Solving bimodular integer programs in strongly polynomial time

Candidate Number: 37599

A Dissertation submitted to the Department of Mathematics of the London School of Economics and Political Science for the degree of Master of Science

December 1, 2020

Abstract

We analyse the paper *A Strongly Polynomial Algorithm for Bimodular Integer Linear Programming* published in 2017 by Artmann, Weismantel and Zenklusen as part of the conference proceedings of $STOC'17$ [2]. Artmann et al. give a strongly polynomial algorithm for solving integer programs of the form $\max\{c^\top x : Ax \leqslant b, x \in \mathbb{Z}^n\}$, with $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$, and where $A \in \mathbb{Z}^{m\times n}$ has rank n and all its $n \times n$ subdeterminants are bounded in absolute value by 2.

We give a complete exposition of the algorithm presented in the paper and we recreate a few proofs which were missing. Furthermore, we give two small contributions to the topic. First, we prove that determining the feasibility and boundedness of a bimodular integer program can be done independently from solving the optimization problem, and second, we show that given any general optimal solution to a bimodular integer program where the feasible region of the LP relaxation is a translated cone, one can find an optimal solution which is a vertex in strongly polynomial time.

Contents

1. Introduction

While it has been proved that linear programs[1] can be solved in polynomial time [10], the integer linear programming (ILP) problem,

$$\max\{c^\top x : Ax \leqslant b, x \in \mathbb{Z}^n\} \tag{ILP}$$

with $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$ is known to be $\mathcal{NP}$-complete[2] (see Section 18.2 in Schrijver [11]). Nevertheless, this does not mean that all ILP instances are $\mathcal{NP}$; in fact, there are classes of integer linear programs which can be solved in polynomial time. Probably the most well-known example of such classes is the ILP problem with totally unimodular (TU) constraint matrix, that is, all square submatrices of A have determinants equal to $-1, 0$ or 1. The total unimodularity of A ensures that the vertices of the region $\{x \in \mathbb{R}^n : Ax \leqslant b\}$ are integral, allowing one to solve the problem by solving its LP relaxation.

Work has been done to extend the result beyond ILP problems with TU matrices. Veselov and Chirkov proved that non-degenerate 2-modular ILP's can be solved in polynomial time [14], and their result was later extended to include all non-degenerate Δ-modular ILP's [1], where Δ-*modular* means that all $n \times n$ subdeterminants of A have absolute value bounded by Δ and where *non-degenerate* means that all $n \times n$ subdeterminants are non-zero.

Unfortunately, degeneracy is quite common in mathematical optimization (see [15]), and therefore one has to take it into account to properly extend the class of combinatorial optimization problems which can be solved efficiently. In 2017, Artmann, Weismantel and Zenklusen proved that 2-modular (also called bimodular) integer linear programs can be solved in polynomial time [2]. In their paper, they present a *strongly polynomial* time algorithm to solve *bimodular integer programs*.

[1]We shall assume familiarity with the geometry of linear programming; for an introduction we refer the reader to the first two chapters of Bertsimas and Tsitsiklis's book [3].

[2]We refer the reader to Chapter 34 in Cormen et al. [6] for a formal introduction to complexity theory.

Definition 1.1. *Let $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$ and $A \in \mathbb{Z}^{m\times n}$ with rank$(A) = n$ such that each $n\times n$ subdeterminant of A is at most 2 in absolute value. A* ***bimodular integer program*** *(BIP) is an optimization problem of the form*

$$\max\{c^\top x : Ax \leqslant b, x \in \mathbb{Z}^n\} \tag{BIP}$$

Definition 1.2. *Assuming the elementary operations of addition, subtraction, multiplication, division and comparison of two integers can be done in unit time, then an algorithm is called* ***strongly polynomial*** *if the number of elementary operations performed is polynomially bounded by the input size.*

As explained in Artmann et al., some classical combinatorial optimization problems can be formulated as bimodular integer programs. Among these, we have the minimum odd $s - t$ cut problem and the one of finding a maximum weight independent set in a graph with an odd cycle packing number of 1.

We note that one cannot expect to extend the result of Artmann et al. to all Δ-modular ILP problems with $\Delta > 0$, since for large Δ, the problem approaches the setting of a general ILP which, as already mentioned, is $\mathcal{NP}$-complete.

1.1 Focus of the dissertation

This dissertation is based on the work of Artmann, Weismantel and Zenklusen [2], *A Strongly Polynomial Algorithm for Bimodular Integer Linear Programming.* Their main result is the following

Theorem 1.3 (Theorem 1.1 of Artmann et al. [2])**.** *There is a strongly polynomial algorithm for solving bimodular integer programs.*

The algorithm is elaborate and composed of many parts. Due to size limitations, the authors skipped several proofs with the intention of publishing them in an extended version of the paper. Since a longer version of the paper has yet to be published, we decided to take it on ourselves to recreate the missing proofs and give a complete exposition of the algorithm.

The authors dedicate a chapter to each main component of the algorithm. As they put it, their chapters address "the reduction of BIP to a better-structured problem, the recursive decomposition of this well-structured problem into simpler base block problems, and the efficient resolution of these base block problem". We follow the same structure as Artmann et al., but deviate from their approach on some occasions to offer a clearer exposition of the result. At each such deviation, we also offer an explanation of their alternative approach.

In what follows, we present a modified version of the algorithm by Artmann et al. and a complete proof that such an algorithm solves BIP in strongly polynomial time. Furthermore, we discuss Veselov and Chirkov's [14] results on bimodular matrices which are fundamental for the realization of the algorithm. Our original contributions can be summarized as follows:

(*i*) In Chapter 3, we prove that the polynomial procedure for finding a feasible point of a BIP proposed by Veselov and Chirkov [14] can be improved to run in strongly polynomial time. This allows us to determine the feasibility and boundedness of BIP independently from solving the optimization problem.

(*ii*) In Lemma 4.5, we prove that for a specific case of BIP, given any optimal solution, one can find an optimal solution which is a vertex in strongly polynomial time. To do so, we exploit the fact that for this particular class of BIP problems, each vertex satisfies $n-1$ inequalities of the constraint matrix at equality.

(*iii*) We provide the missing proofs of Lemma 3.1, Lemma 3.3 and Theorem 4.7 in Artmann et al., which, in this dissertation, correspond to Lemma 3.1, Lemma 4.6 and Theorem 5.4 respectively.

The dissertation is organized as follows. Chapter 2 presents the necessary background material. In Chapter 3 we show how to determine the feasibility and boundedness of a BIP problem in strongly polynomial time. Chapter 4, 5 and 6 discuss the main steps of the algorithm; these are the reduction from BIP to a TU optimization problem with an additional parity constraint, the decomposition of the TU optimization problem into *base block* problems using a variation of Seymour's theorem [12], and the efficient resolution of the *base block* problems. Finally, in Chapter 7 we assemble the results of the previous chapters to give a proof of Theorem 1.3.

2. Background

The exposition of Section 2.1 is based on Chapter 4 of Schrijver's book [11], while Section 2.2 takes concepts from Chapter 19 and 20 of the same book.

2.1 Lattices, Hermite normal form and unimodular matrices

While it is common to study linear programming as part of an undergraduate or post-graduate course in mathematics, it is less common to study the theory of *lattices* and linear *diophantine* equations. Since both topics are relevant when discussing integer linear programs, in this section, we aim to give a brief introduction of the less studied topics.

A system of linear *diophantine* equations, is a system of linear equations in integer variables, if such a system is feasible, then the set of feasible solutions forms a *lattice*.

Definition 2.1. *Let L be an additive group, with $L \subseteq \mathbb{R}^n$, then L is a* ***lattice*** *if it is generated by linearly independent vectors, in other words if:*

(i) $0 \in L$,

(ii) *if $x \in L$, then $-x \in L$,*

(iii) *if $x, y \in L$, then $x + y \in L$,*

and furthermore all $x \in L$ can be written as $\sum_{i=1}^{m} \lambda_i \alpha_i$ with $m \leqslant n$, $\forall i\ \lambda_i \in \mathbb{Z}^n$ and the α_i are linearly independent vectors in $\mathbb{R}^n$. The set $\{a_i : i \in [m]\}$ is called a ***basis*** *for the lattice.*

Note that if a matrix $A \in \mathbb{R}^{n \times n}$ has $\operatorname{rank}(A) = n$, then the column vectors of A generate a lattice. Furthermore, if the matrix B is constructed from A by performing *elementary column operations*, then the column vectors of B generate the same lattice as the column vectors of A. With *elementary column operations* we mean:

1. exchanging two columns,
2. multiplying a column by -1,
3. adding an integral multiple of one column to another column.

A matrix which correspond to a series of elementary column operations is called *unimodular*. Formally, a unimodular matrix is defined as follows:

Definition 2.2. *Let A be a square non-singular matrix with entries in $\mathbb{Z}$, if* $\det(A) \in \{-1, 1\}$ *then A is called **unimodular**.*

We also introduce the notion of a matrix in *Hermite normal form*.

Definition 2.3. *Let A be an $m \times n$ matrix with full row rank. We say A is in **Hermite normal form** if $A = \left[\begin{array}{c|c} B & 0 \end{array}\right]$ with B a non-singular, lower triangular matrix, such that all entries of B are non-negative and the entries of B along the diagonal are strictly larger than the other entries of B on the same row.*

It can be proved that if the matrix $A \in \mathbb{R}^{m \times n}$ has full row rank, then there exists a unimodular matrix U such that AU is in Hermite normal form. This also shows that A and its Hermite normal form generate the same lattice.

Unimodular matrices will be used in the next chapters to show equivalences between optimization problems, it is therefore worth mentioning two other of their properties, which are:

(i) if U is unimodular, then U^{-1} is unimodular,

(ii) U represents a bijection from $\mathbb{Z}^n$ to $\mathbb{Z}^n$.

2.2 Totally unimodular matrices and Seymour's decomposition theorem

In 1980, Seymour proved a decomposition result about matroids [12], which can also be stated in terms of totally unimodular matrices. The aim of this section is to present a variation of Seymour's theorem which is used in Chapter 5. We start by formally introducing totally unimodular matrices and *network matrices.*

Definition 2.4. *A matrix $A \in \mathbb{Z}^{n \times m}$ is called **totally unimodular** if, for any square submatrix B of A, $det(B) \in \{-1, 0, 1\}$.*

Furthermore the following are all actions which preserve total unimodularity:

- permuting rows or columns,
- multiplying rows or columns by -1,
- adding a unit vector or the transpose of a unit vector,
- adding an all-zero row or column, $(\star)$
- repeating a row or a column,
- taking the transpose,
- performing a *pivot*,
- performing a 1-, 2- or 3-*sum*.

Pivoting is the action of replacing the matrix $\begin{bmatrix} \alpha & b \\ c & D \end{bmatrix}$ with $\begin{bmatrix} -\alpha^{-1} & \alpha^{-1}b \\ \alpha^{-1}c & D-\alpha^{-1}cb \end{bmatrix}$, where α is a non-zero scalar, b is a row vector, c is a column vector and D is a matrix. Moreover, the operations of 1-, 2- and 3-*sum* are defined as follows:

$$L \oplus_1 R := \begin{bmatrix} L & 0 \\ 0 & R \end{bmatrix} \tag{1-sum}$$

$$\begin{bmatrix} L & a \end{bmatrix} \oplus_2 \begin{bmatrix} d^\top \\ R \end{bmatrix} := \begin{bmatrix} L & ad^\top \\ 0 & R \end{bmatrix} \tag{2-sum}$$

$$\begin{bmatrix} L & a & a \\ f^\top & 0 & 1 \end{bmatrix} \oplus_3 \begin{bmatrix} 1 & 0 & d^\top \\ g & g & R \end{bmatrix} := \begin{bmatrix} L & ad^\top \\ gf^\top & R \end{bmatrix} \tag{3-sum}$$

where L and R are matrices and a, d, f and g are column vectors. We now define *network matrices* which are a special case of totally unimodular matrices.

