boldsymbol{x} \in \{0,1\}^{n\times m},\ \boldsymbol{z} \in \{0,1\}^m.
\end{aligned}$$

However, this model can only be solved very inefficiently. The reason for this is the weakness of the LP relaxation: Optimum solutions $(x^\star, z^\star)^\top$ of the LP relaxation satisfy (6.15) with equality, therefore

$$\mathbf{1}^\top \boldsymbol{z}^\star = \sum_{j=1}^{m}\sum_{i=1}^{n} \frac{w_i x_{ij}^\star}{W} = \sum_{i=1}^{n} \frac{w_i}{W} \sum_{j=1}^{m} x_{ij}^\star = \sum_{i=1}^{n} \frac{w_i}{W} \to 0 \text{ for } W \to \infty. \tag{6.16}$$

So, if W becomes very large, the optimum value of the LP relaxation is small (value 0) and says almost nothing about the integral solution (value 1 for $W \geq \sum_{i=1}^{n} w_i$).

As a remedy, we can use Dantzig-Wolfe decomposition to eliminate the knapsack conditions (6.15). Every opened bin yields the same knapsack polytope

$$\mathcal{P} := \operatorname{conv}(\{\boldsymbol{y} \in \{0,1\}^n : \boldsymbol{w}^\top \boldsymbol{y} \leq W\}) = \operatorname{conv}(\{\mathbf{0}, \boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(k)}\}) \subset \mathbb{R}^n,$$

where $\mathbf{0}, \boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(k)} \in \{0,1\}^n$ are the vertices of the (bounded) polytope $\mathcal{P}$, while every unopened bin only allows the vector $\mathbf{0}$. Each such vertex describes one *configuration* of items that can be packed into a bin. We introduce variables $\lambda_{j1}, \dots, \lambda_{jk}$ for every bin $j \in \{1,\dots,m\}$, that indicate, which vertex/configuration is selected for each bin, i.e., $\boldsymbol{x}_{\cdot j} = \sum_{\ell=1}^{k} \lambda_{j\ell} \boldsymbol{v}^{(\ell)}$. We obtain the master problem

$$\begin{aligned}
\min \quad & \sum_{j=1}^{m}\sum_{\ell=1}^{k} \lambda_{j\ell} \\
\text{s.t.} \quad & \sum_{j=1}^{m}\sum_{\ell=1}^{k} \lambda_{j\ell} v_i^{(\ell)} = 1 && \forall i \in \{1,\dots,n\} && (6.17)\\
& \sum_{\ell=1}^{k} \lambda_{j\ell} \leq 1 && \forall j \in \{1,\dots,m\} && (6.18)\\
& \boldsymbol{\lambda} \in \{0,1\}^{n\times k}.
\end{aligned}$$

Note that in (6.18) we did not require equality to allow the vertex $\mathbf{0}$. Furthermore, we replaced the variable z_j by $\sum_{\ell=1}^{k} \lambda_{j\ell}$, as it makes no sense to use a bin that does not contain any items, i.e., we encode unopened bins j

via $\sum_{\ell=1}^{k} \lambda_{j\ell} = 0$. Similarly, we can observe that, because of $\boldsymbol{v}^{(\ell)} \neq \mathbf{0}$, it follows from (6.17) that $\sum_j \sum_\ell \lambda_{j\ell} \leq n$, since each $\lambda_{j\ell}$ belongs to a single bin and occurs in exactly one row of (6.17). Therefore, in each feasible solution, a maximum of n vertices (except $\mathbf{0}$) are used. Because of $m \geq n$, we can therefore omit (6.18), since we can convert every solution in such a way that no bin is used by more than one vertex by introducing separate bins for every vertex.

In addition, we define the set $S_i := \{\ell : v_i^{(\ell)} = 1\}$ of the indices of all vertices that contain item i, and introduce the variables

$$\delta_\ell := \sum_{j=1}^{m} \lambda_{j\ell},$$

which indicate whether or not the vertex $\boldsymbol{v}^{(\ell)}$ is used. Because of (6.17) we have $\boldsymbol{\delta} \in \{0,1\}^k$. Using

$$\sum_{j=1}^{m} \sum_{\ell=1}^{k} \lambda_{j\ell} v_i^{(\ell)} = \sum_{\ell=1}^{k} v_i^{(\ell)} \sum_{j=1}^{m} \lambda_{j\ell} = \sum_{\ell \in S_i} \delta_\ell,$$

we obtain

$$\begin{aligned} \min \quad & \mathbf{1}^\top \boldsymbol{\delta} \\ \text{s.t.} \quad & \sum_{\ell \in S_i} \delta_\ell = 1 && \forall i \in \{1,\dots,n\} \\ & \boldsymbol{\delta} \in \{0,1\}^k. \end{aligned} \tag{6.19}$$

Here $\boldsymbol{\delta}$ encodes a partition of the items into configurations. The exact assignment to the bins is implicit.

Example 6.21. Let $m = n = 4$, $w_1 = \frac{3}{4}$, $w_2 = \frac{1}{4}$, $w_3 = \frac{1}{2}$, $w_4 = \frac{1}{2}$ and $W = 1$. A possible solution with two bins is

$$(x_{ij}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The vertices of the knapsack polytope are

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{v}^{(1)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{v}^{(2)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \boldsymbol{v}^{(8)} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

Thus, $S_1 = \{1,5\}$, $S_2 = \{2,5,6,7\}$, $S_3 = \{3,6,8\}$ and $S_4 = \{4,7,8\}$. The solution is obtained by combining the vertices $\boldsymbol{v}^{(5)}$ and $\boldsymbol{v}^{(8)}$, i.e.,

$$\boldsymbol{\delta} = (\ 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1\ )^\top,$$

which corresponds to opening 2 bins, one containing the items $\{1,2\}$ (since $5 \in S_1 \cap S_2$), the other containing the items $\{3,4\}$ (since $8 \in S_3 \cap S_4$). △

The pricing problem aims to find a configuration to include in the current basis, or, equivalently, a violated constraint of the dual of (6.19). The dual is given by

$$\max\{\mathbf{1}^\top \boldsymbol{y} : \sum_{i:\, \ell \in S_i} y_i \leq 1 \ \forall \ell \in \{1,\dots,k\}\},$$

where $\{i : \ell \in S_i\}$ are the items used by the configuration corresponding to vertex $\boldsymbol{v}^{(\ell)}$. We can find a vertex violating its constraint for the current dual solution $\tilde{\boldsymbol{y}}$ via

$$\begin{aligned} \max \quad & \tilde{\boldsymbol{y}}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{w}^\top \boldsymbol{x} \le W \\ & \boldsymbol{x} \in \{0,1\}^m. \end{aligned}$$

This problem is bounded. If the optimum vertex solution $\bar{\boldsymbol{x}}$ satisfies $\tilde{\boldsymbol{y}}^\top \bar{\boldsymbol{x}} > 1$, it yields a column to add to the basis for the master problem, otherwise $\tilde{\boldsymbol{y}}$ is dually feasible and the current solution of the master problem is optimal.

In practice, this Dantzig-Wolfe formulation results in a very strong bound: Often the rounded value of the LP relaxation of the master problem (at the root of the branch-and-bound tree) already gives the optimum number of bins for currently allowed configurations (cf. (6.16) for the LP relaxation of the original formulation).

6.3 Benders' decomposition

Finally, we consider the *Benders' decomposition* (see [4]). Benders' decomposition also eliminates part of the constraint matrix, but, unlike the Dantzig-Wolfe decomposition, where we delete a part of the constraints and reinsert them by means of column generation, we now delete part of the variables and reinsert them using cutting planes. From this perspective, Benders' decomposition is equivalent to Dantzig-Wolfe decomposition applied to the dual problem (see Section 6.4). Consider a MIP of the form

$$\begin{aligned} \max \ & (\boldsymbol{c^{(1)}})^\top \boldsymbol{x^{(1)}} + (\boldsymbol{c^{(2)}})^\top \boldsymbol{x^{(2)}} \\ \text{s.t.} \ & A^{(1)} \boldsymbol{x^{(1)}} + A^{(2)} \boldsymbol{x^{(2)}} \le \boldsymbol{b}, \\ & \boldsymbol{x^{(1)}} \in \mathbb{Z}^{n_1},\ \boldsymbol{x^{(2)}} \in \mathbb{R}^{n_2}, \end{aligned} \tag{6.20}$$

where $A = [A^{(1)}, A^{(2)}] \in \mathbb{Q}^{m \times n}$, $A^{(1)} \in \mathbb{Q}^{m \times n_1}$, $A^{(2)} \in \mathbb{Q}^{m \times n_2}$, $\boldsymbol{c^{(1)}} \in \mathbb{Q}^{n_1}$, $\boldsymbol{c^{(2)}} \in \mathbb{Q}^{n_2}$ with $n_1 + n_2 = n$. Note that, for the sake of simplicity, we assume that all integer variables are in $\boldsymbol{x^{(1)}}$. We want to eliminate the variables $\boldsymbol{x^{(2)}}$ via projection.

To apply projection we start by reformulating (6.20) to

$$\begin{aligned} \max \quad & z \ (+ \boldsymbol{0}^\top \boldsymbol{x^{(2)}}) \\ \text{s.t.} \quad & z \le (\boldsymbol{c^{(1)}})^\top \boldsymbol{x^{(1)}} + (\boldsymbol{c^{(2)}})^\top \boldsymbol{x^{(2)}} \\ & A^{(1)} \boldsymbol{x^{(1)}} + A^{(2)} \boldsymbol{x^{(2)}} \le \boldsymbol{b} \\ & z \in \mathbb{R},\ \boldsymbol{x^{(1)}} \in \mathbb{Z}^{n_1},\ \boldsymbol{x^{(2)}} \in \mathbb{R}^{n_2}. \end{aligned} \tag{6.21}$$

By strong duality, this is equivalent to

$$\begin{aligned} & \max_{\boldsymbol{x^{(1)}} \in \mathbb{Z}^{n_1}, z \in \mathbb{R}} \left\{ z + \max_{\boldsymbol{x^{(2)}} \in \mathbb{R}^{n_2}} \{ \boldsymbol{0}^\top \boldsymbol{x^{(2)}} : -(\boldsymbol{c^{(2)}})^\top \boldsymbol{x^{(2)}} \le (\boldsymbol{c^{(1)}})^\top \boldsymbol{x^{(1)}} - z,\ A^{(2)} \boldsymbol{x^{(2)}} \le \boldsymbol{b} - A^{(1)} \boldsymbol{x^{(1)}} \} \right\} \\ = & \max_{\boldsymbol{x^{(1)}} \in \mathbb{Z}^{n_1}, z \in \mathbb{R}} \left\{ z + \min_{\gamma \in \mathbb{R}, \boldsymbol{v} \in \mathbb{R}^m} \{ \gamma \big((\boldsymbol{c^{(1)}})^\top \boldsymbol{x^{(1)}} - z \big) + \boldsymbol{v}^\top (\boldsymbol{b} - A^{(1)} \boldsymbol{x^{(1)}}) : \begin{pmatrix} \gamma \\ \boldsymbol{v} \end{pmatrix} \in \mathcal{C} \} \right\}, \end{aligned} \tag{6.22}$$

with

$$\mathcal{C} = \{ \begin{pmatrix} \gamma \\ \boldsymbol{v} \end{pmatrix} \in \mathbb{R}^{m+1} : \boldsymbol{v}^\top A^{(2)} - \gamma (\boldsymbol{c^{(2)}})^\top = \boldsymbol{0}^\top,\ \gamma \ge 0,\ \boldsymbol{v} \ge \boldsymbol{0} \}. \tag{6.23}$$

