DISS. ETH NO. 26683

Optimization of bimodular integer programs and feasibility for three-modular base block IPs

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by
STEPHAN ARTMANN
MSc ETH Mathematics, ETH Zurich

born on August 25, 1987
citizen of Austria

accepted on the recommendation of
Prof. Dr. Robert Weismantel
Prod. Dr. Friedrich Eisenbrand

2020

Acknowledgement

I want to thank Robert Weismantel for introducing me to this fascinating topic, providing me this exciting research question, investing as much of his time as he did for countless explanations and valuable suggestions whenever I got stuck, and for his overall commitment and enthusiasm for the project. I am grateful that he was always available whenever I needed help and allowed me to profit from his experience and creativity to tackle the problems I encountered. Without him, this thesis would not have been possible.

I also wish to thank Christoph Glanzer, who is working on a similar topic, for many hints and discussions that were both fun and had a great influence on this work.

Many thanks also to Rico Zenklusen, who put much work and effort into the bimodular case.

Finally, I am grateful to all the members of the IFOR institute for contributing towards an extremely productive and enjoyable workplace.

Abstract

We consider the optimization problem $\max\{c^\mathsf{T} x \colon Ax \leqslant b,\ x \in \mathbb{Z}^n\}$, called an integer linear optimization problem (ILP), where $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$. We study the special case that all $(n \times n)$−sub-determinants of A are, in absolute value, bounded by a constant Δ.

This thesis presents results for several special cases which indicate that such integer optimization problems are poly-time solvable.

Veselov and Chirkov [41] found an efficient algorithm for this problem in the bimodular case, that is, if $\Delta = 2$, under the additional assumption that all $(n \times n)$−sub-determinants are non-zero. We generalize this result by providing a strongly-polynomial algorithm for the binomial case. This is based on solving an odd-parity constrained TU problem.

Furthermore, we consider the strictly 3-modular case when all $(n \times n)$-sub-determinants are in $\{0, \pm 3\}$. We give an equivalent TU-description with one congruence constraint. We show, for important special cases, how to decide feasibility, and that the underlying TU-problem admits a flat direction of width 1 if it is infeasible.

Finally, we conclude by giving a Python implementation of the core ingredients of our bimodular optimization algorithm.

Zusammenfassung

Wir untersuchen das Optimierungsproblem $\max\{c^\mathsf{T}x\colon Ax \leqslant b,\ x \in \mathbb{Z}^n\}$, bekannt unter dem Namen 'integer linear optimization problem (ILP)', wobei $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$ und $c \in \mathbb{Z}^n$. Vor allem der Spezialfall in dem alle $(n \times n)-$sub-Determinanten von A im Absolutwert durch eine Konstante Δ beschränkt sind, ist für uns von Interesse.

In dieser Dissertation stellen wir Ergebnisse für diverse Spezialfälle vor, welche einen Hinweis geben, dass solche ILPs in polynomieller Laufzeit gelöst werden können.

Veselov und Chirkov [41] fanden einen effizienten Algorithmus für den bimodularen Fall, in dem $\Delta = 2$, unter der zusätzlichen Annahme, dass alle $(n \times n)$-sub-Determinanten nicht null sind. Wir verallgemeinern dieses Resultat mit einem Algorithmus mit strongly-polynomial Laufzeit für den bimodularen Fall. Dieser basiert auf der Reduktion des Problems auf ein ungeradzahliges TU-Problem.

Zusätzlich studieren wir den strikt 3-modularen Fall, in dem alle $(n \times n)$-sub-Determinanten in $\{0, \pm 3\}$ liegen. Auch hier existiert eine äquivalente TU-Beschreibung mit einer Kongruenz-Bedingung. Wir zeigen, für wichtige Spezialfälle, wie man $\{x \in \mathbb{R}^n\colon Ax \leqslant b,\ x \in \mathbb{Z}^n\} = \varnothing$ entscheiden kann, und dass das zugrundelegende TU-Problem eine flache Richtung von Weite 1 besitzt, wenn obige Menge leer ist.

Zum Ende der Arbeit wird eine Implementierung des Hauptbestandteils des Algorithmus für den bimodularen Fall in der Programmiersprache Python vorgestellt.

Contents

Chapter 1

Introduction

In this chapter, we will introduce integer linear problems and give some context on them. Parts of it have appeared in [6][1] and [5].

1.1 Motivation and overview

This thesis is concerned with integer linear optimization problems (ILP) with bounded sub-determinants. We postpone details and a formal introduction to Section 1.3, and mention only that integer linear problems can be stated as

$$\max\{c^\mathsf{T} x\colon Ax \leqslant b,\ x \in \mathbb{Z}^n\},$$

where $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$, and $c \in \mathbb{Z}^n$.

Such problems have been studied extensively (see, e.g.,[36]) and occur in many interesting discrete optimization problems, see [37] for an overview.

The wealth of problems that can be stated as ILPs comes at a cost, though: Solving ILPs is $\mathcal{NP}$-hard. This motivates the search for further assumptions which we can place on integer linear problems such that they can still be used on a wide variety of problems, yet be solved efficiently. A breakthrough in this direction was made by [32], who showed that the polyhedron's dimension serves as a parameter of the problem's complexity: When kept fixed, the problem can be solved efficiently.

Another parameter is the maximal minor, also known as maximal sub-determinant, of the constraint matrix. There are many hints in the literature that it might serve a similar purpose as the dimension. These include geometric observations: If, for example, a polyhedron contains a vertex, then the sub-determinants define a grid in $\mathbb{R}^n$ which contains it. Furthermore, there is a proximity result for integer-feasible polyhedra (cf. [36], Theorem 17.2, see Section 1.4.1 for details). It states that for every optimal point x in the polyhedron there is an optimal integer point y whose distance, with respect to the infinity-norm, to x is bounded by a function in the dimension and the maximal sub-determinant.

It is well-known that when all minors are bounded in absolute value by 1, for example, solving ILPs reduces to solving linear problems, which is due to a simple reduction to linear programming, exploiting that the LP relaxation that corresponds to the ILP is integral. There are numerous discrete optimization problems that are naturally described by an ILP with a TU constraint matrix. This includes problems like finding maximum flows, minimum cost flows, maximum bipartite matching, and interval packing and covering, just to name a few (we refer the interested reader to [38] for a much more extensive account).

In the bimodular case, that is, when all $(n \times n)$-sub-determinants of A are at most two in absolute value, [41] showed that full-dimensionality of the polytope implies feasibility, and that an optimal integer point can be found along an edge containing the optimal fractional vertex. They also argued how a bimodular integer linear problem can be solved in the special case that there are no sub-determinants of a certain kind which are zero. We postpone the details to Section 1.4.1.

Interestingly, bimodular ILPs already capture some classical combinatorial optimization problems, like finding a minimum odd s-t cut in a graph, which is an s-t cut containing an odd number of vertices. Also the problem of finding a minimum T-cut can be solved through bimodular integer programming, where the minimum T-cut problem is defined as follows. Given is an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Z}^E_{\geqslant 0}$, and $T \subseteq V$ with $|T|$ even. The goal is to find a set $S \subseteq V$ such that $|S \cap T|$ is odd, and the total weight of edges crossing the cut S is as small as possible. These reductions follow from the fact that bimodular ILPs allow for solving any ILP with a totally unimodular constraint matrix A and an additional parity constraint, which requires that $\sum_{i \in S} x_i$ is odd for some subset S of the components of x; we will come back to this observation later and expand on it. Another interesting problem naturally captured by bimodular ILPs regards the odd cycle packing number of an undirected graph G, which is the maximum number of vertex-disjoint odd cycles in G. The problem of finding a maximum weight independent set in a graph with an odd cycle packing number of 1 can be modeled as a bimodular ILP. This follows by an observation in [25], showing that the maximum subdeterminant of the vertex-edge incidence matrix of a graph G is equal to $2^{\mathrm{ocp}(G)}$, where $\mathrm{ocp}(G)$ is the odd cycle packing number of G. Prior to our work, it was open whether this problem can be solved efficiently, and a PTAS was known for finding the maximum weight independent set in G if G satisfies $\mathrm{ocp}(G) = O(\sqrt{\frac{\log n}{\log \log n}})$, where n is the number of vertices of G [11].

1.2 Contribution

In this thesis we add to the growing list of hints that sub-determinants may be a parameter that, when being fixed, allows for efficient algorithms for ILPs.

In Chapter 2, we describe an efficient algorithm for bimodular IPs which is based on two things: First, on the observations that a bimodular IP can equivalently be stated as a conic odd-parity constrained TU problem, and second, that the latter can recursively be solved using Seymours's TU decomposition [39].

Next, in Chapter 3, we prove that when a strictly 3-modular base block IP is infeasible, it has a flat direction of width 1. Furthermore, we give polynomial-time algorithms to test feasibility in certain special cases.

Finally, in Chapter 4, we conclude by providing a Python implementation of the main routine of the bimodular optimization algorithm.

1.3 Preliminaries

This section provides the background and concepts used throughout this thesis.

1.3.1 Notation

We will use the following notation throughout this thesis.

·) For an integer $n \geqslant m \geqslant 1$, denote by $m : n$, the set $\{m, \dots, n\}$, and by $[n]$ the set $1 : n$.

·) For a matrix $X \in \mathbb{Q}^{m\times n}$, and a set $S \subseteq \{1, \dots, m\}$, let $A_{S,\cdot} \in \mathbb{Q}^{|S|\times n}$ be the matrix consisting of the rows of A indexed by S. The same notation $x \in \mathbb{Q}^n$ is used for vectors, where $b_S \in \mathbb{Q}^{|S|}$ consists of the components of x indexed by S. Similarly, for a set $T \subseteq \{1, \dots, n\}$, let $A_{\cdot,T} \in \mathbb{Q}^{m\times|T|}$ consist of the rows of A indexed by T.

·) For a matrix $X \in \mathbb{Q}^{m\times n}$, the entry in row i and column j is denoted by X_{ij}. The ith entry of a vector $x \in \mathbb{Q}^n$ is denoted by x_i.

·) We compare vectors element-wise: We write, for two vectors x and y in $\mathbb{R}^n$,

$$x \leqslant y :\Leftrightarrow \forall i \in \{1, \dots, n\} \colon x_i \leqslant y_i.$$

A strict inequality $x < y$ is defined element-wise as well.

·) Let $\|x\|$ denote the 2-norm of a vector x, i.e., $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$.

·) $\mathbf{0}$ shall be the all-zero vector of appropriate dimension, and $e^i \in \mathbb{R}^n$, $1 \leqslant i \leqslant n$, the ith standard unit vector, i.e., $e^i_j = 0$, for all $1 \leqslant j \leqslant n$ with $j \neq i$, and $e^i_i = 1$.

·) Let X, Y be finite sets. If the rows and columns of a matrix A are indexed by the elements in X and Y, we write $A \in \mathbb{R}^{X\times Y}$, and $A_{x,y}$ for the entry in A that is indexed by $x \in X$ and $y \in Y$. We use similar notation for row and column selections and vectors.

·) Let $x \in \mathbb{R}^Z$, for some finite set Z, and $Y \subseteq Z$. We denote by $x(Y) := \sum_{y\in Y} x_y$.

Finally, we define the maximal sub-determinant:

·) $\Delta_{\max}(A) := \max\{|\delta| \colon \delta$ is the determinant of an $(n \times n)$−sub-matrix of $A\}$.

1.3.2 Complexity and polyhedral theory

To begin with, let us recapitulate some essentials of complexity theory and basic polyhedral theory. For a more extensive discussion, the reader is referred to [36], for example.

Let $\mathcal{F}$ be a family of functions from $\mathbb{R}^n$ to $\mathbb{R}$ and $\mathcal{X}$ a family of subsets of $\mathbb{R}^n$. An **optimization problem** is the task of finding

$$\begin{aligned} &\max\ f(x) \\ &\text{s. t. } x \in X \end{aligned}$$

for any given $f \in \mathcal{F}$ and $X \in \mathcal{X}$.

(f, X) is called an **instance** of the optimization problem, where f is often referred to as the **objective function** and X as the **set of feasible solutions**. We call a point x **feasible** if $x \in X$, and an x^* achieving the optimum, i.e., $x^* \in X$ and $f(x^*) = \max\{f(x) \colon x \in X\}$, is **optimal**, or an **optimal solution**. An instance is **bounded** if $\sup\{f(x) \colon x \in X\} < \infty$. It is **feasible** if it has a solution, or **infeasible** otherwise.

An algorithm **solves an optimization problem** if given $f \in \mathcal{F}$ and $X \in \mathcal{X}$, it returns an optimal point if such a point exists, and decides that there is no such point, otherwise. The encoding size of an instance (f, X) is the number of bits necessary to describe $f \in \mathcal{F}$ and $X \in \mathcal{X}$. We call an algorithm **efficient** if its running time is bounded by a polynomial in the encoding size of f and X.

Let p be a parameter of a problem (P) (for example the maximal $(n \times n)$−sub-determinant $\Delta_{\max}(A)$ in an integer linear program, as described in Section 1.4). Sometimes, (P) can be solved efficiently with p **fixed**, or equivalently, with p a **constant**. This means that (P) is solved by an algorithm with running time bounded by a polynomial in the other parameters of (P), and p is assumed to be a constant. For some input data $D \in \mathbb{Z}^{m \times n}$, $m, n \in \mathbb{N}_{>0}$, denote by $\langle D \rangle := \max_{1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n} |\log_2 D_{ij}|$ its **encoding size**, to which we will also refer to as input size.

It will suffice for our purposes to say that a problem (P) can be **reduced** to a problem (Q) if the existence of an efficient algorithm that solves (Q) implies the existence of an efficient algorithm to solve (P). This is a special case of the more general notion of a Cook-reduction (cf. Chapter 2.2.1. in [22]).

Let $x, y \in \mathbb{R}^n$ be two vectors. In what follows, we write $x \leqslant y$ if for all $i \in \{1, \ldots, n\} : x_i \leqslant y_i$. Analogously, the strict inequality $x < y$ shall be defined element-wise as well.

A **polyhedron** is a subset of $\mathbb{R}^n$ which can be written as $P = \{x \in \mathbb{R}^n \colon Ax \leqslant b\}$ for a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$. We call A a **constraint matrix** and each pair $(A_{i,\cdot}, b_i)$, for $i \in \{1, \ldots, m\}$, a constraint.

A polyhedron is a **rational polyhedron** if A and b can be chosen to be in $\mathbb{Z}^{m\times n}$ and Z^m, respectively. Note that $\mathbb{R}^n$ is a rational polyhedron itself and that polyhedra are closed and convex. A **polytope** is a bounded polyhedron.

For a polyhedron P, $\operatorname{conv}(P \cap \mathbb{Z}^n)$ is the **integer hull** of P.

An **extreme point** x of a convex set P cannot be written as a convex combination of two different points of this set. Put differently, if $x = \lambda x^1 + (1-\lambda)x^2$ for $0 < \lambda < 1$ and x^1, $x^2 \in P$, then $x = x^1 = x^2$.

For a polyhedron $P = \{x \in \mathbb{R}^n \colon Ax \leqslant b\}$ and a point $x \in P$, the **set of tight constraints** at x is the subset $I \subseteq \{1,\dots,m\}$ such that $\forall i \in I : A_{i,\cdot}x = b_i$. If $\operatorname{rank}(A_{I,\cdot}) = n$, x is called a **vertex** of P. This is the case if and only if x is an extreme point of P (see [9], Theorem 2.3). A polyhedron which has an extreme point is called **pointed**.

A polyhedron $C = \{x \in \mathbb{R}^n \colon Ax \leqslant \mathbf{0}\}$ is called a **polyhedral cone**. We will just refer to it as a cone in this thesis.

An element $r \neq \mathbf{0}$ of a cone C that satisfies $A_{I,\cdot}r = 0$, for $I \subseteq \{1,\dots,m\}$ such that $A_{I,\cdot}$ has rank $n-1$ is called an **extreme ray** of C. As any positive multiple of r is an extreme ray again, one usually scales r such that it is integral and primitive (we call an integral vector $x \in \mathbb{Z}^n$ **primitive** if $\gcd(|x_1|,\dots,|x_n|) = 1$).

Caratheodory's Theorem states that all elements of C can be written as a non-negative linear combination of at most n extreme rays:

Theorem 1.1 (Caratheodory [14]). Let $r^1,\dots,r^l$ be the extreme rays of a cone C. Then for all $x \in C$, there is $i_1,\dots,i_n$ such that

$$x = \sum_{j=1}^{n} \lambda_j r^{i_j},$$

where $\lambda_j \geqslant 0$ for all $j \in \{1,\dots,n\}$.

For a vertex v of a polyhedron P, the polyhedron $C := \{x \in \mathbb{R}^n \colon A_{I,\cdot}x \leqslant b_I\}$, where I is the set of tight constraints at v, is called the **supporting cone at** v. C is a translated cone, as it can be written as $C = v + \{x \in \mathbb{R}^n \colon A_{I,\cdot}x \leqslant \mathbf{0}\}$. An **extreme ray** of C is defined as an extreme ray of $\{x \in \mathbb{R}^n \colon A_{I,\cdot}x \leqslant \mathbf{0}\}$.

For a supporting cone C, take a subset $J \subseteq I$ such that $|J| = n$ and $\det(A_{J,\cdot}) \neq 0$. Then v is uniquely defined by the equation $A_{J,\cdot}v = b_J$, and can be calculated by Cramer's rule as

$$v_i = \frac{\det A^{(i)}_{J,\cdot}}{\det A_{J,\cdot}}, \tag{1.1}$$

where $A^{(i)}_{J,\cdot}$ is the matrix that results from replacing column i in $A_{J,\cdot}$ by b_J.

The **width** of a polyhedron P is defined as

$$w(P) := \min_{d \in \mathbb{Z}^n \setminus \{0\}} w(P,d), \text{ where} \tag{1.2}$$

$$w(P, d) := \max\{d^\mathsf{T} x \mid x \in P\} - \min\{d^\mathsf{T} x \mid x \in P\}.$$

We call a polyhedron flat if it has a small (which usually means bound by a constant) width, and in this case we call a vector d achieving the maximum in (1.2) a **flat direction**. The flatness theorem due to Khinchine [30] shows that integer infeasible polyhedra in dimension n have a direction such that the width along it is bounded by a function only of n, see [7] for a proof of $\mathcal{O}(n^{\frac{5}{2}})$.

1.3.3 Linear and integer linear problems

We summarize the most important concepts and definitions in linear and integer linear programming.

We consider the following type of problems, called **integer linear programming** (ILP) problems:

$$\begin{aligned} \max\ & c^\top x \\ \text{s.t. } & x \in P, \\ & x \in \mathbb{Z}^n, \end{aligned} \tag{1.3}$$

where $P = \{x \colon Ax \leqslant b\} \subseteq \mathbb{R}^n$, for $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, and $c \in \mathbb{Z}^n$, is a rational polyhedron. We will also refer to ILPs as integer programming problems or integer problems.

If we drop the integrality constraints in (1.3), we arrive at a **linear programming** (LP) problem:

$$\begin{aligned} \max\ & c^\top x \\ \text{s.t. } & x \in P. \end{aligned} \tag{1.4}$$

(1.4) is an **LP-relaxation** of (1.3). Analogously, we call an instance of (1.4) an LP-relaxation of an instance of (1.3).

Similarly to the integer case, a synonym that we use for linear programming problems is **linear problem**.

To be explicit, whenever we say that an algorithms **solves** a problem like (1.3) or (1.4), or related problems to be introduced later, we mean the following. The algorithm determines infeasibility if the problem is infeasible. It returns an optimal solution if the problem has a finite optimum. Finally, if the problem is unbounded, then a certificate of unboundedness is returned. In the case of (1.3) or (1.4) – and similarly for related problems – this consists of a feasible point x and a direction $v \in \mathbb{Z}^n$ such that

1. $Av \leqslant 0$, and
2. $c^\mathsf{T} v > 0$.

This clearly shows unboundedness since $x + \lambda v$ will be feasible for any $\lambda \in \mathbb{Z}_{\geqslant 0}$, and the objective value linearly increases with increasing lambda.

For linear problem (1.4) efficient algorithms exist, e.g. Khachiyan's method [29]. Lenstra showed in [32] that (1.3) can be solved efficiently when n is fixed.

The existence of polynomial-time algorithm for IP-problems seems unlikely, as they are $\mathcal{NP}-$hard [28].

1.4 The problem setting: ILPs with bounded sub-determinants

We now introduce our problem setting.

We call the determinant of any square sub-matrix of A a sub-determinant. The determinant of an $(n \times n)-$sub-matrix of A is an $(n \times n)-$sub-determinant of A. We introduce the following definitions to abbreviate notation.

Definition 1.2. Let $A \in Z^{m\times n}$, then

i) the determinant of a square sub-matrix of A as called a **sub-determinant**,

ii) the determinant of an $(n\times n)-$sub-matrix of A is called an $(n\times n)-$**sub-determinant of** A,

iii) $\Delta_{\max}(A) := \max\{|\delta| \colon \delta$ determinant of an $(n \times n)-$sub-matrix of $A\}$ and

iv) $\Delta_{\min}(A) := \min\{|\delta| \colon \delta$ determinant of an $(n \times n)-$sub-matrix of $A\}$.

Throughout this thesis, we are interested in the maximization problem of finding

$$\begin{aligned} \max\ & c^\top x \\ \text{s.t. } & x \in P, \\ & x \in \mathbb{Z}^n, \end{aligned} \tag{1.5}$$

where $P := \{x \in \mathbb{R}^n \colon Ax \leqslant b\}$ is a rational polyhedron, $A \in \mathbb{Z}^{m\times n}$, $b \in Z^m$ and $c \in \mathbb{Z}^n$. If $\Delta_{\max}(A) \leqslant 2$, we call (1.5) a **bimodular ILP**. If all $(n \times n)-$sub-determinants of A are zero or two in absolute value, we call the problem **strictly bimodular**. Accordingly, we talk about a **strictly k-modular** ILP if all such sub-determinants are 0 or $\pm k$, for some $k \in \mathbb{N}$.

Definition 1.3 (Bimodular integer linear problem)**.** Let $b \in \mathbb{Z}^m$, $c \in Z^n$, and $A \in \mathbb{Z}^{m\times n}$ with $\text{rank}(A) = n$ and $\Delta_{\max}(A) = 2$. Find $x \in \mathbb{R}^n$ attaining

$$\max\{c^\mathsf{T} x \colon Ax \leqslant b,\ x \in \mathbb{Z}^n\}, \tag{BIP}$$

or decide that no such x exists.

We remark that [42] contains a nice discussion of matrices with $(n \times n)$-sub-determinants within $\{-k, 0, k\}$ for $k \in \mathbb{Z}_{>0}$.

1.4.1 Related results and observations

There are many results that link polyhedral properties to sub-determinants.

We know, for example (cf. [36], Theorem 17.2) that if y^* achieves $\max\{c^\mathsf{T}x : x \in P\}$ and an optimal solution to (1.3) exists, then there is an optimal solution $x^* \in \mathbb{Z}^n$ for (1.3) such that $\|y^* - x^*\| \leqslant n\widetilde{\Delta_{\max}}(A)$, where $\widetilde{\Delta_{\max}}(A)$ is the maximal absolute value of any square sub-determinant of A.

In addition, Bonifas et al. studied the polyhedral graph of P in [12]. This is an undirected graph with the extreme points of P as vertices. It has an edge between two vertices if the corresponding extreme points are connected by an edge in P. Bonifas et al. [12] were interested in the diameter of P, that is the smallest integral bound on the length of a shortest path between any pair of vertices. They showed that it is bounded by $\mathcal{O}(\widetilde{\Delta_{\max}}(A)^2 n^4 \log_2(n\widetilde{\Delta_{\max}}(A))$.

There are also connections between integer feasibility and the width of a polyhedron. The flatness theorem due to Khinchine [30] shows that integer infeasible polyhedra in dimension n have a direction such that the width along it is bounded by a function only of n, see [7] for a proof of $\mathcal{O}(n^{\frac{5}{2}})$. Gribanov and Veselov showed that if the width is larger than a function f that only depends on Δ and n and is linear in n, then the polyhedron contains $n+1$ integer points [24, Theorem 4]. It was shown in [23] that if P is strictly 3-modular and $P \cap \mathbb{Z}^n = \varnothing$, then there exists $c \in \mathbb{Z}^n \backslash \{0\}$ such that $\max\{c^\mathsf{T}x \mid x \in P\} - \min\{c^\mathsf{T}x \mid x \in P\} \leqslant 2(n+1)$.

Optimization problems under bounded sub-determinants have been studied as well. Linear programs with constraint matrices whose subdeterminants are bounded by a constant $\Delta > 0$ can be solved in strongly polynomial time. This follows by a seminal result of Tardos [40], which shows the existence of an efficient linear programming algorithm whose runtime does not depend on the entries of the right-hand side or the objective function. Clearly, if all sub-determinants of the constraint matrix are bounded by Δ in absolute value, then so are all entries of the constraint matrix.

Interest also arose in obtaining simplex-type linear programming algorithms with similar guarantees as the one of Tardos. To this end, Bonifas et al. [12] showed an important structural result, namely that polyhedra defined by a constraint matrix that is totally Δ-modular have small diameter, i.e., the diameter is bounded by a polynomial in Δ and the number of variables. Dyer and Frieze [16] presented a strongly polynomial randomized simplex-type linear programming algorithm when the constraint matrix is totally unimodular. Very recently, Eisenbrand and Vempala [17] showed a randomized simplex-type linear programming algorithm, whose running time is strongly polynomial even if all subdeterminants of the constraint matrix are bounded by any constant. It remains open whether strongly polynomial linear programming algorithms exist without restrictions on the constraint matrix.

We further want to highlight prior work on integral binet matrices, which is a generalization of a subclass of TU matrices, known as network matrices. Integral binet matrices also lead to ILPs that can be solved efficiently by a reduction to a matching problem (see [1, 2]). The class of binet matrices is not directly comparable to bimodular matrices; more precisely, it neither contains the set of bimodular matrices nor is it contained in it.

In the totally unimodular (TU) case, when all sub-determinants of A lie in $\{-1, 0, 1\}$, problem (1.3) can be solved efficiently. The reason is that then, P has integral extreme points, and so (1.3) can be reduced to finding a point achieving

$$\max\{c^{\mathsf{T}}x \colon x \text{ lies on a minimal face of } P\},$$

or deciding that no such point exists. This can be done efficiently, see, for example, [9].

TU matrices have been studied extensively, as covered in [36], Sections 19 − 21, for example. We will treat the TU decomposition theorem of [39] in greater detail later as it is a key ingredient for our bimodular problem algorithm. Another result which we will use repeatedly in Chapter 3 extends the statement of Caratheodory's Theorem for TU matrices.

Theorem 1.4. Let $C := \{x \in \mathbb{R}^n : Tx \leqslant 0\}$, for $T \in \mathbb{Z}^{m \times n} TU$, be a totally unimodular cone, $r^1, \ldots, r^k$ be its extreme rays. Then, for every $z \in C \cap \mathbb{Z}^n$, there exist $\lambda_1, \ldots, \lambda_n \in \mathbb{N}$ such that

$$x = \sum_{i=1}^{n} \lambda_i r^{j_i},$$

where $j_1, \ldots, j_n \in \{1, \ldots, k\}$.

The proof follows from a similar statement in [20]. Alternatively, Theorem 1.4 can be shown using an inductive argument.

In [41], Veselov and Chirkov proved the existence of an efficient algorithm for bimodular ILPs if all $(n \times n)$-sub-determinants are non-zero. This has later been generalized in [4] to the case that all sub-determinants of A are nonzero and bounded by a constant.

Another result in [41] concerns the feasibility of (BIP) and is treated in greater detail here, as it is essential for our bimodular algorithm. It states that if $\Delta(A) \leqslant 2$ and P is a full-dimensional pointed polyhedron, then P contains an integer point.

The argument of Veselov and Chirkov goes along the following lines: First, assume w.l.o.g. that P has a fractional vertex v, as otherwise, it clearly contains an integer vertex, which can be found efficiently [29]. Let $I \subseteq \{1, \ldots, m\}$ be the set of tight constraints for v and consider the supporting cone $C = \{x \in \mathbb{R}^n \colon A_{I,\cdot}x \leqslant b_I\}$. Then, in [41] an extreme ray $r \in \mathbb{R}^n$ is written as

$$r = -B^{-1}e_1, \tag{1.6}$$

where B is some invertible $(n \times n)$−sub-matrix of $A_{I,\cdot}$ and e^1 is the first unit vector. Note that unlike it is often done, they do not scale r such that it is an integral primitive vector.

Their feasibility result then is:

Lemma 1.5 ([41], Theorem 2)**.** Let y be an extreme point of $\mathrm{conv}(C \cap \mathbb{Z}^n)$. Then, $y \in \mathrm{conv}(P \cap \mathbb{Z}^n)$ and there exists a fractional extreme ray r of C such that $y = v + r$.

For the purpose of proving Lemma 1.5, the following lemma is used in [41] implicitly. We shall give a proof for it here.

Lemma 1.6 (cmp. [41])**.** Let Δ be the maximal $(n \times n)$-subdeterminant of the constraint matrix A, v be a vertex and let r be an extreme ray of the supporting cone C at v as in (1.6), where $\delta := \det(B)$. Then,

i)

$$r_i = -\frac{1}{\delta}(-1)^{i+1}\det(B^{(i)}) \text{ for all } i \in \{1, \dots, n\}, \tag{1.7}$$

where $B^{(i)}$ is the sub-matrix of B obtained after removing the first row and column i.

ii) If $\delta = \Delta$, then for all $i \in \{1, \dots, m\}$

$$|A_{i,\cdot}r| \leqslant 1. \tag{1.8}$$

Proof.

i) We have

$$B^{-1}_{ij} = \frac{1}{\det B}(-1)^{i+j}\det(\bar{B}^{ji}),$$

where $\bar{B}^{ji}$ is the matrix B without row j and column i. Thus, for $j = 1$,

$$B^{-1}_{i1} = \frac{1}{\delta}(-1)^{i+1}\det(B^{(i)}).$$

ii)

$$|A_{i,\cdot}r| = \left|\sum_{j=1}^{n} A_{ij}r_j\right| = \left|\sum_{j=1}^{n} -\frac{A_{ij}}{\Delta}(-1)^{j+1}\det(B^{(j)})\right| = \frac{1}{\Delta}\underbrace{\left|\det\left(\begin{bmatrix} A_{i,\cdot} \\ \widetilde{B} \end{bmatrix}\right)\right|}_{\leqslant \Delta} \leqslant 1,$$

where $\widetilde{B}$ is the $(n-1 \times n)-$sub-matrix obtained from B after deleting the first row.

□

Proof of Lemma 1.5. Let $r = -B^{-1}e_1$ be an extreme ray of C (with B as in (1.6)) which satisfies

$$\forall i \text{ s. t. } A_{i,\cdot}y = b_i : A_{i,\cdot}r = 0. \tag{1.9}$$

Assume, for sake of contradiction, that $r \in \mathbb{Z}^n$. Then,

- ·) $\forall i \in I$ s. t. $A_{i,\cdot}y = b_i : A_{i,\cdot}r = 0$, and
- ·) $\forall i \in I$ s. t. $A_{i,\cdot}y < b_i : A_{i,\cdot}(y \pm r) \leqslant b_i$,

which implies that $y \pm r \in \operatorname{conv}(C \cap \mathbb{Z}^n)$, contradicting that y is a vertex of $\operatorname{conv}(C \cap \mathbb{Z}^n)$. Thus, $r \notin \mathbb{Z}^n$.