Definition 2.5. *Let $D = (V, A)$ be a directed graph and let $T = (V, U)$ be a directed tree. Let N be a matrix in $\mathbb{Z}^{U\times A}$ defined as follows. Take $u \in U$ and $a = (x, y) \in A$ and let P be the undirected $x - y$ path in T. Define*

$$N_{u,a} := \begin{cases} +1 \text{ if } u \text{ occurs in forward direction in } P, \\ -1 \text{ if } u \text{ occurs in backward direction in } P, \\ 0 \text{ if } u \text{ does not occur in } P, \end{cases} \tag{2.1}$$

then the matrix N is called a ***network matrix*** *generated by T and D.*

Furthermore, recognizing network matrices can be done efficiently:

Theorem 2.6. *Consider the matrix $A \in \mathbb{Z}^{m\times n}$, one can recognize in strongly polynomial time if A is a network matrix, and if this is the case one can find in strongly polynomial time a representation of A in terms of a directed graph and a tree.*

We now state Seymour's decomposition theorem for totally unimodular matrices.

Theorem 2.7. *A matrix A is totally unimodular if and only if A arises from network matrices and from the matrices* (2.2) *by applying the actions which preserve total unimodularity listed in* ($\star$) *and where a k-*sum *operation, $k \in [3]$, is applied only if, for both L and R, the sum of the number of rows plus the number of columns is $\geqslant 4$.*

$$\begin{bmatrix} 1 & -1 & 0 & 0 & -1 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

Cunningham and Edmonds proved in 1980 that one can recognize such a decomposition efficiently [7]. Here we state the result in terms of matrices.

Theorem 2.8. *There is a polynomial algorithm to check if a matrix M, after possible row and column permutations, can be written as* 1-, 2- *or* 3-sum*, where, for both L and R, the number of rows plus the number of columns is at least* 4.

Using Theorem 2.7 and Theorem 2.8, Artmann et al. prove a variation of the result on decomposition. They first define the *core* of a matrix T:

Definition 2.9. *Let $T \in \mathbb{Z}^{m \times n}$ be a totally unimodular matrix. We call a submatrix of T a* ***core*** *of T, if it arises from T by iteratively deleting*

(i) *any row or column with at most one non-zero entry,*

(ii) *any row or column appearing twice or whose negation is also in the matrix.*

Since up to row and column permutations, and up to multiplying some rows or columns by -1, a core of a matrix T is unique, it makes sense to talk about *the* core of a matrix.

Theorem 2.10 (Theorem 2.8 of Artmann et al. [2])**.** *Let T be totally unimodular. Then one of the following cases holds:*

(i) core(T) *or* core$(T^\top)$ *is a network matrix.*

(ii) core(T) *is, possibly after row and column permutations, and multiplications of some rows and columns by -1, one of the matrices* (2.2).

(iii) *T can, possibly after row and column permutations, be decomposed into a* 1-, 2- *or* 3-sum *with $m_L, m_R \geqslant 2$.*

(iv) *T can after pivoting once and permuting rows and columns, be decomposed into a* 1-, 2- *or* 3-sum *with $m_L, m_R \geqslant 2$.*

Furthermore, we can efficiently decide in which case we are and find a corresponding k*-sum decomposition efficiently. Here,* m_L *and* m_R *indicate the number of rows of the matrices* L *and* R *which are part of the definition of* 1-, 2- *and* 3-sum*.*

Remark. Notice that (*iii*) ensures that one can decompose a matrix T without first deleting repeated columns and rows, or unit vectors and their transpose. This fact is essential for Chapter 5, where we will apply the decomposition to the constraint matrix of an optimization problem.

3. Feasibility and boundedness

In Chapter 3, we prove that one can determine the feasibility and boundedness of a BIP problem independently from solving the optimization problem. In Section 3.1 we prove that it is possible to solve the linear relaxation of BIP in strongly polynomial time, and in Section 3.2 we show how to find the Hermite normal form of a submatrix of a bimodular matrix in strongly polynomial time. These results are then used in Section 3.3 to prove that the polynomial procedure for finding a feasible point of BIP proposed by Veselov and Chirkov [14] can be improved to run in strongly polynomial time. Finally, we observe that determining whether BIP is bounded given it is feasible, becomes a case of checking whether the linear relaxation of BIP is bounded or not.

This is a deviation from the approach taken by Artmann, Weismantel and Zenklusen. They do not explicitly explain how they treat the infeasible or unbounded case, although most of the statements they present require the problem to be feasible and bounded. We note that checking feasibility and boundedness at the beginning of the algorithm is not strictly necessary; an infeasible or unbounded BIP problem can also be detected when reconstructing an optimal solution for the BIP problem by combining the optimal solutions of the *base block* problems.

In the later chapters, we will assume the BIP problem is both feasible and bounded.

3.1 Solving the linear relaxation of BIP in strongly polynomial time

Lemma 3.1. *Let A be a $m \times n$ bimodular matrix with rank$(A) = n$, then the optimization problem*

$$\max\{c^\top x : Ax \leqslant b, x \in \mathbb{R}^n\} \tag{3.1}$$

can be solved in strongly polynomial time.

Proof. Since $\text{rank}(A) = n$, we can find a square submatrix, Q, of A such that Q has n linearly independent rows. Notice that $|\det(Q)| = 1, 2$. Let $A' := AQ^{-1}$, then problem (3.1) is equivalent to

$$\begin{aligned} &\max\{c^\top x : AQ^{-1}Qx \leqslant b, Qx \in \mathbb{R}^n\} \\ &= \max\{c^\top Q^{-1}y : A'y \leqslant b, y \in \mathbb{R}^n\} \end{aligned} \tag{3.2}$$

with $y = Qx$.

Notice that $A' := AQ^{-1}$ contains the identity matrix, since A contains Q. Furthermore all $n \times n$ submatrices of A' have determinant $0, \pm 1/|\det(Q)|, \pm 2/|\det(Q)|$. It follows that all entries of A' are equal to 0, $\pm 1/|\det(Q)|$ or $\pm 2/|\det(Q)|$.

Since all entries of A' can be written in the form p/q with $|p|, |q| \leqslant 2$, then, by Tardos [13], we can solve (3.2) in strongly polynomial time. We can retrieve an optimal solution x^* for (3.1) by calculating $x^* = Q^{-1}y^*$, where y^* is an optimal solution for (3.2). □

3.2 Finding the Hermite normal form in strongly polynomial time

Lemma 3.2. *Let A be an $m \times n$ bimodular matrix with rank$(A) = n$. Let B be a full row rank submatrix of A of size $m' \times n$, where $m' \leqslant n \leqslant m$. Then we can find in strongly polynomial time a unimodular matrix U, such that BU is the Hermite normal form of B.*

Proof. Let A' be an $n \times n$ submatrix of A with $\text{rank}(A') = n$ and such that B is a submatrix of A' (set B to be the first m' lines of A'). Note $|\det(A')| \in \{1, 2\}$.

By Theorem 5.3 in Schrijver's book [11], one can calculate the Hermite normal form, H', of A' in time polynomially bounded by the size of A' and the value of $|\det(A')| \in \{1, 2\}$. Therefore, one can find the Hermite normal form of A' in strongly polynomial time and, by Corollary 4.3b in Schrijver's book [11], there exists a unimodular matrix U such that $H' = A'U$. Thus one can calculate U using the equation $U = H'A'^{-1}$, and since B is the submatrix of A' consisting of the first m' rows, then BU is the Hermite normal form of B. □

Remark. As stated in Theorem 5.3 of Schrijver's book [11], finding the Hermite normal form of a full row rank matrix A with size $m \times n$, takes time polynomially bounded by the size of A and the absolute value of the determinant of an arbitrary $m \times m$ full rank submatrix of A. In this case, we are able to find the Hermite normal form in strongly polynomial time because A is bimodular. If B is the submatrix of a general matrix A, then we cannot determine whether A has an $m \times m$ full rank submatrix with *small* determinant. In the general case, the algorithm depends on the values of A and therefore it runs in polynomial time.

Corollary 3.2.1. *Let A be an $m \times n$ bimodular matrix with rank$(A) = n$. Let d be the greatest common divisor among the entries of a_1, the first row of A. Then we can find in strongly polynomial time a unimodular matrix U such that:*

$$d \cdot e_1 = a_1 U$$

where $e_1 \in \mathbb{Z}^n$ is the first row of the identity matrix.

Proof. By Lemma 3.2, we can find in strongly polynomial time a matrix U such that $a_1 U$ is in Hermite normal form. Since a_1 is a vector, then $a_1 U$ is of the form $d' \cdot e_1$ for some value $d' \in \mathbb{N}$.

Notice that d' is the smallest non-negative value which can be achieved by applying elementary column operations to a_1, but this is also another way to define the greatest common divisor of the entries of a_1 (see Proposition 6.1 in the book *Integer Programming* by Conforti et al. [5]), we must therefore have that $d' = d$. $\square$

3.3 Determining feasibility and boundedness in strongly polynomial time

Theorem 3.3 (Theorem 1 of Veselov and Chirkov [14]). *Suppose the feasible region of the LP relaxation of* (BIP) *is full-dimensional, then* (BIP) *is feasible. Furthermore, one can find a feasible point for* (BIP) *in strongly polynomial time.*

Proof. We denote with $S_{\mathbb{Z}}$ and S the feasible regions of (BIP) and the relaxation of (BIP) respectively:

$$S_{\mathbb{Z}} = \{x \in \mathbb{Z}^n : Ax \leqslant b\}$$
$$S = \{x \in \mathbb{R}^n : Ax \leqslant b\}$$

We prove the theorem using induction on n.

If $n = 1$, then $S = \{x \in \mathbb{R} : \alpha x \leqslant \beta\}$, with $|\alpha| \in \{1, 2\}$ and $\beta \in \mathbb{Z}$. Notice that we have the following set inclusions:

$$S \supseteq \{x \in \mathbb{R} : \beta - 1 \leqslant \alpha x \leqslant \beta\} \supseteq \left\{\frac{\beta - 1}{\alpha}, \frac{\beta}{\alpha}\right\}$$

and at least one of the values $\frac{\beta}{\alpha}$ and $\frac{\beta-1}{\alpha}$ is an integer. $\Rightarrow S_{\mathbb{Z}} \neq \emptyset$.

Now assume that $n > 1$. Since $\text{rank}(A) = n$ and $S \neq \emptyset$, S has at least one vertex u. If $u \in \mathbb{Z}^n$, then $u \in S_{\mathbb{Z}}$. $\Rightarrow S_{\mathbb{Z}} \neq \emptyset$.

If $u \notin \mathbb{Z}^n$, then we can assume without loss of generality that the first n inequalities of A are linearly independent and tight for u. Let a_1 indicate the first row of the matrix A and let d be the greatest common divisor between the entries of a_1. By corollary 3.2.1, we can find in strongly polynomial time an $n \times n$ unimodular matrix, U, such that $de_1 = a_1 U$, where e_1 is the first row of the identity matrix. Also notice that U represents a bijection from $\mathbb{Z}^n$ to $\mathbb{Z}^n$, therefore $Ax \leqslant b$ has an integer solution x if and only if $AUy \leqslant b$ has an integer solution y. $AUy \leqslant b$ can be rewritten as

$$\begin{bmatrix} d & 0 \\ h & \bar{A} \end{bmatrix} \begin{pmatrix} y_1 \\ \bar{y} \end{pmatrix} \leqslant \begin{pmatrix} b_1 \\ \bar{b} \end{pmatrix} \tag{3.3}$$

with h a column vector, $\bar{A}$ a bimodular matrix, $\bar{y} = (y_2, \ldots, y_n)^\top$ and $\bar{b} = (b_2, \ldots, b_m)^\top$. Since u is a feasible solution for $Ax \leqslant b$ with the first n constraints satisfied at equality, then (3.3) has a solution where the first n inequalities are satisfied at equality, and in particular there exists a point $\bar{y} \in \mathbb{R}^{n-1}$ satisfying

$$hy_1 + \bar{A}\bar{y} \leqslant \bar{b} \tag{3.4}$$

with $y_1 = b_1/d$. By the induction hypothesis, there exists a vector $\bar{y}^* \in \mathbb{Z}^{n-1}$ which satisfies $hy_1 + \bar{A}\bar{y}^* \leqslant \bar{b}$. Now we shall consider two cases depending on the value of b_1/d.