Since the primal objective function is $\mathbf{0}^\top \boldsymbol{x}^{(2)} = 0$, we have that (6.21) is feasible for some z if and only if the value of the inner minimization problem in (6.22) is 0. Otherwise, since $\mathbf{0} \in \mathcal{C} \neq \emptyset$ the inner minimum is unbounded. Thus, (6.22) is equivalent to

$$\max_{\boldsymbol{x}^{(1)} \in \mathbb{Z}^{n_1}, z \in \mathbb{R}} \{z : \gamma\big((\boldsymbol{c}^{(1)})^\top \boldsymbol{x}^{(1)} - z\big) + \boldsymbol{v}^\top (\boldsymbol{b} - A^{(1)} \boldsymbol{x}^{(1)}) \geq 0 \quad \forall \begin{pmatrix} \gamma \\ \boldsymbol{v} \end{pmatrix} \in \mathcal{C}\}. \tag{6.24}$$

Since $\mathcal{C}$ is a polyhedral cone, by Theorem 2.17, there are vectors $\binom{\gamma_1}{\boldsymbol{v}^{(1)}}, \ldots, \binom{\gamma_s}{\boldsymbol{v}^{(s)}}$ with

$$\mathcal{C} = \operatorname{cone}(\{\binom{\gamma_1}{\boldsymbol{v}^{(1)}}, \ldots, \binom{\gamma_s}{\boldsymbol{v}^{(s)}}\}).$$

Because $\mathcal{C} \subseteq \mathbb{R}^m_{\geq 0}$, we can assume by rescaling, that $\boldsymbol{\gamma} \in \{0,1\}^s$, i.e.

$$C = \operatorname{cone}(\{\binom{0}{\boldsymbol{v}^{(k)}} : k \in K\}) + \operatorname{cone}(\{\binom{1}{\boldsymbol{v}^{(j)}} : j \in J\}),$$

with $K \cup J = \{1, \ldots, s\}$ and $K \cap J = \emptyset$. With this description of $\mathcal{C}$, (6.24) can be rewritten as

$$\begin{aligned} \max \quad & z \\ \text{s.t.} \quad & z \leq (\boldsymbol{c}^{(1)})^\top \boldsymbol{x}^{(1)} + (\boldsymbol{v}^{(j)})^\top (\boldsymbol{b} - A^{(1)} \boldsymbol{x}^{(1)}) && \forall j \in J, \\ & 0 \leq (\boldsymbol{v}^{(k)})^\top (\boldsymbol{b} - A^{(1)} \boldsymbol{x}^{(1)}) && \forall k \in K, \\ & z \in \mathbb{R},\ \boldsymbol{x}^{(1)} \in \mathbb{Z}^{n_1}. \end{aligned} \tag{6.25}$$

The problem (6.25) is the *Benders' master problem*. It arises from (6.21) by orthogonal projection in the sense that z, $\boldsymbol{x}^{(1)}$ are feasible for (6.25) if and only if $\boldsymbol{x}^{(2)}$ exists, so that z, $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$ are feasible for (6.21). The master problem only has $n_1 + 1$ variables instead of the $n_1 + n_2$ variables in (6.20). Furthermore, the MIP (6.20) has been converted into an integer program (6.25) with an additional continuous variable z. Nevertheless, (6.25) contains a large number of constraints – generally exponentially many in n.

To circumvent this problem, we solve Benders' master problem by means of an iterative procedure as follows: We start with a small subset of extreme rays of $\mathcal{C}$ (possibly with the empty set) and optimize (6.25) over this subset (e.g., via Branch-and-Bound, see Chapter 3). We obtain an optimum solution $\hat{\boldsymbol{x}}^{(1)}$, $\hat{z}$ of the relaxed problem and have to check whether this solution satisfies all other inequalities in (6.25). This can be done using the *Benders' subproblem* (see (6.23) and (6.24))

$$\begin{aligned} \min \quad & \boldsymbol{v}^\top (\boldsymbol{b} - A^{(1)} \hat{\boldsymbol{x}}^{(1)}) - \gamma(\hat{z} - (\boldsymbol{c}^{(1)})^\top \hat{\boldsymbol{x}}^{(1)}) \\ \text{s.t.} \quad & \boldsymbol{v}^\top A^{(2)} - \gamma (\boldsymbol{c}^{(2)})^\top = \mathbf{0}^\top \\ & \gamma \geq 0,\ \boldsymbol{v} \geq \mathbf{0}. \end{aligned} \tag{6.26}$$

This problem is feasible since $\mathbf{0} \in \mathcal{C}$, and has an optimum solution of value 0 or is unbounded (see discussion of (6.22)). In the first case, $(\hat{\boldsymbol{x}}^{(1)}, \hat{z})$ satisfies all inequalities in (6.25) and we have solved (6.25) optimally and thus (6.20). In the other case, we obtain (e.g., via the Simplex method) an extreme ray $\binom{\hat{\gamma}}{\hat{\boldsymbol{v}}}$ from (6.26) with

$$\hat{\boldsymbol{v}}^\top (\boldsymbol{b} - A^{(1)} \hat{\boldsymbol{x}}^{(1)}) - \hat{\gamma}(\hat{z} - (\boldsymbol{c}^{(1)})^\top \hat{\boldsymbol{x}}^{(1)}) < 0,$$

which, after rescaling to $\hat{\gamma} \in \{0,1\}$, yields an inequality for (6.25), which is violated by $\hat{\boldsymbol{x}}^{(1)}$, $\hat{z}$. We add this cut to the Benders master problem (6.25) and iterate.

6.3.1 Application in stochastic optimization*

We return to the example of stochastic optimization from Section 6.1.3 and apply Benders' decomposition directly. Here, $A^{(1)}\boldsymbol{x}^{(1)} + A^{(2)}\boldsymbol{x}^{(2)} \leq \boldsymbol{b}$ consists of all blocks $T^{(\omega)}\boldsymbol{x} + W^{(\omega)}\boldsymbol{y}^{(\omega)} \leq \boldsymbol{h}^{(\omega)}$ for all $\omega \in \Omega$. This leaves $\boldsymbol{x}^{(1)} = \boldsymbol{x}$ in the master problem and $\boldsymbol{x}^{(2)} = (\boldsymbol{y}^{(\omega)})_{\omega\in\Omega}$ goes to the subproblem. The master problem can then be written as

$$\begin{aligned}
\min \quad & \boldsymbol{c}^\top \boldsymbol{x} + \sum_{\omega\in\Omega} p_\omega z_\omega \\
\text{s.t.} \quad & z_\omega \geq (\boldsymbol{v}^{(\omega,j)})^\top (\boldsymbol{h}^{(\omega)} - T^{(\omega)}\boldsymbol{x}), && \forall\, \omega \in \Omega,\ j \in J_\omega, \\
& 0 \leq (\boldsymbol{v}^{(\omega,k)})^\top (\boldsymbol{h}^{(\omega)} - T^{(\omega)}\boldsymbol{x}), && \forall\, \omega \in \Omega,\ k \in K_\omega, \\
& A\boldsymbol{x} \leq \boldsymbol{b}, \quad \boldsymbol{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}.
\end{aligned}$$

Note that we minimize, which leads to slight adjustments. The vectors $(\boldsymbol{v}^{(\omega,j)})_{j\in J_\omega}$ and $(\boldsymbol{v}^{(\omega,k)})_{k\in K_\omega}$ are the extreme rays and vertices of

$$\mathcal{C}_\omega := \{\begin{pmatrix}\gamma \\ \boldsymbol{v}\end{pmatrix} \in \mathbb{R}^{m+1} : \boldsymbol{v}^\top W^{(\omega)} - \gamma(\boldsymbol{d}^{(\omega)})^\top = \boldsymbol{0}^\top,\ \gamma \geq 0,\ \boldsymbol{v} \geq \boldsymbol{0}\}.$$

The resulting *L-method* (see [41]) can be accelerated using various techniques and is very effective in many applications. However, it is applicable in the presented form only if the second-stage variables are all continuous.

6.3.2 Application in combinatorial optimization*

We consider the *maximum feasible subsystem problem*: Given an *infeasible* system $\tilde{A}\boldsymbol{x} \leq \tilde{\boldsymbol{b}}$ ($\tilde{A} \in \mathbb{R}^{m\times n}$, $\tilde{\boldsymbol{b}} \in \mathbb{R}^m$), identify a feasible subsystem of largest possible size. This can be expressed as MIP via

$$\begin{aligned}
\max \quad & \boldsymbol{1}^\top \boldsymbol{x}^{(1)} + \boldsymbol{0}^\top \boldsymbol{x}^{(2)} \\
\text{s.t.} \quad & \tilde{A}\boldsymbol{x}^{(2)} \leq \tilde{\boldsymbol{b}} + M(\boldsymbol{1} - \boldsymbol{x}^{(1)}) \\
& \boldsymbol{x}^{(1)} \in \{0,1\}^m,\ \boldsymbol{x}^{(2)} \in \mathbb{R}^n.
\end{aligned} \tag{6.27}$$

Here, M is a very large number that guarantees the following: If $x_i^{(1)} = 0$, then $\tilde{\boldsymbol{A}}_{i\cdot}\boldsymbol{x} - \tilde{\boldsymbol{b}}_i \leq M$ for all interesting $\boldsymbol{x}$. This means that the binary variables $\boldsymbol{x}^{(1)}$ activate/deactivate the respective inequalities. Note that the auxiliary variable z is not used here, because $\boldsymbol{c}^{(2)} = \boldsymbol{0}$.