Consequently, $|\det(B)| = 2$. Consider the lattice $\mathcal{L} = \mathcal{L}(B^{-1})$. $\det(L) = \frac{1}{2}$ and thus, $\mathcal{L}$ consists of two cosets, $\mathbb{Z}^n$ and $v + Z^n$. As $r \in \mathcal{L}$ but $r \notin \mathbb{Z}^n$, $r \in v + \mathbb{Z}^n$, which in turn implies $v + r \in \mathbb{Z}^n$.

Additionally, $y = v + r$ as for $p := v + r \in \mathbb{Z}^n$ and $q := 2y - p \in \mathbb{Z}^n$,

- ·) $|A_{i,\cdot}r| \leqslant 1$ implies that $p \in C$,
- ·) $\forall i \in I : A_{i,\cdot}q = 2A_{i,\cdot}y - A_{i,\cdot}u - A_{i,\cdot}r \leqslant A_{i,\cdot}y - A_{i,\cdot}r \leqslant b_i$ and thus $q \in C$, and
- ·) $y = \frac{1}{2}p + \frac{1}{2}q$,

which by the fact that y is a vertex implies that $y = p = q$. Finally, by (1.8), (1.9) and the integrality of y: $\forall i \in \{1, \ldots, m\}$: $A_{i,\cdot}y \leqslant b_i$. □

This statement can be extended a little.

Lemma 1.7 (cmp. [41]). Let P be a full-dimensional, pointed polyhedron, $c \in \mathbb{Z}^n \backslash \{0\}$, $k \in \mathbb{N}$. Then $\exists x, y \in P \cap \mathbb{Z}^n$ s.t. $c^\mathsf{T} x \not\equiv c^\mathsf{T} y \pmod{k}$.

Proof. As P is full-dimensional, there exists $l \in \mathbb{N}$ such that $lP := \{lx \mid x \in P\}$ contains two integral points y, y' of different congruence in its interior. Let $i \in [n]$ such that $c_i \not\equiv 0 \pmod{k}$. Assume that there exists $\alpha \in \mathbb{Z}$ such that for all $x \in P \cap \mathbb{Z}^n$, $c^\mathsf{T} x \equiv \alpha \pmod{k}$. Similarly as in [8], T being `TU` implies that $y = \sum_{i=1}^k y_i$ for $y_1, \ldots, y_k \in P \cap \mathbb{Z}^n$ and $y' = \sum_{i=1}^k y_i'$ for $y_1', \ldots, y_k' \in P \cap \mathbb{Z}^n$. Thus, $\alpha \equiv c^\mathsf{T} y \pmod{k} \equiv l\alpha \pmod{k} \equiv c^\mathsf{T} y' \pmod{k} \not\equiv \alpha \pmod{k}$, a contradiction. □

Analogous to totally bimodular ILPs, one can, for any $\Delta > 0$, consider ILPs where the constraint matrix A is totally Δ-modular, i.e., all subdeterminants of A are at most Δ in absolute value. Clearly, for large enough Δ, ILPs with totally Δ-modular constraint matrices will become $\mathcal{NP}$-hard, as we approach the setting of general ILPs. More precisely, there are examples of $\mathcal{NP}$-hard ILPs with a constraint matrix that is a TU matrix with an additional $\{0, 1\}$-row (see, e.g.,[13, 15]). Such a constraint matrix can easily be seen to be totally n-modular. This implies that for any $\varepsilon > 0$, ILPs with totally n^ε-modular constraint matrices are $\mathcal{NP}$-hard.

1.5 Seymour's TU decomposition

We primarily follow Schrijver's [36] excellent exposition of Seymour's decomposition.

If a matrix is TU, this allows for employing powerful structural and algorithmic results about totally unimodular matrices. One of the most celebrated results in this context is a technique by Seymour [39] to recursively decompose TU matrices into simpler matrices, which are often called base blocks.

The smallest instances of TU matrices, which cannot be decomposed any further, fall into three categories: so-called network matrices, transposed network matrices, and two special types of constant-size matrices.

Definition 1.8 (Network matrix)**.** A matrix N is called a **network matrix**, if there exists a directed graph (V, A) and a directed tree (V, U) satisfying that the rows of N can be indexed by U and its columns by A such that the following holds. Let $a = (v, w) \in A$ and $u \in U$, and let P be the unique v-w path in U. Then

$$N_{u,a} = \begin{cases} +1 & \text{if } P \text{ passes through } u \text{ forwardly,} \\ -1 & \text{if } P \text{ passes through } u \text{ backwardly,} \\ 0 & \text{if } P \text{ does not pass through } u. \end{cases} \tag{1.10}$$

We highlight that the arcs U of the directed tree do not need to be a subset of A, and the directions of U can be arbitrary.

The decision problem of whether a TU matrix is a network matrix can be solved efficiently.

Lemma 1.9 (see, for example, [36, Chapter 20.1])**.** One can efficiently recognize whether a given matrix is a network matrix, and if this is the case, a directed graph and directed tree as in Definition 1.8 can be found in polynomial-time.

The remaining non-decomposable matrices are, up to row-/column permutations and sign changes,

$$\begin{bmatrix} 1 & -1 & 0 & 0 & -1 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{1.11}$$

We refer to (transposed) network matrices and the constant-size matrices (1.11) as the **base blocks** of Seymour's TU decomposition.

Three key operations of Seymour's decomposition are so-called k**-sums**, for $k \in [3]$, which are matrix operations defined as follows.

Definition 1.10 (k-sums)**.** Let $L \in \{-1,0,1\}^{m_L \times n_L}$ and let $R \in \{-1,0,1\}^{m_R \times n_R}$ be two matrices, and let $a \in \mathbb{Z}^{m_L}$, $d \in \mathbb{Z}^{n_R}$, $f \in \mathbb{Z}^{n_L}$, and $g \in \mathbb{Z}^{m_R}$. Then

- **1-sum**: $L \oplus_1 R = \begin{bmatrix} L & 0 \\ 0 & R \end{bmatrix}$,
- **2-sum**: $\begin{bmatrix} L & a \end{bmatrix} \oplus_2 \begin{bmatrix} d^\top \\ R \end{bmatrix} = \begin{bmatrix} L & ad^\top \\ 0 & R \end{bmatrix}$, and
- **3-sum**: $\begin{bmatrix} L & a & a \\ f^\top & 0 & 1 \end{bmatrix} \oplus_3 \begin{bmatrix} 1 & 0 & d^\top \\ g & g & R \end{bmatrix} = \begin{bmatrix} L & ad^\top \\ gf^\top & R \end{bmatrix}$.

Another useful operation employed in TU-decompositions is **pivoting**, which is defined as follows.

Definition 1.11 (Pivoting)**.** Let $T = \left[\begin{smallmatrix} \varepsilon & c^\top \\ b & D \end{smallmatrix}\right]$ be a TU matrix, where $\varepsilon \in \{-1,1\}$, $c \in \mathbb{Z}^n$, $b \in \mathbb{Z}^m$, and $D \in \mathbb{Z}^{m \times n}$. Then the matrix obtained from T by **pivoting** on the element in the first row and first column is

$$p_{1,1}(T) := \begin{bmatrix} -\varepsilon & \varepsilon c^\top \\ \varepsilon b & D - \varepsilon b c^\top \end{bmatrix}.$$

Analogously, pivoting is defined for an arbitrary entry $(i,j) \in [m] \times [n]$ of the matrix that satisfies $T_{i,j} \neq 0$. Formally, a pivot on (i,j) corresponds to first exchanging rows 1 and i and columns 1 and j, applying the pivot operation as defined above, and permuting the rows and columns back. We denote the resulting matrix by $p_{i,j}(T)$.

All of these operations preserve TU-ness:

Lemma 1.12 ([36], Section 19.4)**.** The k-sum, for $k \in [3]$, of two TU matrices is totally unimodular. Moreover, applying a pivot operation to a TU matrix results in a TU matrix.

Schrijver's exposition [36] of Seymour's decomposition is presented in the context of recognizing whether a $\{-1,0,1\}$-matrix is TU. In this context, further simplification is achieved by applying successively the following simple TU-preserving reductions whenever possible: deleting rows or columns with at most one nonzero entry that is either 1 or -1, and deleting a row or column that either appears twice or whose negation is also contained in the matrix to be checked for TU-ness. Unfortunately, these reductions will need some further consideration in our context: When T is the constraint matrix of an IP or ILP, these operations correspond to deleting variables and constraints, respectively. To provide a clean way to go between a matrix after and before such reductions, we define the following notion of a **core** of a TU-matrix, as a matrix obtained after these reductions.

Definition 1.13 (Core of a TU-matrix)**.** Let $T \in \mathbb{Z}^{m \times n}$ be totally unimodular. We call a submatrix of T a **core** of T if it arises from T by iteratively deleting

1. any row or column with at most one non-zero entry,
2. any row or column appearing twice or whose negation is also in the matrix.

Observe that a core of a matrix is unique up to row and column permutations and sign changes of rows and columns.

In the rest of this thesis, it makes no difference which core is chosen in case there are multiple options. Therefore, we denote by $\mathrm{core}(T)$ any one such core and, with a slight abuse of terminology, we will also refer to **the core** of a TU matrix.

The following theorem can be seen as the backbone of Seymour's TU decomposition. It states that the core of any TU matrix is either a base block, or its rows and columns can be permuted to obtain a matrix with a particular block structure (case 3) that, as shown by Seymour, can be further decomposed by a k-sum for $k \in [3]$). Network matrices, and the constant-size matrices that are referred to in point 2, are also called **base blocks** of Seymour's decomposition.

Theorem 1.14 ([39], Schrijver [36] Corollary 19.6b and Theorem 20.2)**.** Let T be totally unimodular. Then, one of the following cases holds for $\mathrm{core}(T)$:

1. $\mathrm{core}(T)$ or $\mathrm{core}(T)^\mathsf{T}$ is a network matrix.
2. $\mathrm{core}(T)$ is, possibly after row and column permutations and multiplication of some rows and columns by -1, one of the two matrices in (1.11).
3. $\mathrm{core}(T)$ is, possibly after row and column permutations, of the form $\left[\begin{smallmatrix} L & D_1 \\ D_2 & R \end{smallmatrix}\right]$, where $L \in \mathbb{Z}^{m_L \times n_L}$, $R \in \mathbb{Z}^{m_R \times n_R}$, $m_L + n_L \geqslant 4$, $m_R + n_R \geqslant 4$, and $\mathrm{rank}(D_1) + \mathrm{rank}(D_2) \leqslant 2$.

Moreover, there is a (strongly) polynomial algorithm to check which of the above cases is true.

It is not hard to observe that $\mathrm{core}(T)$ is a network matrix if and only if T is a network matrix (see, e..g, [36]). Hence, this first condition could equivalently be stated in terms of T instead of $\mathrm{core}(T)$. Still, we sometimes refer to the core in such situations for clarity, to obtain statements only depending on the core of T.

The third case can be distinguished even further:

Lemma 1.15 (Seymour [39], Schrijver [36], Proof of Theorem 20.2)**.** Assume that we are in case 3 of Theorem 1.14. Then, we can further distinguish the following cases, where a, d, f and g are column vectors of appropriate size with entries in $\{0, \pm 1\}$:

1. $\operatorname{rank}(D_1) = \operatorname{rank}(D_2) = 0$: Then, $\operatorname{core}(T) = L \oplus_1 R$ and $m_L + n_L$, $m_R + n_R \geqslant 4$.
2. $\operatorname{rank}(D_1) = 1$, $\operatorname{rank}(D_2) = 0$ and there are no row/column sign changes and permutations after which $\operatorname{core}(T)$ can be written as a 1-sum: Then, $D_1 = ab^\mathsf{T}$, $D_2 = 0$ and $\operatorname{core}(T) = \begin{bmatrix} L & a \end{bmatrix} \oplus_2 \begin{bmatrix} d^\mathsf{T} \\ R \end{bmatrix}$ is the 2-sum of two TU-matrices, for which $m_L + n_L$, $m_R + n_R \geqslant 4$, that we can find efficiently.
3. $\operatorname{rank}(D_1) = \operatorname{rank}(D_2) = 1$ and there are no row/column sign changes and permutations after which $\operatorname{core}(T)$ can be written as a 1- or 2-sum: Then, $D_1 = ad^\mathsf{T}$, $D_2 = gf^\mathsf{T}$ and $\operatorname{core}(T) = \begin{bmatrix} L & a & a \\ f^\mathsf{T} & 0 & 1 \end{bmatrix} \oplus_3 \begin{bmatrix} 1 & 0 & d^\mathsf{T} \\ g & g & R \end{bmatrix}$ is the 3-sum of two TU-matrices that fulfill $m_L + n_L$, $m_R + n_R \geqslant 4$ and that we can find efficiently.
4. $\operatorname{rank}(D_1) = 2$, $\operatorname{rank}(D_2) = 0$ and there are no row/column sign changes and permutations after which $\operatorname{core}(T)$ can be written as a 1-, 2 or 3-sum: Then, with a single pivot operation, row and column sign changes and row and column permutations, the resulting matrix can be written as a 3-sum as in case 3.

Furthermore, there is a polynomial time algorithm to check which case we are in, and find the corresponding k-sum decomposition.

Chapter 2

A strongly polynomial-time algorithm for (BIP)

Generalizing the statement of Veselov and Chirkov [41], we answer the main open question regarding (BIP), namely whether they can be solved efficiently, in the affermative. This chapter appeared in [6][1], and its main result is the following theorem.

Theorem 2.1 (cmp. [6])**.** There is an algorithm solving (BIP) in strongly polynomial time.

2.1 Introduction

We provide an outline of our approach in Section 2.2, and present further details in Section 2.3, 2.4, and 2.5, which cover three main steps of our approach, namely the reduction of (BIP) to a better-structured problem, the recursive decomposition of this well-structured problem into base block problems, and the efficient resolution of these, respectively. Finally, Section 2.6 presents a formal proof of how the different ingredients imply our main result.

2.2 Outline of our Approach

Our approach to solve (BIP) crucially exploits the elegant result of Veselov and Chirkov [41] mentioned in the introduction, exhibiting a close relation between (BIP) and its linear programming relaxation. In particular, the result essentially shows that it suffices to only consider constraints of (BIP) that are tight at an optimal vertex solution of the corresponding LP relaxation. Recall that for a matrix $A \in \mathbb{R}^{m \times n}$ and a row index $i \in [m] := \{1, \dots, m\}$, we denote by $A_{i,\cdot}$ the i-th row of A.

Theorem 2.2 ([41])**.** Consider a (BIP) problem, and assume that its natural LP relaxation $\max\{c^\mathsf{T}x : Ax \leqslant b, x \in \mathbb{R}^n\}$ is feasible and bounded. Let v be an optimal vertex solution to the LP relaxation of (BIP).

1. If (BIP) is feasible, then there is an optimal solution y for (BIP) such that $|A_{i,\cdot}(y-v)| \leqslant 1 \;\; \forall i \in [m]$.
2. Moreover, if v is the unique optimal solution to the LP-relaxation of (BIP), then the following holds. Let A' be the submatrix of A only consisting of the rows that are tight with respect to v. Then, if y is an optimal solution to $\max\{c^\mathsf{T}x : A'x \leqslant 0, v + x \in \mathbb{Z}^n\}$, then $y + v$ is optimal for (BIP).

Both of the above statements follow from Lemma 1.5 and its proof. More precisely, point 2 is a rephrasing of Lemma 1.5, and point 1 is shown in its proof.

In a first step of our approach, we show that Theorem 2.2 can be exploited to reduce (BIP) to an ILP over a TU constraint matrix with an additional parity constraint. For this we define two auxiliary problems that are defined by TU matrices and, as we will show next, are equivalent to (BIP). Recall that we use for any vector $x \in \mathbb{R}^n$ and index set $S \subseteq [n]$ the shorthand $x(S) := \sum_{i\in S} x_i$.

Parity TU-optimization: Given $T \in \mathbb{Z}^{m\times n}$ totally unimodular with $\operatorname{rank}(T) = n$, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$, $\alpha \in \{0,1\}$, and $S \subseteq [n]$, solve

$$\max\{c^\mathsf{T}x\colon Tx \leqslant b,\; x \in \mathbb{Z}^n_{\geqslant 0},\; x(S) \equiv \alpha \pmod 2\}. \tag{PTU}$$

Conic parity TU-optimization: Given $T \in \mathbb{Z}^{m\times n}$ totally unimodular with $\operatorname{rank}(T) = n$, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$, and $S \subseteq [n]$, solve

$$\max\{c^\mathsf{T}x\colon Tx \leqslant 0,\; x \in \mathbb{Z}^n_{\geqslant 0},\; x(S) \text{ odd}\}. \tag{CPTU}$$

The following lemma shows the reducibility of the three problem (BIP), (PTU), and (CPTU) to each other.

Lemma 2.3. Given an algorithm $\mathcal{A}$ for one of (BIP), (PTU) or (CPTU), one can solve any of the other two problems using

1. operations taking strongly polynomial time, and
2. a single call to $\mathcal{A}$.

Furthermore, when solving (PTU) with an algorithm $\mathcal{A}$ for (CPTU), the call to $\mathcal{A}$ is on a (CPTU) problem whose constraint matrix T is a submatrix of the one of the given (PTU) problem.

The TU structure we obtain by moving from (BIP) to (PTU) or (CPTU) allows for employing powerful structural and algorithmic results about totally unimodular matrices. One of the most celebrated results in this context, which is also a key element of our procedure, is a technique by Seymour [39] to recursively decompose TU matrices into simpler matrices, which are often called **base blocks**.

Our main focus is on solving (CPTU). A key technical step of our approach to solve (CPTU) is that we will be able to show that Seymour's TU-decomposition can be used to reduce a (CPTU) problem to a constant number of smaller instances of (PTU) and (CPTU) problems. Moreover, by Lemma 2.3, the (PTU) problems we obtain can again be mapped to (CPTU) problems of no larger size, which will leave us with a constant number of strictly smaller (CPTU) problems. Recursively continuing this approach, we will end up with (CPTU) problems whose constraint matrices are base blocks of Seymour's decomposition. We complete our approach by providing strongly polynomial algorithms to solve a (CPTU) problem whose constraint matrix corresponds to a base block.

In the following we will fill in the details of this high-level approach. To begin with, there are several technical hurdles to transform Theorem 1.14 and Lemma 1.15 into a form that is useful to decompose (CPTU) problems. First, as already mentioned, the above theorem is stated in terms of cores instead of the full matrix. Moreover, to make sure that our decomposition approach decomposes a (CPTU) problem into strictly smaller subproblems, we need that, after a k-sum, the matrices L and R of the subproblems are sufficiently small. It turns out that the conditions $m_L + n_L \geqslant 4$ and $m_R + n_R \geqslant 4$ as guaranteed by Theorem 1.14 are not sufficient for us. However, we show that the decomposition can be adjusted such that $m_L, m_R \geqslant 2$, which is all we need for our procedure to make progress and run efficiently.

We show that all these problems can be addressed. In particular, we obtain the following refinement of Theorem 1.14. Apart form addressing the issue with the core and the sizes of the submatrices L and R after a k-sum decomposition, the points 3–6 of Theorem 2.4 are an expansion of point 3 of Theorem 1.14, where we explicitly state how the matrix can be decomposed as a k-sum.

Theorem 2.4. Let T be totally unimodular. Then, one of the following cases holds:

1. $\mathrm{core}(T)$ or $\mathrm{core}(T)^\mathsf{T}$ is a network matrix.
2. $\mathrm{core}(T)$ is, possibly after row and column permutations and multiplication of some rows and columns by -1, one of the following two matrices:

$$\begin{bmatrix} 1 & -1 & 0 & 0 & -1 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{2.1}$$

3. T can, possibly after row and column permutations, be decomposed into a 1-sum with $m_L, m_R \geqslant 2$.
4. There are no row and column permutations such that T can be written as a 1-sum, and T can after permuting rows and columns be decomposed into a 2-sum with $m_L, m_R \geqslant 2$.
5. There are no row and column permutations such that T can be written as a 1- or 2-sum, and T can after permuting rows and columns be decomposed into a 3-sum with $m_L, m_R \geqslant 2$.
6. There are no row and column permutations such that T can be written as a 1-, 2- or 3-sum, and T can after pivoting once and row and column permutations be written as a 3-sum where $m_L, m_R \geqslant 2$.

Furthermore, we can efficiently decide in which case we are and, if we are in one of the cases 3-6, find a corresponding k-sum decomposition efficiently.

The proof of Theorem 2.4 is deferred to the appendix, Section B.2.

Starting with a (CPTU) problem with constraint matrix T, we first check which of the cases of Theorem 2.4 applies. For cases 1 and 2 we explicitly present strongly polynomial time algorithms to solve the (CPTU) problem. This is summarized in the following lemma, which is shown in Section 2.5.

Lemma 2.5. There exists a strongly polynomial algorithm for solving (CPTU) if the core of the constraint matrix $T \in \mathbb{Z}^{m\times n}$ is a base block, i.e., the core is either a network matrix, its transpose, or—up to sign changes of rows/columns and row/column permutations—one of the two matrices shown in (2.1).

The most involved case among (CPTU) problems on a constraint matrix T that is base block, is when T is the transpose of a network matrix. As we show, this problem can be reduced to a submodular function minimization problem with a parity constraint, which is one of the few constraint classes over which it is known how to efficiently minimize submodular functions (see [21]).

In the remaining cases, we show that the decomposition of T in terms of a k-sum for $k \in [3]$ can be used to reduce (CPTU) to several strictly smaller versions of the same problem, as stated in the following theorem, which is proven in Section 2.4.

Theorem 2.6. Consider a (CPTU) problem with constraint matrix T such that neither core(T) nor its transpose is a network matrix, and core(T) is not, up to row/column permutations and sign changes, one of the constant matrices in (2.1). Then we can solve (CPTU) by

1. solving at most 14 problem of type (CPTU) with at most m_1 rows and at most $n-1$ columns,
2. solving 1 problem of type (CPTU) with at most m_2 rows and at most

$n + 13$ columns, and

3. using further operations taking strongly polynomial time,

where m_1, m_2 satisfy $m_1 \leqslant m_2 < m$, and $m_1 + m_2 \leqslant m + 2$.

Our main result, Theorem 2.1, is now obtained by combining the above results. In particular, by Lemma 2.3, it suffices to show that (CPTU) can be solved in strongly polynomial time to show that (BIP) can be solved in strongly polynomial time. We then use Theorem 2.6 to break a (CPTU) problem into smaller ones, until we end up with (CPTU) problems on constraint matrices whose cores are either network matrices, their transposes, or are, up to row and column permutations and sign changes of rows and columns, one of the two (5×5) matrices shown in (2.1). All these resulting (CPTU) problems, which we call **base block problems**, in analogy to the base blocks in Seymour's decomposition, can be solved efficiently by Lemma 2.5. In Section 2.6, we present a formal proof of Theorem 2.1, showing that the approach as sketched above indeed leads to a strongly polynomial time algorithm for (BIP).

2.3 Reductions between (BIP), (PTU), and (CPTU)

The primary goal of this section is to prove Lemma 2.3. However, some of the results we introduce here will also be useful later on. We will start with reducing (BIP) to (CPTU). A key ingredient in this reduction will be Theorem 2.2, which requires an optimal vertex solution of the LP relaxation of (BIP). The following lemma shows that such an optimal LP vertex solution can be obtained in strongly polynomial time. It is obtained by applying a linear transformation, that transforms the linear program into one with bounded entries, which can be solved in strongly polynomial time by a result of Tardos [40]. The proof is deferred to the Appendix, Section B.1.

Lemma 2.7. The LP relaxation $\max\{c^{\mathsf{T}}x : Ax \leqslant b, x \in \mathbb{R}^n\}$ of (BIP) can be solved in strongly polynomial time. More precisely:

1. If the LP relaxation has a finite optimum, an optimal vertex solution can be determined in strongly polynomial time.
2. If the LP relaxation is unbounded, one can determine in strongly polynomial time a feasible vertex $v \in \mathbb{Q}^n$ and an **improving ray** $r \in \mathbb{Z}^n$, i.e., v satisfies $Av \leqslant b$, and r satisfies $Ar \leqslant 0$ and $c^{\mathsf{T}}r > 0$.

A technicality we have to deal with, is that the linear relaxation of (BIP) may be unbounded, in which case (BIP) is either infeasible or unbounded. This allows for an easy reduction of such a (BIP) instance to another one that is bounded.

Moreover, because we want to apply point 2 of Theorem 2.2, we also need to take care of (BIP) problems with a corresponding LP relaxation that has multiple optimal solutions. The following lemma, using mostly standard techniques, shows that these cases can be reduced to another (BIP) problem with a different objective function and a unique optimal solution.

Lemma 2.8. Consider a (BIP) problem whose LP relaxation is either unbounded or has multiple optimal solutions. Then we can reduce this problem in strongly polynomial time to a (BIP) problem with the same constraint matrix A and right-hand side b, whose LP relaxation has a unique optimum.

The proof is deferred to the appendix, Section B.1. Note, though, that a (not necessarily strongly poly-time) way to check for feasibility is to use a polynomial time procedure presented in [41] to check whether a (BIP) problem is feasible. However, this procedure, as presented, is not strongly polynomial. Moreover, it simplifies the presentation of our algorithm when we can always reduce to the bounded case.

To reduce (BIP) to (CPTU) we will perform a transformation of variables, through column operations, to obtain a TU constraint matrix. The following basic property of the inverse of an integer matrix Q with $|\det(Q)| = 2$ is crucial in our reduction that comes next. It states that the half-integral values of Q^{-1} form a rectangle.

Lemma 2.9. Let $Q \in \mathbb{Z}^{n\times n}$ be a matrix with $|\det(Q)| = 2$. Then, its inverse Q^{-1} has the following structure. There exist row indices $I \subseteq [n]$ and column indices $J \subseteq [n]$ with $I, J \neq \emptyset$, such that

- $(Q^{-1})_{ij} \in \frac{1}{2} + \mathbb{Z} \qquad \forall (i,j) \in I \times J$, and
- $(Q^{-1})_{ij} \in \mathbb{Z} \qquad \forall (i,j) \notin I \times J$.

The proof can again be found in Section B.1.

We are now ready to prove the reduction from (BIP) to (CPTU), which is one of the statements implied by Lemma 2.3.

Lemma 2.10. Given an algorithm $\mathcal{A}$ for (CPTU), one can solve any (BIP) problem using operations taking strongly polynomial time and a single call to $\mathcal{A}$.

Proof. Using Lemma 2.7, we solve the LP relaxation of (BIP) in strongly polynomial time. If it is infeasible, then so is (BIP) and we are done. Otherwise if the LP relaxation of (BIP) is either unbounded or has multiple optimal solutions, then we can invoke Lemma 2.8 to transform the problem to another one with a bounded LP relaxation. Hence, we are left with the case of a (BIP) problem whose LP relaxation has a unique optimum. In this case, let v be an optimal vertex solution of the LP relaxation $\max\{c^\mathsf{T}x : Ax \leqslant b, x \in \mathbb{R}^n\}$. If $v \in \mathbb{Z}^n$, then v is an optimal solution to (BIP) and we are done. Hence, assume $v \notin \mathbb{Z}^n$.

Let $C \in \mathbb{Z}^{m'\times n}$ be the submatrix of A only consisting of the rows that are tight with respect to v. By Theorem 2.2 we have that for any optimal solution y to

$$\max\{c^\mathsf{T}x : Cx \leqslant 0, v + x \in \mathbb{Z}^n\} \ , \tag{2.2}$$

the vector $y + v$ is an optimal solution to (BIP). Clearly, if (2.2) is infeasible, then so is (BIP). Hence, the (BIP) problem is reduced to solving (2.2).

Let Q be a full-rank square submatrix of C, and let b_Q be the part of b that corresponds to the rows in Q. Hence,

$$v = Q^{-1} b_Q \ . \tag{2.3}$$

Because all $(n \times n)$-subdeterminants of A are bounded in absolute value by 2, we have $\det Q \in \{-2, -1, 0, 1, 2\}$. Moreover $v \notin \mathbb{Z}^n$ implies that $|\det Q| = 2$; for otherwise $Q^{-1} \in \mathbb{Z}^{n \times n}$, and (2.3) would thus imply integrality of v, which does not hold. Since Q is an arbitrary $n \times n$ full-rank matrix of C we have the following:

$$\text{Each } (n \times n)\text{-subdeterminant of } C \text{ is either } -2,\ 0, \text{ or } 2. \tag{2.4}$$

Consequently, the matrix $\bar{C} := CQ^{-1}$ is a TU matrix (we remark that this also follows from results in [42]), because:

1. by $|\det(Q^{-1})| = \frac{1}{|\det(Q)|} = \frac{1}{2}$ and statement (2.4), the determinant of any $n \times n$ submatrix of $\bar{C}$ is within $\{-1, 0, 1\}$, and
2. $\bar{C} = CQ^{-1}$ contains an identity matrix because Q is a submatrix of C.

We can thus transform the problem (2.2) into one with a TU matrix as follows, where we set $\bar{c} := (Q^{-1})^{\mathsf{T}} c$, and use $z := Qx$:

$$\begin{aligned}
&\max\{c^{\mathsf{T}} x : Cx \leqslant 0, v + x \in \mathbb{Z}^n\} \\
&\qquad = \max\{c^{\mathsf{T}} Q^{-1} Q x : CQ^{-1} Q x \leqslant 0, v + Q^{-1} Q x \in \mathbb{Z}^n\} \\
&\qquad = \max\{\bar{c}^{\mathsf{T}} z : \bar{C} z \leqslant 0, Q^{-1}(b_Q + z) \in \mathbb{Z}^n\} \\
&\qquad = \max\{\bar{c}^{\mathsf{T}} z : \bar{C} z \leqslant 0, Q^{-1}(b_Q + z) \in \mathbb{Z}^n, z \in \mathbb{Z}^n\} \ .
\end{aligned} \tag{2.5}$$

Notice that the condition $z \in \mathbb{Z}^n$ added in the expression after the last equation is indeed redundant because

$$Q^{-1}(b_Q + z) \in \mathbb{Z}^n \implies b_Q + z \in \mathbb{Z}^n \iff z \in \mathbb{Z}^n,$$

where the first implication follows from $Q \in \mathbb{Z}^{n \times n}$ and the second one from $b_Q \in \mathbb{Z}^n$. Hence, there is a one-to-one relation between optimal solutions z to (2.5) and optimal solutions $x = Q^{-1} z$ of (2.2). We thus reduced (BIP) to (2.5). It remains to reduce (2.5) to (CPTU).