If $b_1/d \in \mathbb{Z}$, then $x^* = U \begin{pmatrix} b_1/d \\ \bar{y}^* \end{pmatrix}$ is a feasible integral solution to $Ax \leqslant b$ and therefore $S_{\mathbb{Z}}$ is non-empty.

Suppose instead that $b_1/d \notin \mathbb{Z}$, then we must have $d = 2$ and b_1 odd. Let $v_1 = \min\{y_1 : AUy \leqslant b\}$. Since S is full-dimensional, we have that $v_1 < b_1/d = b_1/2$. $\Rightarrow v_1 \leqslant \frac{b_1-1}{2}$. It follows that (3.4) has a feasible solution $\bar{y} \in \mathbb{R}^{n-1}$ where $y_1 = \frac{b_1-1}{2}$. By the induction hypothesis, there exists an integral solution $\bar{y}^*$ satisfying (3.4) with $y_1 = \frac{b_1-1}{2}$. This means that $y^* = \begin{pmatrix} \frac{b_1-1}{2} \\ \bar{y}^* \end{pmatrix}$ is an integral solution for (3.3), and $x^* = U \begin{pmatrix} \frac{b_1-1}{2} \\ \bar{y}^* \end{pmatrix}$ is an integral solution for $Ax \leqslant b$. $\square$

Remark. Note that the proof of Theorem 3.3 is itself a procedure for finding a feasible solution of the BIP problem. Moreover, all operations in the procedure can be done in strongly polynomial time.

Theorem 3.4. *One can determine in strongly polynomial time if* (BIP) *is feasible. If* (BIP) *is feasible, then one can find a feasible point in strongly polynomial time.*

Proof. The first step is to understand if $S := \{x \in \mathbb{R}^n : Ax \leqslant b\}$ is full-dimensional. One can do this by checking whether any of the inequalities of the constraint matrix are implicit equalities. This can be done by solving the optimization problems

$$\min\{a_i^\top x : Ax \leqslant b, x \in \mathbb{R}^n\} \tag{3.5}$$

for all rows a_i of A, with $1 \leqslant i \leqslant m$. The optimal solution of (3.5) is b_i if and only if a_i is an implicit equality. And as we have seen in Lemma 3.1, the optimization problem (3.5) can be solved in strongly polynomial time.

Let $B \subseteq [n]$ be the set of indices of the implicit equalities, If $B = \emptyset$, then S is full-dimensional and, by Theorem 3.3, (BIP) is feasible and one can find a feasible solution in strongly polynomial time. If $B \neq \emptyset$, we construct a set $B' \subseteq B$, where B' is a maximal set of indices corresponding to linearly independent implicit equalities.

B' can be found in strongly polynomial time in the following way. Initially we set $B' = \emptyset$, we then consider the implicit equalities one by one. We check whether equality i, with $i \in B$, is linearly independent from the equalities with indices in B'. If this is the case, then we add i to B'. If instead equality i is linearly dependent, we check whether the system of equalities with indices in $\{i\} \cup B'$ is feasible. If it is feasible then we do not add i to B' and we move to consider the next implicit equality. If the system of equalities is infeasible, then (BIP) is infeasible, and therefore we can stop here.

Let $A_{B'}$ indicate the submatrix of A consisting of the equalities with indices in B'. $A_{B'}$ is a full row rank matrix and by Lemma 3.2, we can find in strongly polynomial time a unimodular matrix U such that $A_{B'}U$ is in Hermite normal form. Thus, (BIP) is equivalent to

$$\begin{aligned} &\max\{c^\top Uy : AUy \leqslant b, y \in \mathbb{Z}^n\} \\ =&\max\{c^\top Uy : A_{B'}Uy = b_{B'}, AUy \leqslant b, y \in \mathbb{Z}^n\} \end{aligned} \tag{3.6}$$

where we set $x = Uy$. And consequently, determining the feasibility of (BIP) is equivalent to determining the feasibility of (3.6). Notice that we can solve $A_{B'}Uy = b_{B'}$ in strongly polynomial time which will give a unique solution $\bar{y}_{B'}$. If $\bar{y}_{B'} \notin \mathbb{Z}^{B'}$, then (BIP) is infeasible.

If $\bar{y}_{B'} \in \mathbb{Z}^{B'}$, then the optimization problem (3.6) is equivalent to

$$\max\left\{c^\top U \begin{pmatrix} \bar{y}_{B'} \\ y_N \end{pmatrix} : C_2 y_N \leqslant b - C_1\bar{y}_{B'},\ y_N \in \mathbb{Z}^N\right\} \tag{3.7}$$

where the subscript $_N$ indicates the set of indices which do not correspond to an implicit equality, i.e. $N = [n] \setminus B$. Moreover C_1 and C_2 are matrices defined as shown in (3.8)

$$\left[\begin{array}{c} A_{B'} \\ \hline A_N \end{array}\right] U = \left[\begin{array}{c|c} H & \mathbf{0} \\ \hline C_1 & C_2 \end{array}\right] \tag{3.8}$$

where $\left[\begin{array}{c|c} H & \mathbf{0} \end{array}\right]$ is the Hermite form of $A_{B'}$, with H a lower triangular matrix.

Notice furthermore that C_2 is bimodular; let C' be an $|N| \times |N|$ submatrix of C_2, we have

$$\det(C') \cdot \det(H) = \det(A') \in \{0, \pm 1, \pm 2\} \tag{3.9}$$

where A' is the $n \times n$ submatrix of $\left[\begin{array}{c} A_{B'} \\ \hline A_N \end{array}\right] U$ containing both H and C'. (3.9) implies that $\det(C') \in \{0, \pm 1, \pm 2\}$. Thus, we can conclude that the optimization problem (3.7) is a bimodular integer program whose relaxation has a full-dimensional feasible region. We can therefore use the procedure explained in the proof of Theorem 3.3 to find a feasible point $\bar{y}_N$ to (3.7). Using $\bar{y}_{B'}$ and $\bar{y}_N$ we can derive a feasible solution x for (BIP). □

Corollary 3.4.1. *One can determine in strongly polynomial time if* (BIP) *is unbounded.*

Proof. By Theorem 3.4, one can determine in strongly polynomial time if (BIP) is feasible. By Theorem 3.1, one can determine in strongly polynomial time if the LP relaxation of (BIP) is unbounded.

Since the entries of the constraint matrix are rational, (BIP) is unbounded if and only if (BIP) is feasible and the LP relaxation of (BIP) is unbounded (see Theorem 1 by Byrd et al. [4]) — both conditions can be checked in strongly polynomial time. □

4. Reducing BIP to CPTU

In Chapter 4, we prove that one can reduce BIP to *CPTU* in strongly polynomial time.

Definition 4.1. *Let $T \in \mathbb{Z}^{m\times n}$ be totally unimodular with rank$(T) = n$, $c \in \mathbb{Z}^n$ and $S \subseteq [n]$. A* ***Conic parity TU-optimization*** *(CPTU) is an optimization problem of the form*

$$\max\left\{ c^\top x : Tx \leqslant 0, x \geqslant 0, x \in \mathbb{Z}^n, \sum_{i\in S} x_i \text{ odd} \right\}. \tag{CPTU}$$

In their reduction from BIP to CPTU, Artmann et al. change the objective function of any BIP problem whose LP relaxation is unbounded or has multiple optima in order to have an equivalent BIP problem whose relaxation has a unique optimum (see Lemma 3.2 in Artmann et al. [2]). They then prove that given an algorithm $\mathcal{A}$ for CPTU, they can find an optimal solution for a BIP problem whose relaxation has a unique optimum with a single call to $\mathcal{A}$ and operations taking strongly polynomial time.

We deviate from their approach; we do not transform the BIP problem to a BIP problem whose LP relaxation has a unique optimum. Instead, we perform a reduction where we focus on optimal solutions which are vertices. A reason for such a deviation is that Artmann et al. do not prove Lemma 3.2 furthermore, only one part of the lemma is relevant to us since we have already dealt with the unbounded case in Chapter 3. Finally, working with vertices instead of transforming BIP seemed to us a neater way to go about the problem.

In Section 4.1 we introduce important results regarding the structure of BIP; we discuss Theorem 2 by Veselov and Chirkov [14] and its corollaries. Furthermore, in Lemma 4.5, we prove that for a specific case of BIP, if one has an optimal solution, then they can find an optimal solution which is a vertex in strongly polynomial time. In Section 4.2, we prove Lemma 3.3 in Artmann et al. [2] for which a proof was missing. In Lemma 4.7, we combine the aforementioned results to reduce BIP to CPTU.

4.1 Properties of BIP

It is worth setting specific notation for two sets which we will often mention throughout this section.

Definition 4.2. *Let u be a vertex of the feasible region of the LP relaxation of* (BIP) *and let I be the set of indices of constraints that are tight for u, we define*

$$N(u) := \{x \in \mathbb{R}^n : A_I x \leqslant b_I\} \tag{4.1}$$

and

$$N_{\mathbb{Z}}(u) := \operatorname{conv}(\{x \in \mathbb{Z}^n : A_I x \leqslant b_I\}). \tag{4.2}$$

Theorem 4.3 (Theorem 2 of Veselov and Chirkov, [14])**.** *Let $S = \{x \in \mathbb{R}^n : Ax \leqslant b\}$ be the feasible region of the LP relaxation of* (BIP)*, let u be a vertex of S, then all vertices of $N_{\mathbb{Z}}(u)$ belong to an edge of S.*

Proof. If $u \in \mathbb{Z}^n$, then u is the only vertex of $N_{\mathbb{Z}}(u)$ and it is in S. Suppose now that $u \notin \mathbb{Z}^n$. Let $y \in \mathbb{Z}^n$ be a vertex of $N_{\mathbb{Z}}(u)$ and let $J = \{i \in I : \sum_{j=1}^n a_{ij} y_j = b_i\}$ be the set of indices corresponding to the constraints that are tight for y in the matrix A_I.

Define the cone $C := \{x \in \mathbb{R}^n : A_I x \leqslant 0\}$. We have that $A_I y \leqslant A_I u$ and $A_I y \neq A_I u$, therefore C has dimension greater than zero since both 0 and $y - u \neq 0$ belong to C. It follows that C has an extremal ray. Let r be an extremal ray of C adjacent to the face containing y; since r is an extremal ray, it satisfies $n-1$ linearly independent rows of A_I at equality. Let H be the matrix consisting of these rows (i.e $Hr = 0$), we can use H to write r explicitly. We define:

$$r_j = \frac{1}{2}\sigma(-1)^{1+j}\det(H_j) \qquad \text{for } j = 1, \dots, n \tag{4.3}$$

where $\sigma = \pm 1$, and H_j is the matrix H without column j. In the equation below we check that (4.3) indeed defines an extremal ray,

$$(Hr)_i = \sum_{j=1}^n h_{ij}\frac{1}{2}\sigma(-1)^{1+j}\det(H_j) = \frac{1}{2}\sigma\det\begin{pmatrix} h_i \\ H \end{pmatrix} = 0 \qquad \text{for } i = 1, \dots, n-1$$

where h_i indicates the i^{th} row of H.