We can now reformulate (6.27) to

$$\max_{\boldsymbol{x}^{(1)}\in\{0,1\}^m} \left\{\boldsymbol{1}^\top \boldsymbol{x}^{(1)} + \max_{\boldsymbol{x}^{(2)}\in\mathbb{R}^n} \{\boldsymbol{0}^\top \boldsymbol{x}^{(2)} : \tilde{A}\boldsymbol{x}^{(2)} \leq \tilde{\boldsymbol{b}} + M\,(\boldsymbol{1} - \boldsymbol{x}^{(1)})\}\right\}$$

$$= \max_{\boldsymbol{x}^{(1)}\in\{0,1\}^m} \left\{\boldsymbol{1}^\top \boldsymbol{x}^{(1)} + \min_{v\in\mathbb{R}^m} \{\boldsymbol{v}^\top(\tilde{\boldsymbol{b}} + M\,(\boldsymbol{1} - \boldsymbol{x}^{(1)})) : \boldsymbol{v}^\top \tilde{A} = \boldsymbol{0}^\top,\ \boldsymbol{v} \geq \boldsymbol{0}\}\right\}.$$

We then have the cone $\mathcal{C} = \{\boldsymbol{v} : \boldsymbol{v}^\top \tilde{A} = \boldsymbol{0}^\top,\ \boldsymbol{v} \geq \boldsymbol{0}\}$ with rays $(\boldsymbol{v}^{(k)})_{k\in K}$, and therefore $\mathcal{C} = \text{cone}(\{\boldsymbol{v}^{(k)} : k \in K\})$. The Benders' master problem (6.25) is then

$$\begin{aligned}
\max \quad & \boldsymbol{1}^\top \boldsymbol{x}^{(1)} \\
\text{s.t.} \quad & 0 \leq (\boldsymbol{v}^{(k)})^\top (\tilde{\boldsymbol{b}} + M\,(\boldsymbol{1} - \boldsymbol{x}^{(1)})) \quad \forall k \in K, \\
& \boldsymbol{x}^{(1)} \in \{0,1\}^m,
\end{aligned}$$

and the Benders' subproblem (6.26) becomes

$$\begin{aligned} \min \quad & \boldsymbol{v}^\top(\tilde{\boldsymbol{b}} + M\,(\mathbf{1} - \hat{\boldsymbol{x}}^{(\mathbf{1})})) \\ \text{s.t.} \quad & \boldsymbol{v}^\top \tilde{A} = \mathbf{0}^\top,\ \boldsymbol{v} \geq \mathbf{0}. \end{aligned}$$

This can then be used to apply the solution scheme described above. We discuss this implementation further. Let $\hat{\boldsymbol{x}}^{(\mathbf{1})} \in \{0,1\}^m$ be the solution of the (incomplete) Benders' master problem and $\hat{\boldsymbol{v}}$ be the solution of the corresponding Benders' subproblem. If the optimum value of the subproblem is 0, we are done. Otherwise it is smaller 0 and the following applies.

Lemma 6.22. If $\hat{\boldsymbol{v}}^\top(\tilde{\boldsymbol{b}} + M\,(\mathbf{1} - \hat{\boldsymbol{x}}^{(\mathbf{1})})) < 0$, then the rows of $\tilde{A}x \leq \tilde{\boldsymbol{b}}$ form an invalid subsystem with respect to $S := \{i : \hat{v}_i > 0\}$.

Proof. According to the Farkas lemma, it holds that $\tilde{\boldsymbol{A}}_{\boldsymbol{S}}.x \leq \tilde{\boldsymbol{b}}_{\boldsymbol{S}}$ is infeasible (exactly) if $\boldsymbol{y} \geq \mathbf{0}$ exists with $\boldsymbol{y}^\top \tilde{\boldsymbol{b}}_{\boldsymbol{S}} < 0$ and $\boldsymbol{y}^\top \tilde{\boldsymbol{A}}_{\boldsymbol{S}}. = \mathbf{0}^\top$. With $\boldsymbol{y} := \hat{\boldsymbol{v}}_{\boldsymbol{S}}$ we have $\boldsymbol{y} \geq \mathbf{0}$ and $\boldsymbol{y}^\top \tilde{\boldsymbol{A}}_{\boldsymbol{S}}. = \mathbf{0}^\top$, because of feasibility of $\hat{\boldsymbol{v}}$ for the Benders' subproblem. By assumption, $\hat{\boldsymbol{v}}^\top \tilde{\boldsymbol{b}} + M\,\hat{\boldsymbol{v}}^\top(\mathbf{1} - \hat{\boldsymbol{x}}^{(\mathbf{1})}) < 0$. Because M is sufficiently large, $\hat{\boldsymbol{x}}^{(\mathbf{1})}_{\boldsymbol{S}} = \mathbf{1}$ must hold. It follows that $0 > \hat{\boldsymbol{v}}^\top \tilde{\boldsymbol{b}} + M\,\hat{\boldsymbol{v}}^\top(\mathbf{1} - \hat{\boldsymbol{x}}^{(\mathbf{1})}) = \hat{\boldsymbol{v}}^\top \tilde{\boldsymbol{b}} = \hat{\boldsymbol{v}}^\top \tilde{\boldsymbol{b}}_{\boldsymbol{S}} = \boldsymbol{y}^\top \tilde{\boldsymbol{b}}_{\boldsymbol{S}}$, which proves infeasibility. □

A corresponding ray $\hat{\boldsymbol{v}}$ of the Benders' subproblem thus yields an infeasible subsystem in the part of $\tilde{A}\boldsymbol{x} \leq \tilde{\boldsymbol{b}}$ in which $\hat{v}_i > 0$ holds. The corresponding inequalities in the master problem can therefore be reduced to

$$\sum_{i \in S} x_i^{(1)} \leq |S| - 1,$$

because at least one $x_i^{(1)}$ must be set to 0 (i.e., must be removed from the system).

6.4 Connections between these approaches

At first glance, Lagrangian relaxation, Dantzig-Wolfe decomposition and Benders' decomposition seem to be different approaches relying on different relaxations. However, they are strongly related to each other. Consider again (6.13), which, for a fixed $\bar{\boldsymbol{y}} \leq \mathbf{0}$, can be written as

$$\begin{array}{lcll} \max\,(\boldsymbol{c}^\top - \bar{\boldsymbol{y}}^\top A^{(1)})\boldsymbol{x} & = & \max\,\boldsymbol{c}^\top \boldsymbol{x} + \bar{\boldsymbol{y}}^\top(\boldsymbol{b}^{(\mathbf{1})} - A^{(1)}\boldsymbol{x}) - \bar{\boldsymbol{y}}^\top \boldsymbol{b}^{(\mathbf{1})} & = \quad L(\bar{\boldsymbol{y}}) - \bar{\boldsymbol{y}}^\top \boldsymbol{b}^{(\mathbf{1})}, \\ \text{s.t. } \boldsymbol{x} \in \mathcal{P}^{(2)} & & \text{s.t. } \boldsymbol{x} \in \mathcal{P}^{(2)} & \end{array}$$

i.e., (6.3) and (6.13) define the same problems except for a constant shift $-\bar{\boldsymbol{y}}^\top \boldsymbol{b}$ in the objective. Furthermore, by replacing $\mathcal{P}^{(2)}$ with $\text{conv}(\{\boldsymbol{v}^{(\mathbf{1})}, \ldots, \boldsymbol{v}^{(\boldsymbol{k})}\}) + \text{cone}(\{\boldsymbol{r}^{(\mathbf{1})}, \ldots, \boldsymbol{r}^{(\boldsymbol{\ell})}\})$, it can be shown that (6.12) coincides with the right-hand side in Theorem 6.1 and thus with $\min_{\bar{y}} L(\bar{\boldsymbol{y}})$. In other words, the Dantzig-Wolfe decomposition and the Lagrangian relaxations compute the same bound. The only differences are that, for the update of the dual variables, i.e., $\boldsymbol{\lambda}$ in the Lagrangian relaxation and $\bar{\boldsymbol{y}}$ in the Dantzig-Wolfe decomposition, subgradient methods are used in the former case, whereas LP techniques are applied in the latter.

Analogously, Benders' decomposition is nothing other than the Dantzig-Wolfe decomposition applied to the dual of (6.20). To see this, consider its dual LP

$$\begin{aligned} \min \quad & \boldsymbol{y}^\top \boldsymbol{b} \\ \text{s.t.} \quad & \boldsymbol{y}^\top A^{(1)} = (\boldsymbol{c}^{(\mathbf{1})})^\top, \\ & \boldsymbol{y}^\top A^{(2)} = (\boldsymbol{c}^{(\mathbf{2})})^\top, \\ & \boldsymbol{y} \geq \mathbf{0}. \end{aligned} \tag{6.28}$$

Now we write

$$\bar{\mathcal{P}}^{(2)} = \{\boldsymbol{y} \in \mathbb{R}^m \,:\, \boldsymbol{y}^\top A^{(2)} = (\boldsymbol{c^{(2)}})^\top,\ \boldsymbol{y} \geq \boldsymbol{0}\}$$

as

$$\bar{\mathcal{P}}^{(2)} = \operatorname{conv}(\{\boldsymbol{v^{(j)}} \,:\, j \in J\}) \,+\, \operatorname{cone}(\{\boldsymbol{v^{(k)}} \,:\, k \in K\}),$$

where K, J and $\boldsymbol{v^{(\ell)}}$ for $\ell \in K \cup J$ are exactly the quantities of (6.25) (set $\boldsymbol{v} = \gamma \boldsymbol{y}$, and note that $\operatorname{hog}(\bar{\mathcal{P}}^{(2)}) := \{\binom{1}{y} : \boldsymbol{y} \in \bar{\mathcal{P}}^{(2)}\}^{\circ\circ} = \mathcal{C}$ (*exercise*)), and write (6.28) as

$$\begin{array}{ll} \min & \sum_{j\in J}((\boldsymbol{v^{(j)}})^\top \boldsymbol{b})\,\lambda_j + \sum_{k\in K}((\boldsymbol{v^{(k)}})^\top \boldsymbol{b})\,\mu_k \\ \text{s.t.} & \sum_{j\in J}((\boldsymbol{v^{(j)}})^\top A^{(1)})\,\lambda_j + \sum_{k\in K}((\boldsymbol{v^{(k)}})^\top A^{(1)})\,\mu_k = (\boldsymbol{c^{(1)}})^\top, \\ & \sum_{j\in J}\lambda_j = 1, \\ & \boldsymbol{\lambda} \in \mathbb{R}^J_{\geq 0}, \boldsymbol{\mu} \in \mathbb{R}^K_{\geq 0}. \end{array} \tag{6.29}$$

We now conclude from the results of Section 6.2 that (6.29) is the master problem of (6.28). Dualization of (6.29) results in

$$\begin{array}{lllll} \max & (\boldsymbol{c^{(1)}})^\top \boldsymbol{x^{(1)}} + z & & & \\ \text{s.t.} & (\boldsymbol{v^{(j)}})^\top (A^{(1)}\boldsymbol{x^{(1)}} - \boldsymbol{b}) & \leq & -z & \forall\ j \in J, \\ & (\boldsymbol{v^{(k)}})^\top (A^{(1)}\boldsymbol{x^{(1)}} - \boldsymbol{b}) & \leq & 0 & \forall\ k \in K, \end{array}$$

which is equivalent to (6.25), i.e., to Benders' master problem (6.20). In other words, for LPs, Benders' and Dantzig-Wolfe decomposition result the same bound, which, according to the considerations above, is equal to the value of the Lagrangian relaxation (6.3).

7 Heuristics

In this chapter, we tackle the question how to quickly find good feasible solutions for a mixed integer program, e.g., for the acceleration of the branch-and-bound method (see Chapter 3). Also, the size of practical instances often does not allow for exact but exponential-time solution methods. In both cases, we are willing to sacrifice optimality for the sake of efficiency. Fast algorithms that come without guarantees regarding solution quality are called *heuristics*. In contrast, in Chapter 8, we will see *approximation algorithms,* which, similiarly, sacrifice optimality for the sake of efficieny, but come with a provable solution quality.