By Lemma 2.9, there are sets of row and column indices $I, J \subseteq [n]$, respectively, such that the half-integral entries of Q^{-1} are precisely those entries Q^{-1}_{ij} with $i \in I$ and $j \in J$. This leads to the following equivalence for any integer vector $y \in \mathbb{Z}^n$:

$$Q^{-1} y \in \mathbb{Z}^n \iff y(J) := \sum_{j \in J} y_j \text{ is even.}$$

Because $v = Q^{-1} b_Q \notin \mathbb{Z}^n$, we have that $b_Q(J)$ is odd. Hence, for $z \in \mathbb{Z}^n$ we obtain

$$Q^{-1}(b_Q + z) \in \mathbb{Z}^n \iff (b_Q + z)(J) \text{ is even} \iff z(J) \text{ is odd,}$$

implying that (2.5) can be rewritten as the problem:

$$\begin{aligned}&\max\{\bar{c}^\mathsf{T} z : \bar{C}z \leqslant 0, Q^{-1}(b_Q + z) \in \mathbb{Z}^n, z \in \mathbb{Z}^n\}\\ &= \max\{\bar{c}^\mathsf{T} z : \bar{C}z \leqslant 0, z \in \mathbb{Z}^n, z(J) \text{ odd}\}.\end{aligned}$$

To arrive at a (CPTU) problem, we need non-negativity constraints. A simple way to achieve this is to replace z by the difference of two non-negative integer vectors $x^+, x^- \in \mathbb{Z}^n_{\geqslant 0}$, i.e., $z = x^+ - x^-$. Consequently, the objective $\bar{c}^\mathsf{T} z$ thus gets replaced by $\bar{c}^\mathsf{T} x^+ - \bar{c}^\mathsf{T} x^-$ and the constraints become $\bar{C}x^+ - \bar{C}x^- \leqslant 0$.

Finally, we remark that the presented reduction can be performed in strongly polynomial time because finding a full-rank matrix Q in C and inverting it can both be done in strongly polynomial time. □

To complete the proof of the reductions claimed in Lemma 2.3 between any two problems among (BIP), (PTU), and (CPTU), it suffices to present reductions from (CPTU) to (PTU), and from (PTU) to (BIP), which are readily derived. In particular, the reduction from (CPTU) to (PTU) immediately follows from the fact that any (CPTU) problem is a (PTU) problem; it suffices to choose $\alpha = 1$ and $b = 0$ in the definition of (PTU). The following lemma shows the remaining reduction.

Lemma 2.11. Given an algorithm $\mathcal{A}$ for (BIP), one can solve any (PTU) problem using

1. operations taking strongly polynomial time, and
2. a single call to $\mathcal{A}$.

Proof. Starting with a (PTU) problem $\max\{c^\mathsf{T} x : Tx \leqslant b, x \in \mathbb{Z}^n_{\geqslant 0}, x(S) \equiv \alpha \pmod 2\}$, we define

$$\bar{A} := \left[\begin{array}{c|c} T & 0 \\ \hline \chi_S{}^\mathsf{T} & -2 \\ -\chi_S{}^\mathsf{T} & 2 \end{array}\right] \in \mathbb{Z}^{(m+2)\times(n+1)},\ \bar{c} := \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix}, \bar{b} := \begin{bmatrix} b \\ \alpha \\ -\alpha \end{bmatrix} \in \mathbb{Z}^{m+2},$$

where m is the number of rows of T as usual. The TU-ness of T easily implies that all $(n\times n)$-subdeterminants of $\bar{A}$ are bounded by 2 in absolute value. Finally, the (BIP) problem

$$\max\left\{\bar{c}^\mathsf{T} \begin{bmatrix} x \\ z \end{bmatrix} : \bar{A}\begin{bmatrix} x \\ z \end{bmatrix} \leqslant \bar{b}, -x \leqslant 0, x \in \mathbb{Z}^n, z \in \mathbb{Z}\right\}$$

is equivalent to the original (PTU) problem because, by using the definitions of $\bar{A}$, $\bar{c}$, and $\bar{b}$, it can be rewritten as

$$\max\{c^\mathsf{T} x : Tx \leqslant b, x \in \mathbb{Z}^n_{\geqslant 0}, x(S) = \alpha + 2z, z \in \mathbb{Z}\}\ .$$

□

To complete the proof of Lemma 2.3, it remains to show the following statement, whose proof is deferred to Section B.1 in the appendix.

Lemma 2.12. One can solve any (PTU) problem with an algorithm $\mathcal{A}$ for (CPTU) using

1. operations taking strongly polynomial time, and
2. a single call to $\mathcal{A}$ on a (CPTU) problem whose constraint matrix is a submatrix of the one of the (PTU) problem.

2.4 Solving (CPTU) via TU-Decompositions

In this section, we show how to use Theorem 2.4, i.e., our slightly modified version of Seymour's TU matrix decomposition theorem, to break a (CPTU) problem into smaller ones. Throughout this section we consider a (CPTU) problem with a constraint matrix T that can be decomposed through a k-sum for $k \in [3]$, as stated in cases 3–6 of Theorem 2.4. After briefly discussing the decomposition of a (CPTU) problem whose constraint matrix can be written as a 1-sum, which is straightforward, we explain our main decomposition techniques first in the context of a 2-sum. The 3-sum then follows. It is technically more involved, but follows similar ideas.

The following lemma implies that we can restrict ourselves to (CPTU) problems that are bounded. This simplifies the discussion of our decomposition techniques. The proof is deferred to Section B.3 in the appendix.

Lemma 2.13. Let an instance of (CPTU) be given. Then, it can be strongly polynomially reduced to a bounded (CPTU) problem that has the same constraint matrix.

Hence, throughout this section, we furthermore assume that the (CPTU) problem we consider is bounded.

2.4.1 Decomposition Approach for 1-Sums

Assume that T can be written as a 1-sum, as stated in point 3 of Theorem 2.4, i.e.,

$$T = L \oplus_1 R = \begin{bmatrix} L & 0 \\ 0 & R \end{bmatrix},$$

where $L \in \{-1,0,1\}^{m_L \times n_L}$, $R \in \{-1,0,1\}^{m_R \times n_R}$ with $m_L, m_R \geqslant 2$. Clearly, the (CPTU) problem immediately decomposes into one (CPTU) problem with constraint matrix L and one with constraint matrix R. Hence, an optimal solution to the original problem is found by solving these two independent subproblems and concatenating the resulting optimal solution vectors.

2.4.2 Decomposition Approach for 2-Sums

We now consider the case when T can be written as a 2-sum, as stated in point 4 of Theorem 2.4, i.e.,

$$T = [L\ a] \oplus_2 \begin{bmatrix} d^\mathsf{T} \\ R \end{bmatrix} = \begin{bmatrix} L & ad^\mathsf{T} \\ 0 & R \end{bmatrix},$$

for $L \in \{-1,0,1\}^{m_L \times n_L}$, $R \in \{-1,0,1\}^{m_R \times n_R}$ with $m_L, m_R \geqslant 2$. Our goal is to decompose the (CPTU) problem into smaller subproblems such that optimal solutions to the subproblems allow for obtaining one particularly well-structured optimal solution of the original (CPTU) problem. The following lemma shows that there is an optimal solution with a very useful structure that we will exploit later. A similar result in a slightly different context was also used in [41].

Lemma 2.14. Consider a feasible and bounded (CPTU) problem with constraint matrix $T \in \{-1,0,1\}^{m \times n}$. Then there exists an optimal solution x^* to (CPTU) satisfying

$$w^\mathsf{T} x^* \in \{-1,0,1\}\ ,$$

for all vectors $w \in \{-1,0,1\}^m$ such that appending w^T as an additional row to T preserves TU-ness. For brevity, we call such an optimal solution x^*, a **well-structured** optimal solution to the (CPTU) problem.

Proof. We show that any optimal solution x^* to (CPTU) with minimum ℓ_1-norm fulfills the claimed property. Assume for the sake of deriving a contradiction that there is an optimal solution x^* with minimum ℓ_1-norm violating the conditions of the lemma. Hence, there is a vector $w \in \{-1,0,1\}^m$ such that $\left[\begin{smallmatrix} T \\ w^\mathsf{T} \end{smallmatrix}\right]$ is TU, and $w^\mathsf{T} x^* \geqslant 2$ (the case $w^\mathsf{T} x^* \leqslant -2$ transforms to this case by replacing w by $-w$). Consider the polytope

$$P := \big\{x \in \mathbb{R}^n \colon Tx^* \leqslant Tx \leqslant 0,\\ 1 \leqslant w^\mathsf{T} x \leqslant w^\mathsf{T} x^* - 1,\ 0 \leqslant x \leqslant x^*\big\}.$$

Notice that $P \neq \emptyset$ because $\frac{1}{2}x^* \in P$. Moreover, the constraint matrix defining P is TU, and all right-hand sides are integral. Thus, P is an integral non-empty polytope and therefore there exists an integral point $u \in P \cap \mathbb{Z}^n$. Let $v = x^* - u$. Notice that both u and v fulfill all constraints of (CPTU) with the possible exception of the parity constraint; this follows immediately from $u \in P$. Moreover, again using $u \in P$, we have $u, v \leqslant x^*$ and $u, v \neq x^*$. More precisely, $u, v \leqslant x^*$ follows from $0 \leqslant u \leqslant x^*$, and $u \neq x^*$ follows from $w^\mathsf{T} u \leqslant w^\mathsf{T} x^* - 1$; finally, $v \neq x^*$ follows from $u \neq 0$ which holds due to $w^\mathsf{T} u \geqslant 1$. Hence, both vectors u and v have strictly smaller ℓ_1-norm than x^*. Furthermore, $x^* = u + v$ implies that either u or v has the correct parity, i.e., $u(S)$ or $v(S)$ is odd. Assume that $u(S)$ is odd (the case $v(S)$ odd is analogous). Thus, u is a feasible solution to (CPTU) with smaller ℓ_1-norm than x^*. Therefore, we must have $c^\mathsf{T} u < c^\mathsf{T} x^*$, which, due to $x = u + v$, implies $c^\mathsf{T} v > 0$. However, the vector $x^* + 2v$ is also a feasible solution to (CPTU) with $c^\mathsf{T}(x^* + 2v) > c^\mathsf{T} x^*$, violating optimality of

x^*. Notice that the factor of 2 in front of v makes sure that $x^* + 2v$ fulfills the parity constraint of (CPTU) because $(x^* + 2v)(S)$ is odd since $2v(S)$ is even and $x^*(S)$ is odd. $\square$

Returning to the 2-sum, a key implication of Lemma 2.14 is that a well-structured optimal solution $x^* = \left[\begin{smallmatrix} x_L^* \\ x_R^* \end{smallmatrix}\right]$ to our (CPTU) problem, where x_L^* are the first n_L coordinates of x^* and x_R^* the last n_R, satisfies

$$d^\mathsf{T} x_R^* \in \{-1, 0, 1\} \ . \tag{2.6}$$

This follows from Lemma 2.14 by observing that the row $[0\ d^\mathsf{T}]$ can be added to T without violating TU-ness. One way to observe this is as follows. Let Q be the matrix obtained from R by appending the row d^T, and consider replacing the matrix R by Q in the second summand of the 2-sum of T. Clearly, $\left[\begin{smallmatrix} d^\mathsf{T} \\ Q \end{smallmatrix}\right]$ is TU, because it is identical to the TU matrix $\left[\begin{smallmatrix} d^\mathsf{T} \\ R \end{smallmatrix}\right]$ with the only difference that we doubled a row. The resulting 2-sum with the new second summand leads to the matrix T with an additional row that corresponds to $[0\ d^\mathsf{T}]$. This new 2-sum is TU, because the 2-sum of any two TU-matrices is TU (see Lemma 1.12). Hence, adding $[0\ d^\mathsf{T}]$ as an additional row to T does not destroy TU-ness, as desired.

Equation 2.6 shows that there is only very limited interaction between x_L^* and x_R^* within the constraints of our (CPTU) problem. This limited interaction is what we exploit to decompose the problem. More precisely, we will solve several versions of one of the two parts (corresponding to L or R) of the problem, one for each possible interaction with the other part. These solutions will then be integrated to solve a smaller problem on the other part that allows for recovering an optimal solution. For our algorithm to be efficient, we have to make sure to solve multiple subproblems only on the smaller (in terms of number of rows) of the two parts L and R. We start by showing how to realize this idea when $m_R \leqslant m_L$, and then show how it can be adjusted when L is the smaller part, i.e., $m_L < m_R$.

Solving first the R-subproblem (case $m_R \leqslant m_L$):

Consider the following family of (PTU) problems with constraint matrix R parameterized by $\beta \in \{0, 1\}$ and $\alpha \in \{-1, 0, 1\}$:

$$\begin{aligned} \max\{{c_R}^\mathsf{T} x_R \colon\ & R x_R \leqslant 0,\ d^\mathsf{T} x_R = \alpha, \\ & x_R \in \mathbb{Z}_{\geqslant 0}^{n_R},\ x_R(S_R) \equiv \beta \pmod 2\} \ , \end{aligned} \tag{2.7}$$

where $S_R \subseteq S$ corresponds to all columns of S within the right problem, i.e., these are the columns of S within the last n_R indices, and $S_L = S \backslash S_R$. Similarly, we partition the coordinates of the objective c into a left and right part, i.e., $c = \left[\begin{smallmatrix} c_L \\ c_R \end{smallmatrix}\right]$. We solve (2.7) for all 6 combinations of $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$, by a reduction to (CPTU) problems through Lemma 2.3. Let $\rho_R(\alpha, \beta)$ be the optimal objective value of (2.7), where we set $\rho_R(\alpha, \beta) = -\infty$ if (2.7) is infeasible for this choice of α and β, and $\rho_R(\alpha, \beta) = \infty$ if the problem is unbounded. We now incorporate these optimal ways of solving the right part of our original (CPTU) problem into the left part as follows. We define a modified version $\bar{L}$

of the constraint matrix L, where, for every combination of $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$ with $\rho_R(\alpha, \beta) \notin \{-\infty, \infty\}$, we add a column. The column we add depends on the value of α, and is equal to $\alpha \cdot a$. To simplify the exposition, assume that $\rho_R(\alpha, \beta) \notin \{-\infty, \infty\}$ for all combinations of $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$; hence, we will add 6 columns and obtain

$$\bar{L} := [L \mid -a \mid 0 \mid a \mid -a \mid 0 \mid a] \ ,$$

where the first three added columns correspond to $\beta = 0$ and the last three to $\beta = 1$. Let J correspond to the indices of the last three columns, those corresponding to $\beta = 1$. The extended objective $\bar{c}$ is defined by

$$\begin{aligned}\bar{c}^{\mathsf{T}} := \big({c_L}^{\mathsf{T}} \ \mid \ \rho_R(-1,0) \quad \rho_R(0,0) \quad \rho_R(1,0) \\ \rho_R(-1,1) \quad \rho_R(0,1) \quad \rho_R(1,1)\big) \ .\end{aligned}$$

Again, in the general case if $\rho_R(\alpha, \beta) \in \{-\infty, \infty\}$ for some values of α and β, we do not add the corresponding entry to $\bar{c}$. We then solve the following (CPTU) problem with only m_L rows:

$$\max\left(\bar{c}^{\mathsf{T}}x \colon \bar{L}x \leqslant 0,\ x \in \mathbb{Z}_{\geqslant 0}^{n_L+6}, x(S_L \cup J) \text{ odd}\right) \ . \tag{2.8}$$

Notice that (2.8) is indeed a (CPTU) problem because $\bar{L}$ is TU, which follows immediately from TU-ness of $[L \mid a]$. Based on the fact that a well-structured optimal solution x^* satisfies (2.6), it is not hard to see that the optimal value of (2.8) is at least as good as $c^{\mathsf{T}}x^*$. Conversely, we can transform any solution x of (2.8) into one of the original problem of the same value. This is achieved by interpreting the value x_i, where i is one of the 6 added columns corresponding to some $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$, as the number of times we use the optimal solution to (2.7) for this pair α, β. The following lemma formalizes this discussion.

Lemma 2.15.

1. Given optimal solutions to (2.7) for all $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$, one can in strongly polynomial time transform any feasible solution to (2.8) into a feasible solution of the original (CPTU) problem with same objective value.
2. If the original (CPTU) problem is feasible and bounded, then (2.8) is feasible and has the same optimal value as the original (CPTU) problem.

Proof. We start by proving 1. Let

$$I = \{(\alpha, \beta) : \alpha \in \{-1, 0, 1\}, \beta \in \{0, 1\}, \rho_R(\alpha, \beta) \notin \{-\infty, \infty\}\}$$

be the pairs of parameters (α, β) for which (2.7) has a finite optimal solution. Moreover, for $(\alpha, \beta) \in I$, let $y(\alpha, \beta)$ be an optimal solution to 2.7 for this choice of α and β. Let z be a feasible solution to (2.8); hence $z \in \mathbb{Z}^{n_L+|I|}$, and we denote its first n_L coordinates by z_L, and each remaining coordinate corresponds to a pair $(\alpha,\beta) \in I$, and we denote the z-value of this coordinate by $z(\alpha,\beta)$. We now define a feasible solution x for (CPTU) as follows

$$x := \begin{bmatrix} x_L \\ x_R \end{bmatrix} = \begin{bmatrix} z_L \\ \sum_{(\alpha,\beta)\in I} z(\alpha,\beta)\cdot y(\alpha,\beta) \end{bmatrix} \ .$$

We start by showing that x is indeed a feasible solution to (CPTU). Clearly, $x \geqslant 0$ because $z \geqslant 0$ and $y(\alpha,\beta) \geqslant 0$ for $(\alpha,\beta) \in I$. Moreover

$$\begin{aligned} Lx_L + ad^\mathsf{T} x_R &= Lz_L + a \cdot \sum_{(\alpha,\beta)\in I} z(\alpha,\beta) d^\mathsf{T} y(\alpha,\beta) \\ &= Lz_L + a \sum_{(\alpha,\beta)\in I} z(\alpha,\beta)\cdot \alpha \leqslant 0 \ , \end{aligned}$$

where the second equality follows from $d^\mathsf{T} y(\alpha,\beta) = \alpha$, which holds because $y(\alpha,\beta)$ is a solution to (2.7), and the inequality follows from z being a solution to (2.8). Furthermore,

$$Rx_R = R\left(\sum_{(\alpha,\beta)\in I} z(\alpha,\beta)\cdot y(\alpha,\beta)\right) = \sum_{(\alpha,\beta)\in I} z(\alpha,\beta) Ry(\alpha,\beta) \leqslant 0 \ ,$$

where the inequality follows from $Ry(\alpha,\beta) \leqslant 0$ for $(\alpha,\beta) \in I$, because $y(\alpha,\beta)$ is a solution to (2.7), and by using $z(\alpha,\beta) \geqslant 0$ for $(\alpha,\beta) \in I$. To show that x is a feasible solution to (CPTU), it remains to prove that x fulfills the parity constraint, which holds due to

$$\begin{aligned} x(S) &= x_L(S_L) + x_R(S_R) \\ &= z_L(S_L) + \sum_{\substack{\alpha\in\{-1,0,1\} \\ \text{s.t. } (\alpha,1)\in I}} z(\alpha,1)\cdot y(\alpha,1) \equiv 1 \pmod 2 \ , \end{aligned}$$

which holds because z fulfills the parity constraint of (2.8). Finally, the objective value of x indeed matches the one of z because

$$\begin{aligned} c^\mathsf{T} x &= {c^\mathsf{T}}_L z_L + {c^\mathsf{T}}_R \sum_{(\alpha,\beta)\in I} z(\alpha,\beta)\cdot y(\alpha,\beta) \\ &= {c^\mathsf{T}}_L z_L + \sum_{(\alpha,\beta)\in I} z(\alpha,\beta)\rho_R(\alpha,\beta) = \bar{c}^\mathsf{T} z \ , \end{aligned}$$

where the second equality follows from $y(\alpha,\beta)$ being an optimal solution to (2.7), and $\rho_R(\alpha,\beta)$ is by definition its optimal value. Notice that x is clearly obtained in strongly polynomial time given z and $y(\alpha,\beta)$ for $(\alpha,\beta) \in I$.

We now show point 2. Hence, consider a (CPTU) problem that is feasible and bounded. Point 1 implies that the optimal value of (CPTU) is at least as large as the optimal value of (2.8). Hence, it remains to show that the optimal value of (2.8) is least the one of (CPTU). Let $x^* = \left[\begin{smallmatrix} x_L^* \\ x_R^* \end{smallmatrix}\right]$ be a solution to (CPTU) with $\alpha^* := d^\mathsf{T} x_R^* \in \{-1,0,1\}$. Let $\beta^* \in \{0,1\}$ be the parity of $x_R^*(S_R)$, i.e.,

$x_R^*(S_R) \equiv \beta^*$ (mod 2). We finish the proof by constructing a solution z to (2.8) with the same objective value as x^*, i.e., $\bar{c}^\mathsf{T} z = c^\mathsf{T} x^*$. Again, we define

$$I = \{(\alpha, \beta) : \alpha \in \{-1, 0, 1\}, \beta \in \{0, 1\}, \rho_R(\alpha, \beta) \notin \{-\infty, \infty\}\} \ .$$

First observe that $(\alpha^*, \beta^*) \in I$: Indeed, because x_R^* is a feasible solution to (2.7) with parameters $\alpha = \alpha^*$ and $\beta = \beta^*$, we have $\rho_R(\alpha^*, \beta^*) \neq -\infty$; moreover, we cannot have $\rho_R(\alpha^*, \beta^*) = \infty$ because this would imply that there is a solution x_R to (2.7) with $c^\mathsf{T} x_R > c^\mathsf{T} x_R^*$, which would lead to a solution $\left[\begin{smallmatrix} x_L^* \\ x_R \end{smallmatrix}\right]$ to (CPTU) with strictly better objective value than x^*, violating optimality of x^*.

We define $z \in \mathbb{Z}_{\geqslant 0}^{n_L + |I|}$ as follows. The first n_L coordinates are set to be equal to x_L^*, i.e., $z_L = x_L^*$. For the remaining coordinates, which correspond to the pairs in I, we set all of them to 0 except for the pair (α^*, β^*) which is set to 1. Again, one can easily observe that z is a feasible solution to (2.8), which follows immediately from feasibility of x^* for (CPTU). Moreover, $\bar{c}^\mathsf{T} z = c^\mathsf{T} x^*$ as desired. □

Solving first the L-subproblem (case $m_R > m_L$):

We now discuss how a similar reduction can be performed by first solving the L-part of our (CPTU) problem. This is what we will do if $m_R > m_L$. We again define subproblems for $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$:

$$\max\{{c_L}^\mathsf{T} x_L \colon L x_L \leqslant \alpha \cdot a,\ x_L \in \mathbb{Z}_{\geqslant 0}^{n_L},\ x_L(S_L) \equiv \beta \ (\text{mod } 2)\} \ . \tag{2.9}$$

As before, we denote by $\rho_L(\alpha, \beta)$ the optimal value of (2.9). To incorporate these solutions of the L-part of our (CPTU) problem into the R-part, we will add one additional constraint and up to one additional variable for each of the 6 combinations of $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$. Again, we only add a variable for some pair α, β if $\rho_L(\alpha, \beta) \notin \{-\infty, \infty\}$. For simplicity of exposition, we assume that $\rho_L(\alpha, \beta) \notin \{-\infty, \infty\}$ for all 6 pairs α, β. The additional constraint we add is defined by the following coefficient vector:

$$h^\mathsf{T} := \begin{pmatrix} -1 & 0 & 1 & -1 & 0 & 1 & | & d^\mathsf{T} \end{pmatrix} \ ,$$

and, as before, we define an extended objective $\bar{c}$ by

$$\bar{c}^\mathsf{T} := \big(\rho_L(-1,0) \quad \rho_L(0,0) \quad \rho_L(1,0) \\ \qquad\qquad \rho_L(-1,1) \quad \rho_L(0,1) \quad \rho_L(1,1) \quad | \quad {c_R}^\mathsf{T}\big) \ .$$

The combined problem is now given by

$$\max\left\{\bar{c}^\mathsf{T} x \colon R x_R \leqslant 0,\ h^\mathsf{T} x = 0,\ x \in \mathbb{Z}_{\geqslant 0}^{6 + n_R},\ x(S_R \cup J) \text{ odd}\right\} \ , \tag{2.10}$$

where x_R represents the components of x that correspond to the R-part, i.e., these are the last n_R components of x. In this setting, we need the equality constraint to make sure that a solution to the combined problem can be transformed into one of the original problem. This is due to the fact that the interaction between the left and right problem is not as explicit anymore as in the case $m_R \leqslant m_L$, where we first solved subproblems of the R-problem. In the setting we have now, we must for example make sure that a solution x_L to (2.9) with $\alpha = -1$, i.e, $Lx_L \leqslant -a$, gets combined with a solution x_R to the right problem with $d^\mathsf{T} x_R = 1$, which will lead to a contribution of $ad^\mathsf{T} x_R = a$ to the constraints within the L-part of the (CPTU) problem. It is crucial that x_L gets combined with a R-solution satisfying $d^\mathsf{T} x_R = 1$, and not just say $d^\mathsf{T} x_R \leqslant 1$, because a may have negative components.

Note that (2.10) is not yet a (CPTU) problem, because of the equality constraint. Simply replacing the equality constraints by two inequalities does not work out, because it would lead to a problem with $m_R + 2$ constraints, which may be as large as the original number of constraints m, and we would not achieve our goal to end up with a strictly smaller problem, which we need to make progress. A standard approach would be to eliminate one variable through the equality constraints. When doing so, one has to take care that we again end up with a (CPTU) problem which, by definition, requires non-negativity constraints on the variables. The following lemma shows that this is possible, by adding one inequality constraint back after eliminating one variable. Its proof is deferred to Section B.3 in the appendix.

Lemma 2.16. Consider a (CPTU) problem with a constraint matrix T of the form

$$T = \begin{bmatrix} h^\mathsf{T} \\ -h^\mathsf{T} \\ M \end{bmatrix}.$$

Then, one can in strongly polynomial time reduce the (CPTU) problem to another (CPTU) problem with one variable and one constraint less.

Even though this is technically not needed to obtain an efficient algorithm, we note that Lemma 2.16 can also be used to reformulate a subproblem of type (2.7) into a (CPTU) problem with only $m_R + 1$ many rows (instead of $m_R + 2$).

Finally, similarly to the case $m_R \leqslant m_L$, we can show that the combined problem (2.10) is tightly linked to the original (CPTU) problem with constraint matrix T. The proof of the following lemma is deferred to Section B.3.

Lemma 2.17.

1. Given optimal solutions to (2.9) for all $\alpha \in \{-1, 0, 1\}$ and $\beta \in \{0, 1\}$, one can in strongly polynomial time transform any feasible solution to (2.10) into a feasible solution of the original (CPTU) problem with same objective value.
2. If the original (CPTU) problem is feasible and bounded, then (2.10) is feasible and has the same optimal value as the original (CPTU) problem.

We summarize our findings for both variants to decompose the (CPTU) problem, i.e., either by first solving subproblems on the R-part or L-part, in the following theorem.

Theorem 2.18. Consider a (CPTU) problem with a constraint matrix $T \in \{-1, 0, 1\}^{m \times n}$ that can be written as a 2-sum, where the matrices L and R of the 2-sum have each at least two rows. Then an optimal solution to (CPTU) can be obtained by:

1. Solving up to 6 (CPTU) problems with a constraint matrix with m_1 rows and n_1 columns, and one (CPTU) problem with a constraint matrix with m_2 rows and n_2 columns, where $m_1 \leqslant m_2 < m$, $m_1+m_2 \leqslant m+1$, and $n_1, n_2 \leqslant n+5$.
2. Further operations taking strongly polynomial time.

2.4.3 Decomposition Approach for 3-Sums

The decomposition approach for 3-sums follows similar ideas as the one for 2-sums, but is technically more involved. We prove the following theorem, which, together with our discussion for 1-sums and Theorem 2.18, and the discussion of pivoting in the next section, implies Theorem 2.6, as desired.

Theorem 2.19. Consider a (CPTU) problem with a constraint matrix $T \in \{-1, 0, 1\}^{m \times n}$ as in case 5 in Theorem 2.4, where the matrices L and R of the 3-sum have each at least two rows. Then an optimal solution to (CPTU) can be obtained by:

1. Solving up to 14 (CPTU) problems with a constraint matrix with m_1 rows and n_1 columns, and one (CPTU) problem with a constraint matrix wth m_2 rows and n_2 columns, where $m_1 \leqslant m_2 < m$, $m_1+m_2 \leqslant m+2$, and $n_1, n_2 \leqslant n+13$.
2. Further operations taking strongly polynomial time.

Proof. Let us lay out our proof plan first. We assume that we are in case 5 of Theorem 2.4.

Recall from Definition 1.10 that the 3-sum looks as follows.

$$T := \begin{bmatrix} L & a & a \\ f^\mathsf{T} & 0 & 1 \end{bmatrix} \oplus_3 \begin{bmatrix} 1 & 0 & d^\mathsf{T} \\ g & g & R \end{bmatrix} = \begin{bmatrix} L & ad^\mathsf{T} \\ gf^\mathsf{T} & R \end{bmatrix},$$

with $L \in \mathbb{Z}^{m_L \times n_L}$, $R \in \mathbb{Z}^{m_R \times n_R}$ and $m_L,\ m_R, n_R \geqslant 2$, and we assume that there is no way of permuting rows and columns of T to write it as a 1- or 2-sum. We wish to solve (CPTU), that is, $\max\{c^\mathsf{T}x \colon Tx \leqslant 0,\ x \in \mathbb{Z}^n_{\geqslant 0},\ x(S) \text{ odd}\}$.

For the 3-sum, we will, similarly to the proof of Theorem 2.18, do the following:

1. Restate the problem as pairs of two smaller sub-problems.
   a) Append rows to the summands preserving their total unimodularity.
   b) Invoke Lemma 1.12 to show that a certain vector h is TU-appendable to T, i.e., that $\left[\begin{smallmatrix} T \\ h^\mathsf{T} \end{smallmatrix}\right]$ is TU.

c) Invoke Lemma 2.14 to see that, given the problem is feasible, the corresponding right-hand-sides are, for a well-structured solution x^*, ± 1 or zero.

For each of the cases $m_L \leqslant m_R$ and $m_L > m_R$, we will then proceed as follows:

2. For each pair, solve the smaller (wrt. number of rows) problem. Then incorporate the optimal values into the larger problem by appending columns to (a modified version of) its constraint matrix.
3. Argue why the new constraint matrix is TU.
4. Show how each feasible solution gives rise to a feasible solution for the original problem (CPTU) of the same cost, and that the optimal value is the same if both problems are feasible.
5. Argue that the new problem has a smaller number of rows (by invoking Lemma B.2).

Let us now show how to do each of these steps.