To show that y is on an edge of S, we are going to prove that $y = u + r$ and that $A(u + r) \leqslant b$.

First notice that

$$\sum_{j=1}^{n} a_{ij} r_j = 0 \qquad \text{for } i \in J, \tag{4.4}$$

which holds because r satisfies at equality all constraints of the cone that are satisfied at equality by the points in the faces adjacent to r. And, by definition of J, we have that for all $i \in J$

$$(A\,(y-u))_i = \sum_{j=1}^{n} a_{ij}(y_j - u_j) = \sum_{j=1}^{n} a_{ij} y_j - \sum_{j=1}^{n} a_{ij} u_j = b_i - b_i = 0$$

Furthermore, using the fact that A is bimodular, we get

$$|(Ar)_i| = \left|\sum_{j=1}^{n} a_{ij} \frac{1}{2} \sigma (-1)^{1+j} \det(H_j)\right| = \frac{1}{2} \cdot \left|\sigma \det \begin{pmatrix} a_i \\ H \end{pmatrix}\right| \leqslant \frac{1}{2} \cdot 2 = 1 \quad \text{for } i \in [n]. \tag{4.5}$$

Using the results just obtained, we can prove that $r \notin \mathbb{Z}^n$. Assume by contradiction that $r \in \mathbb{Z}^n$, then $y + r, y - r \in \mathbb{Z}^n$. Furthermore, it follows from (4.4) and (4.5) that $A_I(y + r) \leqslant A_I u$ and $A_I(y - r) \leqslant A_I u$, which implies that both $y + r$ and $y - r$ belong to $N_{\mathbb{Z}}(u)$. But this is a contradiction since y is a vertex of $N_{\mathbb{Z}}(u)$ and cannot be written as a convex combination of points in $N_{\mathbb{Z}}(u)$.

Let B be an $n \times n$ submatrix of A_I with non-zero determinant containing H. Notice that $|\det(B)| = 2$ if not $u \in \mathbb{Z}^n$. This means that $|\det(B^{-1})| = 1/2$ and the lattice L generated by the columns of B^{-1} is divided into two classes: $\mathbb{Z}^n$ and $u + Z^n$. Furthermore $r \notin \mathbb{Z}^n$ and in equation (4.6) we show that $Br \in \mathbb{Z}^n$. This implies that $r \in u + \mathbb{Z}^n$, which means that $r + u \in \mathbb{Z}^n$.

$$Br = \begin{pmatrix} a_i \\ H \end{pmatrix} r = \begin{pmatrix} \sum_{j=1}^{n} a_{ij} \frac{1}{2} \sigma (-1)^{1+j} \det(H_j) \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sigma \det(B) \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \pm 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{4.6}$$

where a_i is a row of A_I which is linearly independent of the $n - 1$ rows in H.

Now let $p = r + u$ and $2y = p + q$. Notice that $q \in \mathbb{Z}^n$ since both y and p are integral, furthermore

$$A_I q = A_I(2y - r - u) \leqslant A_I u = b_I$$

which implies $q \in N_{\mathbb{Z}}(u)$. This means that y is the convex combination of p and q which are both in $N_{\mathbb{Z}}(u)$. Since y is a vertex, the only way this is possible is if $p = q = y$. Hence

$y = u + r$, additionally

$$
\begin{aligned}
(Au)_i + (Ar)_i &\leqslant b_i + 0 = b_i && \text{for } i \in I \\
(Au)_i + (Ar)_i &\leqslant (b_i - 1) + 1 = b_i && \text{for } i \notin I
\end{aligned}
$$

which implies that $y \in S$. □

We denote with $V(P)$ the vertices of a polyhedron P.

Corollary 4.3.1 (Corollary 3 of Veselov and Chirkov, [14])**.** $V(S_{\mathbb{Z}}) = \bigcup_{u \in V(S)} V(N_{\mathbb{Z}}(u))$.

Proof. The direction "$\supseteq$" comes from Theorem 4.3. The direction "$\subseteq$" is proved below. Notice that $S_{\mathbb{Z}} \subseteq N_{\mathbb{Z}}(u)$ for all vertices u of S. It follows that $S_{\mathbb{Z}} \subseteq \bigcap_{u \in V(S)} N_{\mathbb{Z}}(u)$ and therefore $V(S_{\mathbb{Z}}) \subseteq \bigcap_{u \in V(S)} N_{\mathbb{Z}}(u)$.

Let y be a vertex of $S_{\mathbb{Z}}$, we have that $y \in \bigcap_{u \in V(S)} N_{\mathbb{Z}}(u)$ and therefore y is the convex combination of elements in $\bigcup_{u \in V(S)} V(N_{\mathbb{Z}}(u))$. The direction "$\supseteq$" implies that y is the convex combination of elements in $V(S_{\mathbb{Z}})$, but this is only possible if y coincides with one of these elements. Therefore, we must have that y coincides with one of the elements in $\bigcup_{u \in V(S)} V(N_{\mathbb{Z}}(u))$. □

Corollary 4.3.2 (Corollary 4 of Veselov and Chirkov, [14])**.** *If $f(x) = c^\top x$ achieves its maximum at a vertex u^* of S, then $\max_{x \in S_{\mathbb{Z}}} f$ is achieved at some $y \in V(N_{\mathbb{Z}}(u^*))$.*

Proof. Let $OPT := c^\top u^*$ be the optimal objective value of f in S. If $u^* \in \mathbb{Z}^n$ then $\max_{x \in S_{\mathbb{Z}}} f$ is achieved at u^* which belongs to $V(N_{\mathbb{Z}}(u^*))$.

If $u^* \notin \mathbb{Z}^n$, then consider the family of hyperplanes of the form $c^\top x = \delta$, with $\delta \leqslant OPT$. These planes are parallel to each other, and they all cut the (translated) cone $N(u^*)$. Let $\delta^* = \max\{\delta : \exists x \in S_{\mathbb{Z}}, c^\top x = \delta\}$, since $S_{\mathbb{Z}}$ is a convex polyhedron, there exists a vertex y of $S_{\mathbb{Z}}$ such that $c^\top y = \delta^*$. By Corollary 4.3.1, $V(S_{\mathbb{Z}}) = \cup_{u \in V(S)} V(N_{\mathbb{Z}}(u))$, therefore y is also a vertex of $N_{\mathbb{Z}}(u)$ for some vertex u of S. And since the family of planes cuts $N(u^*)$, we must have that y is a vertex of $N_{\mathbb{Z}}(u^*)$. □

We now define a particular type of optimization problem strictly related to BIP.

Definition 4.4. *Let u be a vertex of the feasible region of the LP relaxation of a BIP problem, we call* **u-*BIP*** *an optimization problem of the form*

$$\max\{c^\top x : A_I x \leqslant b_I, x \in \mathbb{Z}^n\}$$

Where $I = \{i : \sum_{j=1}^n a_{ij} u_j = b_i\}$, and A, b, c are defined in (BIP).

Notice that $N(u)$ in (4.1) corresponds to the feasible region of the LP relaxation of u-BIP, and $N_{\mathbb{Z}}(u)$ in (4.2) corresponds to the convex hull of u-BIP.

Lemma 4.5. *Consider the u-BIP optimization problem*

$$\max\{c^\top x : A_I x \leqslant b_I, x \in \mathbb{Z}^n\}$$

where $I = \{i : \sum_{j=1}^n a_{ij}u_j = b_i\}$ and u is an optimal vertex solution for the LP relaxation of BIP. Suppose we have an optimal solution x^* to u-BIP, then we can find in strongly polynomial time an optimal vertex solution to u-BIP.

Proof. First notice that since A_I is the matrix of constraints that are tight for u, we have that $\operatorname{rank}(A_I) = n$. Let A_J be the submatrix of A_I of constraints that are tight for x^*, i.e. $J = \{i \in I : \sum_{j=1}^n a_{ij}x_j^* = b_i\}$. We consider three cases depending on the rank of A_J.

If $\operatorname{rank}(A_J) = n$, then $x^* = u$, and therefore x^* is already an optimal vertex solution for u-BIP.

If $\operatorname{rank}(A_J) = n - 1$, then x^* belongs to an (infinite) edge e of $N_{\mathbb{Z}}(u)$. Let r be the only vertex of e; r is optimal since it is a convex combination of u and x^*. Next, we show how to find r in strongly polynomial time. Take Q, a full row rank submatrix of A_J of size $(n-1)\times n$. By Lemma 3.2, we can find in strongly polynomial time a unimodular matrix U such that QU is the Hermite normal form of Q. Since QU is in Hermite normal form, we can find the integral solutions to $QUy = b_Q$ in strongly polynomial time, (where b_Q indicates the entries of b which corresponds to the rows of Q). Then, Uy gives us the parametrized family of the integral points lying on the line containing the edge e. We take r to be the closest point to u which is contained in the segment ux^*.

If $\operatorname{rank}(A_J) = n_0 < n - 1$, then we look for an edge of $N(u)$ adjacent to the face of $N(u)$ containing x^*. This can be done in strongly polynomial time in the following way. We find a non-zero solution to the system $A_J x = 0$ with $x \in \mathbb{R}^n$. We then move in the direction of x by a factor of λ until one of the inequalities in A_I which are not in A_J is satisfied at equality. Since $N(u)$ is a translated cone, we might have to check the direction x and the direction $-x$ to find an additional inequality which is satisfied at equality. The result will be a point $x_1 = x^* + \lambda_1 x$ which is contained in $N(u)$ and which satisfies $n_0 + 1$ inequalities at equality. We can repeat this process until we find a feasible point which satisfies $n - 1$ linearly independent inequalities at equality. These $n - 1$ linearly independent inequalities define an edge of $N(u)$ adjacent to the face containing x^*. As seen above, we can find in strongly polynomial time all integral points lying on the line containing this edge. We set r to be the closest point to u, contained in the translated cone $N(u)$, of such integral points.

We now show that r is indeed optimal. Let F be the face of the $N_{\mathbb{Z}}(u)$ containing x^* defined by $F = \text{conv}\{x \in \mathbb{Z}^n : A_J x = b_J\}$. We consider two different cases depending on whether x^* is in the interior of F or not.

For x^* in the interior of F, we are going to prove that for all $x \in F_{\mathbb{Z}} = \{x \in \mathbb{Z}^n : A_J x = b_J\}$, we have that $c^\top x = c^\top x^*$. Suppose, by contradiction, that there exists $x_1 \in F_{\mathbb{Z}}$ with $c^\top x_1 \neq c^\top x^*$, then we have that $c^\top x_1 < c^\top x^*$. Moreover, we can apply Carathéodory's Theorem to write x^* as a convex combinations of points in $F_{\mathbb{Z}}$:

$$x^* = \sum_{i=1}^{n+1} x_i \lambda_i$$

with $\sum_{i=1}^{n+1} \lambda_i = 1$ and $\forall i, x_i \in F_{\mathbb{Z}}$. It follows that

$$c^\top x^* = \sum_{i=1}^{n+1} c^\top x_i \lambda_i$$

and since $c^\top x_1 < c^\top x^*$, there must be an x_i with $c^\top x_i > c^\top x^*$, but this is a contradiction since x^* is optimal for u-BIP.

Consider now the case where x^* is not in the interior of F. The only way x^* is not in the interior of F, is if x^* belongs to a face of $N_{\mathbb{Z}}(u)$ which arises from the integrality constraint. It follows that x^* can be written as the convex combination of the vertices adjacent to F:

$$x^* = \sum_{v \in V(F)} v \lambda_v$$

with $\sum_v \lambda_v = 1$. Which implies that

$$c^\top x^* = \sum_{v \in V(F)} c^\top v \lambda_v$$

and therefore all vertices of F are optimal solutions to u-BIP.