Heuristics are usually tailored to exploit the specific structure of the problem at hand and to the distribution of the instances that we can expect. That being said, in the following we present some general methods, which are used as a basis for many heuristics in practice.

7.1 The greedy algorithm

The greedy algorithm constructs a solution incrementally by iteratively adding an element to the solution that increases the objective function the most / the least. In order for this approach to be applicable, we need that every feasible solution can be constructed incrementally. This can elegantly be expressed by considering set systems of feasible solutions that are closed under taking subsets.

Definition 7.1. An *independence system* is a pair $(E, \mathcal{I})$ consisting of a finite ground set E and a family of subsets $\mathcal{I} \subseteq 2^E$ such that $\mathcal{I} \neq \emptyset$ and $I \in \mathcal{I}$ for all $I \subseteq J \in \mathcal{I}$. A set $I \subseteq E$ is *independent* if $I \in \mathcal{I}$, and a *basis* of $J \subseteq E$ if $I' \notin \mathcal{I}$ for all $I \subsetneq I' \subseteq J$.

This abstract framework captures many classical optimization problems.

Example 7.2. The following are independence systems with the set of bases of E denoted by $\mathcal{B} \subseteq \mathcal{I}$.

- linear independence: $E = \{\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(k)}\} \subseteq \mathbb{R}^n$ and $\mathcal{I}$ consists of all sets of linearly independent vectors. Then, $\max_{I \in \mathcal{I}} |I|$ is the problem of finding a basis of $\operatorname{lin}(\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(k)})$.
- knapsack: $E = \{1, \dots, n\}$ and $\mathcal{I} = \{I \subseteq E : \sum_{i \in I} a_i \leq \beta\}$. Then, $\max_{I \in \mathcal{I}} \sum_{i \in I} c_i$ is the knapsack problem (see Example 1.6).
- stable sets: $E = V$ are the vertices of a graph G and $\mathcal{I} = \{I \subseteq E : G[I] = (I, \emptyset)\}$. Then, $\max_{I \in \mathcal{I}} |I|$ is the stable set problem (see *Algorithmic Discrete Mathematics*).
- cycles: E are the edges of a graph and $\mathcal{I} = \{I \subseteq E : I \text{ has no cycle}\}$. Then, $\min_{B \in \mathcal{B}} \sum_{e \in B} c(e)$ is the minimum spanning tree problem (see *Algorithmic Discrete Mathematics*).
- tours: E are the edges of a complete graph and $\mathcal{I} = \{I \subseteq E : \exists \text{ tour } T \supseteq I \text{ in the graph}\}$. Then, $\min_{B \in \mathcal{B}} \sum_{e \in B} c(e)$ is the traveling salesperson problem (see Section 6.1.4). △

The greedy algorithm can abstractly be formulated as follows.

Algorithm: greedy algorithm for maximization or minimization

input: independence system $(E, \mathcal{I})$, objective $c : E \to \mathbb{R}_{\geq 0}$
output: basis of $(E, \mathcal{I})$

$I \leftarrow \emptyset$, $E_0 \leftarrow \emptyset$, $j \leftarrow 0$
while $E_j \neq E$ **:**
  $e \leftarrow \arg\max$ [or min] $\{c(e') : e' \in E \setminus E_j\}$
  $E_{j+1} \leftarrow E_j \cup \{e\}$, $j \leftarrow j + 1$
  if $I \cup \{e\} \in \mathcal{I}$ **:**
    $I \leftarrow I \cup \{e\}$
return I

By definition, the greedy algorithm maintains the invariant that, no previously considered element can be included in the solution.

Observation 7.3. The solution computed by the greedy algorithm contains a basis of E_j for all $j \in \{1, \ldots, |E|\}$.

In some cases, the greedy algorithm computes an optimum solution, as we will see in more detail in the next section.

Example 7.4. In the cycle-system of Example 7.2, the greedy algorithm is equivalent to Kruskal's algorithm for computing a minimum spanning tree (see *Algorithmic Discrete Mathematics*). △

On the other hand, the greedy algorithm sometimes performs very poorly.

Example 7.5. In the knapsack-system of Example 7.2 with costs $\boldsymbol{c} = (n-1)\mathbf{1} + \mathbf{e^1}$, weights $\boldsymbol{a} = \mathbf{1} + (n-2)\mathbf{e^1}$ and capacity $\beta = n-1$, the greedy algorithm finds the solution $I = \{1\}$ of value n, while the optimum solution $I^\star = \{2, \ldots, n\}$ has value $(n-1)^2$. △

7.1.1 Matroids

We have seen that the quality of the solutions produced by the greedy algorithm can vary wildly, depending on the application (see Examples 7.4 and 7.5). Remarkably, the cases when the greedy algorithm finds an optimum solution have an elegant characterization in terms of the following structure.

Definition 7.6. An independence system $(E, \mathcal{I})$ is a *matroid* if for all $F \subseteq E$ with bases $\mathcal{B}(F)$ it holds that

$$r(F) := \max\{|B| : B \in \mathcal{B}(F)\} = \min\{|B| : B \in \mathcal{B}(F)\} =: \rho(F),$$

where $r(F)$ is the *rank* of F and $\rho(F)$ is its *lower rank*.

In particular, the independence system underlying the minimum spanning tree problem is a matroid.

Example 7.7. The following independence systems of Example 7.2 are matroids.

- *vector matroid*: $E = \{\boldsymbol{v^{(1)}}, \ldots, \boldsymbol{v^{(k)}}\} \subseteq \mathbb{R}^n$ and $\mathcal{I}$ consists of all sets of linearly independent vectors, since all bases of a subset $F \subseteq E$ have cardinality $\dim(\mathrm{lin}(F))$.

- *cycle matroid*: E are the edges of a graph and $\mathcal{I} = \{I \subseteq E : I \text{ has no cycle}\}$, since all bases of a subset $F \subseteq E$ have cardinality $|F| - k$, where k is the number of connected components of F. △

Recall that the greedy algorithm is optimal for the cycle matroid (see 7.4). On the other hand, the knapsack system, where the greedy algorithm does not yield a good solution (see 7.5), is not a matroid, and neither is the system for TSP.

Example 7.8. The following independence systems of Example 7.2 are not matroids.

- knapsack system: Not all maximal packings involve the same number of items, i.e., not all bases of E have the same cardinality.
- tours system: While all bases of E have the same cardinality, the same is not true for the bases of subsets of E. To see this, consider the complete graph on 4 vertices. The subgraph obtained by ommitting an arbitrary edge has a cycle of length 4 and a path of length 3 as its bases. △

We now more generally characterize the solution quality of the greedy algorithm via the *rank quotient* $q := \max_{F \subseteq E:\, r(F)>0} r(F)/\rho(F)$.

Theorem 7.9 ([27])**.** Let $(E, \mathcal{I})$ be an independence system, $c : E \to \mathbb{R}_{\geq 0}$ be an objective to be maximized, $I^\star \in \arg\max_{I \in \mathcal{I}} c(I)$ be an optimum solution with $c(I^\star) > 0$, and I be the solution found by the greedy algorithm. Then,

$$\frac{c(I^\star)}{c(I)} \leq q := \max_{F \subseteq E:\, r(F)>0} \frac{r(F)}{\rho(F)},$$

and there exists $c : E \to \{0, 1\}$ such that equality holds.

Proof. For all $j \in \{1, \ldots, |E|\}$, Observation 7.3 implies that $|I \cap E_j| \geq \rho(E_j)$ and $I^\star \cap E_j \in \mathcal{I}$ implies $|I^\star \cap E_j| \leq r(E_j)$. Hence, we obtain (even if $r(E_j) = 0$)

$$|I \cap E_j| \geq (1/q) \cdot |I^\star \cap E_j|. \tag{7.1}$$

Let $\{e_j\} := E_j \setminus E_{j-1}$ for $j \in \{1, \ldots, |E|\}$ and $c(e_{|E|+1}) \equiv 0$. By definition of the greedy algorithm and since $c(e_{|E|}) \geq 0$, we have $c(e_j) \geq c(e_{j+1})$ for all $j \in \{1, \ldots, |E|\}$. This allows to apply (7.1) to obtain

$$\begin{aligned}
c(I) &= \sum_{j=1}^{|E|} (|I \cap E_j| - |I \cap E_{j-1}|)\, c(e_j) \\
&= \sum_{j=1}^{|E|} |I \cap E_j|\, (c(e_j) - c(e_{j+1})) \\
&\overset{(7.1)}{\geq} (1/q) \cdot \sum_{j=1}^{|E|} |I^\star \cap E_j|\, (c(e_j) - c(e_{j+1})) \\
&= (1/q) \cdot \sum_{j=1}^{|E|} (|I^\star \cap E_j| - |I^\star \cap E_{j-1}|)\, c(e_j) \\
&= (1/q) \cdot c(I^\star),
\end{aligned}$$

as claimed.

For the second part of the statement, observe that $c(I^\star) > 0$ implies that $F \subseteq E$ exists with $q = \frac{r(F)}{\rho(F)}$. Set $c(e) = 1$ for $e \in F$ and $c(e) = 0$ for $e \notin F$. Since the greedy algorithm does not specify how to break ties in arg max, we may assume that $E_{\rho(F)}$ is a basis of F. But because $c(e) = 0$ for all $e \notin F$, we have $c(I) = \rho(F) = (1/q) \cdot r(F) = (1/q) \cdot c(I^\star)$. □

With this we obtain that matroids capture exactly the independence systems for which the greedy algorithm is optimal.

Corollary 7.10. The following are equivalent for an independence system $(E, \mathcal{I})$:

(a) $(E, \mathcal{I})$ is a matroid.

(b) For all $c: E \to \mathbb{R}_{\geq 0}$, the greedy algorithm yields an optimum solution to $\max\{c(I) : I \in \mathcal{I}\}$.

(c) For all $c: E \to \{0, 1\}$, the greedy algorithm yields an optimum solution to $\max\{c(I) : I \in \mathcal{I}\}$.

(d) For all $c: E \to \mathbb{R}_{\geq 0}$, the greedy algorithm (for minimization) yields an optimum solution to $\min\{c(B) : B \in \mathcal{B}\}$.

Proof. The equivalence of the first three points is immediate from Theorem 7.9.

For the last point, observe that $\max\{\sum_{e \in I} c'(e) : I \in \mathcal{I}\}$ with $c'(e) := -c(e) + \sum_{e \in E} c(e) \geq 0$ has optimum solutions that are a bases of E (this may not be true for all optima, since $c(e) = 0$ is permitted). If $(E, \mathcal{I})$ is a matroid, all bases have the same cardinality, hence $c(B)$ and $c'(B)$ differ by the same constant for all bases B of E. This means that a basis is optimal for the maximization problem if and only if it is optimal for

$$\max\{-c(B) : I \in \mathcal{I}\} = -\min\{c(B) : B \in \mathcal{B}\}.$$

Finally, observe that the minimization version of the greedy algorithm for c produces the same solution as the maximization version for c' and vice-versa. Hence, the greedy algorithm is optimal for minimization for all $c: E \to \mathbb{R}_{\geq 0}$ if and only if it is optimal for maximization for all $c': E \to \mathbb{R}_{\geq 0}$. □

It is remarkable that we obtain a full characterization for the optimality of the greedy algorithm. In particular, Corollary 7.10 yields an alternate proof for the correctness of Kruskal's algorithm for the minimum spanning tree problem (see *Algorithmic Discrete Mathematics*). Note that the bound of Theorem 7.9 only carries over to minimization problems on matroids. The TSP problem has bounded rank quotient, but the greedy algorithm can produce arbitrarily bad solutions (*exercise*).