1. By appending the row $[0\ 0\ d^\intercal]$ to the right summand, it stays totally unimodular (this can, for example, be seen with Ghouila-Houri's argument [19]). Thus we conclude by Lemma 2.14 that if a well-structured optimal solution x^* for the original (CPTU) problem exists, it fulfills $d^\intercal x^*_R \in \{-1,0,1\}$. By a similar argument with the left summand, $f^\intercal x^*_L \in \{-1,0,1\}$. Since $gf^\intercal x^*_L + Rx^*_R \leqslant 0$, we have that $Rx^*_R \leqslant -gf^\intercal x^*_L$. Analogously, $Lx^*_L \leqslant -ad^\intercal x^*_R$. By Ghouila-Houri, we see that also $[f^\intercal\ 1\ 1]$ is TU-appendable to the left summand. Thus,

$$[f^\intercal\ d^\intercal]x^* \in \{-1,0,1\}. \tag{2.11}$$

We can then define the pairs of problems as

$$\max\{{c_L}^\intercal x\colon Lx \leqslant \alpha_L a,\ f^\intercal x = \beta_L,\ x(S_L) \equiv \gamma \pmod 2\}, \tag{2.12}$$

and

$$\max\{{c_R}^\intercal x\colon Rx \leqslant \beta_R g,\ d^\intercal x = \alpha_R,\ x(S_R) \equiv 1-\gamma \pmod 2\}, \tag{2.13}$$

for $\alpha,\ \beta \in \{-1,0,1\}^2$ and $\gamma \in \{0,1\}$. Note that any $w_L \in \mathbb{Z}^{n_L}$ that is feasible for (2.12) for some α_L, β_L and γ, together with a feasible solution w_R for (2.13) with $\alpha_R = -\alpha_L$, $\beta_R = -\beta_L$ and the same value of β as w_L had, yields a feasible solution $\begin{bmatrix} w_L \\ w_R \end{bmatrix}$ for the original (CPTU) problem.

First, assume that $m_L \geqslant m_R$.

2. The constant number of smaller problems to solve are given by (2.13). Recall that all of these can be reduced in polynomial time to a problem of the form (CPTU). Denote by $\rho(\alpha_R, \beta_R, \gamma)$, for $\alpha_R,\ \beta_R \in \{-1,0,1\}$ and $\gamma \in \{0,1\}$, the respective optimal values of those which are feasible and

bounded, set it to ∞ for those which are unbounded, and to $-\infty$ if one is infeasible. As we did for the 2-sum, we first restrict ourselves to the case that all of them are finite, and discuss how to solve the general case in the end.

However, because of (2.11), there is, given the original (CPTU) problem is feasible, an optimal solution x^* with x^*_L feasible for (2.12), for some α_L, β_L, and x^*_R feasible for (2.13), for some α_R, β_R, such that

$$|\alpha_R + \beta_L| \leqslant 1. \tag{2.14}$$

We may therefore restrict ourselves to pairs of problems which fulfill (2.14). Define

$$X = \begin{bmatrix} a & a & 0 & 0 & 0 & -a & -a \end{bmatrix},$$

and denote by h^T and $\bar{c}^\mathsf{T}$ the following vectors:

$$\begin{aligned} h^\mathsf{T} = \big(1 \quad 0 \quad 1 \quad 0 \quad -1 \quad 0 \quad -1 \mid \; 1 \quad 0 \\ 1 \quad 0 \quad -1 \quad 0 \quad -1\big) \end{aligned}$$

and

$$\begin{aligned} \bar{c} = \big(&\rho(1,1,0), \rho(1,0,0), \rho(0,1,0), \rho(0,0,0), \rho(0,-1,0), \rho(-1,0,0), \rho(-1,-1,0), \\ &\rho(1,1,1), \rho(1,0,1), \rho(0,1,1), \rho(0,0,1), \rho(0,-1,1), \rho(-1,0,1), \rho(-1,-1,1), c_L\big)^\mathsf{T}. \end{aligned}$$

Denote by $I = \{1, \ldots, 14\}$ the column indices corresponding to X in the matrix $[\,X\ X\ L\,]$, and by I^c its complement. Let J denote the column indices of the first seven columns in the above matrix. We then consider the problem of finding

$$\begin{aligned} \max \big\{ \bar{c}^\mathsf{T} x \colon \begin{bmatrix} X & X & L \end{bmatrix} x \leqslant 0,\ [h^\mathsf{T}\ f^\mathsf{T}]x = 0, \\ x \in \mathbb{Z}^{n+14}_{\geqslant 0},\ x(S_L \cup J) \text{ odd} \big\}. \end{aligned} \tag{2.15}$$

3. Clearly, the new constraint matrix is TU, as we just appended standard unit vectors or copies of the column $\begin{bmatrix} a \\ 1 \end{bmatrix}$ or its negative to $\begin{bmatrix} L \\ f^\mathsf{T} \end{bmatrix}$.

4. We first show how to transform a feasible solution z to (2.15) into one for the original (CPTU) problem in polynomial time.

   For each of the problems (2.13), denote by $y(\alpha_R, \beta_R, \gamma)$ an optimal solution, i.e., one which achieves that ${c_R}^\mathsf{T} y(\alpha_R, \beta_R, \gamma) = \rho(\alpha_R, \beta_R, \gamma)$. Similarly as in the case for the 2-sum, define Y to be the matrix that contains as columns the $y(\alpha_R, \beta_R, \gamma)$'s in the order corresponding to the objective function vector in (2.15).

   Consider the vector $y := \begin{bmatrix} z_{15:(n_L+14)} \\ \sum_{i \in I} z_i Y_{\cdot,i} \end{bmatrix}$, which is feasible for the original problem (CPTU) for the following reasons:

   - $y \geqslant 0$.

- By our choice of z and Y, we have that for any $\omega \in \mathbb{Z}$,

$$h^{\mathsf{T}} z_I = \omega \Rightarrow R \sum_{i \in I} z_i Y_{\cdot,i} \leqslant \omega g,$$

and therefore, since z fulfills $h^{\mathsf{T}} z_I = -f^{\mathsf{T}} z_{15:(n_L+14)}$, we have that $R \sum_{i \in I} z_i Y_{\cdot,i} \leqslant -g f^{\mathsf{T}} z_{15:(n_L+14)}$, and thus

$$g f^{\mathsf{T}} y_L + R y_R = g f^{\mathsf{T}} z_{15:(n_L+14)} + R \sum_{i \in I} z_i Y_{\cdot,i} \leqslant 0.$$

- Also by our choice of z and Y, $-L y_L = -L z_{15:(n_L+14)} \geqslant a h^{\mathsf{T}} z_{15:(n_L+14)} = a d^{\mathsf{T}} \sum_{i \in I} \bar{z}_i Y_{\cdot,i} = a d^{\mathsf{T}} y_R$.

-

$$\begin{aligned} y(S) =& y_L(S_L) + y_R(S_R) = z_{15:(n_L+14)}(S_L) + \sum_{i \in I} z_i Y_{\cdot,i}(S_R) \\ \equiv& z_{15:(n_L+14)}(S_L) + \sum_{i \in J} z_i \pmod 2 \equiv 1 \pmod 2. \end{aligned}$$

Let us now assume that the original (CPTU) problem is feasible. Denote by x^* an optimal solution that fulfills all the additional conditions on its right-hand-side imposed above. We write it as $x^* = \begin{bmatrix} x_L^* \\ x_R^* \end{bmatrix}$. x_R^* is a feasible solution to (2.13) for $\alpha_R = f^{\mathsf{T}} x_L$, $\beta_R = d^{\mathsf{T}} x_R^*$ and $\gamma \equiv x_R(S^R) \pmod 2$. Let l be the index of this problem in $[X\ X\ L]$, then $\begin{bmatrix} e_l \\ x_L^* \end{bmatrix}$ is feasible for (2.15) and $c^{\mathsf{T}} x^* = \bar{c}^{\mathsf{T}} \begin{bmatrix} e_L \\ x_L^* \end{bmatrix}$.

Finally, let us discuss the case that some of the $\rho_R(\alpha_R, \beta_R, \gamma)$ are $\pm\infty$. Then, analogously as in the case for the 2-sum, we delete the corresponding variables in (2.15), thus effectively constraining them to zero. This way, we impose additional constraints and thus, we will still have that any solution for (2.15) can be transformed into one of the same cost for the original (CPTU) problem. On the other hand, let x^* be an optimal solution for the original (CPTU) problem, and choose l as above. Then the corresponding problem (2.13) cannot have optimal value ∞, since then, there would be a w for which ${c_R}^{\mathsf{T}} w > {c_R}^{\mathsf{T}} x_R^*$. But then, $\begin{bmatrix} x_L^* \\ w \end{bmatrix}$ would also be feasible for the original (CPTU)-problem and $c^{\mathsf{T}} \begin{bmatrix} x_L^* \\ w \end{bmatrix} > c^{\mathsf{T}} x^*$, a contradiction.

5. We now wish to argue that this problem has a lesser number of rows. Assume it did not, implying that $m_R = 1$. But this contradicts our choice of the k-sum.

We may thus return the vector y from above as an optimal solution. Note that it can be computed in polynomial running time. By symmetry, the case $m_L \leqslant m_R$ is solved analogously. $\square$

2.4.4 Pivoting

Finally, we discuss how to efficiently reduce case 6 in Theorem 2.4 to the one above. In the following lemma, we show how to obtain a (CPTU) problem whose constraint matrix we can decompose.

Lemma 2.20. Consider a TU-matrix T and the matrix M obtained from pivoting T at element $T_{i,j}$. Let $\widehat{M}$ be the matrix obtained from M by multiplying column j by -1. Assume that there exists an algorithm to solve (CPTU) with constraint matrix $\widehat{M}$ that is efficient, i.e., whose running time is polynomially bounded by m and n. Then, there also exists an efficient algorithm for (CPTU) with constraint matrix T.

Proof of Lemma 2.20. Let us write the non-negativity constraints of the (CPTU) problem explicitly, and restate the problem as solving

$$\max\{c^\mathsf{T} x\colon \begin{bmatrix} T \\ -\mathcal{I} \end{bmatrix} \leqslant 0,\ x(S) \text{ odd}\},$$

where $\mathcal{I}_n$ is the $(n \times n)$-identity matrix. Without loss of generality, M arises from T by pivoting at the element in the first row and column. Write $T = \begin{bmatrix} \varepsilon & c^\mathsf{T} \\ b & D \end{bmatrix}$. Consider the unimodular matrix $Q \in \mathbb{Z}^{n\times n}$ which corresponds to columns operations such that $(TQ)_{1,\cdot} = [-1, 0, \dots, 0]$. Then,

$$\begin{bmatrix} T \\ -\mathcal{I}_n \end{bmatrix} Q = \begin{bmatrix} \varepsilon & c^\mathsf{T} \\ b & D \\ -1 & 0 \\ 0 & -\mathcal{I}_{n-1} \end{bmatrix} Q = \begin{bmatrix} -1 & 0 \\ -\varepsilon b & D - \varepsilon b c^\mathsf{T} \\ \varepsilon & \varepsilon c^\mathsf{T} \\ 0 & -\mathcal{I}_{n-1} \end{bmatrix}.$$

Thus, by column operations and a single swap of rows, we arrive at the matrix $\begin{bmatrix} \hat{M} \\ -\mathcal{I}_n \end{bmatrix}$, i.e., we can write $\hat{M} = PTQ$ for a permutation matrix P.

We can therefore reformulate our problem as follows, where $\chi_S \in \{0,1\}^n$ is the characteristic vector of set S:

$$\begin{aligned}
&\max\{c^\mathsf{T} x\colon Tx \leqslant 0,\ x(S) \text{ odd},\ x \in \mathbb{Z}^n_{\geqslant 0}\} \\
&= \max\{c^\mathsf{T} Q Q^{-1} x\colon \begin{bmatrix} T \\ -\mathcal{I}_n \end{bmatrix} Q Q^{-1} x \leqslant 0,\ x(S) \text{ odd},\ x \in \mathbb{Z}^n\} \\
&= \max\{c^\mathsf{T} Q Q^{-1} x\colon P \begin{bmatrix} T \\ -\mathcal{I}_n \end{bmatrix} Q Q^{-1} x \leqslant 0,\ {\chi_S}^\mathsf{T} x \text{ odd},\ x \in \mathbb{Z}^n\} \\
&= \max\{c^\mathsf{T} Q Q^{-1} x\colon \hat{M} Q^{-1} x \leqslant 0,\ {\chi_S}^\mathsf{T} Q Q^{-1} x \text{ odd},\ x \in \mathbb{Z}^n_{\geqslant 0}\} \\
&= \max\{c^\mathsf{T} Q x\colon \hat{M} x \leqslant 0,\ {\chi_S}^\mathsf{T} Q x \text{ odd},\ x \in \mathbb{Z}^n_{\geqslant 0}\}, \\
&\qquad \text{which, with } \bar{S} := \left\{ i \in \{1, \dots, n\} \colon ({\chi_S}^\mathsf{T} Q)_i \equiv 1 \pmod 2 \right\}, \\
&= \max\{c^\mathsf{T} Q x\colon \hat{M} x \leqslant 0,\ x(\bar{S}) \text{ odd},\ x \in \mathbb{Z}^n_{\geqslant 0}\}.
\end{aligned}$$

$\square$

Consequently, we can find a 3-sum decomposition of $\hat{M}$ defined as in the Lemma above, apply our recursion strategy to solve the (CPTU) problem with constraint matrix $\hat{M}$, and invoke Lemma 2.20 to solve the original problem.

We have thus shown the following Lemma.

Theorem 2.21. Consider a (CPTU) problem with a constraint matrix $T \in \{0, \pm 1\}^{m \times n}$ as in case 5 or case 6 in Theorem 2.4, where the matrices L and R of the 3-sum have each at least two rows. Then an optimal solution to (CPTU) can be obtained by:

1. Solving up to 14 (CPTU) problems with a constraint matrix with m_1 rows and n_1 columns, and one (CPTU) problem with a constraint matrix wth m_2 rows and n_2 columns, where $m_1 \leqslant m_2 < m$, $m_1 + m_2 \leqslant m + 2$, and $n_1, n_2 \leqslant n + 13$.
2. Further operations taking strongly polynomial time.

2.5 Solving Base Block (CPTU)s

In this section we provide details regarding the proof of Lemma 2.5 by presenting strongly polynomial time algorithms to solve a (CPTU) problem with a constraint matrix T that either falls in case 1 or 2 of Theorem 2.4, i.e., core(T) is either a network matrix, the transpose of a network matrix, or is up to sign changes and permutations of rows and columns one of the two matrices in (2.1). As mentioned previously, core(T) is a network matrix or transpose thereof, if and only if T is a network matrix or transpose thereof, respectively. Due to Lemma 2.13, we can assume that all (CPTU) problems we consider in this section are bounded, i.e., they either have a finite optimum or are infeasible.

In what follows in this section, we start by discussing in Section 2.5.1 how to solve (CPTU) problems with a constraint matrix T being a network matrix. Section 2.5.2 then deals with the case of T being a transpose of a network matrix, and Section 2.5.3 with T being a matrix of constant core.

2.5.1 Solving (CPTU)s with T Being a Network Matrix

Consider a bounded (CPTU) problem

$$\max\{c^\mathsf{T} x : Tx \leqslant 0,\ x \in \mathbb{Z}^A_{\geqslant 0},\ x(S) \text{ odd}\},$$

where T is a network matrix, and we assume to have a representation of T as described in Definition 1.8 in terms of a graph (V, A) and tree (V, U). In particular, the rows of T are indexed by U and its columns by A, i.e., $T \in \{-1, 0, 1\}^{U \times A}$. Consequently, we have $c \in \mathbb{Z}^A$ and $S \subseteq A$. We will first focus on the LP relaxation

$$\max\{c^\mathsf{T} x : Tx \leqslant 0, x \in \mathbb{R}^A_{\geqslant 0}\} \ ,$$

which we rewrite as

$$\min\{-c^\mathsf{T} x : Tx + y = 0, x \in \mathbb{R}^A_{\geqslant 0}, y \in \mathbb{R}^U_{\geqslant 0}\} \ . \tag{2.16}$$

Let $G = (V, A \cup U)$ be the graph on the vertices V containing both arc sets V and U. A key observation is that the linear program (2.16) describes a (flow) circulation problem on G.

Lemma 2.22. Let $x \in \mathbb{R}^A_{\geqslant 0}$, $y \in \mathbb{R}^U_{\geqslant 0}$. Then the following two statements are equivalent:

1. $Tx + y = 0$,
2. (x, y) is a feasible circulation in G (without capacity constraints).

Proof. Consider any arc $u = (v, w) \in U$, and let $W_u \subseteq V$ be all vertices in the connected component of $(V, U \backslash \{u\})$ that contains v. We now consider the entry of Tx that corresponds to u. By (1.10), we have

$$(Tx)_u = x(\delta^+_A(W_u)) - x(\delta^-_A(W_u)) \ ,$$

where $\delta^+_A(W_u)$ and $\delta^-_A(W_u)$ are all arcs of A that are leaving W_u and entering W_u, respectively. Hence, $Tx + y = 0$ is equivalent to

$$x(\delta^+_A(W_u)) - x(\delta^-_A(W_u)) + y_u = 0 \qquad \forall u \in U \ ,$$

which implies that, when interpreting (x, y) as a flow vector in G, there is no net flow crossing the cut W_u, and this holds for all $u \in U$. Because U is a spanning tree in G (when disregarding orientations), this is easily seen to be equivalent to (x, y) being a circulation in G. □

Hence, the considered (CPTU) problem can be rewritten as

$$\min\{-c^\mathsf{T} x : Tx + y = 0,\ x \in \mathbb{Z}^A_{\geqslant 0},\ y \in \mathbb{Z}^U_{\geqslant 0}, x(S) \text{ odd}\} \ , \tag{2.17}$$

and is equivalent to finding an integer circulation in G, minimizing $-c^\mathsf{T} x$, and satisfying $x(S)$ odd. Let $\ell \in \mathbb{Z}^{A \cup U}$ be the vector obtained by extending $-c$ with a zero-vector of dimension $|U|$, such that we have $\ell^\mathsf{T} \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] = -c^\mathsf{T} x$. We continue with the following observation.

Lemma 2.23. If (2.17) is feasible, then any circuit $C \subseteq A \cup U$ in the graph G satisfies $\ell(C) \geqslant 0$.

Proof. If there was a circuit $C \in A \cup U$ with $\ell(C) < 0$, then one could strictly decrease the value of any feasible solution to (2.17) by adding $2\chi^C$ to it. This would imply unboundedness of (2.17), and therefore also unboundedness of the (CPTU) problem we started with, which is a contradiction with the assumption of (CPTU) being bounded.

□

Lemma 2.24. If (2.17) is feasible, then there is an optimal solution $(x^*, y^*) \in \mathbb{Z}^{A \cup U}$ to (2.17) that corresponds to a single circuit, i.e., there is a circuit $C \subseteq A \cup U$ in G such that $(x^*, y^*) = \chi^C$, where χ^C is the characteristic vector of C.

Proof. We start with an arbitrary optimal solution $\widetilde{z}^\mathsf{T} = (\widetilde{x}^\mathsf{T}, \widetilde{y}^\mathsf{T})$ to (2.17) which, by Lemma 2.22, corresponds to a circulation in G, and can thus be written as

$$\widetilde{z} = \sum_{i=1}^{k} \chi^{C_i} \ ,$$

where C_i for $i \in [k]$ are circuits in G, and we allow circuits to appear multiple times. There must be at least one circuit $j \in [k]$ such that $|C_j \cap S|$ is odd for $\widetilde{z}$ to satisfy that $\widetilde{z}(S)$ is odd. Hence, $z^* = \binom{x^*}{y^*} = \chi^{C_j}$ is feasible to (2.17), and moreover, because all circuits have non-negative ℓ-length by Lemma 2.23, z^* has objective value $\ell^\mathsf{T} z^* \leqslant \ell^\mathsf{T}\widetilde{z}$, which shows optimality of z^* and completes the proof. □

By Lemma 2.24, problem (2.17) thus reduces to the following combinatorial optimization problem

$$\min\{\ell(C) : C \subseteq A \cup U \text{ is a directed circuit in } G \text{ with } |C \cap S| \text{ odd}\} \ . \quad (2.18)$$

We complete our discussion of how to solve a (CPTU) problem whose constraint matrix is a network matrix with the following lemma.

Lemma 2.25. There is a strongly polynomial algorithm to solve (2.18).

Proof. We construct an auxiliary graph $G' := (V \cup V', A')$, where V' is a copy of the vertex set V. For $v \in V$, denote by v' its duplicate in V'. We define A' in a way such that only the arcs in S connect V and V'. To this end, let $A' := A_1 \cup A_2 \cup A_3$, where

$$\begin{aligned} A_1 &:= \{(v,w) \mid (v,w) \in (A \cup U)\backslash S\}, \\ A_2 &:= \{(v',w') \mid (v,w) \in (A \cup U)\backslash S\}, \\ A_3 &:= \{(v,w') \cup (v',w) \mid (v,w) \in S\} \ . \end{aligned}$$

Moreover, we define lengths $\ell' : A' \to \mathbb{Z}$ on the arcs as follows: The length of any arc (v,w), (v',w), (v,w') or (v',w') is set to $\ell((v,w))$. One can easily observe that for any $v \in V$, we have that v–v' walks in G' correspond one-to-one to closed walks in G containing v and using the arcs in S an odd number of times. Hence, the two duplicates V and V' of the original vertex set can be interpreted as representing even and odd S-parities, respectively.

Observe that the edge lengths ℓ' in G' are **conservative**, i.e., any circuit in G' has non-negative length. This follows from the observation that any circuit in G' corresponds to a closed walk in G, which is a disjoint union of circuits in G, all of which have non-negative length by Lemma 2.23. We can thus apply a strongly polynomial shortest path algorithm for conservative lengths to G', like the Floyd-Warshall algorithm, or the Moore-Bellman-Ford Algorithm (see [31] for an excellent exposition of these algorithms).

To solve (2.18), we compute for every $v \in V$ a shortest v–v' path in G. Each such path corresponds to a closed walk in G, which can be decomposed into circuits, at least one of which uses the arcs of S an odd number of times. Let C_v be such a circuit. Among all C_v for $v \in V$, we return the one with the smallest length $\ell(C_v)$. This will indeed solve problem (2.18) because the optimal circuit in G is a candidate solution to one of the shortest path problems that we solve in G'. □

2.5.2 Solving (CPTU)s with T Being the Transpose of a Network Matrix

We now consider a bounded (CPTU) problem

$$\max\{c^\mathsf{T} x : Tx \leqslant 0,\ x \in \mathbb{Z}^U_{\geqslant 0},\ x(S) \text{ odd}\}, \tag{2.19}$$

where T is the transpose of a network matrix. As before we assume to have a representation of T^T as described in Definition 1.8 in terms of a graph (V, A) and tree (V, U). Hence, the rows of T are indexed by A and its columns by U, i.e., $T \in \{-1, 0, 1\}^{A \times U}$, $c \in \mathbb{Z}^U$, and $S \subseteq U$. This case is the most involved one among the base cases. We present a solution relying on submodular function minimization subject to parity constraints.

Consider a well-structured optimal solution x^* for this (CPTU) problem, as claimed by Lemma 2.14. Again, since every row of the identity matrix can be appended to T without destroying TU-ness, and because x^* must be non-negative, we have that x^* is binary. We can thus focus on solutions of (2.19) that can be represented as arc-sets. In particular, we denote by $X^* \subseteq U$ the arc-set corresponding to a well-structured optimal solution x^*, i.e., $x^* = \chi^{X^*}$, and we also call $X^* \subseteq U$ a well-structured optimal solution to (2.19).

Lemma 2.26. Let X^* be a well-structured solution. Then there are no two arcs $u_1 = (v_1, w_1), u_2 = (v_2, w_2) \in X^*$ such that the unique (undirected) v_1–w_2 path $P \subseteq U$ in (V, U) contains both u_1, u_2 but no other arc of X^*.

Proof. We will show that if such arcs u_1, u_2 exist, then X^* cannot be a well-structured solution. To this end we define a new constraint that can be added to T without violating TU-ness. For this consider the way how T would expand into a bigger transpose of a network matrix if we added a new arc $a = (v_1, w_2)$ to the set A. The new constraint that corresponds to a has a coefficient of 1 in both entries corresponding to the columns of u_1 and u_2, and it has 0-coefficients for all other entries of X^*. Hence, the left-hand side value of this constraint with respect to X^* (or more precisely, χ^{X^*}) is 2, which shows that X^* cannot be well-structured. □

Lemma 2.27. Let $Q \subseteq V$ such that $\delta^+_U(Q) = \varnothing$, $\delta^-_A(Q) = \varnothing$, and $|\delta^-_U(Q) \cap S|$ is odd. Then χ^X, where $X = \delta^-_U(Q)$, corresponds to a feasible solution to (CPTU).

Proof. The parity constraint is clearly satisfied. It remains to check the constraints corresponding to arcs $a = (v, w) \in A$. Let $P \subseteq U$ be the unique v–w path in (V, U). If both $v, w \in Q$ or both $v, w \in V \backslash Q$, then P leaves Q the same number of times as it enters it. Due to $\delta_U^+(Q) = \emptyset$, whenever a path enters Q, it uses one of the arcs of X forwardly, and whenever it leaves Q, it uses one arc of X backwardly. Hence, the left-hand side of the constraint corresponding to a with respect to the solution X is 0, and the constraint is thus satisfied. Moreover, if $v \in Q$ and $w \notin Q$ then the same argument holds with the difference that P leaves Q once more than it enters Q. Thus the constraint corresponding to a is feasible with a slack of 1. Finally, $v \notin Q$ and $w \in Q$ is not possible since $\delta_A^-(Q) = \emptyset$. □

The next lemma shows that there is also an **optimal solution** to (2.19) that satisfies the conditions of Lemma 2.27. For any set of vertices $Q \subseteq V$, we use the notation

$$U[Q] := \{u \in U : u \text{ has both endpoints in } Q\}.$$

Lemma 2.28. If (2.19) is feasible, then any well-structured solution X^* is of the form $X^* = \delta_U^-(Q)$ for some $Q \subseteq V$ with $\delta_U^+(Q) = \emptyset$, and $\delta_A^-(Q) = \emptyset$.

Proof. Observe first that any arc set $F \subseteq U$ that is not of the form $F = \delta_U^-(Q)$ for some $Q \subseteq V$ has two arcs u_1, u_2 as stated in Lemma 2.26, which is forbidden. Hence, there is a vertex-set $Q' \subseteq V$ such that $X^* = \delta_U^-(Q')$. We show how Q' can be modified to obtain a set Q with the desired properties. Consider the connected components of $(Q', U[Q'])$, where $U[Q']$ are all arcs of U with both endpoints in Q', and let $Q'_1, \ldots, Q'_k \subseteq V$ be the corresponding vertex sets. We have

$$\delta_U^-(Q') = \bigcup_{i=1}^{k} \delta_U^-(Q'_i) \ .$$

Notice that we can assume $\delta_U^-(Q'_i) \neq \emptyset$ for $i \in [k]$; for otherwise we can replace Q' by $Q' \backslash Q'_i$. Now assume that there is an arc $u = (w_1, w_2) \in \delta_U^+(Q'_i)$ for some $i \in [k]$. Consider the vertex set $W_u \subseteq V \backslash Q'$ of all vertices that are in the same connected component as w_2 in the graph $(V \backslash Q', U[V \backslash Q'])$. Observe that $\delta_U^+(W_u) = \emptyset$; for otherwise any arc $a_1 \in \delta_U^+(W_u)$, which must satisfy $a_1 \in \delta_U^-(Q')$ due to our definition of W_u, forms together with any arc $a_2 \in \delta_U^-(Q_i)$ a configuration that is forbidden by Lemma 2.26. Hence, we can replace Q' by $Q' \cup W_u$ to obtain a bigger set $Q'' \subseteq V$ satisfying $\delta_U^-(Q'') = \delta_U^-(Q')$ and $u \notin \delta_U^+(Q)$. We can now iterate this approach to eliminate any arc of U that is leaving Q'' until we end up with a set $Q \subseteq V$ satisfying $\delta_U^-(Q) = \delta_U^-(Q') = X^*$ and $\delta_U^+(Q) = \emptyset$. We claim that this set also satisfies $\delta_A^-(Q) = \emptyset$. Indeed, any arc $a = (w_1, w_2) \in \delta_A^-(Q)$ would be a violated constraint for X^*, because the w_1–w_2 path $P \subseteq U$ in (V, U) would enter Q once more than it leaves it. This completes the proof. □

Hence, Lemma 2.28 shows that it suffices to restrict ourselves to (binary) solutions to (2.19) that are characteristic vectors of sets $\delta_U^-(Q)$ where $Q \subseteq V$ satisfies $\delta_U^+(Q) = \varnothing$ and $\delta_A^-(Q) = \varnothing$.

Our goal is to find an optimal solution to (2.19) by optimizing over such cuts Q. For this we map the parity constraint, which is defined on a set $S \subseteq U$, to the vertices V. More precisely, let $K \subseteq V$ be defined as follows. For any vertex $v \in V$ we have

$$v \in K \iff |(\delta_U(v) \cap S)| \equiv 1 \pmod 2 \ ,$$

where $\delta_U(v) = \delta_U^+(v) \cup \delta_U^-(v)$. The set K indeed allows for mapping the parity constraint to the vertices as the following lemma shows.

Lemma 2.29. Let $Q \subseteq V$ with $\delta_U^+(Q) = \varnothing$. Then the following two statements are equivalent:

1. $|\delta_U^-(Q) \cap S| \equiv 1 \pmod 2$, and
2. $|Q \cap K| \equiv 1 \pmod 2$.

Proof. Because $\delta_U^+(Q) = \varnothing$ we have $\delta_U^-(Q) = \delta_U(Q)$. And hence,

$$|\delta_U^-(Q) \cap S| \equiv |\delta_U(Q) \cap S| \equiv \sum_{q \in Q} |\delta_U(q) \cap S| \equiv |Q \cap K| \pmod 2,$$

where the second equality holds because each arc $u \in S$ with both endpoints in Q is counted twice in the above sum (thus having no impact mod 2), and each $u \in \delta_U(Q) \cap S$ is counted once, as desired. Moreover, the third equality follows from the definition of K. ☐

Summarizing the above discussion we have the following.

Lemma 2.30. Assume that (2.19) is feasible, and let $Q \subseteq V$ be a set that maximizes $c(\delta_U^-(Q))$ subject to

1. $\delta_U^+(Q) = \varnothing$,
2. $|Q \cap K| \equiv 1 \pmod 2$, and
3. $\delta_A^-(Q) = \varnothing$.

Then $\chi^{\delta_U^-(Q)}$ is an optimal solution to (2.19).

Proof. Lemma 2.27 and Lemma 2.29 show that any set Q with the conditions stated in Lemma 2.30 leads to a feasible solution $X = \delta_U^-(Q)$ to (2.19). Finally, Lemma 2.28 guarantees that there is an optimal solution that corresponds to a set Q satisfying the conditions of the lemma. ☐

Hence, it remains to solve the following problem, where we inverted the objective to obtain a minimization problem for convenience.

$$\begin{aligned} \min\{-c(\delta_U^-(Q))) : Q \subseteq V,\ \delta_U^+(Q) = \varnothing, \\ \delta_A^-(Q) = \varnothing,\ |Q \cap K| \equiv 1 \pmod 2\} \end{aligned} \tag{2.20}$$

Notice that the above objective function, when restricted to sets Q with $\delta^+_U(Q) = \varnothing$, is modular due to

$$-c(\delta^-_U(Q)) = c(\delta^+_U(Q)) - c(\delta^-_U(Q)) = \sum_{q\in Q} \big(c(\delta^+(q)) - c(\delta^-(q))\big) \ .$$

Moreover, we can encode the constraints $\delta^+_U(Q) = \varnothing$ and $\delta^-_A(Q) = \varnothing$ using the objective by introducing submodular penalty functions.Note that, alternatively, one could also keep the objective modular, and optimize over a lattice family which enforces the conditions $\delta^+_U(Q) = \varnothing$ and $\delta^-_A(Q) = \varnothing$. We use submodular penalties mostly for ease of presentation.