We have therefore proved, that whether x^* is in the interior of F or not, the vertices of $N_{\mathbb{Z}}(u)$ adjacent to F are optimal solutions for u-BIP. □

Remark. Observe that finding the convex hull of (ILP) is $\mathcal{NP}$-complete. In fact, if we know the convex hull of an ILP, then we can write the ILP in its ideal formulation and find an optimal solution by solving the LP relaxation. For u-BIP, we can efficiently find the vertices of the convex hull because we can take advantage of the fact that all vertices of the convex hull satisfy $n-1$ rows of the constraint matrix at equality.

4.2 The reduction

We prove Lemma 4.6 in preparation for the reduction from BIP to CPTU.

Lemma 4.6. *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*

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

Proof. Let H be the Hermite normal form of Q, then H has the following form.

- There is a $k \in [n]$ such that $H_{kk} = 2$. Furthermore, $H_{ii} = 1$ for all $i \neq k$, $i \in [n]$. This follows from $|\det(Q)| = 2$.
- For all rows $i \neq k$, $i \in [n]$, we have $H_{ij} = 0$ for all $j \neq i$.
- For row k, we have $H_{kj} \in \{0, 1\}$ for $j < k$, and $H_{kj} = 0$ for $j > k$.

Let $\bar{J} = \{j \in [n] : H_{kj} = 1\}$, then it is easy to check that the inverse of H has the following form.

- $(H^{-1})_{kk} = 1/2$, and $(H^{-1})_{ii} = 1$ for all $i \neq k$, $i \in [n]$.
- For all rows $i \neq k$, $i \in [n]$ we have $(H^{-1})_{ij} = 0$ for all $j \neq i$.
- For row k, we have $(H^{-1})_{kj} = -1/2$ if $j \in \bar{J}$, and $(H^{-1})_{kj} = 0$ otherwise.

Let $J = \bar{J} \cup \{k\}$, note that $(H^{-1})_{ij} \in \frac{1}{2} + \mathbb{Z}$ if and only if $(i,j) \in \{k\} \times J$. Let U be a unimodular matrix such that $H = QU$, then $Q^{-1} = UH^{-1}$. Notice that multiplying by U from the left corresponds to performing elementary row operations. In particular, a row of Q^{-1} has entries in $1/2 + \mathbb{Z}$ if and only if the corresponding entry in the column vector U_k is odd. Let $I = \{i \in [n] : U_{ik} \text{ is odd}\}$, notice that $I \neq \emptyset$ since if all entries of the column vector U_k were even, then U would not be unimodular. Therefore the entries of Q^{-1} satisfy

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

where I and J are non empty sets. □

Lemma 4.7. *Let $\mathcal{A}$ be an algorithm which outputs an optimal solution for any feasible and bounded CPTU, then one can solve any feasible and bounded BIP with a single call to $\mathcal{A}$ and operations taking strongly polynomial time.*

Proof. If the BIP is feasible and bounded, then the LP relaxation of BIP is as well feasible and bounded. Using Lemma 3.1, one can find in strongly polynomial time an optimal vertex solution u to the LP relaxation of the BIP. If $u \in \mathbb{Z}^n$, then u is an optimal solution to BIP and we are done.

Consider the case where $u \notin \mathbb{Z}^n$. By Corollary 4.3.2, BIP achieves its maximum at a vertex of $N_{\mathbb{Z}}(u)$. Furthermore, by Lemma 4.5, given an optimal solution to u-BIP, we can find in strongly polynomial time an optimal solution which is a vertex. Therefore, if we solve u-BIP, we can solve BIP.

Let I be the set of indices corresponding to the constraints in A that are tight for u. u-BIP is equivalent to the same optimization problem but translated so that u coincides with the origin:

$$\begin{aligned} &\max\{c^\top x : A_I x \leqslant b_I, x \in \mathbb{Z}^n\} \\ =&\max\{c^\top (x+u) : A_I x + A_I u \leqslant b_I, x+u \in \mathbb{Z}^n\} \\ =&\max\{c^\top x : A_I x \leqslant 0, x+u \in \mathbb{Z}^n\} \end{aligned} \tag{4.7}$$

Let Q be a square submatrix of A_I, with $\operatorname{rank}(Q) = n$. Let b_Q be the part of b that corresponds to the rows in Q, then

$$u = Q^{-1} b_Q$$

Since $u \notin \mathbb{Z}^n$, some entries of Q^{-1} must be fractional, $\Rightarrow |\det(Q)| = 2$. This means that for every submatrix Q of A_I with $\operatorname{rank}(Q) = n$, we have that $|\det(Q)| = 2$.

Now let $\bar{A} := A_I Q^{-1}$. $\bar{A}$ contains the identity matrix, since A_I contains Q. Furthermore, every full rank square submatrix of $\bar{A}$ is equal to $Q'Q^{-1}$ where Q' is some full rank square submatrix of A_I. Therefore we have

$$\det(Q'Q^{-1}) = \det(Q') \cdot \det(Q^{-1}) = 2 \cdot \frac{1}{2} = 1,$$

this implies that $\bar{A}$ is totally unimodular.

Hence the optimization problem (4.7) is equivalent to

$$\begin{aligned}
&\max\left\{c^\top Q^{-1}Qx : A_I Q^{-1}Qx \leqslant 0, u + Q^{-1}Qx \in \mathbb{Z}^n\right\}\\
=&\max\left\{\bar{c}^\top z : \bar{A}z \leqslant 0, Q^{-1}(b_Q + z) \in \mathbb{Z}^n\right\}\\
=&\max\left\{\bar{c}^\top z : \bar{A}z \leqslant 0, Q^{-1}(b_Q + z) \in \mathbb{Z}^n, z \in \mathbb{Z}^n\right\}
\end{aligned}$$

with $\bar{c}^\top = c^\top Q^{-1}$ and $z = Qx$. Notice that adding the condition $z \in \mathbb{Z}^n$ does not change the maximization problem, since this condition was already implied by $Q^{-1}(b_Q+z) \in \mathbb{Z}^n$:

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

We now show that the condition $Q^{-1}(b_Q + z) \in \mathbb{Z}^n$ is equivalent to a parity constraint. By Lemma 4.6, we have that

$$\begin{aligned}
&(Q^{-1})_{ij} \in \frac{1}{2} + \mathbb{Z} && \forall (i,j) \in I \times J\\
&(Q^{-1})_{ij} \in \mathbb{Z} && \forall (i,j) \notin I \times J
\end{aligned}$$

for $I, J \neq \emptyset$ with $I, J \subseteq [n]$, which implies the equivalence at line (4.8),

$$\begin{aligned}
Q^{-1}(b_Q + z) \in \mathbb{Z}^n &\iff \sum_{j\in J}(b_Q + z)_j \text{ is even} && (4.8)\\
&\iff \sum_{j\in J} z_j \text{ is odd} && (4.9)
\end{aligned}$$

while line (4.9) follows from $Q^{-1}b_Q = u \notin \mathbb{Z}^n$ which means that $\sum_{j\in J}(b_Q)_j$ is odd.

The BIP is equivalent to solving

$$\max\left\{\bar{c}^\top z : \bar{A}z \leqslant 0, \sum_{j\in J} z_j \text{ is odd}, z \in \mathbb{Z}^n\right\}. \tag{4.10}$$

Furthermore, $\bar{A}$ contains the identity matrix, therefore the system of inequalities $\bar{A}z \leqslant 0$ implies $z \leqslant 0$. Therefore, the non-negativity constraint of CPTU can be achieved by substituting z with $-z$. □

5. Decomposing CPTU problems

In Chapter 5, we show how to decompose a CPTU problem when the constraint matrix can be written as a 3-sum. Since 1-sum and 2-sum are special cases of 3-sum, the same procedure is valid for 1- and 2-sum as well.

While from an algorithmic perspective using the same procedure for 1-, 2- and 3-sum is not computationally efficient[1], from a theoretical point of view this is enough to prove that a CPTU problem can be decomposed in strongly polynomial time. For a specific procedure for 2-sum we refer the reader to Theorem 4.6 of Artmann et al. [2].

The main result of this chapter is Theorem 5.4 which corresponds to Theorem 4.7 in the paper of Artmann et al., for which a proof was not provided. Theorem 5.4 gives a bound on the number of CPTU subproblems one needs to solve in order to find an optimal solution for the original CPTU problem.

5.1 Limiting the number of subproblems

Consider a constraint matrix T for CPTU which can be decomposed as a 3-sum:

$$T = \begin{bmatrix} L & ad^\top \\ gf^\top & R \end{bmatrix} = \begin{bmatrix} L & a & a \\ f^\top & 0 & 1 \end{bmatrix} \oplus_3 \begin{bmatrix} 1 & 0 & d^\top \\ g & g & R \end{bmatrix}. \tag{5.1}$$

Depending on whether the matrix L or the matrix R have fewer rows, the algorithm solves a parametrized family of $L-$*subproblems* or $R-$*subproblems*. The idea is two consider all possible subproblems for the $L-$part (or the $R-$part) which lead to an optimal solution for the original problem. Lemma 5.1 allows us to limit the number of subproblems we have to look at, since it states that if a CPTU problem is feasible and bounded then there exists a *well-structured* optimal solution. Therefore, to find an optimal solution for the

[1] Artmann et al. [2] show that to find an optimal solution for 2-sum one needs to solve 7 smaller CPTU problems, while to find an optimal solution for 3-sum one needs to solve 15 smaller CPTU problems.

original problem, it is enough to consider all subproblems which lead to a *well-structured* optimal solution.

Lemma 5.1 (Lemma 4.2 in Artmann et al. [2]). *Consider a feasible and bounded CPTU problem with constraint matrix T. Then there exists an optimal solution x^* to CPTU satisfying*

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

for all vectors w such that appending $w^\top$ as an additional row to T preserves TU-ness. We call such an optimal solution x^*, a ***well-structured*** *optimal solution to the CPTU problem.*

We refer to Artmann et al. [2] for a proof of Lemma 5.1.

Lemma 5.2. *Let T be a totally unimodular matrix. If T can be decomposed as a 3-sum as shown in (5.1), then one can append vectors $\begin{pmatrix} 0 & d^\top \end{pmatrix}$, $\begin{pmatrix} f^\top & 0 \end{pmatrix}$ and $\begin{pmatrix} f^\top & d^\top \end{pmatrix}$ to T and create totally unimodular matrices.*

Proof. Notice that the newly originated matrices can be decomposed with a 3-sum as follows:

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

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

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

furthermore, since T is a 3-sum of totally unimodular matrices, we have that

$$\begin{bmatrix} L & a & a \\ f^\top & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & d^\top \\ g & g & R \end{bmatrix}$$

are totally unimodular. What is left to prove is that also the matrices M_1, M_2 and M_3 below are totally unimodular.

$$M_1 = \begin{bmatrix} f^\top & 1 & 1 \\ L & a & a \\ f^\top & 0 & 1 \end{bmatrix}, \; M_2 = \begin{bmatrix} f^\top & 0 & 0 \\ L & a & a \\ f^\top & 0 & 1 \end{bmatrix}, \; M_3 = \begin{bmatrix} 1 & 0 & d^\top \\ 0 & 0 & d^\top \\ g & g & R \end{bmatrix}.$$

Consider the matrix M_1. Notice that up to row and column permutations, most of the submatrices of M_1 are also submatrices of

$$M_0 = \begin{bmatrix} L & a & a & a \\ f^\top & 0 & 1 & 1 \end{bmatrix},$$

and therefore their determinant is in $\{-1, 0, 1\}$ since M_0 is totally unimodular.