We further give a sufficient condition for an independence system to have bounded rank quotient, namely, when it arises as an intersection of matroids.

Theorem 7.11. Let $(E, \mathcal{I}_1), \ldots, (E, \mathcal{I}_k)$ be matroids. Then, $(E, \mathcal{I}) := (E, \bigcap_{i=1}^{k} \mathcal{I}_i)$ is an independence system with rank quotient $q = \max_{F \subseteq E:\, r(F) > 0} r(F)/\rho(F) \leq k$.

Proof. The fact that $(E, \mathcal{I}_1), \ldots, (E, \mathcal{I}_k)$ are independence systems immediately implies that $(E, \mathcal{I})$ is an independence system. Let $F \subseteq E$ and let $B, B' \subseteq F$ be any two bases of F with respect to $(E, \mathcal{I})$. It is sufficient to show that $|B'| \leq k \cdot |B|$.

Since $B \in \mathcal{I} = \bigcap_{i=1}^{k} \mathcal{I}_i$, we can choose bases $\bar{B}_i$ with $B \subseteq \bar{B}_i \subseteq B \cup B'$ with respect to $(E, \mathcal{I}_i)$ for $i \in \{1, \ldots, k\}$. We claim that $\bigcap_{i=1}^{k} \bar{B}_i = B$. To see this, suppose there exists $e \in \bigcap_{i=1}^{k} \bar{B}_i \setminus B$. Then, $B \cup \{e\} \subseteq \bigcap_{i=1}^{k} \bar{B}_i \in \mathcal{I}_j$ for all $j \in \{1, \ldots, k\}$ and thus $B \cup \{e\} \in \mathcal{I}$. Because of $B \cup \{e\} \in B \cup B' \subseteq F$, this contradicts B being a basis of F.

Now, our claim implies that every $e \in B' \setminus B$ is contained in at most $k-1$ of the sets $\bar{B}_i \setminus B$, which means that

$$\left(\sum_{i=1}^{k} |\bar{B}_i| \right) - k\,|B| = \sum_{i=1}^{k} |\bar{B}_i \setminus B| \leq (k-1)\,|B' \setminus B| \leq (k-1)\,|B'|. \tag{7.2}$$

As before, we can choose additional bases $\bar{B}_i'$ with $B' \subseteq \bar{B}_i' \subseteq B \cup B'$ with respect to $(E, \mathcal{I}_i)$ for $i \in \{1, \dots, k\}$. Since $(E, \mathcal{I}_i)$ is a matroid, we have

$$|\bar{B}_i| = |\bar{B}_i'| \qquad \forall i \in \{1, \dots, k\}. \tag{7.3}$$

Overall, we obtain

$$|B'| \leq |B'| + \sum_{i=1}^{k} |\bar{B}_i' \setminus B'| = \left(\sum_{i=1}^{k} |\bar{B}_i'| \right) - (k-1)|B'| \overset{(7.3)}{=} \left(\sum_{i=1}^{k} |\bar{B}_i| \right) - (k-1)|B'| \overset{(7.2)}{\leq} k|B|. \qquad \square$$

For most practical problems, the greedy algorithm can produce arbitrarily bad solutions. Nevertheless, due to its simplicity and efficiency, it is a popular choice in practice.

7.2 Local search

Once we found some feasible solution, we can try to improve it by modifications. In other words, we search a neighborhood for better solutions, update our solution, and iterate.

Algorithm: local search

input: feasible solutions $\mathcal{F}$, neighborhoods $N : \mathcal{F} \to \mathcal{F}$
objective $c : \mathcal{F} \to \mathbb{R}$, initial solution $S \in \mathcal{F}$

output: solution $S' \in \mathcal{F}$

while $\exists S' \in N(S)$ *with* $c(S') \gtrless c(S)$**:**
  $S \leftarrow S'$

return S

The quality of this algorithm critically depends on the definition of the neighborhood N.

Example 7.12.

- 2-opt heuristic for TSP: Here $\mathcal{F}$ is the set of all tours and the neighborhood of a tour T consists of the tours T' obtained from T via a pairwise exchange of two edges $\{u_1, v_1\}, \{u_2, v_2\}$ by the edges $\{u_1, v_2\}, \{u_2, v_1\}$.
- 1-exchange/2-exchange for the equipartition problem: We are given an undirected, edge-weighted graph $G = (V, E)$ with $|V|$ even and are looking for $S \subseteq V$ with $|S| = |V \setminus S| = |V|/2$, while maximizing the weights of the cut $\delta(S)$. The neighborhood of a solution $S \subseteq V$ could then, for example, be all $S' = S \setminus \{v\} \cup \{v'\}$ for $v \in S$, $v' \notin S$ (2-exchange). Alternatively, we can add a penalty term of the form $\alpha(|S| - |V \setminus S|)^2$ to the objective and allow 1-exchanges $S' = S \setminus \{v\}$, $v \in S$ or $S' = S \cup \{v'\}$, $v \notin S$. $\triangle$

A general problem of local search is that, inherently, it gets stuck in local extrema. We now discuss modifications of local search that alleviate this flaw to some extent.

7.2.1 Tabu search

The idea of *tabu search* is to escape local minima by forcing, in every step, to switch to a solution in the neighborhood that we didn't already visit – even if that means getting worse in the objective. However, it is generally computationally too expensive to track all previously encountered solutions, and we instead limit ourselves to a *tabu list* of limited length that only contains recently encountered solutions.

Algorithm: tabu search

input: feasible solutions $\mathcal{F}$, neighborhoods $N : \mathcal{F} \to \mathcal{F}$
objective $c : \mathcal{F} \to \mathbb{R}$, initial solution $S^{(0)} \in \mathcal{F}$
output: solution $S' \in \mathcal{F}$

$T \leftarrow \emptyset$ (tabu list)
for $i \leftarrow 1, 2, \ldots$ **:**
  $S^{(i)} \leftarrow \arg\max \mathit{or} \min\{c(S) : S \in N(S^{(i-1)}) \setminus T\}$
  $T \leftarrow (T_2, T_3, \ldots) \oplus (S^{(i)})$ (discard oldest)
return $\arg\max \mathit{or} \min\{S^{(0)}, \ldots, S^{(i)}\}$

Example 7.13. Consider the problem of, for given $k \in \mathbb{N}$, finding a coloring $\phi : V \to \{1, 2, \ldots, k\}$ of the vertices of an undirected graph $G = (V, E)$, such that $\phi(u) \neq \phi(v)$ for all edges $\{u, v\} \in E$. We minimize

$$c(\phi) := |\{\{u, v\} \in E : \phi(u) = \phi(v)\}|,$$

and define the neighborhood of ϕ to consist of the colorings that differ from ϕ in the color of a single vertex, i.e.,

$$N(\phi) := \{\varphi : V \to \{1, \ldots, k\} : |\{v \in V : \phi(v) \neq \varphi(v)\}| = 1\}. \qquad \triangle$$

The parameters of tabu search are the length of the tabu list and the stopping condition. A longer tabu list or execution potentially improves the result, but comes at the cost of deteriorating the running time. Natural stopping conditions are fixing the number of iterations without improvement or in total. The best choices of the associated parameters are highly problem-dependent and usually determined by experimental tuning.

7.2.2 Simulated annealing

Simulated annealing also allows to deteriorate the objective, but does so probabilistically instead of using a deterministic tabu list. More specifically, we allow worse solutions with a certain probability that depends on how much worse they are and decreases over the course of the execution of the algorithm. The method is inspired by the physics of crystal formation, which requires a gradual decrease in temperature to allow time for the individual particles to find their place in the crystaline structure. Accordingly, it maintains a temperature and uses a probability according to the Boltzmann distribution.

Algorithm: simulated annealing (for maximization)

input: feasible solutions $\mathcal{F}$, neighborhoods $N : \mathcal{F} \to \mathcal{F}$
objective $c : \mathcal{F} \to \mathbb{R}$, initial solution $S^{(0)} \in \mathcal{F}$
initial temperature $T \geq 1$, cooling rate θ
output: solution $S' \in \mathcal{F}$

for $i \leftarrow 1, 2, \ldots$ **:**
  randomly pick $S^{(i)} \in N(S^{(i-1)})$ and $\alpha \in [0, 1]$
  if $c(S^{(i)}) - c(S^{(i-1)}) < 0$ *and* $\mathrm{e}^{(c(S^{(i)}) - c(S^{(i-1)}))/T} < \alpha$ **:**
    $S^{(i)} \leftarrow S^{(i-1)}$ (discard solution)
  $T \leftarrow T \cdot \theta$
return $\arg\max\{c(S) : S \in \{S^{(0)}, \ldots, S^{(i)}\}\}$

Remark 7.14. It can be shown that there always exists parameters for which, with high probability, simulated annealing eventually finds an optimum solution (see [30]).

As for the previous heuristics, a lot of experimentation is required for tuning the parameters of simulated annealing, such as the starting temperature, the cooling rate, and the stopping criterion.

7.2.3 Genetic algorithms

While simulated annealing is physics inspired, *genetic algorithms* take their inspiration from biology. The basic idea is to simultaneously consider a set (called *population*) of candidate solutions and to improve that set by combining individual solutions. In each iteration (called *generation*) the method selects a series pairs of solutions (called *parents*) which are *crossed* to generate new solutions (called *offspring*). In addition, the resulting solutions are randomly perturbed (*mutated*) with a small probability to avoid getting stuck in local optima. A subset of the generated solutions is then selected, favoring solutions with better objective values (their *fitness*).

Algorithm: genetic algorithm

input: feasible solutions $\mathcal{F}$, objective $c\colon \mathcal{F} \to \mathbb{R}$
initial population $\mathcal{S}^{(0)} \subset \mathcal{F}$

output: solution $S \in \mathcal{F}$

for *generations* $i \leftarrow 1, 2, \ldots$ **:**
  for *parents* $\{S^{\bigcirc}, S^{\bullet}\} \subseteq \mathcal{S}^{(i-1)}$ **:**
    $S^{\CIRCLE} \leftarrow S^{\bigcirc} \times S^{\bullet}$ (reproduction)
    $\tilde{S}^{\CIRCLE} \leftarrow \begin{cases} \text{random perturbation of } S^{\CIRCLE}, & \text{with low probability,} \\ S^{\CIRCLE} & \text{otherwise.} \end{cases}$ (mutation)
    $\mathcal{S}^{(i)} \leftarrow \mathcal{S}^{(i)} \cup \{S^{\CIRCLE}\}$
  while $|\mathcal{S}^{(i)}|$ *too large* **:**
    choose $S \in \mathcal{S}^{(i)}$ with probability depending on $c(S)$ (selection)
    $\mathcal{S}^{(i)} \leftarrow \mathcal{S}^{(i)} \setminus \{S\}$
return $\arg\max[or\ \min]\{c(S) : S \in \mathcal{S}^{(0)} \cup \cdots \cup \mathcal{S}^{(i)}\}$

As with the previous methods, the parameters such as choice of parents, crossing method, mutation rate and process, selection criterion, stopping criterion, etc., are heavily problem dependent and require careful tuning.