To encode $\delta^+_U(Q) = \varnothing$ and $\delta^-_A(Q) = \varnothing$ we choose a large value $M > 0$. In particular, $M = 1 + \sum_{u\in U} |c(u)|$ suffices. We now define two submodular functions $g_1, g_2 : V \to \mathbb{Z}_{\geqslant 0}$ as follows: For $Q \subseteq V$ we set

$$\begin{aligned} g_1(Q) &= M \cdot |\delta^+_U(Q)| \ , \text{ and} \\ g_2(Q) &= M \cdot |\delta^-_A(Q)| \ . \end{aligned}$$

Both g_1 and g_2 are indeed submodular, because they are scaled versions of directed cut functions. Problem (2.20) can now be rephrased as the following submodular function minimization problem with a parity constraint:

$$\begin{aligned} \min \Big\{ \sum_{q\in Q} \big(c(\delta^+(q)) - c(\delta^-(q))\big) + g_1(Q) + g_2(Q) \\ : \ Q \subseteq V, |Q \cap K| \equiv 1 \pmod 2 \Big\} \ . \end{aligned} \tag{2.21}$$

Due to the large constant M, any optimal solution Q to problem (2.21) must satisfy $\delta^+_U(Q) = \varnothing$ and $\delta^-_A(Q) = \varnothing$, except if (2.20) is infeasible. Problem (2.21) is a submodular minimization problem subject to a parity constraint. Efficient algorithms are known to solve such problems (see [26, 21]). In particular, the approach in [21] strongly polynomially reduces the problem to $O(|V|^2)$ many submodular function minimization problems. Hence, by using a strongly polynomial algorithm to minimize submodular functions (see [37, 27, 35]), we can solve (2.21) in strongly polynomial time, and therefore also any (CPTU) problem with a constraint matrix that is the transpose of a network matrix.

We remark that there are alternative ways to solve (2.21). In particular, (2.21) can also be rephrased as a minimum T-cut problem. We chose the description in terms of parity-constrained submodular minimization for clarity and ease of presentation.

2.5.3 Solving (CPTU) with constraint matrix of small core

Lemma 2.31. There exists a strongly polynomial algorithm for solving (CPTU) with constraint matrix T if core(T) is, up to sign changes of rows/columns and row/column permutations, one of the two matrices in (1.11).

Proof. Let us divide the rows and columns of T into two parts, and reorder the rows/columns of T accordingly: Let J be the index of columns which do not originate from $C := \operatorname{core}(T)$ by iteratively copying columns and multiplying columns by -1 in $\operatorname{core}(T)$, and I be the index of those rows which do originate from C. Then we can, after row and column permutations, write T as

$$T = \begin{bmatrix} T^1 & \bar{C} \\ T^2 & T^3 \end{bmatrix},$$

where $T^1 = T_{I,J}$, $T^2 = T_{\bar{I},J}$, $\bar{C} = T_{I,\bar{J}}$, $T^3 = T_{\bar{I},\bar{J}}$, and $\bar{I} := [m]\backslash I$, $\bar{J} := [n]\backslash J$. By Lemma 2.14, if the problem is feasible, there exists an optimal solution $x^* \in \{0,1\}^n$ to the (CPTU) problem we consider because each row of the $(n \times n)$-identity matrix can be appended to T without destroying TU-ness, which implies that we can require $x^* \in \{-1,0,1\}^n$; moreover, x^* must be non-negative.

Let $J_i \subseteq [n]$ be the column indeces of columns in $\bar{C}$ that correspond to column i in $[C,-C]$. Then, since the (5×5)-unit matrix is TU-appendable to C, each row vector ${\chi_{J_i}}^\mathsf{T} \in \{0,1\}^n$ is TU-appendable to T. Thus, it suffices to look for a solution $x^* \in \{0,1\}^n$ with $x^*(J_i) \leqslant 1$. For such x^* there are at most polynomially many possible values for $x^*_{\bar{J}}$. For each of these, we obtain a (PTU) problem in at most n variables with constraint matrix $\left[\begin{smallmatrix} T^1 \\ T^2 \end{smallmatrix}\right]$.

Observe that $\left[\begin{smallmatrix} T^1 \\ T^2 \end{smallmatrix}\right]$ is a network matrix: To see this, replace $\bar{C}$ by any network matrix of the same size as $\bar{C}$, for example the all-zero matrix. T can be generated from $\bar{C}$ by iteratively appending standard unit vectors (or their negatives) as well as all-zero vectors. Appending such vectors to a network matrix yields again a network matrix, which implies that $\left[\begin{smallmatrix} T^1 & 0 \\ T^2 & T^3 \end{smallmatrix}\right]$ is a network matrix. Finally, a sub-matrix of a network matrix is a network matrix, too, so that $\left[\begin{smallmatrix} T^1 \\ T^2 \end{smallmatrix}\right]$ is a network matrix.

By Lemma 2.12, we can strongly polynomially reduce this (PTU)-problem to a (CPTU) problem whose constraint matrix is a sub-matrix of $\left[\begin{smallmatrix} T^1 \\ T^2 \end{smallmatrix}\right]$. Therefore, we have at most $\binom{n}{2}$ many problems of type (CPTU) with a network matrix as their constraint matrix, which we can solve in strongly polynomial time as shown in Section 2.5.1. □

2.6 Proof of Main Theorem

In this section, we provide a formal proof of our main theorem, Theorem 2.1.

Proof of Theorem 2.1. By Lemma 2.3, it suffices to show that (CPTU) can be solved in strongly polynomial time. Hence, consider a (CPTU) problem with constraint matrix $T \in \{-1,0,1\}^{m \times n}$, right-hand side $b \in \mathbb{Z}^m$, objective $c \in \mathbb{Z}^n$, and parity constraint $x(S) \equiv 1 \pmod 2$ for some set $S \subseteq [n]$. Primarily to simplify the analysis, we will first simplify the problem by removing unnecessary columns. More precisely, whenever there are two identical columns in T, say with indices $i, j \in [m]$, such that either both $i, j \in S$ or both $i, j \notin S$, then we can remove the column i if $c_i \leqslant c_j$; otherwise we remove j. Clearly, if $c_i \leqslant c_j$, then column i is **dominated** by column j in the sense that any solution x to

(CPTU) with $x_i > 0$ can be transformed into a new solution z such that $z_i = 0$, $z_j = x_i + x_j$, and $z_\ell = x_\ell$ for $\ell \in [n] \setminus \{i, j\}$ with objective value at least as good as x. Hence, without loss of generality, we can assume that the (CPTU) we start with has no dominated columns. For brevity, we call such a (CPTU) problem **slim**. Slim (CPTU) problems have the property that each column appears at most twice in T, once being part of S and once not being contained in S. This allows us to bound the number of columns n of T in terms of m, its number of rows, by using that any TU matrix on m rows without repeated columns has at most $m^2 + m + 1$ columns [10]. Because T is a TU matrix where each column may appear twice, we have $n \leqslant 2(m^2 + m + 1)$. In what follows, we will therefore bound running times in terms of m only, to show that there is a strongly polynomial algorithm for (CPTU).

We will show by induction on the size of m that, for appropriately chosen constants $C, \gamma \geqslant 1$, one can solve any slim (CPTU) problem with a constraint matrix with m rows in time bounded by $C \cdot m^\gamma$. Theorem 2.6 covers the most important case of our analysis, showing how the problem decomposes into smaller ones when core(T) is neither a network matrix nor a transpose of it, and core(T) is, up to sign changes of rows/columns and row/column permutations, not one of the matrices in (2.1). In this case we talk about a **non-base block** (CPTU); otherwise we talk about a **base-block** (CPTU). Let $\alpha, \beta \geqslant 1$ be constants such that, for a slim non-base block (CPTU), one can in time at most $\alpha \cdot m^\beta$ perform all of the following operations:

1. decompose the (CPTU) problem into at most 15 smaller subproblems according to Theorem 2.6,
2. making those subproblems slim, and
3. transforming optimal solutions of the subproblems to one of the original (CPTU) problem.

Indeed, Theorem 2.6 guarantees that these operations can be performed in strongly polynomial time; hence, such constants $\alpha, \beta \geqslant 1$ exist. We now specify how the constants C, γ are chosen. We want C to be large enough to cover the time to solve several small (constant-size) (CPTU) problems. To define small problems, let

$$\ell = \max\left\{200, \frac{\alpha\beta}{4} + 2, 2^\beta\right\} \quad . \tag{2.22}$$

The constants C and γ are chosen such that

1. One can, in time Cm^γ, solve any slim base block (CPTU) with a constraint matrix with m rows. Lemma 2.5 guarantees that this can be done in strongly polynomial time;
2. One can, in time C, solve 14 (CPTU) problems, each having a constraint matrix with at most ℓ rows;
3. $\gamma \geqslant \max\{4, \beta + 1\}$;
4. $C \geqslant 2\alpha$.

We now start with the inductive proof that any slim (CPTU) problem with a constraint matrix with m rows can be solved in time bounded by Cm^γ. Clearly, if $m \leqslant \ell$ or (CPTU) is a base block problem, then the problem can be solved in time at most Cm^γ, due to 2 and 1, respectively. Hence, from now on we assume to deal with a non-base block (CPTU). By Theorem 2.6, this problem decomposes into at most 14 problems with m_1 rows and one problem with m_2 rows, where m_1, m_2 satisfy $m_1 \leqslant m_2 < m$ and $m_1 + m_2 \leqslant m + 2$. We make sure that these subproblems are slim. As mentioned above, decomposing the problem, making the subproblems slim, as well as all additional operations needed to recover an optimal solution to the original (CPTU) take altogether time at most αm^β. Hence, it remains to bound the time to solve the 15 slim (CPTU) subproblems. We distinguish between $m_1 < \ell$ and $m_1 \geqslant \ell$.

Case $m_1 < \ell$. By 2, we can solve the up to 14 slim (CPTU) problems with constraint matrices with at most m_1 rows in time C. Hence, what is left to show is the following inequality:

$$\alpha m^\beta + C m_2^\gamma + C \leqslant C m^\gamma \ .$$

This holds due to

$$\begin{aligned}
\alpha m^\beta + C m_2^\gamma + C &\leqslant \alpha m^\beta + C(m-1)^\gamma + C \\
&\leqslant \alpha m^\beta + C m^\gamma - C m^{\gamma-1} + C \\
&\leqslant C m^\gamma + \alpha m^\beta - 2\alpha(m^{\gamma-1} - 1) \\
&\leqslant C m^\gamma - \alpha m^{\gamma-1} + 2\alpha \\
&\leqslant C m^\gamma \ ,
\end{aligned}$$

where the third inequality follows by 4, the forth by 3, and the last one by using $m > \ell \geqslant 2$ and $\gamma \geqslant 4$, which holds due to 3.

Case $m_1 \geqslant \ell$. We have to show

$$\alpha m^\beta + C m_2^\gamma + 14 C m_1^\gamma \leqslant C m^\gamma \ .$$

Notice that $m_1 \leqslant \frac{m}{2} + 1$, which follows from $m_1 + m_2 \leqslant m + 2$ and $m_1 \leqslant m_2$.

$$\begin{aligned}
&\alpha m^\beta + C m_2^\gamma + 14 C m_1^\gamma \\
&\quad \leqslant \alpha m^\beta + C(m + 2 - m_1)^\gamma + 14 C m_1^\gamma \\
&\quad \leqslant \alpha m^\beta + C\left(\max\left\{(m+2-\ell)^\gamma + 14\ell^\gamma, 15\left(\frac{m}{2}+1\right)^\gamma\right\}\right) \ ,
\end{aligned}$$

where we use $m_2 \leqslant m + 2 - m_1$ for the first inequality, and the second one follows from the fact that $C(m + 2 - m_1)^\gamma + 14Cm_1^\gamma$ is a convex function in m_1 for $m_1 < m$, and hence, its maximum is achieved for either the smallest or largest possible value of m_1. Since m_1 satisfies $\ell \leqslant m_1 \leqslant \frac{m}{2} + 1$, we bounded the function $C(m + 2 - m_1)^\gamma + 14Cm_1^\gamma$ by the maximum of the two values achieved at $m_1 = \ell$ and $m_1 = \frac{m}{2} + 1$. Thus, it remains to show the following inequalities:

$$\alpha m^\beta + C\left((m+2-\ell)^\gamma + 14\ell^\gamma\right) \leqslant C m^\gamma \ , \text{ and} \tag{2.23}$$

$$\alpha m^\beta + 15C\left(\frac{m}{2}+1\right)^\gamma \leqslant C m^\gamma \ . \tag{2.24}$$

To show (2.23), observe that $g(m) := Cm^\gamma - C\left((m+2-\ell)^\gamma + 14\ell^\gamma\right) - \alpha m^\beta$ is monotone increasing in m. Indeed, the derivative $g'(m)$ of $g(m)$ satisfies

$$\begin{aligned}
g'(m) &= C\gamma\left(m^{\gamma-1} - (m+2-\ell)^{\gamma-1}\right) - \alpha\beta m^{\beta-1} \\
&\geqslant C\gamma m^{\gamma-2}(\ell-2) - \alpha\beta m^{\beta-1} \\
&\geqslant m^{\beta-1}\left(4C(\ell-2) - \alpha\beta\right) \\
&\geqslant 0 \ ,
\end{aligned}$$

where the first inequality follows by using $(m+2-\ell)^{\gamma-1} \leqslant m^{\gamma-2}(m+2-\ell)$, the second one by $\gamma \geqslant \max\{4, \beta+1\}$ due to 3, and the last one by $C \geqslant 1$ and $\ell \geqslant \frac{\alpha\beta}{4} + 2$.

Thus, it suffices to show (2.23) for the smallest possible value of m. Notice that $m \geqslant m_1 + m_2 - 2 \geqslant 2\ell - 2$. Hence, (2.23) reduces to

$$\alpha(2\ell-2)^\beta + 15C\ell^\gamma \leqslant C(2\ell-2)^\gamma \ ,$$

and the above inequality holds due to

$$\begin{aligned}
C(2\ell-2)^\gamma &= C2^\gamma\left(\frac{\ell-1}{\ell}\right)^\gamma \ell^\gamma \\
&\geqslant 15.5 \cdot C\ell^\gamma && \text{(using } \ell \geqslant 200 \text{ and } \gamma \geqslant 4) \\
&\geqslant 15C\ell^\gamma + \alpha\ell^\gamma && (C \geqslant 2\alpha \text{ by } 4) \\
&\geqslant 15C\ell^\gamma + \alpha(2\ell)^\beta && (\ell \geqslant 2^\beta, \text{ and } \gamma \geqslant \beta+1 \text{ by } 3) \\
&\geqslant 15C\ell^\gamma + \alpha(2\ell-2)^\beta \ .
\end{aligned}$$

Finally, it remains to show (2.24), which follows from

$$\begin{aligned}
&\alpha m^\beta + 15C\left(\frac{m}{2}+1\right)^\gamma \\
&\quad\leqslant \alpha m^\beta + 15Cm^\gamma 2^{-\gamma}\left(\frac{\ell+2}{\ell}\right)^\gamma && \text{(using } m \geqslant \ell) \\
&\quad\leqslant \alpha m^\beta + \frac{15}{16}Cm^\gamma\left(\frac{\ell+2}{\ell}\right)^4 && (\gamma \geqslant 4 \text{ by } 3) \\
&\quad\leqslant \alpha m^\beta + 0.98Cm^\gamma && (\ell \geqslant 200) \\
&\quad\leqslant Cm^\gamma + \left(\alpha - \frac{Cm}{50}\right)m^\beta && (\gamma \geqslant \beta+1 \text{ by } 3) \\
&\quad\leqslant Cm^\gamma \ ,
\end{aligned}$$

where the least inequality follows from $m \geqslant \ell \geqslant 200$ and $C \geqslant 2\alpha$ due to 4. □

Chapter 3

Strictly 3-modular base block ILPs

The goal of this chapter is to contribute to extending parts of the theory for $\Delta \leqslant 2$ to higher values of Δ. In Chapter 2 we showed that solving the bimodular case essentially boils down to solving a TU-problem with one parity constraint. One of the most straight-forward extensions is to consider TU-systems with one congruence-3-constraint, where we focus on feasibility and flatness results.

Congruence-3 TU-feasibility: Given $T \in \mathbb{Z}^{m\times n}$ totally unimodular with $\operatorname{rank}(T) = n$, $b \in \mathbb{Z}^m$, $c \in \{0,1,2\}^n$ with $c \not\equiv 0 \pmod 3$, $\gamma \in \{0,1,2\}$, and $P = \{x \in \mathbb{R}^n : Ax \leqslant b\}$,

$$\text{find } x \in P \cap \mathbb{Z}^n \text{ s.t. } c^\mathsf{T} x \equiv \gamma \pmod 3, \tag{C3TU}$$

or decide that no such x exists.

Denote by (CkTU) the generalization of (C3TU) for optimizing, and for general prime integers $k \in \mathbb{N}$:

Congruence-k TU-optimization: Given $T \in \mathbb{Z}^{m\times n}$ totally unimodular with $\operatorname{rank}(T) = n$, $b \in \mathbb{Z}^m$, $k \in \mathbb{N}$ prime, $c \in \{0,\ldots,k-1\}^n$ with $c \not\equiv 0 \pmod k$, $\gamma \in \{0,\ldots,k-1\}$, $d \in \mathbb{Z}^n$, and $P = \{x \in \mathbb{R}^n_{\geqslant 0} : Tx \leqslant b\}$, find x maximizing

$$\max\{d^\mathsf{T} x : Tx \leqslant b, x \in \mathbb{Z}^n, c^\mathsf{T} x \equiv \gamma \pmod k\} \tag{CkTU}$$

or decide that no such x exists.

Not surprisingly, solving ILPs with strictly k-modular constraint matrix can be reduced to this setting, as the following lemma shows.

Lemma 3.1. Consider an ILP of the form (1.3), and assume that all $(n\times n)$-

submatrices of the constraint matrix A have a determinant in $\{0, \pm k\}$, for $k \in \mathbb{N}$ prime. Then, (1.3) can be reduced to (CkTU).

Proof. The proof goes along similar lines as the proof of the equivalence of (BIP) and (PTU). The constraint matrix of P has full column rank, meaning that P, if it is non-empty, has a vertex v. W.l.o.g., assume that $v \notin \mathbb{Z}^n$, that $A_{1:n,\cdot}$ is invertible and that $A_{1:n,\cdot}v = b_{1:n}$. Via column operations, we can bring $A_{1:n,\cdot}$ in Hermite Normal Form H, which can be done in polynomial time [18]. Call the corresponding unimodular matrix $U \in \mathbb{Z}^{n\times n}$ and let $A' := AU$.

Next, we consider $T = A'H^{-1}$. Note that $T_{1:n,\cdot}$ is the identity matrix, and that since $v \notin \mathbb{Z}^n$, $\det(H) = k$ and thus all $(n \times n)$-subdeterminants of T are in $\{0, \pm 1\}$. As H is in HNF, there is a unique entry $H_{j,j}$ on its diagonal which is not 1. Then $H_{j,j} = k$ and H^{-1} is equal to the identity matrix, apart from row j which is equal to

$$(H^{-1})_{j,l} = \begin{cases} -\frac{1}{k}H_{j,l}, & \text{if } j \neq l, \\ \frac{1}{k}, & \text{otherwise} \end{cases}.$$

Then, $H^{-1}y \in \mathbb{Z}^n \Leftrightarrow \tilde{H}y \equiv 0 \pmod 3$, where $\tilde{H} = -k \cdot (H^{-1})_{j,\cdot}$.

We can now write (1.3) as

$$\begin{aligned}
&\max\{c^\mathsf{T}x\colon Ax \leqslant b, x \in \mathbb{Z}^n\} \\
=\ &\max\{c^\mathsf{T}x\colon AUH^{-1}HU^{-1}x \leqslant b, x \in \mathbb{Z}^n\} \\
=\ &\max\{c^\mathsf{T}UH^{-1}x\colon Tx \leqslant b, UH^{-1}x \in \mathbb{Z}^n\} \\
=\ &\max\{c^\mathsf{T}UH^{-1}x\colon Tx \leqslant b, H^{-1}x \in \mathbb{Z}^n\} \\
=\ &\max\{c^\mathsf{T}UH^{-1}x\colon Tx \leqslant b, \tilde{H}x \equiv 0 \pmod 3\}.
\end{aligned}$$

□

We will, in the rest of this chapter, focus on the case where T in (C3TU) is a matrix corresponding to one of the base blocks in Seymour's TU-decomposition. Recall that if T is a network matrix, it comes with a directed graph representation $G = (V, E)$, with a subset $U \subseteq E$ of the edges forming a spanning tree. If we denote the complement as $A = E \backslash U$, then the rows of T correspond to the edges in U, while the columns correspond to the edges in A. Put differently, $T \in \mathbb{Z}^{U\times A}$.

Definition 3.2. If T in (C3TU) is a network matrix, a transposed network matrix or, up to row/column sign changes/permutations, one of the matrices in (1.11), we refer to the problem as a **base block (C3TU) problem**.

In Section 3.1, we show that if such a problem is infeasible, one of the constraints defines a flat direction of width at most 1. In Section 3.2, we show that for a TU system whose constraints are defined by a transposed network matrix or one of the constant-size matrices, we can solve (C3TU).

The following proximity theorem will be useful for us later on.

Lemma 3.3 (Proximity)**.** Le x be a vertex of P in (C3TU). Then if (C3TU) has a solution y, there exists a solution y^* to (C3TU) s.t. $\|x - y^*\|_\infty \leqslant 2$.

Proof. Let $I := \{i \in [m] \mid T_{i,\cdot}x \geqslant T_{i,\cdot}y\}$ and $J := [m]\backslash I$. Consider the cone $C := \{x \in \mathbb{R}^n \mid T_{I,\cdot}x \leqslant 0, T_{J,\cdot}x \geqslant 0\}$, so that $y - x \in C$. Assume that $\|x - y\|_\infty > 2$. By Lemma 1.4, we can write $y - x$ as the sum of integral extreme rays $r^1, \dots, r^k$ of C with $\|r^i\|_\infty \leqslant 1$ $\forall i \in [k]$. Note that since the extreme rays are orthant-compatible, $\forall S \subseteq [k] : x + \sum_{i \in S} r^i \in P$. If there exists an i such that $c^\mathsf{T} r^i \equiv \gamma$, let $y^* := x + r^i$, and note that $x + r^i$ is a solution to (C3TU). Otherwise, pick r^i and r^j, $1 \leqslant i, j \leqslant m$ with $c^\mathsf{T} r^i \equiv c^\mathsf{T} r^j \equiv -\gamma$ and let $y^* := x + r^i + r^j$. ☐

3.1 Flat direction in infeasible base block (C3TU) problems

The goal of this section is to develop the following flatness result. It states that if a base block (C3TU) problem is infeasible, the underlying polyhedron P has a flat direction of width at most 1 which is defined by one of the rows in T.

Theorem 3.4. Consider a base block (C3TU) problem. If it is infeasible, then

$$\exists i \in \{1, \dots, m\} : P \subseteq \{x \in \mathbb{R}^n \colon b_i - 1 \leqslant T_{i,\cdot}x \leqslant b_i\}.$$

Theorem 3.4 immediately gives us a polynomial-time algorithm to find a flat direction when (C3TU) is infeasible: For each constraint $T_{i,\cdot}x \leqslant b_i$, $1 \leqslant i \leqslant m$, solve two LPs $\max\{T_{i,\cdot}x \colon x \in P\}$ and $\min\{T_{i,\cdot}x \colon x \in P\}$, which can be done in polynomial time [29]. For some i, the two corresponding LPs will have an optimal value differing by at most one, which means that $T_{i,\cdot}$ defines the flat direction.

To simplify the setting when dealing with (transposed) network matrices, we will assume that if $P \neq \varnothing$, then $0 \in P$ is a vertex. We can do so without loss of generality, as the following lemma shows.

Definition 3.5. We call a (C3TU) problem $(\mathcal{A})$ with underlying polyhedron $P_{(\mathcal{A})} = \{x \in \mathbb{R}^n \colon T_{(\mathcal{A})}x \leqslant b_{\mathcal{A}}\}$ an (sC3TU) problem if

- there are no duplicate rows in $T_{(\mathcal{A})}$,
- all points in $P_{(\mathcal{A})}$ are non-negative, and
- 0 is a vertex of $P_{(\mathcal{A})}$.

We call it a base block (sC3TU) problem if in addition the constraint matrix is a base block in Seymour's TU decomposition.

To shorten notation, we introduce the following definition: We call a point $x \in P \cap \mathbb{Z}^n$ congruence-3-feasible or (c, γ)-feasible if it is feasible for (sC3TU).

Lemma 3.6. Let $(\mathcal{A})$ be a (C3TU) problem with underlying polyhedron $P_{(\mathcal{A})} = \{x \in \mathbb{R}^n : T_{(\mathcal{A})}x \leqslant b_{\mathcal{A}}\}$ and whose constraint matrix $T_{(\mathcal{A})}$ is a network matrix (resp. transposed network matrix).

Then there is a base block (sC3TU) problem $(\mathcal{B})$ with underlying polyhedron $P_{(\mathcal{B})} = \{x \in \mathbb{R}^n : T_{(\mathcal{B})}x \leqslant b_{\mathcal{B}}\}$ such that

- $T_{(\mathcal{B})}$ is a network matrix (resp. transposed network matrix) as well, has the same number of columns and at most as many rows as $T_{(\mathcal{A})}$,
- $(\mathcal{A})$ is feasible if and only if $(\mathcal{B})$ is,
- and both polyhedra $P_{(\mathcal{A})}$ and $P_{(\mathcal{B})}$ have the same width $w(P_{(\mathcal{A})}) = w(P_{(\mathcal{B})})$.

Proof. Since T has full column rank, if P is non-empty, it has a vertex q. Consider an invertible sub-matrix T' of T corresponding to tight constraints at q. Then we can apply a unimodular transformation to T that corresponds to column operations which turn T' into the $(n \times n)$-identity matrix, and adjust c accordingly. Finally, we translate q to 0 by adapting b and γ.

Note that the resulting constraint matrix S is a (transposed) network matrix as well. For transposed network matrices, the statement follows from the proof of statement (36) in Example 4, Chapter 19.3 in [36]. For network matrices, this follows from the fact that the pivot of a network matrix is a network matrix again. □

3.1.1 Network matrix

We show that (sC3TU) can be reduced to the combinatorial problem of finding a feasible circulation of given congruence in a certain weighted digraph with capacities. We begin by defining this graph.

Definition 3.7. Let $(\mathcal{A})$ be an (sC3TU) problem whose constraint matrix T is a network matrix represented by a directed graph $G = (V, E)$ with spanning tree $U \subseteq E$ and $A = E \backslash U$. For an edge $e = (u, v) \in E$, denote by $-e = (v, u)$ its inverse arc. For a set $S \subseteq E$, denote by $-S$ the set of reversed edges $-S := \{-e : e \in S\}$.

- We define the graph

$$\operatorname{rev}((\mathcal{A})) := (V, E \cup -U),$$

with capacities $\bar{b} \in \mathbb{R}_{\geqslant 0}$ and weights $\bar{c} \in \mathbb{R}_{\geqslant 0}$ for the edges, as follows:

$$\bar{c}_a = \begin{cases} 0, & \text{if } a \in U \cup -U, \\ c_a, & \text{if } a \in A \end{cases}, \quad \bar{b}_a = \begin{cases} b_{-a}, & \text{if } a \in -U, \\ \infty, & \text{otherwise} \end{cases}.$$

- We call a circulation C in rev $((\mathcal{A}))$ **simplified** if for each pair (v, w), $(w, v) \in E \cup -U$, $C((v, w)) = 0$ or $C((w, v)) = 0$.

Lemma 3.8. Let $(\mathcal{A})$ be an (sC3TU) problem with T a network matrix. Then:

1. There is a bijection M which maps points in $P \cap \mathbb{Z}^n$ onto simplified integral circulations in rev$((\mathcal{A}))$ while preserving congruence, i.e., for all $x \in P \cap \mathbb{Z}^n$, $c^\mathsf{T} x \equiv \bar{c}(M(x)) \pmod 3$.
2. M maps extreme rays (of the supporting cone at 0) onto simplified cycles.
3. $(\mathcal{A})$ can be reduced to the following combinatorial problem: Given rev$((\mathcal{A}))$ and $\gamma \in \{0, 1, 2\}$, find a feasible circulation C such that $\bar{c}(C) \equiv \gamma \pmod 3$.
4. If $(\mathcal{A})$ is infeasible, there is a constraint in $Tx \leqslant b$ of width at most 1.

At the first glimpse, it might be surprising that we go in this direction, as a very similar problem is known to be NP-hard [3]: Given a graph G and $\gamma \in \{1, 2\}$, is there a cycle of length $\gamma \pmod 3$? Note, however, that we ask for any kind of circulation of congruence γ here.

To simplify notation, in what follows we consider a cycle or a circulation to be a vector $C \in \mathbb{R}^E_{\geqslant 0}$, rather than a subset of E.

Proof of Lemma 3.8. 1. We perform the proof by adapting proofs and statements that appeared in Section 2.5.1. To shorten notation, we write $\bar{G} := \text{rev}((\mathcal{A}))$.

Let $x \in P$ be given, and consider $U^+ := \{u \in U : T_u x > 0\}$ and $U^- := \{u \in U : T_u x \leqslant 0\}$. Note that

$$\begin{bmatrix} T_{U^-} \\ -T_{U^+} \end{bmatrix} x \leqslant 0,$$

that $\hat{T} := \begin{bmatrix} T_{U^-} \\ -T_{U^+} \end{bmatrix}$ is a network matrix again, and that the graph $\hat{G}$ representing $\hat{T}$ can be obtained from G by replacing all edges in U^+ by their inverse arcs, i.e., by replacing U^+ by $-U^+$. Let $\hat{z} := -\hat{T}x \geqslant 0$, then by Lemma 2.22, $(x, \hat{z})$ is a circulation in $\hat{G}$. Furthermore, for all $u \in U^+$, $\hat{z}_u \leqslant b_u$. Since the edge set of $\bar{G}$ is a superset of the edge set of $\hat{G}$, we can, via the inclusion map, interpret $(x, \hat{z})$ as a simplified circulation $(x, y, z) \in \mathbb{R}^A_{\geqslant 0} \times \mathbb{R}^U_{\geqslant 0} \times \mathbb{R}^{-U}_{\geqslant 0}$ in $\bar{G}$. More precisely, (x, y, z) is given by the equations

$$\begin{aligned} z_{(-U^+)} &= \hat{z} \\ Tx + y - z &= 0 \\ \forall u \in U : \min\{y_u, z_{-u}\} &= 0. \end{aligned} \tag{3.1}$$

Since $\forall(-u) \in -U^+ : z_{-u} \leqslant b_u$, this circulation fulfills the capacity constraints, and since the congruence of (x, y, z) only depends on x, $\bar{c}((x, y, z)) = c^\mathsf{T} x$.