If a submatrix Q of M_1, is not a submatrix of M_0, then Q must contain entries from the first and last row of M_1. Notice that the last and first row of M_1 are identical apart from the presence of a 1 in the first row instead of a 0. Let i be the index of the column in which the first and last row differ. If Q has no entries from column i, then $\det(Q) = 0$, since Q has two identical rows. If Q has entries from column i, then $\det(Q) = 0 + (-1)^{1+i} \det(Q_{1i})$. Where Q_{1j} is the submatrix of Q without row 1 and column i. Q_{1j} is a submatrix of M_0, hence $\det(Q_{1j}) \in \{-1, -, 1\}$. It follows that $\det(Q) \in \{-1, 0, 1\}$, and this concludes the proof that M_1 is TU. With a similar reasoning, one can prove that also M_2 and M_3 are TU. □

Let x^* be a well-structured optimal solution to a CPTU problem with constraint matrix (5.1), Lemma 5.2 implies that x^* satisfies:

$$\begin{aligned} \begin{pmatrix} 0 & d^\top \end{pmatrix} \cdot x^* &\in \{-1, 0, 1\}, \\ \begin{pmatrix} f^\top & 0 \end{pmatrix} \cdot x^* &\in \{-1, 0, 1\}, \\ \begin{pmatrix} f^\top & d^\top \end{pmatrix} \cdot x^* &\in \{-1, 0, 1\}. \end{aligned} \tag{5.2}$$

Therefore, the pair $\left(\begin{pmatrix} 0 & d^\top \end{pmatrix} \cdot x^*, \begin{pmatrix} f^\top & 0 \end{pmatrix} \cdot x^* \right)$ can only take values from the set $H = \{(-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0)\}$. This result is applied in the next section to define the subproblems.

5.2 The decomposition

We use the subscripts $_L$ and $_R$ to refer to the variables, indices and values associated with the L-part and the R-part of (CPTU). We consider separately the case where L has fewer rows than R, and vice versa.

In the case where $m_L \leqslant m_R$, the algorithm solves a family of *L-subproblems* parametrized by $(\alpha, \beta) \in H$ and $\gamma \in \{0, 1\}$:

$$\max\{c_L^\top y : Ly + \alpha \cdot a \leqslant 0,\ y \in \mathbb{Z}^{n_L},\ f^\top y = \beta,\ y(S_L) \equiv \gamma \mod 2\}. \tag{5.3}$$

We denote with $\rho_L(\alpha,\beta,\gamma)$ the optimal objective value of (5.3) with the convention that $\rho_L(\alpha,\beta,\gamma) = -\infty$ if the problem is infeasible and $\rho_L(\alpha,\beta,\gamma) = +\infty$ if the problem is unbounded. For the bounded feasible subproblems, we denote the optimal solution found with $y^*(\alpha,\beta,\gamma)$. We let I to be the set of tuples (α,β,γ) for which the subproblems are feasible and bounded:

$$I = \{(\alpha,\beta,\gamma) : (\alpha,\beta) \in H, \gamma \in \{0,1\}, \rho_L(\alpha,\beta,\gamma) \neq \pm\infty\}\,.$$

Additionally, we define the combined optimization problem:

$$\max\{\bar{c}^\top z : \bar{R}z \leqslant 0,\ h^\top z = 0,\ z(S_R \cup J) \text{ odd},\ z \in \mathbb{Z}_{\geqslant 0}^{n_R+|I|}\} \tag{5.4}$$

where,

$$\begin{aligned}
&\bar{c}^\top := \left(\, c_R^\top \,\middle|\, \tilde{c}^\top \,\right) && \text{with } \tilde{c}_{\alpha,\beta,\gamma} = \rho_L(\alpha,\beta,\gamma),\ \forall(\alpha,\beta,\gamma) \in I,\\
&\bar{R} := \left[\, R \,\middle|\, \widetilde{R} \,\right] && \text{with } \widetilde{R}_{\alpha,\beta,\gamma} = \beta \cdot g,\ \forall(\alpha,\beta,\gamma) \in I,\\
&h^\top := \left(\, d^\top \,\middle|\, \tilde{h}^\top \,\right) && \text{with } \tilde{h}_{\alpha,\beta,\gamma} = -\alpha,\ \forall(\alpha,\beta,\gamma) \in I,\\
&J = \{(\alpha,\beta,\gamma) \in I : \gamma = 1\}.
\end{aligned}$$

Lemma 5.3. *Let* (CPTU) *be feasible and bounded. Suppose the constraint matrix of* (CPTU) *can be decomposed into a 3-sum as in* (5.1)*. Then the optimal value of* (CPTU) *is equal to the optimal value of the combined optimization problem* (5.4) *and*

$$f(z^*,y^*) = x^* = \begin{pmatrix} x_L^* \\ x_R^* \end{pmatrix} = \begin{pmatrix} \sum_{(\alpha,\beta,\gamma)\in I} y^*(\alpha,\beta,\gamma) z^*(\alpha,\beta,\gamma) \\ z_R^* \end{pmatrix}$$

is an optimal solution to (CPTU)*, where z^* is an optimal solution to the combined optimization problem* (5.4)*.*

Proof. The proof can be summarized in the following way. We prove that whenever (CPTU) is feasible and bounded then the combined problem is feasible and at least one subproblem is feasible and bounded. Next, we prove that the combined problem achieves an objective value at least as big as (CPTU). On the other hand, we prove that if z is feasible for the combined problem, then $x = f(z,y^*)$ is feasible for (CPTU), and z and x achieve the same objective value. Which implies that if z^* is optimal for (5.4), then $x^* = f(z,y^*)$ must be optimal for (CPTU). We now start with the proof.

Since (CPTU) is feasible and bounded, then (CPTU) has an optimal solution x^*. Set $\alpha^* := (0\ \ d^\top) \cdot x^*$, $\beta^* := (f^\top\ \ 0) \cdot x^*$ and $\gamma^* := x_L^*(S_L)$. Notice that the $L-$subproblem with parameters $(\alpha^*,\beta^*,\gamma^*)$ is feasible, since x_L^* is a feasible solution for it. Furthermore, we must have that $\rho_L(\alpha^*, \beta^*, \gamma^*) \neq +\infty$, otherwise we could find a feasible point y such that $c_L^\top y > c_L^\top x_L^*$, but this would imply that the vector $(y \;\; x_R^*)^\top$ is feasible for (CPTU) and has greater objective value than x^*, leading to a contradiction. Let z be defined by $z_R := x_R^*$ and

$$z(\alpha, \beta, \gamma) := \begin{cases} 1 & \text{if } \alpha = \alpha^*, \beta = \beta^*, \gamma = \gamma^* \\ 0 & \text{otherwise} \end{cases},$$

then z is feasible for the combined problem, and it corresponds to a combination of only feasible and bounded subproblems. This means that whenever (CPTU) is feasible and bounded, then the set of feasible and bounded L-subproblems is non-empty (i.e. $I \neq \emptyset$), and the function $f(z, y^*)$ is well-defined for all feasible points z of (5.4).

Furthermore z has objective value greater or equal than x^*:

$$\begin{aligned}
\bar{c}^\top z &= c_R^\top z_R + \sum_{(\alpha,\beta,\gamma)\in I} \rho_L(\alpha, \beta, \gamma) z(\alpha, \beta, \gamma) \\
&= c_R^\top x_R^* + \rho_L(\alpha^*, \beta^*, \gamma^*) \\
&\geqslant c_R^\top x_R^* + c_L^\top x_L^* \qquad (5.5) \\
&= c^\top x^*
\end{aligned}$$

where line (5.5) follow from the fact that x_L^* is feasible for (5.3).

On the other hand, if z is a feasible point for (5.4), then $x = f(z, y^*)$ is feasible for (CPTU). Notice, in-fact, that the matrix constraints hold:

$$\begin{aligned}
Lx_L + ad^\top x_R &= L \left(\sum_{(\alpha,\beta,\gamma)\in I} y^*(\alpha, \beta, \gamma) z(\alpha, \beta, \gamma) \right) + ad^\top z_R \\
&\leqslant \sum_{(\alpha,\beta,\gamma)\in I} -(\alpha \cdot a) z(\alpha, \beta, \gamma) + ad^\top z_R \qquad (5.6) \\
&= a \left(\sum_{(\alpha,\beta,\gamma)\in I} -\alpha z(\alpha, \beta, \gamma) + d^\top z_R \right) \\
&= ah^\top z \qquad (5.7) \\
&= 0
\end{aligned}$$

where the inequality at line (5.6) holds because $y^*(\alpha,\beta,\gamma)$ is an optimal solution of (5.3) and z is non negative. In line (5.7) we apply the definition of h.

$$\begin{aligned}
gf^\top x_L + Rx_R &= gf^\top \left(\sum_{(\alpha,\beta,\gamma)\in I} y^*(\alpha,\beta,\gamma) z(\alpha,\beta,\gamma) \right) + Rz_R \\
&= \sum_{(\alpha,\beta,\gamma)\in I} g\beta z(\alpha,\beta,\gamma) + Rz_R \qquad (5.8) \\
&= \bar{R} z \qquad (5.9) \\
&\leqslant 0
\end{aligned}$$

(5.8) arises from $f^\top y^*(\alpha,\beta,\gamma) = \beta$ which is a constraint of (5.3), and (5.9) comes from the definition of $\bar{R}$.

The parity constraint holds as well:

$$\begin{aligned}
x(S) &= x_L(S_L) + x_R(S_R) \\
&= \sum_{(\alpha,\beta,\gamma)\in I} z(\alpha,\beta,\gamma)\left[y^*(\alpha,\beta,\gamma)(S_L)\right] + z_R(S_R) \\
&\equiv \sum_{(\alpha,\beta,\gamma)\in I} z(\alpha,\beta,\gamma)\gamma + z_R(S_R) \pmod 2 \qquad (5.10) \\
&\equiv \sum_{(\alpha,\beta,1)\in I} z(\alpha,\beta,1) + z_R(S_R) \pmod 2 \\
&= z(S_R \cup J) \\
&\equiv 1 \pmod 2
\end{aligned}$$

where (5.10) follows from the fact that $y^*(\alpha,\beta,\gamma)$ satisfies the optimization problem (5.3).

Finally notice that $x = f(z, y^*)$ achieves the same objective value for (CPTU) as z for (5.4):

$$\begin{aligned}
c^\top x &= c_L^\top x_L + c^\top x_R \\
&= c_L^\top \sum_{(\alpha,\beta,\gamma)\in I} y^*(\alpha,\beta,\gamma) z(\alpha,\beta,\gamma) + c_R^\top z_R \\
&= \sum_{(\alpha,\beta,\gamma)\in I} \rho_L(\alpha,\beta,\gamma) z(\alpha,\beta,\gamma) + c_R^\top z_R \\
&= \bar{c}^\top z
\end{aligned}$$

□

Notice that the case $m_L > m_R$ is symmetrical to the case $m_L \leqslant m_R$. This is because we can permute the rows and columns of T so that R becomes L, $gf^\top$ becomes $ad^\top$, and vice versa.

We are now able to prove the main theorem regarding decomposition.

Theorem 5.4. *Let* (CPTU) *be feasible and bounded. Suppose the constraint matrix of* (CPTU) *can be decomposed into a* 3*-sum as in* (5.1)*. Then one can solve* (CPTU) *by solving:*

- *at most* 14 *CPTU problems with at most* m_1 *rows,*
- 1 *CPTU problem with* m_2 *rows,*

where $m_1 \leqslant m_2 \leqslant m-1$ *and* $m_1 + m_2 = m+2$*.*

Proof. Without loss of generality we consider the case where L has no more rows than R. Let $m_1 - 1$ be the number of rows of L, and let $m_2 - 1$ be the number of rows of R. Then $m_1 + m_2 = m + 2$ and $m_1 \leqslant m_2$. Furthermore, Theorem 2.10 implies that $m_1 - 1 \geqslant 2$ and therefore $m_2 = m + 2 - m_1 \leqslant m - 1$.