8 Approximation Algorithms

The solution methods for MIPs covered in Chapters 3, 5, and 6 generally require exponential running times, and we cannot hope for more efficient methods due to the NP-hardness of solving integer programs (Theorem 3.1). In practice, exponential runtimes are computationally infeasible except for very small problem instances, and we can only hope that input instances are "friendly" in the sense that our methods terminate much faster than their theoretical worst-case running times suggest. Otherwise, we have to abort the solution process prematurely. In general, we cannot make any theoretical guarantees regarding the quality of a solution obtained this way. The same applies to the heuristic solutions covered in Chapter 7.

In this chapter, we focus on *approximation algorithms* that provide (theoretically) efficient running times as well as a provable solution quality.

Definition 8.1. For every instance I of an optimization problem, let $\textsc{Opt}(I)$ denote the optimum solution value and $\textsc{Alg}(I)$ the objective function value of the solution computed by an algorithm Alg. Then, Alg is a *γ-approximation algorithm* if it has polynomial running time and its solution is always within a factor of $\gamma \geq 1$ of the optimum solution, i.e., for all instances I,

$$\textsc{Alg}(I) \leq \gamma\, \textsc{Opt}(I) \text{ for minimization problems, and } \textsc{Alg}(I) \geq (1/\gamma)\, \textsc{Opt}(I) \text{ for maximization problems.}$$

In this sense, Theorem 7.9 gives us a solution guarantee for the greedy algorithm.

Example 8.2. The greedy algorithm is a q-approximation algorithm for independence systems $(E, \mathcal{I})$ of rank quotient $q = \max_{F \subseteq E} r(F)/\rho(F)$. △

In particular, we obtain the following approximation algorithm.

Theorem 8.3. The greedy algorithm is a 2-approximation algorithm for finding a bipartite matching of maximum weight.

Proof. Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. By Theorems 7.9 and 7.11, it suffices to show the existence of two matroids $(E, \mathcal{I}_1)$ and $(E, \mathcal{I}_2)$ such that $\mathcal{I} = \mathcal{I}_1 \cap \mathcal{I}_2$ is the set of matchings in G. Two sets with this property are defined by, for $\ell \in \{1, 2\}$,

$$\mathcal{I}_\ell := \{M \subseteq E : |M \cap \delta_G(v)| \leq 1 \; \forall v \in V_\ell\}.$$

Clearly, $\emptyset \in \mathcal{I}_\ell$ holds, $\mathcal{I}_\ell$ is closed under taking subsets, and $\mathcal{I} = \mathcal{I}_1 \cap \mathcal{I}_2$. Also, for all $F \subseteq E$, the bases of F in $(E, \mathcal{I}_\ell)$ have the same cardinality $|\{v \in V_\ell : F \cap \delta_G(v) \neq \emptyset\}|$. Hence, $(E, \mathcal{I}_\ell)$ is a matroid. □

8.1 Approximation for TSP

To prove lower bounds on the best possible approximation factor of an NP-hard problem, we obviously have to assume $P \neq \text{NP}$ since otherwise even optimal algorithms with polynomial running time would be possible.[1] Under this assumption we can show that TSP is not approximable in general.

Theorem 8.4. There is no γ-approximation algorithm for TSP for any $\gamma = \gamma(n) \geq 1$, provided that $P \neq \text{NP}$.

Proof. We consider the Hamiltonian cycle problem in which an undirected graph $G = (V, E)$ is given and we have to decide whether G contains a Hamiltonian cycle, i.e., a cycle that visits every vertex exactly once. This problem is one of the classical NP-complete problems (see *Algorithmic Discrete Mathematics*).

We construct a TSP instance on a complete graph $G' = (V, E')$ with edge weights $c\colon E' \to \mathbb{R}_{\geq 0}$, such that any γ-approximate solution of the TSP instance is sufficient to decide whether a Hamiltonian cycle exists in G. Since approximation algorithms have polynomial running time by definition, and since G' and c can be generated in polynomial time, this construction immediately implies the statement of the theorem.

To do this, we set the edge weights for $e \in E'$ to

$$c(e) := \begin{cases} 0, & \text{if } e \in E, \\ 1, & \text{otherwise.} \end{cases}$$

Obviously, a Hamiltonian cycle in G corresponds to a tour of weight 0 in G' and vice-versa. Moreover, every tour in G' that has edges from $E' \setminus E$ has weight at least 1. By definition, every γ-approximation algorithm with $\gamma \geq 1$ must produce a solution of weight 0 if one exists, and thus decides whether a Hamiltonian cycle exists. □

Corollary 8.5. There is no γ-approximation algorithm for TSP for any constant $\gamma \geq 1$, provided that $\text{P} \neq \text{NP}$, even if edge weights must be strictly positive.

Proof. We proceed as in the proof of Theorem 8.4, but set

$$c(e) := \begin{cases} \frac{1}{2\gamma n}, & \text{if } e \in E, \\ 1, & \text{otherwise.} \end{cases}$$

Then, a Hamiltonian cycle in G corresponds to a tour of weight $\frac{1}{2\gamma} < 1$ in G' and, conversely, every tour of weight less than 1 must Hamiltonian, since it cannot afford even a single edge of $E' \setminus E$. Deciding whether a Hamiltonian cycle exists in G thus amounts to deciding whether the lightest tour in G' has weight $\frac{1}{2\gamma}$ or at least 1. By definition, every γ-approximation algorithm for TSP can be used to distinguish these cases. □

We have thus shown that we have to accept very bad solutions if we only want to invest polynomial computing time to solve TSP. This result strongly relies on the fact that the edge weights can be arbitrary. Fortunately, TSP instances often have additional structure: Most of the time the edge weights correspond to distances (e.g., in a road network) and are therefore metric. In this case, we speak of *metric TSP*. In particular, the edge weights (or lengths) fulfil the triangle inequality $c(\{u, v\}) \leq c(\{u, w\}) + c(\{w, v\})$ for all $u, v, w \in V$. Note that metric TSP is still NP-hard (see for example [32, Theorem 21.2]).

[1] This statement assumes that the corresponding decision problem is in NP.

8.1.1 Greedy algorithm

We first consider the greedy algorithm that proceeds as follows. We start with a cycle on two vertices v_1, v_2, where $\{v_1, v_2\}$ is a cheapest edge of the graph (the cycle uses this edge twice). We extend the cycle step by step by choosing a cheapest edge e that connects a vertex v of our cycle with a vertex v' outside. To obtain a cycle again, we replace an edge $\{v, v''\}$ from our cycle with $\{v', v''\}$.

Algorithm: greedy algorithm for metric TSP

input: complete graph $G = (V, E)$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$
output: tour T

$\{v_1, v_2\} \leftarrow \arg\min_{e \in E} c(e)$
$T = (V_T, E_T) \leftarrow \{(v_1, v_2), (v_2, v_1)\}$
while $V_T \neq V$ **:**
  $(v, v') \leftarrow \arg\min\{c(\{u, u'\}) : (u, u') \in V_T \times (V \setminus V_T)\}$ (edge selection)
  take $v'' \in V_T$ with $(v, v'') \in T$
  $T \leftarrow T \setminus \{(v, v'')\} \cup \{(v, v'), (v', v'')\}$
return T

Theorem 8.6. The greedy algorithm is a 2-approximation algorithm for metric TSP.

Proof. A closer look reveals that the edge selection $\{v, v'\}$ in each iteration corresponds exactly to the edge selection of Prim's algorithm for finding a minimum spanning tree (see *Algorithmic Discrete Mathematics*). If F is the set of all edges selected during the course of the algorithm, then F is therefore a minimum spanning tree.

In each step the length of our cycle increases by

$$c(\{v, v'\}) + c(\{v', v''\}) - c(\{v, v''\}) \leq 2\, c(\{v, v'\}),$$

where we used the triangle inequality. The greedy algorithm thus ensures that $c(T) \leq 2\, c(F)$.

Now consider an optimum TSP tour $T^\star$. If we remove any edge from $T^\star$, we obtain a spanning tree F'. It follows that

$$c(T) \leq 2\, c(F) \leq 2\, c(F') \leq 2c(T^\star).$$ □

8.1.2 Algorithm of Christofides

Starting from a minimum spanning tree $F^\star$, we can also construct a 2-approximation more directly: We double every edge in $F^\star$ and obtain a *Eulerian (multi)graph*, i.e., a graph that has a cycle using every edge exactly once. It is easy to see (cf. *Algorithmic Discrete Mathematics*) that a graph is Eulerian if and only if it is connected and every vertex has even degree (start somewhere, move along unused edges until a cycle is closed; repeat and join obtained cycles). A Eulerian cycle in the doubled spanning tree is of course not yet a tour, since it uses vertices multiple times. However, the triangle inequality allows to eliminate redundant visits to vertices without increasing the length of the tour, i.e., we skip all but the first occurrence of each vertex.

To summarise, we started from a minimum spanning tree and extended it to a Eulerian graph while doubling its weight. We can improve this algorithm by finding the cheapest extension to a Eulerian graph. To this

end, we first observe that every graph has an even number of vertices of odd degrees, since the sum of all vertex degrees must be even (namely $2|E|$). We want to partition these vertices into pairs and join each pair via the corresponding direct edge to obtain an Eulerian graph. The problem of finding such a partition resulting in the minimum total weight of the corresponding edges is precisely the minimum weight perfect matching problem, which can be solved using the algorithm of Edmonds [15] in polynomial time (cf. lecture *Combinatorial Optimization*). The resulting algorithm is known as *Christofides' algorithm* [8].

Algorithm: Christofides' algorithm for metric TSP

input: complete graph $G = (V, E)$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$
output: tour T

$F^\star \leftarrow$ minimum spanning tree in G
$V' \leftarrow \{v \in V : v \text{ has odd degree in } F^\star\}$
$M^\star \leftarrow$ perfect matching of minimum weight in $G[V']$
$R \leftarrow$ Eulerian cycle in $(V, F^\star \cup M^\star)$
$T \leftarrow$ eliminate redundant vertices in R (replace $\{u,v\},\{v,w\}$ by $\{u,w\}$)
return T

Theorem 8.7 ([8])**.** Christofides' algorithm is a 3/2-approximation algorithm for metric TSP.