This defines an injective map. To see that it is also surjective, let (x, y, z) be any simplified circulation in $\mathbb{Z}^A_{\geqslant 0} \times \mathbb{Z}^U_{\geqslant 0} \times \mathbb{Z}^{-U}_{\geqslant 0}$. Since (x, y, z) is simplified, $\operatorname{supp}(y) \cap \operatorname{supp}(z) = \varnothing$, and thus is the unique solution to the equations in (3.1). Furthermore, $z \leqslant b$, which implies $Tx \leqslant b$. Thus indeed, $(x, y, z) = M(x)$.

2. Let us describe how M maps extreme rays onto cycles in $\bar{G}$.

   Denote by $I := \{u \in U : b_u = 0\}$ the index set of those constraints that are tight at the vertex 0, and let G^I be the graph resulting from G when contracting all arcs in $U \backslash I$. Furthermore, denote by $C := \{x \in \mathbb{R}^n : T_{I,\cdot} \leqslant 0\}$ the supporting cone at the vertex 0. Let $r \in \{0, 1\}^n$ be an extreme ray of C. $M(r) = (r, y_r, z_r)$ is a circulation and since r is well structured, $M(r)$ is a circuit. It is therefore a sum of edge-disjoint cycles. Assume that there was more than one cycle involved, i.e., that we can decompose (r, y_r, z_r) into k many cycles $(r^1, y^1_r, z^1_r), \ldots, (r^k, y^k_r, z^k_r)$. But then, $r^1, \ldots, r^k$ would all be in C and $r = \sum_{i=1}^k r^i$, contradicting the fact that r is an extreme ray. We conclude that $M(r)$ is a simplified cycle in $\bar{G}$.

3. We have seen that there is a 1-on-1 correspondence between solutions to (sC3TU) and simplified circulations in $\bar{G}$. The statement follows from the fact that we can modify any circulation in $\bar{G}$ to be simplified by subtracting cycles of the form $\chi^e + \chi^{-e}$, for some $e \in U$, where necessary, and that this operation does not change the weight of the circulation.

4. To shorten our exposition in the rest of the proof, whenever we speak of cycles in $\bar{G}$, we mean simplified cycles. Furthermore, we define for two cycles C_1 and C_2, $C_1 \cap C_2$ as the vector $\chi^{\operatorname{supp}(C_1) \cap \operatorname{supp}(C_2)}$ in $\mathbb{R}^{(A \cup -E)}_{\geqslant 0}$, as well as $C_1 \cup C_2$ and $C_1 \backslash C_2$ accordingly.

   For a vector $x \in \mathbb{R}^{A \cup U \cup -U}$, denote by $\beta(x) := \{e \in \operatorname{supp}(x) \cap -U : \bar{b}_e = 1\}$. Given two vertices v, w of C, let $C_{v,w}$ denote the path from v to w along C. Assume that no constraint as in the claim exists and that (sC3TU) is infeasible. By Lemma 1.7, there is an integer point in P of non-zero congruence. By Theorem 1.4, this point can be decomposed into extreme rays, one of which is of non-zero congruence as well. Due to what we have proven above, this implies that there is a cycle in $\bar{G}$ of non-zero congruence.

   Let C and D be cycles in $\bar{G}$ (possibly $C = D$) such that $\bar{c}(C)$, $\bar{c}(D) \not\equiv 0 \pmod 3$ and $|\beta(C \cap D)|$ is minimal, cmp. Fig. 3.1. Since $C + D$ is infeasible, there exists $e = (v, w) \in \beta(C) \cap \beta(D)$. As no constraint has width 1, there has to be a point $x^* \in P$ with $T_{e,\cdot} x^* = -1$. This implies that in $\bar{G}$, there is another simplified cycle F with $F(-e) = 1$.

   (i) Assume that $F \cap (C \cup D) = \varnothing$. Then $\bar{c}(F) \not\equiv -\gamma \pmod 3$, since otherwise $F_A + C_A$ is (c, γ)-feasible. Thus, the cycle $(C \cup F) \backslash (\chi^e + \chi^{-e})$ fulfills $\bar{c}((C \cup F) \backslash (\chi^e + \chi^{-e})) \equiv -\gamma \pmod 3$. Then, however, $|\beta((C \cup F) \backslash (\chi^e + \chi^{-e}) \cap D)| < |\beta(C \cap D)|$, contrary to our choice of C and D.

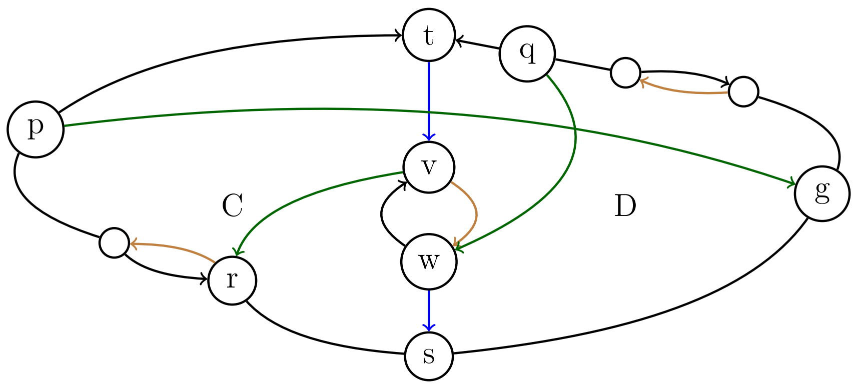


Figure 3.1: Cycles C and D with $\beta(C \cap D)$ minimal. Brown arcs have capacity 1, black/green arrows correspond to paths, blue arrows indicate (possibly intersecting) parallel paths. The green paths belong to the cycle F.

(ii) $F \cap (C \cup D) \neq \varnothing$. This implies that there exists a vertex r in $C \cup D$ (w.l.o.g. r in C) such that $F_{v,r}$ is a non-empty path. Let r be the first such vertex in C. Denote by q the last vertex in F adjacent to an edge in C or D (in Fig. 3.1 chosen to be in $D \backslash C$). Note that also $F_{q,w}$ is a non-empty path. Let $E \in \{C, D\}$ be the cycle q belongs to. Note that $\bar{c}(F_{v,r}) + \bar{c}(C_{r,v}) \equiv 0 \pmod 3$, since otherwise, $\bar{c}(F_{v,r} \cup C_{r,v}) = -\gamma \pmod 3$ and $|\beta((F_{v,r} \cup C_{r,v}) \cap D)| < |\beta(C \cap D)|$, contrary to our choice.

There is no cycle of congruence γ, which implies that

$$\bar{c}(F_{v,r}) \equiv -\bar{c}(C_{r,v}) \pmod 3 \equiv \bar{c}(C_{v,r}) + \gamma \pmod 3$$

and

$$\bar{c}(F_{q,w}) \equiv -\bar{c}(E_{w,q}) \pmod 3 \equiv \bar{c}(E_{q,w}) + \gamma \pmod 3.$$

We then derive a contradiction as follows: If q is a vertex of C, recall that $F = F_{v,r} \cup C_{r,q} \cup F_{q,v}$. Then $\bar{c}(F) \equiv \bar{c}(-C_{r,v} + C_{r,q} - C_{v,q}) \pmod 3 \equiv \bar{c}(-\chi_A^{C_{r,q}} - \chi_A^{C_{q,v}} + \chi_A^{C_{r,q}} - \chi_A^{C_{v,q}}) \pmod 3 \equiv \gamma \pmod 3$. Otherwise, if q is not a vertex of C but a vertex of D, let p be the last vertex that C and F share. Let g be the first vertex of $F \cap D$ after p, and $X := C_{r,p} \cup F_{p,g} \cup D_{g,q}$. Then $\bar{c}\big(F_{v,r} \cup X \cup F_{q,w}\big) \equiv \bar{c}\big(C_{v,r} \cup X \cup F_{q,w}\big) + \gamma \pmod 3 \equiv \bar{c}\big(C_{v,r} \cup X \cup D_{q,v}\big) + 2\gamma \pmod 3$, so one of these three cycles has congruence $\gamma \pmod 3$.

□

3.1.2 Transposed network matrix

Let T be the transpose of a network matrix. While the high-level approach is the same as in the section before, our combinatorial algorithm is quite different, as it operates on vertex sets rather than on edge sets. In the rest of this section, we will make use of the following graph $\bar{G}$:

Let $G = (V, A \cup U)$ be the graph representing T. We define $\bar{G}$ as having the same edge and vertex set as G, but the edges are, in addition, endowed with capacities $\bar{b} : E \to \mathbb{Z}_{\geqslant 0}$ and weights $\bar{c} : E \to \mathbb{Z}_{\geqslant 0}$ defined by

$$\bar{b}(a) = \begin{cases} b_a & \text{if } a \in A, \\ \infty & \text{otherwise} \end{cases}, \bar{c}(a) = \begin{cases} c_a & \text{if } a \in U, \\ 0 & \text{otherwise} \end{cases}.$$

We denote by $A_i := \{a \in A \colon b_a = i\}$ the subset of arcs corresponding to right-hand side i. In particular, A_0 are the edges corresponding to the constraints that are tight at the vertex 0.

Definition 3.9. We call any set $Q \subseteq V$ that fulfills $\delta^+_U(Q) = \delta^-_{A_0}(Q) = \varnothing$ **simple feasible**.

From Lemma 2.28 it follows that any extreme ray r of C can be written as $\delta^-_U(Q)$, for $Q \subseteq V$ simple feasible. Note further that if Q is feasible, then for all $a \in \delta^-_A(Q) : T_{a,\cdot}\delta^-_U(Q) = 1$, and that for all $a \in \delta^+_A(Q), T_{a,\cdot}\delta^-_U(Q) = -1$ (both follows by similar arguments as in the proof of Lemma 2.27), and that $\delta^-_U(Q) \in P \cap \mathbb{Z}^n$.

Also note that if $Q^1, Q^2 \subseteq V$ are simple feasible, then $Q^1 \cap Q^2$ and $Q^1 \cup Q^2$ are, too, and $\delta^-_U(Q^1 \cup Q^2) + \delta^-_U(Q^1 \cap Q^1) = \delta^-_U(Q^1) + \delta^-_U(Q^2)$, which can be seen from counting edges and using the fact that $\delta^+_U(Q^1) = \delta^+_U(Q^2) = \varnothing$.

Lemma 3.10. Let $(\mathcal{A})$ be an (sC3TU) problem with T a transposed network matrix.

1. $(\mathcal{A})$ can be reduced to the following combinatorial problem: Find
   - a vertex set $Q \subseteq V$ with $\delta^+_U(Q) = \delta^-_{A_0}(Q) = \varnothing$ and $c^\mathsf{T}\delta^-_U(Q) \equiv \gamma \pmod 3$, or
   - two sets $Q^1 \subset Q^2 \subseteq V$ with $\delta^+_U(Q^i) = \delta^-_{A_0}(Q^i) = \varnothing$, $\delta^-_{A_1}(Q^1) \cap \delta^-_{A_1}(Q^2) = \varnothing$, and $c^\mathsf{T}\delta^-_U(Q^i) \equiv -\gamma \pmod 3$, $i = 1, 2$.
2. If $(\mathcal{A})$ is infeasible, there is a constraint in $Tx \leqslant b$ of width at most 1.

Proof. For the ease of exposition, we write $\delta^-_U(Q) \in \mathbb{R}^U$, for $Q \subseteq V$, instead of $\chi^{\delta^-_U(Q)} \in \mathbb{R}^U$.

Recall that A_0 denotes the set of tight constraints at 0, and consider the supporting cone $C := \{x \in \mathbb{R}^n : T_{A_0,\cdot}x \leqslant b_{A_0}\}$.

Let $\delta^-_U(Q^1), \dots, \delta^-_U(Q^k)$ correspond to all extreme rays of C, such that any point in $P \cap \mathbb{Z}^n$ is an integer conic combination of $\delta^-_U(Q^1), \dots, \delta^-_U(Q^k)$. Note that these Q_i, for $1 \leqslant i \leqslant k$, are simple feasible.

To shorten notation a bit, we will at times, for two integers a, b, drop the '(mod 3)' and say that $a \equiv b$ or $a \not\equiv b$, when we mean that $a \equiv b \pmod 3$, or $a \not\equiv b \pmod 3$, respectively.

1. Let a vector $x = \sum_{i=1}^k \lambda_i \delta^-_U(Q^i)$ with $\lambda \in \mathbb{Z}^k_{>0}$, be given. Rather than as a conic combination, we write it as the sum of l many extreme rays (thereby

possibly repeating some), $x = \sum_{i=1}^{l} \delta_U^-(Q^i)$, for some $l \in \mathbb{N}$. We have that $x \in P \Leftrightarrow \forall e = (u,v) \in A$:

$$|\{i \in \{1,\ldots,l\}\colon v \in Q^i, u \notin Q^i\}| - |\{i \in \{1,\ldots,l\}\colon u \in Q^i, v \notin Q^i\}| \leqslant b_e.$$

Assume that $\sum_{i=1}^{k} \lambda_i > 2$. First, we find a different way of writing x as a sum based on a laminar family. If there are two sets Q^i, Q^j, $Q^i \cap Q^j \neq \varnothing$, $Q^i \backslash Q^j \neq \varnothing$ and $Q^j \backslash Q^i \neq \varnothing$, we replace them by $Q^i \cup Q^j$ and $Q^i \cap Q^j$. We claim that $\delta_U^-(Q^1 \cap Q^2) + \delta_U^-(Q^1 \cup Q^2) = \delta_U^-(Q^1) + \delta_U^-(Q^2)$, which implies that x remains unchanged:

Let $a = (u,v)$, and let us distinguish two cases:

- Let neither a, nor $-a$ be an edge leaving $Q^1\backslash Q^2$ and entering $Q^2\backslash Q^1$. Then, $\delta_U^-(Q^1 \cap Q^2)_a + \delta_U^-(Q^1 \cup Q^2)_a = \delta_U^-(Q^1)_a + \delta_U^-(Q^2)_a$.
- The remaining case cannot occur: W.l.o.g. assume that $u \in Q^1\backslash Q^2$ and $v \in Q^2\backslash Q^1$ (otherwise, relabel Q^1 and Q^2). But then, $a \in \delta_U^+(Q^1)$, which is supposed to be the empty set, a contradiction.

Thus, w.l.o.g. the family of Q^i's in the representation of x is laminar. If it contains a Q^i with $c^\mathsf{T}\delta_U^-(Q^i) \equiv \gamma \pmod 3$, the proof is complete. Otherwise, there are two sets Q^i, Q^j with $c^\mathsf{T}\delta_U^-(Q^i) \equiv c^\mathsf{T}\delta_U^-(Q^j) \pmod 3 \equiv -\gamma \pmod 3$. If $Q^i \cap Q^j = \varnothing$, then $Q := Q^1 \cup Q^2$ is (c,γ)-feasible.

Otherwise, for all pairs Q^i, Q^j of non-zero congruence we have $Q^i \cap Q^j \neq \varnothing$, which implies that they form a chain of sets containing each other. Choose two, Q^1 and Q^2, say, and let $y = \delta_U^-(Q^1) + \delta_U^-(Q^2)$.

Assume that for some $a \in A_1$, $a \in \delta^-(Q^1) \cap \delta^-(Q^2)$, implying that $T_{a,\cdot}y = 2 > b_a$. But since x was in P, there has to be another set $Q^{(a)}$ in the decomposition of x with $a \in \delta^+(Q^{(a)})$. Because of laminarity, $Q^{(a)} \cap Q^2 = \varnothing$, and thus, $c^\mathsf{T}\delta_U^-(Q^{(a)}) \equiv 0 \pmod 3$.

Consider the laminar family $\mathcal{L}$ of all sets of this form, i.e., of all such $Q^{(a)}$'s. For each chain in $\mathcal{L}$, choose a maximal set. Call this collection of disjoint sets $\bar{L}$. Denote by $\bar{Q}$ the disjoint union $\bar{Q} := Q^2 \cup \bigcup\{Q^i\colon Q^i \in \bar{L}\}$, and then $\delta_U^-(Q^1) + \delta_U^-(\bar{Q})$ is (c,γ)-feasible.

Finally, any simple feasible set Q with $\delta_U^-(Q) \equiv \gamma \pmod 3$, or two feasible sets as in the statement of the lemma, lead to a (c,γ)-feasible solution, which implies the reduction, as claimed.

2. We will make use of the following facts repeatedly: As there is no simple feasible set of congruence γ, one of the following may occur for two simple feasible sets Q^1, Q^2:

   (i) $c^\mathsf{T}\delta_U^-(Q^1) \equiv c^\mathsf{T}\delta_U^-(Q^2) \equiv -\gamma \Rightarrow c^\mathsf{T}\delta_U^-(Q^1 \cup Q^2) \equiv c^\mathsf{T}\delta_U^-(Q^1 \cap Q^2) \equiv -\gamma$.

   (ii) $c^\mathsf{T}\delta_U^-(Q^1) \equiv 0,\ c^\mathsf{T}\delta_U^-(Q^2) \equiv -\gamma \Rightarrow c^\mathsf{T}\delta_U^-(Q^1 \cup Q^2) + c^\mathsf{T}\delta_U^-(Q^1 \cap Q^2) \equiv -\gamma$ and either $c^\mathsf{T}\delta_U^-(Q^1 \cup Q^2)$ or $c^\mathsf{T}\delta_U^-(Q^1 \cap Q^2) \equiv 0$.

Without loss of generality, P is full-dimensional, as otherwise, one of the constraints in the description of P will define a flat direction of width 0. By Lemma 1.7, there is a maximal simple feasible set C with cut $\delta_U^-(C)$ of congruence $-\gamma$ (cmp. Fig. 3.2). Let $D \subseteq C$ be a minimal simple feasible subset with $c^\mathsf{T}\delta_U^-(D) \equiv -\gamma$ (possibly $C = D$). Since $\delta_U^-(C) + \delta_U^-(D) \notin P$, there is an arc $(v, w) \in A_1 \cap \delta^-(C) \cap \delta^-(D)$ and a simple feasible F with $v \in F \backslash C$, $w \in D \backslash F$, as otherwise $\delta_U^-(C) + \delta_U^-(D)$ is (c, γ)-feasible.

We claim that $D \cap F \neq \varnothing$. Suppose not. Since $(\mathcal{A})$ is infeasible, $c^\mathsf{T}\delta_U^-(F) \not\equiv \gamma$. Furthermore, $c^\mathsf{T}\delta_U^-(F) \not\equiv -\gamma$, as otherwise $c^\mathsf{T}\delta_U^-(C \cup F) \equiv -\gamma$, contrary to our choice of C being maximal. So $c^\mathsf{T}\delta_U^-(F) \equiv 0$ and $c^\mathsf{T}\delta_U^-(C \cup F) \equiv -\gamma$, again contrary to our choice of C.

By maximality of C, $c^\mathsf{T}\delta_U^-(C \cup F) \equiv 0$. Therefore, we are in case (ii) above and thus $c^\mathsf{T}\delta_U^-(C \cap F) \equiv -\gamma$. We obtain $c^\mathsf{T}\delta_U^-(D \cup (C \cap F)) + c^\mathsf{T}\delta_U^-(D \cap (C \cap F)) \equiv c^\mathsf{T}\delta_U^-(D) + c^\mathsf{T}\delta_U^-(C \cap F) \equiv -\gamma - \gamma \equiv \gamma$. Since $D \cap (C \cap F) = D \cap F \subsetneq D$, $c^\mathsf{T}\delta_U^-(D \cap (C \cap F)) \equiv 0$. Then, $c^\mathsf{T}\delta_U^-(D \cup (C \cap F)) \equiv \gamma$.

□

3.1.3 Constant-size matrices

Let T in (C3TU) be, up to row/column permutations and sign changes, one of the matrices in (1.11).

Lemma 3.11. Consider $P = \{x \in \mathbb{R}^n : e \leqslant Tx \leqslant b,\ l \leqslant x \leqslant u\}$, where T is up to multiplication of columns with -1 one of the matrices in (1.11), $e,\ l \in (\mathbb{Z} \cup \{-\infty\})^5$, $b,\ l \in (\mathbb{Z} \cup \{\infty\})^5$. If P is congruence-infeasible for $p \in \{0, \pm 1\}^5$, $\gamma \in \{0, \pm 1\}$, then it admits a flat direction of width 1 that is given by a standard unit vector or a row in T.

To prove Lemma 3.11, we will use a couple of technical results. Our proof plan is as follows: First, we show that without loss of generality, all entries in the right-hand sides are finite.

Lemma 3.12. Let $T \in \mathbb{Z}^{m \times n}$ be a TU matrix, and $t \in \mathbb{Z}^n$ be TU-appendable. Consider the polyhedra $P = \{x \in \mathbb{R}^n : Tx \leqslant b\}$ and $\bar{P} = \{x \in \mathbb{R}^n : Tx \leqslant$

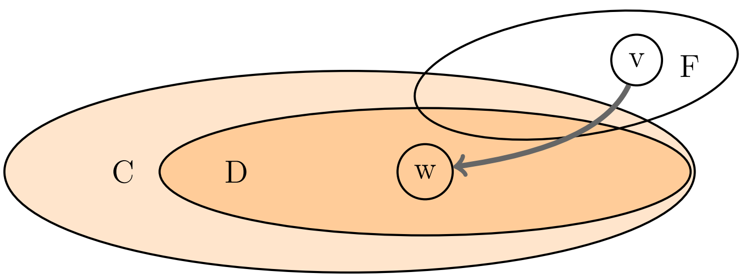


Figure 3.2: Sets C, D and F in the proof of Lemma 3.10.2.

$b,\ tx \leqslant 2\}$. Then if $0 \in P$ and $\omega(P) > 1$, also $\omega(\bar{P}) > 1$.

Proof. As in [23], we rewrite the width of P as follows, where for sake for brevity, we denote by

$$\bar{T} := \begin{bmatrix} T \\ t \end{bmatrix}$$

and $\bar{b}(\gamma) := \begin{bmatrix} b \\ \gamma \end{bmatrix}$, such that $\bar{T}$ and $\bar{b}(2)$ are the matrix and vector describing $\bar{P}$:

Then

$$\begin{aligned}
1 < \omega(P) &= \min_{c \in \mathbb{Z}^n \setminus \{0\}} \big\{ \max\{c^\top x \colon Tx \leqslant b\} - \min\{c^\top x \colon Tx \leqslant b\}\big\} \\
&= \min_{c \in \mathbb{Z}^n \setminus \{0\}} \big\{ \max\{c^\top x \colon Tx \leqslant b\} + \max\{-c^\top x \colon Tx \leqslant b\}\big\} \\
&= \min_{c \in \mathbb{Z}^n \setminus \{0\}} \big\{ \min\{b^\top y \colon y^\top T = c^\top,\ y \geqslant 0\} + \\
&\quad \min\{b^\top z \colon z^\top T = -c^\top,\ z \geqslant 0\}\big\} \\
&= \min_{c \in \mathbb{Z}^n \setminus \{0\}} \big\{ \min\{b^\top y + b^\top z \colon y^\top T = -z^\top T \in \mathbb{Z}^n \setminus \{0\},\ y, z \geqslant 0\}\big\} \\
&= \min\{b^\top y + b^\top z \colon y^\top T = -z^\top T \in \mathbb{Z}^n \setminus \{0\},\ y, z \geqslant 0\} \\
&= \min\{\bar{b}(2)^\top y + \bar{b}(2)^\top z \colon y^\top \bar{T} = -z^\top \bar{T} \in \mathbb{Z}^n \setminus \{0\},\ y, z \geqslant 0, \\
&\quad y_{m+1} = z_{m+1} = 0\}. \qquad (3.2)
\end{aligned}$$

A similar calculation yields

$$\omega(\bar{P}) = \min\{\bar{b}(2)^\top y + \bar{b}(2)^\top z \colon y^\top \bar{T} = -z^\top \bar{T} \in \mathbb{Z}^n \setminus \{0\},\ y, z \geqslant 0\}.$$

Notice that every solution to the last term in (3.2) is also feasible for $\omega(\bar{P})$, and that every (y^*, z^*) with $y^*_{m+1} + z^*_{m+1} = 0$ which is feasible for $\omega(\bar{P})$ is also feasible for (3.2).

Furthermore, let (y^*, z^*) be an optimal solution for $\omega(\bar{P})$ with $y^*_{m+1} + z^*_{m+1} > 0$. As $\bar{T}$ is TU, we may assume (y^*, z^*) to be integral. Since $y^*, z^* \geqslant 0$, this implies that then, $\bar{b}(2)^\top(y^* + z^*) \geqslant 2$.

Taken together, this implies that $\omega(\bar{P}) \geqslant 2$.

□

Next, we show that Lemma 3.11 can be reduced to a simpler setting.

Lemma 3.13. Let P be as in Lemma 3.11. Then the statement of Lemma 3.11 follows if it holds for all of the polyhedra given by

$$\{x \in \mathbb{R}^n \colon e \leqslant Sx \leqslant b,\ 0 \leqslant x \leqslant u\}, \qquad (3.3)$$

with 0 as a vertex, where $e, b, u \in \mathbb{Z}^5$ and S is one of the matrices in (1.11), have width of at most 1.

Proof of Lemma 3.13. First, we replace all infinite entries in e and l by -2 and those of b and u by 2 and call the resulting polyhedron Q. Note that by Lemma 3.12, if $\omega(P) > 1$, then also $\omega(Q) > 1$. Observe that we may exchange any constraint $e_i \leqslant T_{i,\cdot} \leqslant b_i$ by $-b_i \leqslant T_{i,\cdot} \leqslant -e_i$ and change the sign of any column (and replacing the upper and lower bound for the corresponding variable accordingly) without affecting width or congruence feasibility. Furthermore, we may translate the polyhedron so that l become the all-zero vector.

We thus arrive at one of the polyhedra in (3.3), whose feasibility also implies congruence-feasibility of P. □

We are now ready to prove the lemma.

Proof of Lemma 3.11. By Lemma 3.13, it suffices to show the statement for a polyhedron as given in (3.3).

Let us first assume that S is the matrix on the right in (1.11). Since S is non-negative and 0 is a vertex, $e = 0$. By assumption, $b, u \geqslant 2$, and thus, we may choose any $j \in [n]$ such that $p_j \neq 0$. Then $\gamma \cdot \chi^{\{j\}}$ is a congruence-feasible solution.

If S is the matrix on the left in (1.11), then by non-negativity of the first three rows of S, $e_1 = e_2 = e_3 = 0$. By assumption, $u, b \geqslant 2$. As the following case-by-case analysis shows, the constraints $e_4 \leqslant S_{4,\cdot} \leqslant b_4$ and $e_5 \leqslant S_{5,\cdot} \leqslant b_5$ allow for a congruence-feasible solution x^* such that $S_{1:3,\cdot}x^* \leqslant 2$. We perform a case-by-case analysis: If $p_2 \neq 0$: Choose $\gamma \cdot \chi^2$. If $p_1 \neq 0$: Either $\gamma \cdot \chi^1$ can be chosen, or $b_5 \leqslant \gamma - 1$. But then, $(2 - \gamma)\chi^4$ is feasible, implying $b_4 \geqslant 1$, so if $p_4 \neq 0$, choose from variables 1 and 4. Otherwise, take $\gamma \cdot (\chi^1 + \chi^4)$. If $p_3 \neq 0$: Can be reduced to the case above by switching rows 4 and 5, columns 1 and 3 as well as columns 4 and 5. If $p_4 \neq 0, p_1, p_2, p_3 = 0$: If $\gamma \cdot \chi^4$ is not feasible, either $b_4 \leqslant \gamma - 1$ or $e_5 \geqslant -\gamma + 1$. Depending on p_5, we construct a solution from x_4 and x_5. If $p_5 \neq 0$ and $p_1 = p_2 = p_3 = 0$: Can be reduced to the case above by switching rows 4 and 5, columns 1 and 3 as well as columns 4 and 5.

Since $u, b \geqslant 2$, x^* is also congruence-feasible with respect to all constraints. □

3.2 Feasibility in congruence-3-constrained base block TU problems

If T in (C3TU) is a base block of constant size, then feasibility for (C3TU) can be checked using Lenstra's algorithm [33].

We thus concentrate on T being a network matrix or a transposed network matrix. In the latter case, we give an efficient algorithm, while in the first, we give structural results for the associated graph in the hope that it will lead to an efficient algorithm in future work.

3.2.1 Transposed Network Matrix

Theorem 3.14. Let T in (sC3TU) be a transposed network matrix. Then:

1. Given $Q' \subseteq V$ simple feasible, we can find a minimal/maximal simple feasible superset $Q \supsetneq Q'$ of given congruence in polynomial time, or decide that no such set exists.
2. (sC3TU) can be solved in polynomial time.

Proof. Again, to improve readibility, we drop '(mod 3)' at times.

1. Let $S^i := \{e \in U : \bar{c}(e) \equiv i \pmod 3\}$, $i \in [2]$. Define the function $k : 2^V \to \mathbb{Z}$ by

$$k(\{v\}) := |\delta_U^-(v) \cap S^1| - |\delta_U^+(v) \cap S^1| + 2\big(|\delta_U^-(v) \cap S^2| - |\delta_U^+(v) \cap S^2|\big),$$

and $k(Q) = \sum_{v \in Q} k(\{v\})$, for any $Q \subseteq V$.

Let $R^i := \{v \in V : k(v) \equiv i\}, i \in [2]$. Then, for all $Q \subseteq V$ we have

$$k(Q) = \bar{c}^\mathsf{T}(\delta_U^-(Q) - \delta_U^+(Q)) = |Q \cap R^1| + 2|Q \cap R^2|.$$

We want to solve

$$\min\{\varepsilon|Q| : Q \supsetneq Q', \delta_U^+(Q) = \delta_{A_0}^-(Q) = \varnothing, c^\mathsf{T}\delta_U^-(Q) \equiv \alpha \pmod 3\},$$

for $\varepsilon \in \{\pm 1\}$, $\alpha \in [2]$. We proceed similarly as in the bimodular case. For $v \in V$ let $Q^v := Q' \cup \{v\}$. Define the submodular functions $g_1(Q) := M|\delta_U^+(Q)|$, $g_2(Q) := M|\delta_{A_0}^-(Q)|$ and $g_3^v(Q) := -M|Q \cap Q^v|$, for $M > 0$ large enough, yet polynomial in the input. It then suffices to solve the following problems

$$\min\big\{\varepsilon|Q| + g_1(Q) + g_2(Q) + g_3^v(Q) : |Q \cap R^1| + 2|Q \cap R^2| \equiv \gamma\big\},$$

or equivalently, the following $3|V|$ many problems, indexed by $\alpha \in [2]$, $v \in V$:

$$\min\big\{\varepsilon|Q| + g_1(Q) + g_2(Q) + g_3^v(Q) : |Q \cap R^1| \equiv \gamma - 2\alpha, |Q \cap R^2| \equiv \alpha\big\},$$

which can be done in polynomial-time [34].

2. We give a poly-time algorithm for the problem stated in Lemma 3.10.1.

To decide feasibility, we first check whether there is a simple feasible set Q with $c^\mathsf{T}\delta_U^-(Q) \equiv \gamma$. If there is one, the problem is feasible. In case that no such set was found, we look for a minimal simple feasible set Q^1 of congruence $-\gamma$. If we are not able to find such a set, either, the problem is infeasible.