Consider the combined problem (5.4). Notice that $\bar{R}$ is TU since $[R|g]$ is TU. Furthermore, we can use the constraint $h^\top z = 0$ to write one of the variables, say z_i, in terms of the other variables. To guarantee that $z_i \geqslant 0$, we add one row to $\bar{R}$. The resulting constraint matrix is TU, since one can add $h^\top$ to $\bar{R}$ and preserve TU-ness. Therefore the combined problem (5.4) is equivalent to a CPTU problem with m_2 rows.

Consider a parametrized family of L-subproblems as in (5.3). Observe that we have at most 14 L-subproblems, since $|I| \leqslant |H| \cdot |\{0,1\}| = 7 \cdot 2 = 14$. We now prove that we can reduce (5.3) to a CPTU problem with at most m_1 rows. Note that we can use the constraint $f^\top y = \beta$ to write one variable in terms of the other ones. Since the variables are free, we do not have to add a row to L to guarantee the non-negativity of the variables. The resulting constraint matrix L' is as well TU since one can add $f^\top$ to L and preserve TU-ness. Therefore, the optimization problem (5.3) is equivalent to

$$\max\{\bar{c}^\top y : L'y \leqslant -\alpha \cdot a, y \in \mathbb{Z}^{n_L - 1}, y(S_{L'}) \equiv \gamma \mod 2\} \tag{5.11}$$

for some vector $\bar{c}$. Consider the optimization problem (5.12), created by taking the parity constraint out of (5.11):

$$\max\{\bar{c}^\top y : L'y \leqslant -\alpha \cdot a, y \in \mathbb{Z}^{n_L - 1}\}. \tag{5.12}$$

The optimization problem (5.12) is an integer program with TU constraint matrix, therefore it can be solved in strongly polynomial time using Tardos's Theorem [13]. Furthermore, given an optimal solution to (5.12), one can find in strongly polynomial time an optimal solution which is a vertex. This is true, because the feasible region of the LP relaxation of (5.12) coincides with the convex hull of (5.12) and it is well-known that given an optimal solution to LP, one can find in strongly polynomial time an optimal solution to LP which is a vertex.

Let u^* be an optimal vertex solution to (5.12). If u^* satisfies $u^*(S_{L'}) \equiv \gamma \mod 2$, then u^* is an optimal solution to (5.11).

Otherwise, let L'_I be the submatrix of L' consisting of the inequalities which are tight for u^*, and let $L'_{\bar{I}}$ be the submatrix of L' containing the remaining inequalities of L'. By applying the substitution $y \to x + u^*$ to (5.11), we have

$$\max \left\{ \bar{c}^\top x : L'_I x \leqslant 0,\ L'_{\bar{I}} x \leqslant -\alpha \cdot a_{\bar{I}} - L'_{\bar{I}} u^*,\ x \in \mathbb{Z}^{n_L - 1},\ x(S_{L'}) \equiv 1 \mod 2 \right\}. \quad (5.13)$$

Furthermore, we can solve the optimization problem (5.13), by finding an optimal solution x^* to

$$\max \left\{ \bar{c}^\top x : L'_I x \leqslant 0,\ x \in \mathbb{Z}^{n_L - 1},\ x(S_{L'}) \equiv 1 \mod 2 \right\} \quad (5.14)$$

which satisfies $||x^*|| \leqslant 1$. This is equivalent to finding a well-structured optimal solution to the CPTU problem created by replacing x with $x^+ - x^-$ in (5.14), where $x^+, x^- \in \mathbb{Z}^{n_L - 1}_{\geqslant 0}$.

This proves that we can solve the subproblems (5.3) by solving a CPTU problem with $\leqslant m_1$ rows. $\square$

6. Solving the base block problems

The final part of the algorithm deals with solving the base block problems. Solving CPTU where the constraint matrix T is one of the two matrices in (2.2) can be done in a fixed amount of operations and therefore it takes strongly polynomial time. We will now discuss how to solve CPTU when T is a network matrix, and when T is the transpose of a network matrix. In this chapter, we follow the work done by Artmann et al. [2].

6.1 Solving CPTU where T is a network matrix

Let T be a network matrix, by Theorem 2.6, we can find in strongly polynomial time a representation of T in terms of a directed graph (V, A) and a tree (V, U). Then CPTU can be written as

$$\max\{c^\top x : Tx \leqslant 0, x \in \mathbb{Z}^A_{\geqslant 0}, x(S) \text{ odd}\}, \tag{6.1}$$

where $T \in \mathbb{Z}^{U \times A}$ and $S \subseteq A$. The optimization problem (6.1) is equivalent to

$$\min\{-c^\top x : Tx + y = 0, x \in \mathbb{Z}^A_{\geqslant 0}, y \in \mathbb{Z}^U_{\geqslant 0}, x(S) \text{ odd}\} \tag{6.2}$$

As explained below, (6.2) describes a minimum cost flow circulation problem on the graph $G = (V, A \cup U)$ with the condition that the units of flow on the arcs in S have to sum up to an odd number and where arc $a \in A$ has cost $-c_a$ and arcs in U have cost 0.

Lemma 6.1 (Lemma 5.3 in Artmann et al.). *The following statements are equivalent:*

(i) $Tx + y = 0$.

(ii) (x, y) *is a feasible flow circulation on the graph* $G = (V, A \cup U)$.

Proof. Let $u = (i, j) \in U$ be an arc of the spanning tree, and let W_u be the set of vertices of the connected component in $(V, U \setminus \{u\})$ containing i. Consider the row of the constraint matrix in (6.2) which corresponds to the arc u, we have

$$(Tx + y)_u = x(\delta^+_A(W_u)) - x(\delta^-_A(W_u)) + y_u = 0$$

which, if we interpret (x, y) as a flow, it indicates that the net flow leaving W_u in the graph G is zero. This implies that (x, y) is a feasible flow circulation. One way to convince ourselves that this implication is true, is to first look at arcs $u = (i, j) \in U$ where one of the vertices, say i, is a leaf of the spanning tree, then $W_u = \{i\}$ and therefore the net flow leaving i is zero. We then consider the arcs which are one arc away from a leaf and so on. Since U is a spanning tree, we have that for all $v \in V$ the net flow leaving v is 0. □

The problem of finding a circulation in G of minimum cost is equivalent to finding a directed circuit in G of minimum cost. We have therefore reduced the optimization problem (6.1) to

$$\min\{l(C) : C \subseteq A \cup U \text{ is a directed circuit in } G \text{ with } |C \cap S| \text{ odd}\} \tag{6.3}$$

where $l(C)$ is the cost of going through the circuit. Lemma 6.2 concludes the demonstration that one can solve a CPTU problem with a network constraint matrix in strongly polynomial time.

Lemma 6.2. *We can solve the optimization problem* (6.3) *in strongly polynomial time.*

Proof. We define G' by, $G' = (V \cup V', A_1 \cup A_2 \cup A_3)$ where V' is a copy of V (the duplicate of $v \in V$ is denoted by $v' \in V'$), and where A_1, A_2, A_3 are defined as follows:

$$\begin{aligned} A_1 =& \{(u, v) : (u, v) \in (A \cup U) \setminus S\} \\ A_2 =& \{(u', v') : (u, v) \in (A \cup U) \setminus S\} \\ A_3 =& \{(u', v) \cup (u, v') : (u, v) \in S\} \end{aligned}$$

where each arc in G' inherits the cost of the corresponding arc in G it was created from. Note that the only way to go from a vertex in V to a vertex in V' is through an edge of S. Therefore for any walk W from V to V' we have that $|W \cup S|$ is odd, in particular a walk from v to v' in G' corresponds to a circuit C in G with $|C \cup S|$ odd. Furthermore, a walk in G' and the corresponding walk in G have the same cost. Therefore to find the optimal circuit in G we can calculate the shortest walk from v to v' for all $v \in V$ and the one with minimum cost will give a solution to (6.3). Notice that G' does not have cycles with negative costs, because this would imply that CPTU is unbounded, therefore we can use Belman-Ford algorithm to find such shortest paths in G'. For an exposition of Belman-Ford algorithm, we refer the reader to Section 24.1 in [6]. □

6.2 Solving CPTU where T is the transpose of a network matrix

Consider the CPTU optimization problem

$$\max\{c^\top x : Tx \leqslant 0, x \in \mathbb{Z}^U_{\geqslant 0}, x(S) \text{ odd}\} \tag{6.4}$$

where $T \in \mathbb{Z}^{A\times U}$ is the transpose of a network matrix. Let the directed graph (V, A) and the tree (V, U) be a representation of $T^\top$.

Lemma 6.3 (Lemma 5.10 in Artmann et al. [2])**.** *Let Q^* be an optimal solution to the optimization problem*

$$\max\{c(\delta^-_U(Q)) : Q \subseteq V,\ \delta^+_U(Q) = \emptyset,\ \delta^-_A(Q) = \emptyset, |Q \cap K| \equiv 1 \mod 2\} \tag{6.5}$$

with $K = \{v \in V : |\delta_U(v) \cap S| \equiv 1 \mod 2\}$, then $\chi^{\delta^-_U(Q^*)}$ is an optimal solution to (6.4).

Proof. To begin, we prove that if Q is feasible for (6.5), then $\chi^{\delta^-_U(Q)}$ is feasible for (6.4). If $|Q \cap K|$ is odd, then, by definition of K, $|(\delta_U(Q) \cap S)|$ is odd. Furthermore, since $\delta^+_U(Q) = \emptyset$, we have

$$|(\delta_U(Q) \cap S)| = |(\delta^-_U(Q) \cap S)| \equiv 1 \mod 2$$

which implies that $x = \chi^{\delta^-_U(Q)}$ satisfies the parity constraint.

Consider an arc $a = (u, v) \in A$ and let P be the unique path in U from u to v. We cannot have that $u \notin Q$ and $v \in Q$ because $\delta^-_A(Q) = \emptyset$. If u and v are both in Q or both not in Q then the path P crosses the cut $\delta_U(Q)$ an even number of times, and since $\delta^+_U(Q) = \emptyset$ whenever P leaves Q it uses an arc of $\delta^-_U(Q)$ backwardly, and whenever P enter Q it uses an arc of $\delta^-_U(Q)$ forwardly. This gives $(Tx)_a = 0$. While for the case where $u \in Q$ and $v \notin Q$ then P use one more arc of $\delta^-_U(Q)$ backwardly, so in this case we have $(Tx)_a = -1 \leqslant 0$. Observe that Q and $\chi^{\delta^-_U(Q)}$ achieve the same objective value.

By Lemma (5.1), we have a well-structured solution x^* to (6.4). Since we can append rows of the identity matrix to T and preserve TU-ness and since $x^* \geqslant 0$, we have that $x^* \in \{0, 1\}^U$. This means that we always have an optimal solutions x^* which corresponds to a set of arcs in U, i.e. $x^* = \chi^{X^*}$ with $X^* \subseteq U$. We now prove that any well-structured optimal solution to (6.4) has a corresponding set $Q \subseteq V$ of equal objective value, such that $\delta^+_U(Q) = \emptyset$ and $\delta^-_A(Q) = \emptyset$.