Proof. As before, we have $c(F^\star) \leq c(T^\star)$, where $F^\star$ denotes a minimum spanning tree and $T^\star$ an optimum TSP tour. Let V' be the set of all vertices of odd degrees in $F^\star$, and let $M^\star \subset E$ be a perfect matching of minimum weight in $G[V']$. Since, by the triangle inequality, the removal of redundant vertices does not increase the length of the tour, it suffices to show that $c(M^\star) \leq c(T^\star)/2$.

First, let T' be an optimum TSP tour in $G[V']$. We claim that $c(T') \leq c(T^\star)$. To see this, we start with $T^\star$ and gradually remove vertices that are not in V' by replacing consecutive edges with the direct connection. By the triangle inequality the cycle does not become longer, which establishes the claim.

Now, since $|T'|$ is even, T' can be divided into two perfect matchings in $G[V']$ which each have weight at least $c(M^\star)$, because $M^\star$ is optimal. It follows that $c(M^\star) \leq c(T')/2 \leq c(T^\star)/2$. □

The state of the art in approximation guarantees for metric TSP is summarized below.

Theorem 8.8 ([28])**.** There exists a randomized algorithm for metric TSP that outputs a tour whose expected weight is at most $(\frac{3}{2} - \varepsilon) \cdot c(T^\star)$, for some $\varepsilon > 10^{-36}$.

Theorem 8.9 ([29])**.** The metric TSP problem does not admit a γ-approximation algorithm with $\gamma < 123/122$, provided that $P \neq \text{NP}$.

Closing the gap between the best upper and lower bounds is an important open problem in the field of approximation algorithms.

8.2 Polynomial-time approximation schemes

An approximation algorithm guarantees solutions of fixed approximation quality. We can view the best-possible approximation factor as a measure of the difficulty of a (NP-hard) optimization problem. In that sense, the easiest problems allow *every* fixed approximation factor $\gamma > 1$.

Definition 8.10. A *polynomial-time approximation scheme (PTAS)* is a family $\{A_\varepsilon\}$ of algorithms, one for every $\varepsilon > 0$, such that A_ε is a $(1+\varepsilon)$-approximation algorithm.

A PTAS therefore yields, for every fixed $\varepsilon > 0$, an algorithm with polynomial running time. However, our definition allows an arbitrary dependence of the running time on ε, in particular the running time could increase exponentially with increasing $1/\varepsilon$. The following definition is more restrictive.

Definition 8.11. A *fully polynomial-time approximation scheme (FPTAS)* is a polynomial-time approximation scheme $\{A_\varepsilon\}$ for which the running time of A_ε can be bounded by a polynomial in $1/\varepsilon$ (as well as in the instance size).

We first show that we cannot hope for a better dependence on ε for NP-hard problems.

Theorem 8.12. If an optimization problem with integral, non-negative objective function has a PTAS $\{A_\varepsilon\}$, where A_ε has polynomial running time in $\log(1/\varepsilon)$, the problem also has a polynomial-time exact algorithm.

Proof. We show the statement for minimization problems (the proof for maximization problems is analogous). We construct a polynomial-time algorithm as follows. We first apply A_ε with $\varepsilon > 0$ arbitrarily (e.g., $\varepsilon = 1$) to the instance and obtain a solution of value $c_\varepsilon \geq c_{\text{OPT}}$. Since A_ε has polynomial running time in the input size, c_ε is at most simply exponential in the input size.

Set $\varepsilon' := \frac{1}{c_\varepsilon + 1}$. Since c_ε is at most simply exponential in the input size, $\log(1/\varepsilon') \in \mathcal{O}(\log c_\varepsilon)$ is polynomial in the the input size. We apply $A_{\varepsilon'}$ and return the computed solution of value $c_{\varepsilon'}$. The outlined algorithm has polynomial running time since the running time of $A_{\varepsilon'}$ is polynomial in $\log(1/\varepsilon')$.

It remains to show that $c_{\varepsilon'} = c_{\text{OPT}}$. Because c_{OPT} and $c_{\varepsilon'}$ are integers by assumption, this follows from

$$c_{\text{OPT}} \leq c_{\varepsilon'} \leq (1+\varepsilon')\, c_{\text{OPT}} = \left(1 + \tfrac{1}{c_\varepsilon + 1}\right) \cdot c_{\text{OPT}} < \left(1 + \tfrac{1}{c_{\text{OPT}}}\right) \cdot c_{\text{OPT}} \leq c_{\text{OPT}} + 1. \qquad \square$$

8.2.1 Example: knapsack problem

We now show that the knapsack problem can be approximated arbitrarily well, and that even an FPTAS exists. In addition to the pseudopolynomial algorithm from Section 5.2.1, this is an indication that the knapsack problem is an "easy" NP-hard problem.

We consider a knapsack problem of the form $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta,\ \boldsymbol{x} \in \{0,1\}^n\}$ with $\boldsymbol{a}, \boldsymbol{c} \in \mathbb{R}^n_{\geq 0}$, $\beta \in \mathbb{R}_{\geq 0}$, and $0 < a_i \leq \beta$ for all $i \in \{1, \ldots, n\}$. The idea of the FPTAS is simple: We round the entries of $\boldsymbol{c}$ to integer multiples of some $\mu \in \mathbb{R}$, and apply the dynamic program of Section 5.2.1 to the rounded instance. We choose μ in such a way that (a) our algorithm has a polynomial running time and (b) our solution differs by at most a factor of $(1+\varepsilon)$ from c_{OPT}.

Algorithm: FPTAS for the knapsack problem.

input: $\boldsymbol{a}, \boldsymbol{c} \in \mathbb{R}^n_{\geq 0}$; $\beta \in \mathbb{R}_{\geq 0}$, $\varepsilon > 0$
output: approximate solution to $\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta,\ x \in \{0,1\}^n\}$

$M \leftarrow \max_{i \in \{1,\ldots,n\}} c_i$, $\mu \leftarrow \varepsilon M / n$
$\boldsymbol{c'} \leftarrow \lfloor \boldsymbol{c}/\mu \rfloor$ (component-wise)
$\boldsymbol{x} \leftarrow \arg\max\{(\boldsymbol{c'})^\top \boldsymbol{x} : \boldsymbol{a}^\top \boldsymbol{x} \leq \beta, \boldsymbol{x} \in \{0,1\}^n\}$ (via DP of Section 5.2.1)
return $\boldsymbol{x}$

Theorem 8.13. There exists an FPTAS for the knapsack problem with running time $\mathcal{O}(n^3/\varepsilon)$.

Proof. We first consider the rounding error. Rounding the entries of $\boldsymbol{c}$ to the nearest integer multiples of μ influences the solution value by at most μn, so we need to ensure that $\mu n \leq \varepsilon c_{\text{OPT}}$. Because of $M \leq c_{\text{OPT}}$, it is sufficient to set $\mu n = \varepsilon M$, i.e. $\mu := \varepsilon M/n$.

The running time of the algorithm is determined by the running time of the dynamic program on the rounded instance. According to Theorem 5.16, the latter is $\mathcal{O}(nZ)$, where $Z := \sum_{i=1}^{n} \lfloor c_i/\mu \rfloor \leq n \lfloor M/\mu \rfloor \in \mathcal{O}(n^2/\varepsilon)$. Hence, the running time is polynomial in n and $1/\varepsilon$, as desired. □

8.3 LP Rounding

Many approximation algorithms are based on rounding a solution to some LP relaxation. The applicability of this approach depends on whether we can ensure the rounded solution to be feasible and whether we are able to estimate the quality of the rounded solution. Randomization often helps, especially for the latter aspect (cf. the proof of Theorem 4.15). We illustrate the general approach on an example from machine scheduling.

Scheduling problems are generally concerned with assigning jobs $j \in \{1, 2, \ldots, n\}$ to machines $i \in \{1, \ldots, m\}$, where assigning job j on machine i encurs a processing time of $p_{ij} \geq 0$. Many practical problems can be expressed in this way, and, accordingly, there is a large number of variants of this problem, which differ in the objective function, the processing times and other restrictions for valid solutions.

We consider a variant in which every job $j \in \{1, \ldots, n\}$ may be rejected (i.e., not processed at all) for a cost s_j. Machines can only process one job at a time, but jobs can be interrupted (*preemption*), transferred to other machines while being processed (*migration*), and even run simultaneously on several machines (*parallelization*). The objective is to reduce the sum of the completion time of the last job (the *makespan*) and the rejection costs of the unprocessed jobs.

We can formulate this problem as a MIP with the following variables

$$\begin{aligned} x_{ij} &= \text{proportion of job } j \text{ processed by machine } i \\ y_j &= \begin{cases} 1, & \text{if job } j \text{ is processed,} \\ 0, & \text{otherwise} \end{cases} \\ T &= \text{makespan.} \end{aligned}$$

The resulting MIP is

$$\begin{aligned} \min \quad & T + \textstyle\sum_{j=1}^{n} s_j (1 - y_j), \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{n} p_{ij} x_{ij} \leq T && \forall i \in \{1, \ldots, m\}, \\ & \textstyle\sum_{i=1}^{m} x_{ij} = y_j && \forall j \in \{1, \ldots, n\}, \\ & x_{ij} \geq 0 && \forall i \in \{1, \ldots, m\},\ j \in \{1, \ldots, n\}, \\ & \boldsymbol{y} \in \{0,1\}^n. \end{aligned}$$

It can be shown that the above scheduling problem is NP-hard, see [26]. We describe an algorithm that determines a feasible solution by rounding the LP relaxation.

Algorithm: DeterministicRounding(α)

input: threshold $\alpha \in (0,1)$
determine solution $(\boldsymbol{x}^\star, \boldsymbol{y}^\star, T^\star)$ of the LP relaxation
for $j \in \{1,\dots,n\}$**:**
  if $y_j^\star \le \alpha$**:**
    $\hat{y}_j \leftarrow 0$
    $\hat{x}_{ij} \leftarrow 0 \quad \forall i \in \{1,\dots,m\}$
  else $(y_j^\star > \alpha)$
    $\hat{y}_j \leftarrow 1$
    $\hat{x}_{ij} \leftarrow x_{ij}^\star / y_j^\star \quad \forall i \in \{1,\dots,m\}$
$\hat{T} \leftarrow \max_{i\in\{1,\dots,m\}} \sum_{j=1}^n p_{ij}\hat{x}_{ij}$
return $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{T})$

We establish an approximation guarantee for this rounding method.

Theorem 8.14. DeterministicRounding(1/2) is a 2-approximation algorithm.