Otherwise, we claim that there can be no other simple feasible set S of congruence $-\gamma$ that is crossing Q^1 in the sense that $S \backslash Q^1, Q^1 \backslash S, S \cap Q^1 \neq \varnothing$: Assume there was, then by minimality of Q^1, $S \cap Q^1$ is of zero congruence. But then, $c^\mathsf{T}\delta_U^-(Q^1 \cup S) \equiv c^\mathsf{T}\delta_U^-(Q^1 \cup S) + c^\mathsf{T}\delta_U^-(Q^1 \cap S) \equiv c^\mathsf{T}\delta_U^-(Q^1) + c^\mathsf{T}\delta_U^-(S) \equiv -\gamma - \gamma = \gamma$, a contradiction. We conclude that no such S can exist.

Taken together, we observe that

- there is no simple feasible set S of non-zero congruence crossing with Q^1, and
- there is no simple feasible set of non-zero congruence strictly contained in Q^1 (by minimality of Q^1), and further, that
- there is no simple feasible set S of congruence $-\gamma$ with $S \cap Q^1 = \varnothing$ (since otherwise $S \cup Q^1$ is simple feasible with congruence γ).

If there is no arc in A_1 entering Q^1, we return $2 \cdot \delta_U^-(Q^1)$ as a solution.

Otherwise, by the above bullet points, all sets of congruence $-\gamma$ contain Q^1. Using the algorithm described in the first part of this proof, we look for a maximal simple feasible set $Q^2 \supsetneq Q^1$ with $c^\mathsf{T}\delta_U^-(Q^2) \equiv -\gamma$. Observe that similarly as with Q^1, there can be no feasible set S of non-zero congruence that is crossing with Q^2, for the following reason:

Assume there was a feasible set S with $\delta_U^-(S) \equiv -\gamma$ such that $S \cap Q^2, S \backslash Q^2, Q^2 \backslash S \neq \varnothing$. Then either one of $S \cap Q^2$ and $S \cup Q^2$ has congruence γ, or both have congruence $-\gamma$, in contradiction to our choice of Q^2 being maximal.

Consequently, all feasible sets of congruence $-\gamma$ form a chain

$$Q^1 \subset \cdots \subset Q^2,$$

with Q^1 being the minimal, and Q^2 the maximal set in the chain.

If there is no arc in A_1 entering Q^1 and Q^2, we return $\delta_U^-(Q^1) + \delta_U^-(Q^2)$ as a solution. If there is, however, it enters all sets in the chain, and consequently, we cannot choose any two sets to obtain a feasible solution. The problem is thus infeasible.

□

3.2.2 Network Matrix

Again, to simplify notation, we consider a cycle or a circulation to be a vector $C \in \mathbb{R}^E_{\geqslant 0}$, rather than a subset of E.

We begin this section with two observations: First, that given rev$((\mathcal{A}))$, there is an efficient algorithm that returns a cycle of non-zero congruence, and second, that if an edge is contained in all circulations of non-zero congruence, the (sC3TU) problem reduces to another of the same kind in a graph of strictly fewer edges.

Lemma 3.15. Let $G = (V, A)$ be a directed graph with edge weights $c \in \{0, 1, 2\}^A_{\geqslant 0}$ and capacities $b \in \mathbb{Z}^A_{\geqslant 0}$. Then we can find a cycle C with $c(C) \not\equiv 0 \pmod 3$ in polynomial time, or decide that no such cycle exists.

Furthermore, if all cycles in G are of congruence $0 \pmod 3$, then for all

$s, t \in V$, $k \in \{0, 1, 2\}$, a shortest s-t-path of given congruence k can be found efficiently.

Proof. The following procedure is a generalization of the proof of Lemma 2.25.

We start with an auxiliary graph $G' := (V^0 \cup V^1 \cup V^2, A')$, where V^0, V^1, V^2 are three copies of the vertex set V. We denote, for each $v \in V$, by v^0, v^1 and v^2 the corresponding vertices in V^0, V^1 and V^2, respectively, and for each $a \in A$, we introduce three arcs (u^i, u^{i+c_a}), for $i \in \{0, 1, 2\}$, each with the same capacity as (u, v).

Similarly as in the bimodular case, each v^i–v^j walk in G' corresponds to precisely one closed walk C in G with $c(C) \equiv j - i \pmod 3$. To solve the problem, it thus suffices to compute shortest v^i–v^j paths, for all $v \in V$, $i \neq j \in \{0, 1, 2\}$. If we cannot find any such path, we know that no circulation of non-zero congruence exists. If we do, we decompose the corresponding circulations in G into cycles, one of which has non-zero congruence, and return this cycle.

To see that the second part is true, we make use of the same auxiliary graph, and search for a $s^i - t^{i+k}$-path, $i \in \{0, 1, 2\}$. Such paths correspond precisely to sums of one s-t-path and cycles in the original graph G. After subtracting the cycles, all of which have congruence $0 \pmod 3$ by assumption, an s-t-path of congruence k remains. □

The next lemma provides us with an argument why we may, without loss of generality, assume that there is no edge e in G that attains the same value for all circulations of non-zero congruence.

Lemma 3.16. Consider a problem $(\mathcal{A})$ of type (sC3TU) whose constraint matrix T is a network matrix with corresponding graph $\bar{G} := \operatorname{rev}((\mathcal{A})) = (V, E \cup -U)$. Assume that there is an edge $a^* \in A \cup -U$, and a value $z \in \mathbb{N}_{\geqslant 0}$, such that for all circulations C in $\bar{G}$ with $c^\mathsf{T} C \not\equiv 0 \pmod 3$, $C(a^*) = z$.

Then, we can reduce $(\mathcal{A})$ to a (sC3TU) problem whose constraint matrix is a network matrix with strictly fewer rows or columns.

Proof. Let a^* be an edge in $\bar{G} = (V, E \cup -U)$ as in the statement of the lemma. By the first part of Lemma 3.8, simplified integral circulations in $\bar{G}$ correspond precisely to points in $P \cap \mathbb{Z}^n$. We claim that we can add an equality constraint to $(\mathcal{A})$ without affecting feasibility:

If $a^* \in A$, all $x \in P \cap \mathbb{Z}^n$ with $c^\mathsf{T} x \not\equiv 0 \pmod 3$ fulfill $x_{a^*} = z$. Appending this constraint allows us to reduce $(\mathcal{A})$ to deciding feasibility of

$$\left\{x \in \mathbb{R}^{n-1} \colon T_{\cdot,[n]\setminus\{a^*\}} x \leqslant b - z T_{\cdot,a^*},\ {c_{[n]\setminus\{a^*\}}}^\mathsf{T} x \equiv \gamma - z c_{a^*} \pmod 3\right\},$$

where the sub-matrix $T_{\cdot,[n]\setminus\{a^*\}}$ of T is again a network matrix.

If $a^* \in U$, we append the constraint $T_{a^*,\cdot} x = z$, if $a^* \in -U$, $T_{a^*,\cdot} x = -z$. To abbreviate notation, let $z' = z$ in the first case, and $z' = -z$ in the second. This way, we can consider both cases at once and write $T_{a^*,\cdot} x = z'$: Via a unimodular operation Z we transform $T_{a*,\cdot}$ into ${e_1}^\mathsf{T} = T_{a*,\cdot}Z$. Then $z' = T_{a*,\cdot}ZZ^{-1}x = e_1Z^{-1}x = (Z^{-1}x)_1$, so that $(\mathcal{A})$ was reduced to

$$\Big\{x \in \mathbb{R}^{n-1} \colon (TZ)_{[m]\setminus\{a*\},2:n}x \leqslant b - z(TZ)_{[m]\setminus\{a*\},1},$$
$$(Zc)_{2:n}{}^\mathsf{T} x \equiv \gamma - z'(Zc)_1 \pmod 3\Big\}.$$

From the fact that pivot operations preserve the property of a matrix being a network matrix, TZ, and any of its sub-matrices, are network matrices again. □

Lemma 3.17. Let $(\mathcal{A})$ be an (sC3TU) problem with T a network matrix. Then solving it can be reduced to finding a circulation of congruence γ in $\bar{G} := \operatorname{rev}((\mathcal{A}))$, where we can make the following additional assumptions on $\bar{G}$:

1. Each edge has capacity 1, and for each edge a, there is a circulation C with $\bar{c}^\mathsf{T}C \not\equiv 0 \pmod 3$ and $C_a = 0$.
2. We are given three cycles C_1, C_2, C_3 of congruence $-\gamma$ with pairwise non-empty intersection.

Proof of Lemma 3.17. By Lemma 3.8, a problem $(\mathcal{A})$ of type (sC3TU) can be reduced to finding a circulation of proper congruence in the directed graph $\bar{G} := \operatorname{rev}((\mathcal{A}))$ with edge weights $\bar{c}$.

Let us simplify the problem: We replace each edge $e = (u, v)$ with capacity k by k many parallel edges (all pointing from u to v), each with a weight of $\bar{c}_e$ and with a capacity of 1. This new graph has a circulation of congruence γ if and only if the original did, and any such circulation can be transformed into one in $\bar{G}$. Therefore, in what follows we may without loss of generality assume that each edge in $\bar{G}$ has a capacity of 1.

Next, we argue that without loss of generality, for each edge a there is a circulation C with $\bar{c}^\mathsf{T}C \not\equiv 0 \pmod 3$ and $C_a = 0$. Assume this was not the case, and let a^* be an edge such that for each circulation C of non-zero congruence, $C_{a^*} > 0$. This implies that a^* has no parallel edges, and thus, already in the original graph $\bar{G}$ (before we duplicated edges) all circulations of non-zero congruence contained a^* in their support with a value of 1. By Lemma 3.16, $(\mathcal{A})$ can then be reduced to a smaller (sC3TU) problem with a network matrix as its constraint matrix, proving the claim.

Next, we show that w.l.o.g., there are two cycles C_1 and C_2 with $\bar{c}(C_1) \equiv \bar{c}(C_2) \equiv -\gamma \pmod 3$:

We apply Lemma 3.15 to check for a cycle C_1 of non-zero congruence in $\bar{G}$. If no such cycle exists, $(\mathcal{A})$ is infeasible, so we assume it does. If $\bar{c}(C_1) \equiv \gamma \pmod 3$, $(\mathcal{A})$ is feasible. Otherwise, let a be any edge in C_1. By our assumption above, there is a cycle C_2 of non-zero congruence with $C_2(a) = 0$. We can find such a cycle by deleting a from $\bar{G}$ and running the algorithm from Lemma 3.15 again. For the same reason as before, we may assume that $\bar{c}^\mathsf{T}C_2 \equiv -\gamma \pmod 3$. If

$\mathrm{supp}(C_1) \cap \mathrm{supp}(C_2) = \varnothing$, then $C_1 + C_2$ solves the problem and $(\mathcal{A})$ is feasible, so let us assume their support is not disjoint.

Let $a \in \mathrm{supp}(C_1) \cap \mathrm{supp}(C_2)$. Similarly as above, we can delete a from $\bar{G}$ and search for a cycle of non-zero congruence. If none is found, or if it has congruence γ, or if one of $\mathrm{supp}(C_3) \cap \mathrm{supp}(C_1)$, $\mathrm{supp}(C_3) \cap \mathrm{supp}(C_2)$, or $\mathrm{supp}(C_3) \cap \mathrm{supp}(C_1 + C_2)$ is empty, we are done. Otherwise, we found three cycles as claimed. $\square$

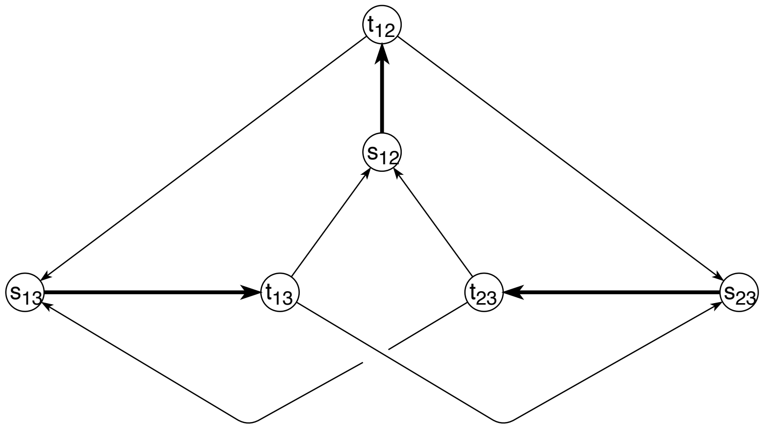
As it turns out, assuming more structure on just these three cycles greatly helps us to deduce more properties of $\mathrm{rev}((\mathcal{A}))$, as the following lemma shows.

Lemma 3.18. Consider the three cycles C_1, C_2, C_3 in Lemma 3.17. Denote by C their union, i.e., $C = \chi^{\mathrm{supp}(C_1) \cup \mathrm{supp}(C_2) \cup \mathrm{supp}(C_3)}$. If $\mathrm{supp}(C_1) \cap \mathrm{supp}(C_2) \cap \mathrm{supp}(C_3) = \varnothing$, the intersection of any two cycles forms a path in $\bar{G}$, and for each cycle, there is at least an edge between one intersecting path and the other, then the problem can be reduced to the following setting:

Find a feasible circulation of congruence γ in $\bar{G}$, where the congruence of an s-t-path, for $s, t \in C_i$, that is internally vertex disjoint from C, depends only on its starting and target vertices s and t.

Proof. Note first that if we assume that there is no cycle of non-zero congruence that is edge-disjoint from C (in any case, the existence of such a cycle would imply feasibility of the (sC3TU) problem), we can compute in polynomial time the congruence of s-t-paths that are internally vertex disjoint from C by considering the graph resulting from $\mathrm{rev}(\mathcal{A})$ after deleting all edges in C.

Denote by s_{ij}, $i, j \in \{1, 2, 3\}$, $i \neq j$, the first vertex in the intersection path of C_i and C_j, and by t_{ij} the last vertex of this path, as in the following illustration.



We first observe that we can form two additional cycles,

$$K_1 := (C_1)_{s_{13}}^{t_{13}} + (C_3)_{t_{13}}^{s_{23}} + (C_2)_{s_{23}}^{t_{23}} + (C_2)_{t_{23}}^{s_{12}} + (C_1)_{s_{12}}^{t_{12}} + (C_1)_{t_{12}}^{s_{13}}$$

and

$$K_2 := (C_2)_{s_{23}}^{t_{23}} + (C_3)_{t_{23}}^{s_{13}} + (C_1)_{s_{13}}^{t_{13}} + (C_1)_{t_{13}}^{s_{12}} + (C_2)_{s_{12}}^{t_{12}} + (C_2)_{t_{12}}^{s_{23}}.$$

K_1 and K_2 originate from C_3 by replacing one path by a parallel one that goes over C_1 and C_2. Let us make a calculation on how this detour affects the congruence of K_1 and K_2. Note first that the circulation which results from taking both detours is $C_1 + C_2$. Thus, at least one of K_1 or K_2 has a different congruence than C_3. W.l.o.g., none of $\bar{c}^\mathsf{T} K_1$ and $\bar{c}^\mathsf{T} K_2$ are $\gamma \pmod 3$, which implies that one of them is $0 \pmod 3$. This means that one of said detours increased the congruence of C_3 by $\gamma \pmod 3$, and consequently, to arrive at $\bar{c}^\mathsf{T}(C_1 + C_2) \equiv \gamma \pmod 3$, the other detour does the same. As a consequence, both K_1 and K_2 fulfill $\bar{c}^\mathsf{T} K_1 \equiv \bar{c}^\mathsf{T} K_2 \equiv 0 \pmod 3$.

This in turn implies that every edge in C belongs to one cycle of congruence $-\gamma$, and another of congruence 0. Let P be a path as given in the statement of this lemma. Doing a case-by-case analysis, the claim follows from the following two facts:

Case 1: P can be complemented with two paths P_1 and P_2, with $\bar{c}^\mathsf{T} P_1 \not\equiv \bar{c}^\mathsf{T} P_2 \pmod 3$, to two cycles $P + P_1$ and $P + P_2$.

Then we can either immediately decide that the problem is feasible, or the congruence of P is fixed to one possible value $(\bmod 3)$ such that $\bar{c}(P + P_1)$,$\bar{c}(P + P_2) \not\equiv \gamma \pmod 3$.

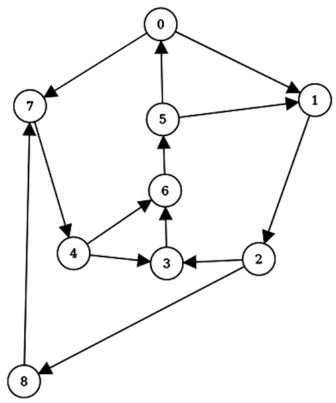
Case 2: There is an internally vertex disjoint path P that is anti-parallel to a path Q in C such that the cycle $P + Q$ is disjoint from some C_j, $j \in \{1, 2, 3\}$.

Then, the problem can immediately be decided to be feasible, unless $\bar{c}^\mathsf{T}(P + Q) \equiv 0 \pmod 3 \Leftrightarrow \bar{c}^\mathsf{T} P \equiv -\bar{c}^\mathsf{T} Q \pmod 3$.

One of the two cases always occurs for P, as a case-by-case analysis shows. □

To be able to immediately read off feasibility from such paths, the assumption that there is no edge that is shared between the three cycles is indeed necessary: In the following example, there is an edge shared by all cycles of non-zero congruence. Only after a reduction to a smaller graph (via Lemma 3.17) we can make more statements about the corresponding problem.

Example 3.19. *Let us assume that* $\gamma = 2$*, and consider the graph depicted below, where the edges* $(5, 1)$*,* $(0, 1)$*,* $(0, 7)$ *shall be endowed with a weight of* 1*, while all others have a weight of zero. Assume further that all edges have a capacity of* 1*. Then there is no circulation of congruence* γ *(mod* 3*).*



Note that for each choice of three cycles of congruence 1*, they overlap at least in* $(6,5)$*. Consider a* 5-1*-path that it internally vertex disjoint to the above graph. Setting its weight to either* 0 *or* 1 *yields no feasible circulation, either.*

Chapter 4

Implementation of the bimodular ILP algorithm

To further illustrate the bimodular ILP algorithm of Chapter 2, its main procedure has been implemented in the Python programming language. The main function takes as its input:

- Data $c \in \mathbb{Z}^n$, $b \in \mathbb{Z}^m$, $A \in \mathbb{Z}^{m \times n}$ bimodular,
- an algorithm $\mathcal{A}$ to solve LPs, and return an optimal vertex if one exists,
- an algorithm $\mathcal{B}$ which returns, for a (transposed) network matrix, its graph representation,
- an algorithm $\mathcal{C}$ which returns Seymour's TU decomposition of a TU matrix (see, e.g., [36] for an algorithm finding a k-sum decomposition),
- a sub-modular function minimization oracle,
- a shortest path algorithm,
- an algorithm which, given a network matrix, returns its graph representation,
- and an oracle which, given a matrix $B \in \mathbb{Z}^{k \times n}$ of full column rank, returns an invertible $(n \times n)$-submatrix of B.

It returns an optimal solution, given it exists, 'math.inf' if the problem is unbounded, or '-math.inf' if it is infeasible.

It can be found on the following link:
`http://doi.org/10.5905/ethz-1007-247`.

Note that the time complexity of this implementation also depends on the oracles as well as the Python packages used and may therefore deviate from what is theoretically possible.

Bibliography

[1] G. Appa and B. Kotnyek. Rational and integral k-regular matrices. **Discrete Mathematics**, 275(1–3):1–15, 2004.

[2] G. Appa, B. Kotnyek, K. Papalamprou, and L. Pitsoulis. Optimization with binet matrices. **Operations Research Letters**, 35(3):345–352, 2007.

[3] E. M. Arkin, C. H. Papadimitriou, and M. Yannakakis. Modularity of cycles and paths in graphs. **Journal of the ACM**, 38(2):255–274, April 1991.

[4] S. Artmann, F. Eisenbrand, C. Glanzer, T. Oertel, S. Vempala, and R. Weismantel. A Note on Non-Degenerate Integer Programs with Small Subdeterminants. **Operations Research Letters**, 44(5):635–639, 2016.

[5] S. Artmann, T. Oertel, and R. Weismantel. Non-degenerate integer linear programs with bounded sub-determinants. Master's thesis, 2016.

[6] S. Artmann, R. Weismantel, and R. Zenklusen. A Strongly Polynomial Algorithm for Bimodular Integer Linear Programming. In **Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing**, New York, NY, USA, 2017. ACM.

[7] A. Barvinok. **A Course in Convexity**. Graduate studies in mathematics. American Mathematical Society, 2002.

[8] S. Baum and L. E. Trotter. Integer rounding and polyhedral decomposition for totally unimodular systems. In Rudolf Henn, Bernhard Korte, and Werner Oettli, editors, **Optimization and Operations Research**, pages 15–23, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

[9] D. Bertsimas and J. N. Tsitsiklis. **Introduction to linear optimization**. Athena Scientific, 1997.

[10] R. E. Bixby and W. H. Cunningham. Short Cocircuits in Binary Matroids. **European Journal of Combinatorics**, (8):213–225, 1987.

[11] A. Bock, Y. Faenza, C. Moldenhauer, and A. J. Ruiz-Vargas. Solving the stable set problem in terms of the odd cycle packing number. In **Proceedings of the 34th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)**, pages 187–198, 2014.

[12] N. Bonifas, M. Di Summa, F. Eisenbrand, N. Hähnle, and M. Niemeier. On sub-determinants and the diameter of polyhedra. **Discrete & Computational Geometry**, 52(1):102–115, 2014.

[13] C. Burch, R. Carr, S. Krumke, M. Marathe, C. Phillips, and E. Sundberg. A decomposition-based pseudoapproximation algorithm for network flow inhibition. In **Network Interdiction and Stochastic Integer Programming**, chapter 3, pages 51–68. Springer, 2003.

[14] C. Carathéodory. Über den variabilitätsbereich der fourierschen konstanten von positiven harmonischen funktionen. **Rendiconti del Circolo Matematico di Palermo (1884-1940)**, 32(1):193–217, 1911.

[15] S. Chestnut and R. Zenklusen. Interdicting structured combinatorial optimization problems with $0, 1$-objectives. **Mathematics of Operations Research**, 42(1):144–166, 2017.

[16] M. Dyer and A. Frieze. Random walks, totally unimodular matrices, and a randomised dual simplex algorithm. **Mathematical Programming**, 64(1):1–16, 1994.

[17] F. Eisenbrand and S. Vempala. Geometric Random Edge, 2016. `https://arxiv.org/abs/1404.1568v5`.

[18] M. A. Frumkin. An algorithm for the reduction of a matrix of integers to triangular form with power complexity of the computations (in Russian). **Ekonomika i Matematicheskie Metody**, 12:173–178, 1976.

[19] A. Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. **Comptes rendus hebdomadaires des séances de l'Académie des Sciences**, 1962.

[20] C. Glanzer, T. Oertel, and R. Weismantel. A combinatorial approach towards integer programming with small sub-determinants. Master's thesis, 2016.

[21] M. X. Goemans and V. S. Ramakrishnan. Minimizing submodular functions over families of sets. **Combinatorica**, 15(4):499–513, 1995.

[22] O. Goldreich. **Computational complexity - a conceptual perspective.** Cambridge University Press, 2008.

[23] D. V. Gribanov and S. I. Veselov. On integer programming with bounded determinants. **Optimization Letters**, 10(6):1169–1177, 2016.

[24] Dmitry Gribanov and Sergey Veselov. On integer programing with bounded determinants. **Arxiv:1505.03132**, 2015.

[25] J. W. Grossman, D. M. Kulkarni, and I. E. Schochetman. On the minors of an incidence matrix and its smith normal form. **Linear Algebra and its Applications**, 218:213 – 224, 1995.

[26] M. Grötschel, L. Lovász, and A. Schrijver. **Geometric Algorithms and Combinatorial Optimization**, volume 2 of **Algorithms and Combinatorics**. Springer, second corrected edition, 1993.

[27] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. **J. ACM**, 48:761–777, July 2001.

[28] R. M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, **Complexity of Computer Computations**, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[29] L. G. Khachiyan. Polynomial algorithms in linear programming (in russian). **USSR Computational Mathematics and Mathematical Physics**, 20(1):53–72, 1980.

[30] A. Y. Khinchine. A quantitative formulation of Kronecker's theory of approximation. **Izvestiya Akademii Nauk SSR Seriya Matematika.**, 12(2):113–122, 1948.

[31] B. Korte and J. Vygen. **Combinatorial Optimization, Theory and Algorithms**. Springer, 5th edition, 2012.

[32] H. W. Lenstra. Integer programming with a fixed number of variables. **Mathematics of operations research**, 8(4):538–548, 1983.

[33] Hendrik W. Lenstra. Integer programming with a fixed number of variables. **Math. Oper. Res.**, 8:538–548, 1983.

[34] M. Nägele, B. Sudakov, and R. Zenklusen. Submodular minimization under congruency constraints. In **Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms**, SODA '18, pages 849–866, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics.

[35] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. **Mathematical Programming**, 118:237–251, 2009.

[36] A. Schrijver. **Theory of Linear and Integer Programming**. John Wiley and Sons, NY, 1986.

[37] A. Schrijver. A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time. **Journal of Combinatorial Theory, Series B**, 80(2):346–355, November 2000.

[38] A. Schrijver. **Combinatorial Optimization - Polyhedra and Efficiency**. Springer, 2003.

[39] P. D. Seymour. Decomposition of regular matroids. **Journal of Combinatorial Theory, Series B**, 28(3):305–359, 1980.

[40] E. Tardos. A Strongly Polynomial Algorithm to Solve Combinatorial Linear Programs. **Operations Research**, 34(2):250–256, 1986.

[41] S. I. Veselov and A. J. Chirkov. Integer program with bimodular matrix. **Discrete Optimization**, 6(2):220–222, 2009.

[42] V. A. Yemelichev, M. M. Kovalev, and M. K. Kravtsov. **Polytopes, Graphs and Optimisation**. Cambridge University Press, New York, NY, USA, 1986.

Appendix A

Copyright

Parts of the introduction appeared in [5].

Parts of the introduction and most of the bimodular chapter of this thesis was previously published in [6]:

Appendix B

Deferred proofs of Chapter 2

B.1 Deferred proofs for the equivalence of (BIP), (CPTU) and (PTU)

When outlining how the three main problem types we deal with can be strongly polynomially reduced to each other, we postponed some technical proofs to this Section.

B.1.1 LP relaxations of (BIP)

We give proofs on how to find an optimal vertex solution for the LP-relaxation of a (BIP)-problem, or a certificate for unboundedness, and how to reduce any (BIP)-problem to one whose LP-relaxation has a unique optimal vertex solution.

Proof of Lemma 2.7. We find an invertible $(n \times n)$ submatrix Q of A and, using Gaussian elimination, which is strongly poly-time (see, for example, [36], Section 3.3), calculate Q^{-1}. AQ^{-1} has the identity matrix as a submatrix, and all $(n \times n)$-determinants of AQ^{-1} are bounded by 1 in absolute value. Furthermore, $|\det(Q)| \leqslant 1$, which implies that all non-zero entries of AQ^{-1} are 1/2 or 1 in absolute value. We may now apply an LP-algorithm, such as the one given by Tardos [40], that finds an optimum x^* for $\max\{c^\mathsf{T} Q^{-1}x \colon AQ^{-1}x \leqslant b\}$ in strongly polynomial time, or decides unboundedness or infeasibility. Strictly speaking, Tardos does not specify whether, if a solution is returned by the algorithm, it is a vertex solution, and whether a certificate of unboundedness is returned, otherwise, both of which can be achieved using standard techniques:

1. If the LP is optimal and bounded, but the optimal solution x^* is not a vertex, we can restrict ourselves to a minimal face that x^* is in. To this end, denote by $A_{I,\cdot}Q^{-1}x \leqslant b_I$, for $I \subseteq [m]$, the maximal sub-system of inequalities that are tight at x^*. If $I = \varnothing$, let $d = [1, 0, \ldots, 0]$. Otherwise, using Gaussian elimination, we can in strongly polynomial time determine the rank l of $A_{I,\cdot}$ and a non-zero solution $d \in \mathbb{Z}^n$ to $A_{I,\cdot}Q^{-1}x = 0$. Let $\lambda = \min\{\frac{b_i - A_{i,\cdot}Q^{-1}x^*}{A_{i,\cdot}Q^{-1}d}\}$. Then $x^* + \lambda d$ fulfills all the constraints indexed

by I, as well as an additional constraint indexed by i^*, with equality, and $\operatorname{rank}(\begin{bmatrix} A_{I,\cdot} \\ A_{i^*,\cdot} \end{bmatrix}) = l+1$. We re-iterate this procedure until $l = n$.

2. If the LP-relaxation is unbounded, we first obtain a feasible solution by replacing the objective function by another which is surely bounded, e.g. by $A_{1,\cdot}$, and obtain a feasible vertex as described above.

   There exists an $r' \in \mathbb{Z}^n$ in the recession cone such that $c^\top Q^{-1} r' > 0$ and $AQ^{-1}r' \leqslant 0$. We can find such an r' e.g. by solving the LP $\max\{c^\top x \colon AQ^{-1}x \leqslant 0,\ 0 \leqslant x \leqslant 1\}$. The improving ray r is then obtained from r' as $r = Q^{-1}r'$. If r is not integral, we rescale it, meaning that if we write each entry as $r_i = \frac{p_i}{q_i}$, for $p_i \in \mathbb{Z}$, $q_i \in \mathbb{Z}_{\neq 0}$, we return $\max\{|q_i| \colon i \in \{1, \dots, n\}\} \cdot r$ together with the feasible solution.

□

Proof of Lemma 2.8. Let us assume first that the LP-relaxation is feasible and bounded, but that there is not a unique optimal solution. By Lemma 2.7, we can find an optimal vertex solution v. We replace the objective function vector c by $2m\bar{c}$, where $\bar{c}^\top := c^\top + \frac{1}{2m}\sum_{i \in I} A_{i,\cdot}$, and $I \subseteq \{1, \dots, m\}$ denotes the set of tight constraints at v. If $v \in \mathbb{Z}^n$, we may return v as a solution for (BIP). So let $v \notin \mathbb{Z}^n$. We now argue that every solution that is optimal for the new problem also was optimal before. To this end, let $z \in \mathbb{Z}^n$ be a well-structured optimal solution fulfilling $|A_I z| \leqslant 1$, and y be some feasible, but not optimal, solution. Since $z \neq v$, we have that $A_I z \neq b_I$, and thus

$$\begin{aligned} \bar{c}^\top (z - v) &= c^\top (z - v) + \frac{1}{2m}\sum_{i \in I} A_{i,\cdot}(z - v) \geqslant c^\top (z - v) - \frac{1}{2}, \\ \bar{c}^\top (y - v) &= \underbrace{c^\top (y - v)}_{\leqslant c^\top (z-v) - 1} + \frac{1}{2m}\sum_{i \in I} \underbrace{A_{i,\cdot}(y - v)}_{\leqslant 0} \leqslant c^\top (z - v) - 1, \end{aligned}$$

and thus, $\bar{c}^\top y < \bar{c}^\top z$.