Suppose the arcs $u = (a, b)$ and $v = (c, d)$ are two arcs in the unique path $P \subseteq U$ from a to d, then it cannot be that u, v and no other arc in P is in X^*. Suppose on the contrary, that u and v are both in X^* and no other arc of P is in X^*, then the graph $(V, A \cup (a,d))$ with the tree (V,U) generate the network matrix T with a new appended row w corresponding to (a,d). Since x^* is a well-structured solution, we should have that $w \cdot x^{*\top} \in \{-1,0,1\}$, but instead we have $|w \cdot x^{*\top}| = 2$, raising a contradiction. This implies that $X^* = \delta_U^-(Q')$ for some $Q' \subseteq V$. In fact, if this was not the case, then there would be a path in U where the only arcs of P which are in X^* are two arcs both used in the same direction, but this is not possible as we have just seen.

We now show how, starting from Q', we can find the desired $Q \subseteq V$. Suppose $\delta_U^+(Q') \neq \emptyset$, then there exists an arc (u,v) with $u \in Q'$ and $v \notin Q'$. Consider the connected component C_v of the graph $(V, U \setminus (u,v))$ containing v, $\delta_U^+(C_v) = \emptyset$ if not we would again end up with a path in U where the only arcs in X^* are two arcs crossed in the same direction. Setting $Q' \leftarrow Q' \cup W_v$, reduces the number of arcs in $\delta_U^+(Q')$. We can repeat the process until $\delta_U^+(Q') = \emptyset$; notice that at any repetition we do not change $\delta_U^-(Q')$ which implies that the objective value does not change. At the end of the process, we reach the set Q with $\delta_U^+(Q) = \emptyset$, furthermore, since x^* satisfies $Tx^* \leqslant 0$ we must have that $\delta_A^-(Q) = \emptyset$.

Therefore if Q^* is optimal for (6.5), then $x^* = \chi^{\delta_U^-(Q^*)}$ is a feasible solution to (6.4) with equal objective value. Furthermore the optimal well-structured solutions to (6.4), correspond to subsets in V which have objective value not grater than Q^*. This implies that $x^* = \chi^{\delta_U^-(Q)}$ is an optimal solution for (6.4). □

We now show how the optimization problem (6.5) can be transformed into a minimization problem with a *submodular* objective function over subsets with odd cardinality. These problems are known to be solvable in strongly polynomial time, (see Groetschel et al. [8] Section 10.4 for an exposition of the fact), therefore Lemma 6.5 concludes the proof that CPTU with $T^\top$ a network matrix, can be solved in strongly polynomial time.

Definition 6.4. *Let E be a finite set, a function $f : 2^E \to \mathbb{R}$ is called* ***submodular*** *on 2^E if for every $X, Y \subseteq E$ with $X \subseteq Y$ and for every $x \in E \setminus Y$ we have that*

$$f(X \cup \{x\}) - f(X) \geqslant f(Y \cup \{x\}) - f(Y).$$

Lemma 6.5. *The optimization problem* (6.5) *is equivalent to the minimization problem*

$$\min\{f(Q) : Q \subseteq V,\ |Q \cap K| \equiv 1 \mod 2\} \tag{6.6}$$

where f is a submodular function.

Proof. Note that if $\delta_U^+(Q) = \emptyset$, we have the following equivalences for the objective of (6.5):

$$-c(\delta_U^-(Q)) = c(\delta_U^+(Q)) - c(\delta_U^-(Q)) = \sum_{q \in Q} \left(c(\delta_U^+(q)) - c(\delta_U^-(q)) \right)$$

Furthermore we can encode the constraints $\delta_U^+(Q) = \emptyset$ and $\delta_A^-(Q) = \emptyset$ of (6.5) into the objective function. We can do so by adding penalties when $\delta_U^+(Q)$ or $\delta_A^-(Q)$ are not the empty set:

$$g_1(Q) = M \cdot |\delta_U^+(Q)|$$
$$g_2(Q) = M \cdot |\delta_A^-(Q)|.$$

If we fix M large enough, for instance $M = 1 + \sum_{u \in U} |c(u)|$, then any optimal solution to (6.6), with $f(Q) = \sum_{q \in Q} \left(c(\delta_U^+(q)) - c(\delta_U^-(q)) \right) + g_1(Q) + g_2(Q)$ must satisfy $\delta_U^+(Q) = \emptyset$ and $\delta_A^-(Q) = \emptyset$. $\sum_{q \in Q} \left(c(\delta_U^+(q)) - c(\delta_U^-(q)) \right)$, g_1 and g_2 are all submodular functions, and therefore their sum is submodular. $\square$

7. Proof of the main theorem

Proving the main theorem is now a question of combining together the results obtained in Chapters 3, 4, 5 and 6.

Proof of Theorem 1.3. If BIP is infeasible or unbounded we can show that this is the case in strongly polynomial time, using respectively Theorem 3.4 and Corollary 3.4.1.

If BIP is feasible and bounded, then by Lemma 4.7 we can solve BIP by solving a feasible and bounded CPTU problem and performing operations taking strongly polynomial time. We are left to prove that we can solve a feasible and bounded CPTU problem in strongly polynomial time.

Notice that if the constraint matrix T of CPTU is a base block, then we have proved in Chapter 6 that we can find an optimal solution in strongly polynomial time. Furthermore, Lemma 5.3 shows how to reconstruct in strongly polynomial time an optimal solution to the original CPTU problem, given optimal solutions for the CPTU problems it was decomposed into.

Note that if the algorithm makes the CPTU problem *slim* any time it has the chance, then the number of variables is polynomially bounded by the number of rows. We say that a totally unimodular matrix A is *slim* if there are no dominated columns. We say column A_i dominates column A_j, if $A_i = A_j$, i and j are both in S or both not in S, and $c_i \geqslant c_j$. By a result from Heller [9], if A is a totally unimodular matrix with no repeated columns and no zero column, then A has at most $m(m+1)$ columns, where m is the number of rows. Because in a CPTU problem with slim constraint matrix $T \in \mathbb{Z}^{m\times n}$, each column appears at most twice (once in S and once not in S), then T has at most $2m(m+1)$ columns. Observe that it takes strongly polynomial time to make a TU matrix slim, furthermore given an optimal solution for the CPTU problem with a slim constraint matrix, one can retrieve an optimal solution for the CPTU with the original matrix in strongly polynomial time (it can be done by setting $x_i = 0$ for all i such that A_i is a dominated column).

Consider a CPTU problem with m rows. We are going to prove by induction on m that the total number of subproblems which are solved (not through decomposition) in order to solve CPTU, is polynomially bounded by m.

Notice that we can fix a value ℓ so that all non-base block CPTU problems with less than ℓ rows are solved without any further decomposition; we fix $\ell = 16$. We call a CPTU problem *small* if it is a base block or if it has less than ℓ rows. Let $\mathcal{B}(m)$ be the maximal number of small problems solved in order to find an optimal solution to a CPTU problem with m rows. Then for $m < \ell$, we have $\mathcal{B}(m) = 1$. We now prove that for all $m \geqslant \ell$, $\mathcal{B}(m) \leqslant Cm^\gamma$ for some fixed C and γ. We additionally set $C \geqslant 100$ and $\gamma \geqslant 4$.

By Theorem 5.4, we have that that

$$\mathcal{B}(m) \leqslant 14\mathcal{B}(m_1) + \mathcal{B}(m_2).$$

We consider two cases depending on the value of m_1.

If $m_1 < \ell$, then

$$\begin{aligned} 14\mathcal{B}(m_1) + \mathcal{B}(m_2) &\leqslant 14 \cdot 1 + \mathcal{B}(m_2) \\ &\leqslant 14 + \mathcal{B}(m-1) \\ &\leqslant 14 + C(m-1)^\gamma \\ &\leqslant Cm^\gamma \end{aligned}$$

where the second inequality follows from $m_2 \leqslant m-1$ and the fourth follows from $C \geqslant 100$.

If $m_1 \geqslant \ell$, we apply the substitution $m_2 = m + 2 - m_1$ and we consider the function

$$f(m_1) = 14\mathcal{B}(m_1) + \mathcal{B}(m + 2 - m_1)$$

with m fixed. Note that $f(m_1)$ is convex for $m_1 \leqslant m + 2$, which contains all possible feasible values of m_1. Furthermore, the smallest and highest possible values for m_1 are ℓ and $\frac{m}{2} + 1$. Therefore, to prove that $14\mathcal{B}(m_1) + \mathcal{B}(m_2) \leqslant Cm^\gamma$, is enough to prove that

$$14\mathcal{B}(\ell) + \mathcal{B}(m + 2 - \ell) \leqslant Cm^\gamma \tag{7.1}$$

and

$$14\mathcal{B}\left(\frac{m}{2} + 1\right) + \mathcal{B}\left(m + 2 - \left(\frac{m}{2} - 1\right)\right) \leqslant Cm^\gamma. \tag{7.2}$$

For (7.2) we have

$$\begin{aligned} 14C\left(\frac{m}{2}+1\right)^{\gamma}+C\left(m+2-\left(\frac{m}{2}-1\right)\right)^{\gamma} &= 14C\left(\frac{m}{2}+1\right)^{\gamma}+C\left(\frac{m}{2}+3\right)^{\gamma} \\ &\leqslant 15C\left(\frac{m}{2}+3\right)^{\gamma} \\ &\leqslant Cm^{\gamma} \end{aligned}$$

where the second inequality holds because $\gamma \geqslant 4$ and $m \geqslant 2\ell+2 > 200$. For (7.1), observe that the function

$$g(m) = Cm^{\gamma} - 14C\ell^{\gamma} - C(m+2-\ell)^{\gamma}$$

is increasing for all feasible values of m. Therefore, to prove inequality (7.1), it is enough to show that $g(m) \geqslant 0$ for a value smaller or equal than all possible values of m. Note $m = m_1 + m_2 + 2 > 2\ell - 2$, furthermore

$$\begin{aligned} g(2\ell-2) &= C(2\ell-2)^{\gamma} - 15C\ell^{\gamma} \\ &\geqslant C\ell^{\gamma-1}(2\ell-2) - 15C\ell^{\gamma} \\ &= C\ell^{\gamma-1}(2\ell-2-15\ell) \\ &\geqslant 0 \end{aligned}$$

where the last inequality holds because $\ell = 16$.

Therefore, for a CPTU problem with m rows, one needs to solve at most Cm^{γ} small problems. The other operations which are performed to solve the CPTU problem with m rows are: decomposing a CPTU problem with $m_0 \leqslant m$ rows, making the newly created subproblems slim and retrieving an optimal solution for the CPTU problem with m_0 rows by combining the optimal solutions of the subproblems. These operations can be done in time polynomially bounded by m_0, and furthermore they are done less than Cm^{γ} times. It follows that the algorithm solves bimodular integer programs in strongly polynomial time. $\square$

Bibliography

[1] S. Artmann, F. Eisenbrand, C. Glanzer, T. Oertel, S. Vempala, and R. Weismantel. A note on non-degenerate integer programs with small sub-determinants. *Operations Research Letters*, 44(5):635–639, 2016.

[2] 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*, pages 1206–1219, 2017.

[3] D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[4] R. H. Byrd, A. J. Goldman, and M. Heller. Recognizing unbounded integer programs. *Operations Research*, 35(1):140–142, 1987.

[5] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer programming*, volume 271. Springer, 2014.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms.* MIT press, 2009.

[7] W. H. Cunningham and J. Edmonds. A combinatorial decomposition theory. *Canadian Journal of Mathematics*, 32(3):734–765, 1980.

[8] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[9] I. Heller et al. On linear systems with integral valued solutions. *Pacific Journal of Mathematics*, 7(3):1351–1364, 1957.

[10] L. G. Khachiyan. A polynomial algorithm in linear programming. In *Doklady Akademii Nauk*, volume 244, pages 1093–1096. Russian Academy of Sciences, 1979.

[11] A. Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998.

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

[13] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986.

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

[15] P. Zörnig. Degeneracy problems in mathematical optimization. In *Degeneracy Graphs and Simplex Cycling*, pages 3–14. Springer, 1991.