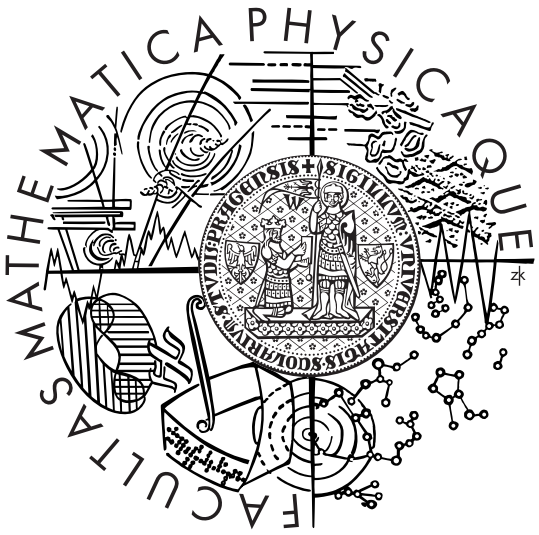
CHARLES UNIVERSITY
FACULTY OF MATHEMATICS AND PHYSICS



HABILITATION THESIS

LINEAR AND EXACT EXTENDED FORMULATIONS

HANS RAJ TIWARY

Department of Applied Mathematics
2016

For Maria and Palomito.

ACKNOWLEDGMENTS

This thesis would not have taken form if not for the numerous wonderful collaborators with whom I had the pleasure and honor to work. My first and foremost thanks to all of them.

I would also like to thank the various mentors I had over the years: Raimund Seidel, Günter M. Ziegler, and Samuel Fiorini. I may not have learnt everything that I could have learnt from them, but the things that I did learn were numerous.

Special thanks also to members of KAM and IUUK for a very friendly and encouraging environment.

Last but not the least, I would like to thank my wife Maria. Without her support I would be utterly lost.

The finger can point to the moon's location.
However, the finger is not the moon.
To look at the moon, it is necessary to gaze beyond the finger, right?"

— Hotei, The laughing Budhha

PREFACE

In recent years there has been a flurry of activity regarding bounds on the extension complexity of combinatorial polytopes. This work covers some of those results in which I took part. At present, this document is part of my habilitation thesis and as such I do not make any attempt to be comprehensive about research on extended formulations. To a knowledgeable reader it may be glaringly obvious that some important aspects of the recent research related to extended formulations are missing from this work. In particular there is essentially no discussion about approximate extended formulations and semidefinite extended formulations. Hopefully a future version of this document will discuss these aspects.

This document – in its current form – is best read as a companion and commentary to ten of the research articles that I have coauthored. These articles are listed after the table of contents. It has been my intent to disassemble and reassemble the contents of these papers to provide the reader with a coherent view of my research in recent years. While individual tastes may differ regarding the value of these lines of inquiry, I have attempted to sew them with a common thread. Naturally, this document also contains results in which I played no part and I have attempted to cite the correct source for those statements. I take full responsibility for any omissions and misattributions, and hope that if a reader notices such an issue they will kindly notify me.

The reader may also notice that this document does not contain any lemmata or theorems, just a sequence of propositions and a few exercises. This choice was made to keep the size of the document reasonable for a habilitation thesis. Listing propositions and exercises had an obvious benefit for me: I could get away with listing only some of the proofs. I have attempted to provide a link to the actual proof where I could find one, but for some propositions and exercises I am not aware of any text that lists them in the same form as stated in this work. This does not mean that they are difficult to prove or are novel, but often a correct proof would require technical discussions that we do not wish to have. I can only hope that the exercises and the propositions that have been stated without proofs are simple enough for someone with basic knowledge of the material. I am also hopeful that a future revision of this work will include the missing proofs.

In its present form, this work assumes a relatively high level of familiarity with polytopes and communication complexity. This does not mean that the reader needs to be an expert in these fields. It is

my estimate that someone pursuing a PhD in theoretical computer science or a related field should be able to follow the text, fill missing proofs, and understand the presented propositions.

It is my hope that anyone interested in extended formulations finds this document helpful. Any comments on how to better this text is most welcome.

Hans Raj Tiwary

CONTENTS

LIST OF FIGURES

PUBLICATIONS

This thesis is based on the following ten research articles that I have coauthored. A copy of each article is attached in the appendices.

[1] David Avis and Hans Raj Tiwary. "A generalization of extension complexity that captures P." In: *Information Processing Letters* 115.6-8 (2015), pp. 588–593. DOI: 10.1016/j.ipl.2015.02.005.