Let us now consider the case that the LP is unbounded, which means that by Lemma 2.8, we have a feasible vertex solution v and an improving ray $r \in \mathbb{Z}^n$. Then (BIP) is either infeasible or unbounded. To reduce the question of feasibility to a bounded problem of type (BIP), we replace c by an objective function vector such that the resulting problem is surely bounded. To this end, we may choose to maximize $A_{1,\cdot}$, and, if the problem is feasible with optimal solution z, return z and r as a certificate for unboundedness of the original (BIP)-problem. □

B.1.2 Further deferred proofs of Section 2.3

In this Section, we first give the proof of Veselov and Chirkov [41] for their theorem, give a technical result and then show how to strongly polynomially reduce (PTU) with constraint matrix T to a problem (CPTU) whose constraint matrix is a submatrix of T.

Proof of Lemma 2.9. The entries of Q^{-1} are half-integral. There exists a unimodular matrix $U \in \mathbb{Z}^{n\times n}$ corresponding to elementary column operations on Q such that QU is in Hermite Normal Form (HNF) [18]. In particular, we may choose U such that $QU \geqslant 0$ and that it is equal to the identity matrix with the exception of one row i, where

$$(QU)_{i,j} \begin{cases} \leqslant 1, & \text{if } j < i, \\ = 2, & \text{if } j = i, \\ = 0, & \text{if } j > i \end{cases}.$$

Thus, a single row of $(QU)^{-1}$ contains all of its non-integral entries, and there is at least one such entry. Since $(QU)^{-1} = \bar{U}^{-1}Q^{-1}$ is generated from Q^{-1} by elementary row operations, there exists a submatrix of Q^{-1} consisting of all of its non-integral entries. Put differently, there is $I, J \subseteq \{1, \ldots, n\}$ such that $Q^{-1}_{i,j} \notin \mathbb{Z} \Leftrightarrow (i \in I, j \in J)$. □

Proof of Lemma 2.12. We will reduce (PTU) to (CPTU) via (BIP) in a way such that the resulting constraint matrix is a submatrix of T in (PTU).

Given a problem of form (PTU) with T as the constraint matrix, we reformulate it as the (BIP) problem

$$\max\left\{ \begin{pmatrix} c \\ 0 \end{pmatrix}^{\mathsf{T}} \begin{bmatrix} x \\ y \end{bmatrix} : \begin{bmatrix} T & 0 \\ \chi^S & 2 \\ -\chi^S & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leqslant \begin{bmatrix} b \\ \alpha \\ -\alpha \end{bmatrix},\ \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{Z}^{n+1} \right\}.$$

By Theorem 2.2, we can solve the LP-relaxation of this problem in strongly polynomial time. First, assume that the relaxed problem is bounded, and let v be an optimal vertex solution. Note that by dividing the last column in the constraint matrix by 2, we obtain a unimodular problem with a vertex equal to v in all but the last variable. Thus, $v_{1:n}$ is integral.

If v_{n+1} is integral, we return $v_{1:n}$. Otherwise, by Lemma 2.14 we may neglect all but those constraints

$$\tilde{T} := \begin{bmatrix} \hat{T} & 0 \\ \chi^S & 2 \\ -\chi^S & -2 \end{bmatrix}$$

that are tight with respect to v, where $\hat{T}$ is a submatrix of T, and solve

$$\max\left\{ c^{\mathsf{T}}x \colon \tilde{T}\begin{bmatrix} x \\ y \end{bmatrix} \leqslant \begin{bmatrix} 0 \\ \alpha \\ -\alpha \end{bmatrix},\ v + \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{Z}^{n+1} \right\}.$$

Since $v_{1:n} \in \mathbb{Z}^n$, we may reformulate the above as

$$\max\{c^{\mathsf{T}}x \colon \hat{T}x \leqslant 0,\ x \in \mathbb{Z}^n,\ x(S) \text{ odd}\},$$

where $\hat{T}$ is now a submatrix of T.

If the LP-relaxation of the (BIP) problem is unbounded, then by Lemma 2.8, we can reduce this to a (BIP) problem with the same constraint matrix and right-hand-side that has a unique LP-optimum v. By the same arguments as above, this problem can then be reduced to a (CPTU) problem. □

B.2 Deferred statements and proofs on our variant of Seymour's TU-decomposition

This subsection is dedicated to strengthening Theorem 1.14 so that it yields Theorem 2.4.

We first want to examine in greater detail the case covered in Lemma 1.15 that core(T) is neither a network matrix, nor one of the two matrices (2.1) of constant size.

(CPTU) problems have non-negativity constraints. Reductions on other optimization problems that involve sign changes are therefore inconvenient for us, and so we find a way to decompose a constraint matrix as a k-sum without applying sign changes of rows or columns.

Lemma B.1. If core(T) in Lemma 1.15 can, after row/column sign changes and permutations, be written as a k-sum with $m_L + n_L$, $m_R + n_R \geqslant 4$, then we can in polynomial time find a way of writing it as a k-sum that also fulfills $m_L + n_L \geqslant 4$, $m_R + n_R \geqslant 4$ after performing row and column permutations only.

Proof. We apply Seymour's decomposition as given in Lemma 1.15: We permute rows and columns and perform sign changes of rows and columns of T and write the resulting matrix as a k-sum, $k \in \{1, 2, 3\}$, of two totally unimodular matrices. We then find an alternative decomposition that does not involve sign changes anymore, where the general plan is as follows: For all row sign changes, we choose a corresponding row in one of the summands and change their signs. We do the same for columns. Then these new summands will still be TU, and once we show that their k-sum is (up to row- and column-permutations) the matrix T, we are done.

We now give an inductive reasoning on how to find the correct sign changes for the summands: assume that (possibly after row/column permutations) $T = \begin{bmatrix} L & D_1 \\ D_2 & R \end{bmatrix}$ and that after a single row or column sign change, it can be written as a k-sum $M_1 \oplus_k M_2$ of two TU-matrices M_1 and M_2. Let us first assume that the sign change involved affects a row $T_{i,\cdot} = [L_{i,\cdot} \mid (D_1)_{i,\cdot}]$ in the first n_L rows of T. Depending on k, we have $D_1 = 0$ or $D_1 = ad^{\mathsf{T}}$. If $k = 1$ and $D_1 = 0$, we change the sign of row i in M_1. Call this new matrix $\widehat{M_1}$. Then, $T = \widehat{M_1} \oplus_1 M_2$, as desired. Otherwise, $T_{i,\cdot} = [L_{i,\cdot} \mid a_i d^{\mathsf{T}}]$. If $k = 2$, replace row i in the left summand by its negative, $[L_{i,\cdot} \mid a_i]$ and call this new matrix $\widehat{M_1}$. Then, $T = \widehat{M_1} \oplus_2 M_2$. Analogously, if $k = 3$, replace row i in the left summand by its negative.

Let us treat the case that we change the sign of a row i, $n_L + 1 \leqslant i \leqslant n$, next. Then $T_{i,\cdot} = [(D_2)_{i,\cdot} \mid R_{i,\cdot}]$. If $k = 1$ or $k = 2$, $D_2 = 0$ and we obtain a decomposition of T via a 1-sum by changing the sign of row $i - n_L$ in M_2.

Otherwise, $k = 3$ and $T_{i,\cdot} = [g_{i-n_L} f^\top \mid R_{i,\cdot}]$. We change the sign of row $i - n_L + 1$ in $\begin{bmatrix} 1 & 0 & d^\top \\ g & g & R \end{bmatrix}$ to obtain a 3-sum decomposition of T.

The next case to be treated is that the sign change affects a column. In case of the 1-sum, an analogous reasoning applies as above. If the column whose sign we change is $T_{\cdot,j} = \begin{bmatrix} L_{\cdot,j} \\ (D_2)_{\cdot,j} \end{bmatrix}$, we can replace, if $k = 2$, $L_{\cdot,j}$ by $-L_{\cdot,j}$ in the left summand of the 2-sum, or, if $k = 3$, $\begin{bmatrix} L_{\cdot,j} \\ f_j g^\top \end{bmatrix}$ by its negative in the left summand of the 3-sum. Finally, if $T_{\cdot,j} = \begin{bmatrix} d_{j-n_L} a \\ R_{j-n_L} \end{bmatrix}$, we can change the sign of the same column in the right summand, which is in position $j - n_L$ for the 2-sum and $j - n_L + 2$ for the 3-sum.

□

To make progress in the recursion, we need that the new (CPTU) problems and (PTU) problems have strictly less rows. This will follow from the following lemma, which says that both summands in the k-sum decomposition have a minimum number of rows:

Lemma B.2. Under the assumptions of case 3 in Theorem 1.14, i.e. in the setting of Lemma 1.15, we have that $m_L \geqslant 2$ and $m_R \geqslant 2$.

Proof of Lemma B.2. We show that there is a standard unit vector, its negative ($\pm e$) or two linearly dependent rows/columns in core(T) if the statement is violated, contradicting the definition of a core.

1. $k = 1$: If $m_L = 1$, $\begin{bmatrix} L \\ D_2 \end{bmatrix}$ consists of at least three standard unit vectors (or their negatives), contradicting that core(T) is a core. The analogous reasoning applies if $m_R = 1$.

2. $k = 2$, $m_L = 1$: L has only one row and thus, all vectors $\begin{bmatrix} L \\ 0 \end{bmatrix}$ are (up to signing) standard unit vectors.

3. $k = 2$, $m_R = 1$: Write $D_1 = ad^\top$. R consists of one row only, and can thus R can be interpreted as a row vector with at least three entries (since $m_R + n_R \geqslant 4$). By TU-ness of core(T), there can be no submatrix of $\begin{bmatrix} D_1 \\ R \end{bmatrix}$ of the form $\pm \begin{bmatrix} a & -a \\ 1 & 1 \end{bmatrix}$, and since the matrix is a core, R has at most one entry that is zero. Then, however, one of the following matrices is a submatrix of core(T), contradicting that it is a core:

$$\pm \begin{bmatrix} a & -a \\ 1 & -1 \end{bmatrix}, \quad \pm \begin{bmatrix} a & a \\ 1 & 1 \end{bmatrix}.$$

4. $k = 3$, $m_L = 1$: L has one row only, and by TU-ness of $\begin{bmatrix} L \\ gf^\mathsf{T} \end{bmatrix}$, there is no submatrix $\pm \begin{bmatrix} 1 & -1 \\ g & g \end{bmatrix}$ of $\begin{bmatrix} L \\ gf^\mathsf{T} \end{bmatrix}$. Since $m_L + n_L \geqslant 4$, there have to be at least three columns in L and thus two that are linearly dependent.

5. $k = 3$, $m_R = 1$: R has one row only, and by TU-ness of $\begin{bmatrix} ad^\mathsf{T} \\ R \end{bmatrix}$, there is no submatrix $\pm \begin{bmatrix} a & -a \\ 1 & 1 \end{bmatrix}$ of $\begin{bmatrix} ad^\mathsf{T} \\ R \end{bmatrix}$. Since $m_R + n_R \geqslant 4$, there have to be at least three columns in R and thus two that are linearly dependent.

□

So far, the matrix decompositions we considered only deal with the core of a matrix. For our purposes, however, we need k-sum decomposition of the entire constraint matrix T. The next lemma shows how to efficiently obtain one from the decomposition of core(T). We first show a statement for one row or column only, which, when iterated, will yield the desired decomposition.

Lemma B.3. Let $T \in \{0, \pm 1\}^{m \times n}$ be a TU matrix which can be written as a k-sum, $T = M_L \oplus_k M_R$, $k \in \{1, 2, 3\}$, such that $m_L + n_L$, $m_R + n_R \geqslant 4$. Consider a column vector $r \in \{0, \pm 1\}^m$ that is linearly dependent on one of the columns of T, or has at most one non-zero entry. Then, the k-sum can in polynomial time be extended onto $\begin{bmatrix} T & r \end{bmatrix}$, i.e., we can (possibly after column permutations) efficiently find another k-sum such that $\begin{bmatrix} T & r \end{bmatrix} = N_L \oplus_k N_R$ for two TU-matrices N_L and N_R such that M_L and M_R are submatrices of N_L and N_R, respectively.

Furthermore, if $s \in \{0, \pm 1\}^n$ has at most one non-zero entry or is linearly dependent of one of the rows of T, then the k-sum of T can (possibly after row permutations) be efficiently extended to a k-sum of $\begin{bmatrix} T \\ s^\mathsf{T} \end{bmatrix}$.

Proof. We find the new k-sum decomposition as follows: If $r = 0$, we append an all-zero column to the left summand. It will stay totally unimodular, the k sum is up to column permutations equal to $\begin{bmatrix} T & d \end{bmatrix}$. Analogously, if r has exactly one non-zero entry d_i in position i, we do the following: If $i \leqslant m_L$, we truncate r and append it as a column to the left summand. If $i > m_L$, we append a column vector consisting of the last m_R (if $k = 1, 2$) or the last $m_R + 1$ (if $k = 3$) entries of r to the right summand. This way, the summands stay TU and their k-sum is (up to column permutations) equal to T.

Assume now that r is linearly dependent of column i, $1 \leqslant i \leqslant n_L$. Then, $r = \alpha T_{\cdot,i} = \alpha \begin{bmatrix} L_{\cdot,i} \\ \sigma g \end{bmatrix}$, where $\alpha \in \{\pm 1\}$ and $\sigma \in \{0, \pm 1\}$. By doubling the ith column of the left summand and multiplying the new column by α, the matrix stays TU, and the k-sum where the left summand is replaced with this modified matrix yields (up to column permutations) T.

If r is linearly dependent of column i, $n_L < i \leqslant n$, then $T_{\cdot,i} = \alpha \begin{bmatrix} \sigma a \\ R_{\cdot,(i-n_L)} \end{bmatrix}$, where $\alpha = \pm 1$ and $\sigma \in \{0, \pm 1\}$. We now append a (signed by α) copy of column $i - n_L$ in the right summand.

Analogously, we find a new k-sum decomposition for $\left[\begin{smallmatrix} T \\ s^\mathsf{T} \end{smallmatrix}\right]$: If it is an all-zero vector, we append an all-zero row to the left summand. If s has exactly one non-zero entry in position i, we append its truncated counterpart to the left summand if $1 \leqslant i \leqslant n_L$, and to the right one, otherwise. Finally, if s is linearly dependent of another row $T_{i,\cdot}$ in T, we append a corresponding row to the left summand if $1 \leqslant i \leqslant m_L$ and to the right, otherwise. □

By definition, T arises from core(T) by iteratively appending rows and columns of the above type. We can therefore apply Lemma B.3 to obtain the k-sum decomposition of T:

Lemma B.4. Let $T \in \mathbb{Z}^{m \times n}$ be a TU-matrix such that core(T) can be written as a k-sum, core(T) $= M_L \oplus_k M_R$, $k \in \{1, 2, 3\}$, of two TU-matrices M_L and M_R with $m_L + n_L \geqslant 4$ and $m_R + n_R \geqslant 4$. Then we can efficiently find a way to write T (up to row and column permutations) as a k-sum that extends the original one, i.e., we can write, possibly after row/column permutations, T as $N_L \oplus N_R$, N_L and N_R TU, where M_L is a submatrix of N_L and M_R is a submatrix of N_R.

Finally, we need to tackle case 4 in Lemma 1.15, where a pivot operation is applied to the constraint matrix before it can be decomposed as a 3-sum. For this we observe that in a certain sense, the core of a matrix is invariant under pivoting, which will then enable us to prove Theorem 2.4 further below.

Lemma B.5 (Invariance of canonical and redundant columns under pivoting)**.** Let $T \in \{0, \pm 1\}^{m \times n}$ be a TU-matrix and $d \in \{0, \pm 1\}^m$ be a vector with at most one non-zero entry that is ± 1, or correspond to (the negative of) a column of T. Consider $\begin{bmatrix} T & d \end{bmatrix}$, and let M be the matrix resulting from a pivot operation at an element that is not in the last column of $\begin{bmatrix} T & d \end{bmatrix}$. Then, the last column of M still is a vector with only one non-zero entry that is ± 1 or (the negative of) a copy of another column in M.

Proof. Recall that pivoting involves row and column permutations as well replacing a matrix $\begin{bmatrix} T & d \end{bmatrix} = \begin{bmatrix} \varepsilon & c^\mathsf{T} \\ b & D \end{bmatrix}$ by $M := \begin{bmatrix} -\varepsilon & \varepsilon c^\mathsf{T} \\ \varepsilon b & D - \varepsilon b c^\mathsf{T} \end{bmatrix}$. Row and column permutations preserve the support of d and linear dependence between columns, and thus, we may assume that we perform a pivot at $T_{1,1}$. Then, the column of interest fulfills $M_{\cdot,n} = \begin{bmatrix} \varepsilon d_1 \\ d_{2:m} - \varepsilon d_1 b \end{bmatrix}$.

Consider first the case that d has at most one non-zero entry. If $d_1 = 0$, $M_{\cdot,n} = d$ and the statement is true. Otherwise, $d_{(2:m)} = 0$ and $d_1 = \pm 1$. This implies that $M_{\cdot,n} = d_1 \begin{bmatrix} \varepsilon \\ -\varepsilon b \end{bmatrix} = -d_1 M_{\cdot,1}$, and thereby linearly dependent of the first column.

Finally, assume that d is linearly dependent of some other column $g = T_{\cdot,i}$, $1 \leqslant i \leqslant n-1$. If $i \neq 1$, d and g will remain linearly dependent after the pivoting. Otherwise, $i = 1$ and $d = \alpha \begin{bmatrix} \varepsilon \\ b \end{bmatrix}$, for $\alpha \in \{\pm 1\}$. This becomes, after the pivoting, $\begin{bmatrix} \varepsilon^2 \\ b - \varepsilon^2 b \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. □

Lemma B.6. Consider a TU-matrix T that can via pivoting be transformed into a TU-matrix M, and let C be the submatrix of M that corresponds to $\operatorname{core}(T)$. Assume that we can write $C = M_L \oplus_3 M_R$ for two TU-matrices M_L and M_R. Then we can efficiently find a 3-sum decomposition of M, $M = N_L \oplus_3 N_R$, which extends the first, i.e., such that M_L and M_R are submatrices of N_L and N_R, respectively.

Proof. Let $I \subset \{1, \ldots, m\}$ be a subset of rows, $J \subseteq \{1, \ldots, n\}$ a subset of columns such that $\operatorname{core}(T) = T_{I,J}$. By Lemma B.5, canonical and redundant columns, and, by symmetry, rows, remain canonical or redundant after pivoting. Therefore, the index sets specifying the core of M remain I and J, i.e., $\operatorname{core}(M) = M_{I,J}$. By assumption, $M_{I,J}$ can be written as a 3-sum, and by Lemma B.4, this can efficiently be extended to a 3-sum decomposition of M. □

We now have all the ingredients in place to prove Theorem 2.4.

Proof of Theorem 2.4. The second case is a restatement of case 2 in Theorem 1.14. The first case corresponds to the first in Theorem 1.14, in which $\operatorname{core}(T)$ is a network matrix or its transpose is, where we made use of the fact that if $\operatorname{core}(T)$ is a network matrix, or its transposed is, then so it T or T^{T}, respectively.

The next three cases resemble the first three in Lemma 1.15. However, we need to get rid of sign changes, which we do by invoking Lemma B.1. This gives as a k-sum decomposition of $\operatorname{core}(T)$, and by Lemma B.2, it fulfills m_L, $m_R \geqslant 2$. We invoke Lemma B.4 to obtain a k-sum decomposition of the entire matrix T in strongly polynomial time.

The final case in Theorem 2.4 corresponds to the last case in Lemma 1.15, namely that $\operatorname{core}(T)$ can, after pivoting once, be written as a 3 sum with $m_L + n_L$, $m_R + n_R \geqslant 4$. By Lemma B.6, we can extend this 3-sum decomposition to the entire matrix that we obtained from T after pivoting, and obtain $m_L \geqslant 2$, $m_R \geqslant 2$ and remove the need for sign changes, as before. □

B.3 Deferred proofs on (CPTU) decompositions

Proof of Lemma 2.13. First, we solve the LP-relaxation of the given (CPTU) problem using the algorithm of Tardos [40], which runs in strongly polynomial time since all entries of T are bounded by 1 in absolute value. If the LP is bounded, we are done. Otherwise, we need to find out whether the original problem is feasible. If so, it is unbounded as well. To check feasibility, we exchange the objective function by one that is surely bounded. For example, we might choose $c^\mathsf{T} = T_{1,\cdot}$. In case that this altered (CPTU) problem has a feasible solution x^*, we provide a certificate for unboundedness as follows: We solve the LP $\max\{c^\mathsf{T} x \colon Tx \leqslant 0, 0 \leqslant x \leqslant 1\}$, again with the method of Tardos [40]. We will obtain an optimal solution r with $c^\mathsf{T} r < 0$ this way, and we rescale it to be integral, meaning that if $r_i = \frac{p_i}{q_i}$, for $p_i \in \mathbb{Z}$, $q_i \in \mathbb{Z}_{\neq 0}$, we replace r by $\max\{|q_i| \colon i \in \{1, \ldots, n\}\} \cdot r$. Then x^* together with r certify unboundedness of the problem. □

We postponed some details in our discussion of our recursion if the k-sum involved is a 2-sum. We prove the missing ingredients here.

Proof of Lemma 2.16. We make the non-negativity constraints implicit by appending $-\mathcal{I}$, the negative of the $(n \times n)$-identity matrix, to T, and write $x \in \mathbb{Z}^n$ instead of $x \in \mathbb{Z}_{\geqslant 0}$ in (CPTU).

After performing a column permutation and possibly swapping the first two constraints as well as replacing h by $-h$, we may assume that $h_1 = 1$. We perform column operations on $\begin{bmatrix} T \\ -\mathcal{I} \end{bmatrix}$ such that the first row becomes $[1, 0, \ldots, 0]$.

Let us take a closer look at the constraint after this operation. The first row was changed to $[1 \mid 0]$, while the second row of the matrix is now $[-1 \mid -h_{2:n} - h_1(-h_{2:n})] = [-1, 0, \ldots, 0]$. The row which, before the application of the column operations, corresponded to the first non-negativity constraint, was changed to $g^\mathsf{T} := [-1 \mid h_{2:n}]$. The other rows of $-\mathcal{I}$ were not affected. We can thus, with a permutation matrix P and a unimodular matrix Q, write

$$P \begin{bmatrix} T \\ -\mathcal{I} \end{bmatrix} Q = \begin{bmatrix} e^1 \\ g^\mathsf{T} \\ M \\ -\mathcal{I} \end{bmatrix},$$

where M is totally unimodular and $e^1 = \mathcal{I}_{1,\cdot}$.

Let us discuss what happens to the parity constraint when pivoting. In particular, we will show that it simply gets transformed into a different parity constraint, which, taken together with the observations above, will allow us to conclude that we indeed reduced to a problem that is of type (CPTU) as well. So let $S \subseteq \{1, \ldots, n\}$ be a set for which we require $x(S) \equiv 1 \pmod 2$ in the original (CPTU) problem. We may write this constraint as

$$\chi_S{}^\mathsf{T} x \equiv 1 \pmod 2, \tag{B.1}$$

where $\chi_S \in \{0,1\}^n$ is the characteristic vector of S. Let Q be the unimodular matrix corresponding to the column operations that were performed, i.e., $h^\mathsf{T} Q = [-1, 0, \ldots, 0]$. Then, the constraint (B.1) becomes, after pivoting, $\chi_S{}^\mathsf{T} Q x \equiv 1 \pmod 2$, and since $\chi_S{}^\mathsf{T} Q \in \mathbb{Z}^n$, this can, using the set $\bar{S} = \{s \in \chi_S{}^\mathsf{T} Q \mid s \equiv$

$1 \pmod 2\}$, be rewritten as

$$1 \pmod 2 \equiv {\chi_S}^\mathsf{T} Qx \equiv \sum_{i=1}^{n} ({\chi_S}^\mathsf{T} Q)_i x_i \pmod 2 \equiv {\chi_{\bar{S}}}^\mathsf{T} x \pmod 2,$$

which yields $x(\bar{S}) \equiv 1 \pmod 2$.

We may thus reformulate the (CPTU)-problem as follows, where $\chi_S \in \{0,1\}^n$ is the characteristic vector of set S:

$$\begin{aligned}
&\max\{c^\mathsf{T} x \colon Tx \leqslant 0,\ x(S) \text{ odd},\ x \in \mathbb{Z}^n_{\geqslant 0}\} \\
&= \max\{c^\mathsf{T} Q Q^{-1} x \colon \begin{bmatrix} T \\ -\mathcal{I} \end{bmatrix} Q Q^{-1} x \leqslant 0,\ x(S) \text{ odd},\ x \in \mathbb{Z}^n\} \\
&= \max\{c^\mathsf{T} Q Q^{-1} x \colon P \begin{bmatrix} T \\ -\mathcal{I} \end{bmatrix} Q Q^{-1} x \leqslant 0,\ {\chi_S}^\mathsf{T} x \text{ odd},\ x \in \mathbb{Z}^n\} \\
&= \max\{c^\mathsf{T} Q Q^{-1} x \colon \begin{bmatrix} e^1 \\ g^\mathsf{T} \\ M \\ -\mathcal{I} \end{bmatrix} Q^{-1} x \leqslant 0,\ {\chi_S}^\mathsf{T} Q Q^{-1} x \text{ odd},\ x \in \mathbb{Z}^n\} \\
&= \max\{c^\mathsf{T} Q x \colon \begin{bmatrix} e^1 \\ g^\mathsf{T} \\ M \\ -\mathcal{I} \end{bmatrix} x \leqslant 0,\ {\chi_S}^\mathsf{T} Q x \text{ odd},\ x \in \mathbb{Z}^n\}, \\
&\qquad \text{which, with } \bar{S} := \left\{ i \in \{1,\dots,n\} \colon ({\chi_S}^\mathsf{T} Q)_i \equiv 1 \pmod 2 \right\}, \\
&= \max\{c^\mathsf{T} Q x \colon \begin{bmatrix} e^1 \\ g^\mathsf{T} \\ M \\ -\mathcal{I} \end{bmatrix} x \leqslant 0,\ x(\bar{S}) \text{ odd},\ x \in \mathbb{Z}^n\} \\
&= \max\{c^\mathsf{T} Q x \colon \begin{bmatrix} g^\mathsf{T} \\ M \end{bmatrix} x \leqslant 0,\ x_1 = 0,\ x(\bar{S}) \text{ odd},\ x \in \mathbb{Z}^n_{\geqslant 0}\} \\
&= \max\{c^\mathsf{T} Q x \colon \begin{bmatrix} g^\mathsf{T} \\ M \end{bmatrix}_{\cdot,(2:n)} x \leqslant 0,\ x(\bar{S} \backslash \{1\}) \text{ odd},\ x \in \mathbb{Z}^{n-1}_{\geqslant 0}\}.
\end{aligned}$$

We have thus eliminated the first variable and have found a (CPTU)-problem with one constraint and one dimension less. □

Proof of Lemma 2.17. Let us treat the case first that all problems (2.9) are bounded and feasible, i.e., that $-\infty < \rho_L(\alpha,\beta) < \infty$ for all $\alpha \in \{0,\pm 1\}$, $\beta \in \{0,1\}$. Let $I = \{1,\dots,6\}$ correspond to the first six entries of $\bar{c}$.

1. Let z be feasible for (2.10). Denote by $y(\alpha,\beta)$ an optimal solution for (2.9) for the pair (α,β) and by Y the matrix

$$Y = \begin{bmatrix} y(-1,0) & y(0,0) & y(1,0) & y(-1,1) & y(0,1) & y(1,1) \end{bmatrix}$$

that contains the $y(\alpha,\beta)$'s as columns in the order corresponding to $\bar{c}$. Consider the vector

$$y := \begin{bmatrix} \sum_{i \in I} z_i Y_{\cdot,i} \\ z_{7:(n_R+6)} \end{bmatrix}.$$

$y \geqslant 0$ and $Ry_R = Rz_{7:(n_R+6)} \leqslant 0$. Furthermore, by our choice of Y, we have that for any $\omega \in \{0, \pm 1\}$, ${h_I}^\mathsf{T} z_I = \omega \Rightarrow L \sum_{i \in I} z_i Y_{\cdot,i} \leqslant \omega a$. Since z fulfills

$$-d^\mathsf{T} z_{7:(n_R+6)} = {h_I}^\mathsf{T} z_I,$$

this implies

$$-ad^\mathsf{T} z_{7:(n_R+6)} \geqslant L \sum_{i \in I} z_i Y_{\cdot,i},$$

and so

$$Ly_L + ad^\mathsf{T} y_R = L \sum_{i \in I} z_i Y_{\cdot,i} + ad^\mathsf{T} z_{7:(n_R+6)} \leqslant 0.$$

y is thus feasible for (CPTU). Furthermore, $c^\mathsf{T} y = \bar{c}^\mathsf{T} z$.

2. Denote by x^* an optimal solution for the original problem (CPTU) with constraint matrix T that fulfills $|d^\mathsf{T} x^*| \leqslant 1$. We write it as $\begin{bmatrix} x_L^* \\ x_R^* \end{bmatrix}$, where $x_L^* = x^*_{(1:n_L)}$ corresponds to the left, and $x_R^* = x^*_{(n_L+1):n}$ to the right part of T.

   We show once more that x^* can be transformed into to a feasible solution for (2.10), which will imply that problem (2.10) cannot have a smaller optimal objective function value than $c^\mathsf{T} x^*$.

   x_L^* is a feasible solution to (2.9) for some $\alpha = \alpha_L$ and some β and x_R^* to (2.7), for $\alpha_R = \alpha_L = -d^\mathsf{T} x_R^*$ and $\beta \equiv x_L^*(S^L) \pmod 2$.

   Let l be the index of the column in the constraint matrix of (2.10) that corresponds to problem (2.9) with $\alpha_L = -d^\mathsf{T} x_R^*$ and $\beta \equiv x_L^*(S^l) \pmod 2$. Then by construction, $\begin{bmatrix} e_l \\ x_R^* \end{bmatrix}$ is feasible and

$$c^\mathsf{T} x^* = \bar{c}^\mathsf{T} \begin{bmatrix} e_{l_L} \\ x_R^* \end{bmatrix},$$

   where e_l is the l-th standard unit vector.

As in the proof of Lemma 2.15, we need to argue that we can treat the case that some $\rho_L(\alpha, \beta)$ are ∞ or $-\infty$ as well. Again, we delete the corresponding columns and entries in $\bar{c}$, which can be interpreted as adding additional constraints that enforce that some variables are set to zero. Since we did not add additional feasible solutions to this problem, the first statement of the lemma still holds. As for the second statement, we need to argue why $\begin{bmatrix} e_l \\ x_R^* \end{bmatrix}$ is still feasible: Assume it was not, then variable l was deleted because the corresponding problem (2.9) is unbounded. Choose any $w \in \mathbb{Z}^{n_L}$ that is feasible for this problem, and that fulfills ${c_L}^\mathsf{T} w > {c_L}^\mathsf{T} x_L^*$. Then $\begin{bmatrix} w \\ x_R^* \end{bmatrix}$ is feasible for the original (CPTU) problem as well with strictly larger value than x^*, a contradiction. □