Proof. The computed solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{T})$ is feasible since either $\hat{y}_j = 0 = \sum_{i=1}^m \hat{x}_{ij}$ or, by feasibility of $(\boldsymbol{x}^\star, \boldsymbol{y}^\star, T^\star)$, $\hat{y}_j = 1 = \sum_{i=1}^m x_{ij}^\star / y_j^\star = \sum_{i=1}^m \hat{x}_{ij}$.
We claim that, for all $i \in \{1,\dots,m\}$, $j \in \{1,\dots,n\}$, it holds that

$$(1-\hat{y}_j) \le \frac{1}{1-\alpha}(1-y_j^\star), \text{ and} \tag{8.1}$$

$$\hat{x}_{ij} \le \frac{1}{\alpha} x_{ij}^\star. \tag{8.2}$$

For $y_j^\star \le \alpha$, we have $\hat{y}_j = \hat{x}_{ij} = 0$ and thus

$$\frac{1}{1-\alpha}(1-y_j^\star) \ge \frac{1}{1-\alpha}(1-\alpha) = 1 = (1-\hat{y}_j),$$

and

$$\frac{1}{\alpha} x_{ij}^\star \ge 0 = \hat{x}_{ij}.$$

For $y_j^\star > \alpha$, we have$\hat{y}_j = 1$ and thus

$$\frac{1}{1-\alpha}(1-y_j^\star) \ge 0 = (1-\hat{y}_j),$$

and

$$\hat{x}_{ij} = \frac{x_{ij}^\star}{y_j^\star} < \frac{1}{\alpha} x_{ij}^\star.$$

Hence the claim holds.
Now, let $i \in \{1,\dots,m\}$ be a machine with completion time $\hat{T}$ in the solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{T})$, i.e., such that $\hat{T} = \sum_{j=1}^n p_{ij}\hat{x}_{ij}$. Then,

$$\frac{1}{\alpha} T^\star \ge \frac{1}{\alpha} \sum_{j=1}^n p_{ij} x_{ij}^\star \overset{(8.2)}{\ge} \sum_{j=1}^n p_{ij}\hat{x}_{ij} = \hat{T}. \tag{8.3}$$

It follows that

$$\hat{T} + \sum_{j=1}^{n} s_j(1-\hat{y}_j) \overset{(8.1),(8.3)}{\leq} \frac{1}{\alpha}\, T^\star + \frac{1}{1-\alpha} \sum_{j=1}^{n} s_j(1-y_j^\star)$$

$$\leq \max\Big\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\Big\}\Big(T^\star + \sum_{j=1}^{n} s_j(1-y_j^\star)\Big) \leq \max\Big\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\Big\} \cdot c_{\text{LP}} \leq \max\Big\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\Big\} \cdot c_{\text{MIP}}.$$

Setting $\alpha = \frac{1}{2}$ completes the proof. □

By randomizing the selection of α we can further improve the (expected) approximation factor. We can derandomize the resulting algorithm by testing a linear number of values for α.

Algorithm: DERANDOMIZEDROUNDING

$\mathcal{S} \leftarrow \{\text{DETERMINISTICROUNDING}(\alpha) : \alpha \in ((\frac{1}{e}, 1) \cap \{y_1^\star, \ldots, y_n^\star\}) \cup \{\frac{1}{\mathrm{e}}\}\}$
return $\arg\min\{\hat{T} + \sum_{i=1}^{m} s_j(1-\hat{y}_j) : (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{T}) \in \mathcal{S}\}$

We obtain the following solution guarantee.

Theorem 8.15 ([26])**.** The derandomized rounding algorithm is a $\frac{\mathrm{e}}{\mathrm{e}-1}$-approximation algorithm, where $\frac{\mathrm{e}}{\mathrm{e}-1} \approx 1.58$.

Proof. First, consider the randomized procedure where $\alpha \sim \mathcal{U}(\frac{1}{\mathrm{e}}, 1)$ is selected uniformly at random from the open interval $(\frac{1}{\mathrm{e}}, 1)$. Let $(\hat{\boldsymbol{x}}(\alpha), \hat{\boldsymbol{y}}(\alpha), \hat{T}(\alpha))$ denote the resulting solutions. Then, using the arguments from the proof of Theorem 8.14, we have

$$\mathbb{E}_\alpha[\hat{T}(\alpha)] = \frac{1}{1-\frac{1}{\mathrm{e}}} \int_{\frac{1}{\mathrm{e}}}^{1} \hat{T}(\alpha)\, \mathrm{d}\alpha \overset{(8.3)}{\leq} \frac{\mathrm{e}}{\mathrm{e}-1} \int_{\frac{1}{\mathrm{e}}}^{1} \frac{1}{\alpha} T^\star \mathrm{d}\alpha = \frac{\mathrm{e}}{\mathrm{e}-1} T^\star [\ln \alpha]_{\frac{1}{\mathrm{e}}}^{1} = \frac{\mathrm{e}}{\mathrm{e}-1} T^\star. \tag{8.4}$$

Furthermore, we have

$$\mathbb{E}_\alpha[\sum_{j=1}^{n} s_j(1-\hat{y}_j(\alpha))] = \sum_{j=1}^{n} s_j \Pr_\alpha[y_j^\star \leq \alpha] = \sum_{j=1}^{n} s_j \cdot \frac{1}{1-\frac{1}{\mathrm{e}}} \int_{\max\{\frac{1}{\mathrm{e}}, y_j^\star\}}^{1} \mathrm{d}\alpha$$

$$\leq \sum_{j=1}^{n} s_j \cdot \frac{\mathrm{e}}{\mathrm{e}-1} \int_{y_j^\star}^{1} \mathrm{d}\alpha = \frac{\mathrm{e}}{\mathrm{e}-1} \sum_{j=1}^{n} s_j(1-y_j^\star). \tag{8.5}$$

Hence,

$$\mathbb{E}_\alpha[\hat{T}(\alpha) + \sum_{j=1}^{n} s_j(1-\hat{y}_j(\alpha))] \overset{(8.4),(8.5)}{\leq} \frac{\mathrm{e}}{\mathrm{e}-1}(T^\star + \sum_{j=1}^{n} s_j(1-y_j^\star)) = \frac{\mathrm{e}}{\mathrm{e}-1} c_{\text{LP}} \leq \frac{\mathrm{e}}{\mathrm{e}-1} c_{\text{MIP}},$$

and thus we have proven an approximation factor of $\frac{\mathrm{e}}{\mathrm{e}-1}$ in expectation.
For derandomization, note that all possible solutions occur for $\alpha \in \{y_1^\star, \ldots, y_n^\star\}$ and thus it is sufficient to compute rounded solutions for $\alpha \in (\frac{1}{\mathrm{e}}, 1) \cap \{y_1^\star, \ldots, y_n^\star\}$ (and $\frac{1}{\mathrm{e}}$ if this set is empty). Furthermore, the best of these solutions must be at least as good as the expected value. □

On the other hand, it is known that the scheduling problem cannot be approximated arbitrarily well. Such problems are referred to as *APX-hard*.

Theorem 8.16 ([26])**.** There is no PTAS for the above scheduling problem, provided that P ≠ NP.

Bibliography

[1] Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Oper. Res. Lett.*, 33(1):42–54, 2005.

[2] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8:146–164, 1975.

[3] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324, 1993.

[4] J.F. Benders. Partitioning procedures for solving mixed variables programming. *Numerische Mathematik*, 4:238–252, 1962.

[5] C. Berge. Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung). In *Wissenschaftliche Zeitschrift*, Mathematisch-Naturwissenschaftliche Reihe. Martin Luther Universität Halle-Wittenberg, 1961.

[6] R. Borndörfer. *Aspects of Set Packing, Partitioning, and Covering*. PhD thesis, Technische Universität Berlin, 1998.

[7] Claus C. Carøe and RüDiger Schultz. Dual decomposition in stochastic integer programming. *Oper. Res. Lett.*, 24(1-2):37–45, 1999.

[8] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, CMU, 1976.

[9] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[10] V. Chvátal. On certain polytopes associated with graphs. *Journal on Combinatorial Theory B*, 18:305–337, 1975.

[11] Michele Conforti, Gérard Cornuéjols, and M. R. Rao. Decomposition of balanced matrices. *J. Comb. Theory, Ser. B*, 77(2):292–406, 1999.

[12] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[13] Guoli Ding, Li Feng, and Wenan Zang. The complexity of recognizing linear systems with certain integrality properties. *Mathematical Programming*, 114(2):321–334, 2008.

[14] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. In P.L. Hammer, editor, *Studies in Integer Programming*, volume 1, pages 185–204. North-Holland, Amsterdam, 1977.

[15] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[16] D. Fulkerson, A.J. Hoffman, and R. Oppenheim. On balanced matrices. *Mathematical Programming Study*, 1:120–132, 1974.

[17] D.R. Fulkerson. Packing rooted directed cuts in a weighted directed graph. *Mathematical Programming*, 6:1–13, 1974.

[18] A. Ghouila-Houri. CaractÃ©risation des matrices totalement unimodulaires. *Comptes Rendus Hebdomadaires des SÃ©ances de l'Academie des Sciences (Paris)*, 254:1192–1194, 1962.

[19] R.E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The RAND Cooperation, 1960.

[20] R.E. Gomory. An algorithm for integer solutions to linear programming. In R.L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302, New York, 1969. McGraw-Hill.

[21] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[22] P.L. Hammer, E.L. Johnson, and U.N. Peled. Facets of regular 0-1 polytopes. *Mathematical Programming*, 8:179–206, 1975.

[23] Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[24] Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1(1):6–25, 1971.

[25] A.J. Hoffman and J.B. Kruskal. Integral boundary points of convex polyhedra. In H. W. Kuhn and A. W. Tucker, editors, *Linear inequalities and related systems*, pages 223–246. Princeton University Press, Princeton, NJ, 1956.

[26] Han Hoogeveen, Martin Skutella, and Gerhard J. Woeginger. Preemptive scheduling with rejection. *Mathematical Programming*, 94:361–374, 2003.

[27] Tom A. Jenkyns. The efficacy of the "greedy" algorithm. *Proc. 7th southeast. Conf. Comb., Graph Theory, Comput.*, pages 341–350, 1976.

[28] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric tsp. *Operations Research*, 2023. To appear.

[29] Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. *Journal of Computer and System Sciences*, 81(8):1665–1677, 2015.

[30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[31] D Klabjan, G.L Nemhauser, and C Tovey. The complexity of cover inequality separation. *Operations Research Letters*, 23(1-2):35–40, 1998.

[32] B. Korte and J. Vygen. *Combinatorial Optimization. Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, Heidelberg, 5th edition, 2012.

[33] H. Marchand, A. Martin, R. Weismantel, and L.A. Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123/124:391–440, 2002.

[34] H. Marchand and L.A. Wolsey. The 0-1 knapsack problem with a single continuous variable. *Mathematical Programming*, 85:15–33, 1999.

[35] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

[36] M.W. Padberg. A note on zero-one programming. *Operations Research*, 23(4):833–837, 1975.

[37] M.W. Padberg. $(1,k)$-configurations and facets for packing problems. *Mathematical Programming*, 18:94–99, 1980.

[38] M.W. Padberg and G. Rinaldi. A branch and cut algorithm for the resolution of large–scale symmetric traveling salesman problems. *SIAM Review*, 33:60–100, 1991.

[39] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.

[40] Paul D. Seymour. Decomposition of regular matroids. *Journal of Combinatorial Theory*, 28:305–359, 1980.

[41] R. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[42] R. Weismantel. On the 0/1 knapsack polytope. *Mathematical Programming*, 77(1):49–68, 1997.

[43] L.A. Wolsey. Faces of linear inequalities in 0-1 variables. *Mathematical Programming*, 8:165–178, 1975.