[2] David Avis and Hans Raj Tiwary. "On the extension complexity of combinatorial polytopes." In: *Mathematical Programming* 153.1 (2015), pp. 95–115. DOI: 10.1007/s10107-014-0764-2.

[3] David Avis and Hans Raj Tiwary. "On the $\mathcal{H}$-free extension complexity of the TSP." In: *Optimization Letters* (2016), pp. 1–11. ISSN: 1862-4480. DOI: 10.1007/s11590-016-1029-1.

[4] David Avis, David Bremner, Hans Raj Tiwary, and Osamu Watanabe. "Polynomial size linear programs for non-bipartite matching problems and other problems in P." In: *CoRR* abs/1408.0807 (2014). eprint: arXiv:1408.0807.

[5] Yuri Faenza, Samuel Fiorini, Roland Grappe, and Hans Raj Tiwary. "Extended formulations, nonnegative factorizations, and randomized communication protocols." In: *Mathematical Programming* 153.1 (2015), pp. 75–94. DOI: 10.1007/s10107-014-0755-3.

[6] Samuel Fiorini, Thomas Rothvoß, and Hans Raj Tiwary. "Extended Formulations for Polygons." In: *Discrete & Computational Geometry* 48.3 (2012), pp. 658–668. DOI: 10.1007/s00454-012-9421-9.

[7] Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. "Exponential Lower Bounds for Polytopes in Combinatorial Optimization." In: *Journal of the ACM* 62.2 (2015), p. 17. DOI: 10.1145/2716307.

[8] Jakub Gajarský, Petr Hliněný, and Hans Raj Tiwary. "Parameterized Extension Complexity of Independent Set and Related Problems." In: *CoRR* abs/1511.08841 (2015). eprint: arXiv:1511.08841.

[9] Petr Kolman, Martin Koutecký, and Hans Raj Tiwary. "Extension Complexity, MSO Logic, and Treewidth." In: *Proceedings of the 15th SWAT* To appear (2016). eprint: arXiv:1507.04907.

[10] Hans Raj Tiwary. "Extension Complexity of Formal Languages." In: *ArXiv e-prints* (Mar. 2016). arXiv: 1603.07786 [cs.CC].

"The time has come," the Walrus said,
"To talk of many things:
Of shoes–and ships–and sealing-wax–
Of cabbages–and kings–
And why the sea is boiling hot–
And whether pigs have wings."

— The Walrus and The Carpenter [17]

0 INTRODUCTION

HISTORY

This thesis deals with recent advances in the theory of extended formulations, with emphasis on linear and exact formulations. A linear and exact extended formulation for a polytope P is another polytope Q such that P is a projection of Q. If one wants to optimize a linear function over P one can also obtain the same result by optimizing instead over an extended formulation with the weights modified according to the projection matrix that produces P from Q (cf. Chapter 3). Many polytopes that require an exponential number of facets to describe can be obtained as a projection of a higher dimensional polytope with just polynomially many facets [19, 39, 62]. The importance of extended formulations, thus, is clear for combinatorial optimization.

The directions that are explored in this thesis come from the reverse perspective. In the late eighties Swart attempted to prove that PTIME = NP by writing a polynomial sized linear program (LP) for the traveling salesman problem. Due to the large size of the LP and its complicated nature, it was difficult to find an error in the construction. In a groundbreaking paper Yannakakis showed that any symmetric[1] LP whose feasible region is an extended formulation for the TSP polytope must have exponentially many facets, thus proving that Swart's LP was erroneous. Whether or not the requirement of symmetry could be removed was left by Yannakakis as an open problem. Yannakakis believed that asymmetry should not help one avoid exponential size [63].

This question remained dormant for about two decades when Kaibel, Pashkovich, and Theis showed that asymmetry does play an important role in reducing the size of an extended formulation [40]. They gave explicit polytopes whose symmetric extended formulations required exponential size but for which asymmetric extensions of polynomial size existed. Two years later Rothvoss showed that there are 0/1 polytopes that require exponential size extended formulations [54]. These two results created a fresh interest in proving that symmetry did not help for polytopes related to NP-hard problems such as MAX-CUT or TSP. Soon afterwards in a paper coauthored by Fiorini, Pokutta, Massar, de Wolf, and the present author, it was shown that the TSP polytope requires exponential sized extended formulation [27].

The final piece of the puzzle initiated by Yannakakis was supplied by Rothvoss again who showed that even the perfect matching polytope – which corresponds to the polynomial time solvable problem

1 The reader need not concern themselves with the meaning of a symmetric LP; it suffices to think of it as a technical requirement that Swart's LP satisfied.

of identifying whether a graph has a perfect matching – also requires exponential size extended formulations [55]. The rusults of Rothvoss and Fiorini et al. created a flurry of activity for proving lower bounds for extension complexity of polytopes. While Rothvoss' result implied that even "easy" problems may require large extended formulations, the unconditional nature of the lower bounds – at least for "hard" problems seemed to match the conventional conditional bounds relying on complexity assumptions such as PTIME $\neq$ NP.

At this point the lines of equiry about bounds on the size of smallest extensions of a polytope had branched into widely different directions. The notion of extended formulations was generalized to arbitrary conic lifts [33]; lower bounds were obtained for approximation [9, 10, 18] and semidefinite extensions [12, 25, 46]; and connections with physical theories [28] and information theory [8, 11] were discovered. The previous list of citations do not do any justice to the widely diverse results that have been obtained since then, and we will not even attempt to have a comprehensive citation in order to keep the focus on works where the present author has taken part.

SUMMARY AND ORGANIZATION OF THE THESIS

This thesis summarizes ten articles that the present author has coauthored and that relate to extended formulations. Out of the ten articles six have appeared in peer-reviewed journals [1–3, 24, 26, 27]. The article [44] has been accepted for presentation in a peer-reviewed coference, and [4, 30] and [60] are under peer-review.

This thesis is best read as a commentary to these ten accompanying articles. The contents of these ten articles have been disassembled and reassembled as smaller parts of a larger picture. The rest of thesis is organized into three parts, each containing three chapters.

0.0.1 *Part One: Ingredients*

Part one of the thesis describes the basic objects that we deal with: polytopes, communication complexity, and extended formulations. It is the intent of the author to present the basic notions that are relevant for the accompanying papers in a comprehensive way. We develop common terminology in which later results can be presented without repeated description of the underlying notions. This part of the thesis refers to some results that the present author coauthored in [2, 24, 26] and [27].

In Chapter 1 we collect the basic terminology related to polytopes. We present basic facts related to polytopes, discuss the role of embeddings and describe common operations on polytopes that become relevant in later chapters. This chapter contains mostly classical results about polytopes. The discussion about embeddings becomes relevant when discussing extended formulations because the same polytope may be described in various ways depending on whether the description is minimal or not, and whether the polytope is embedded in

the smallest possible Euclidean space. We develop the notion of slack equivalence of a polytope that serves as an invariant under the actual embedding of a polytope and allows us to later focus on just the combinatorial structure relevant to the extension complexity of the polytope.

In Chapter 2 we describe basic setting of communication complexity and some tools to prove lower bounds for it. In Section 2.1 we introduce nonnegative rank of a matrix which turns out to be closely related to the size of smallest extension of a polytope. We also discuss some tools to lower bound this quantity, and discuss the lower bound for a specific problem: unique disjointness. This problem serves as a canonical ingredient in proving lower bounds for the extension complexity of the CUT polytope. In Section 2.2 we introduce the notion of communication protocol and in Section 2.3 we describe three models of communication complexity, two of which are the classical deterministic and randomized models while the third one – where it is enough for a protocol to work in expectation – is relevant to extension complexity.

In Chapter 3 we finally introduce the notion of extended formulations and extension complexity. This allows us to talk about the size of the smallest possible extended formulation of a polytope. In Section 3.1 we present the notion of extension complexity for a set of related polytopes since one is usually interested not just in the extension complexity of a single polytope but of a family of related polytopes. We have attempted to provide a general language that allows us to later on talk not just about sets of polytopes but also about notions such as the parameterized extension complexity of polytopes. We also discuss some common tricks to bound the extension complexity of a polytope. In Section 3.2 we discuss the effects of applying some common operations of polytopes on their extension complexity. Finally, in Chapter 3.3 we present some canonical examples of polytope families and bounds on their extension complexity. Later on these examples serve us as building blocks for bounds on other families.

0.0.2 *Part two: Recipes*

Part two of the thesis presents more lower bounds for various families of polytopes. We also develop notions allowing us to talk about extension complexity of binary languages and we define the class of languages that admit polynomial size extended formulations. This part of the thesis refers to some results that the present author coauthored in [2, 27] and [60].

In Chapter 4 we present lower bounds for various classes of polytopes. These polytopes correspond to various NP-hard problems and the bounds on their extension complexity are derived using standard NP-hardness reductions together with some simple observations made in earlier chapters. We also discuss some issues in trying to make a meta statement about arbitrary polynomial time reductions.

In Chapter 5 we describe a convenient way to discuss extension complexity of binary languages. In Section 5.1 we discuss various kinds of problems related to an underlying language, and in Section 5.2 we present the class of languages admitting polynomial size extended formulations. We also discuss the impact of small extensions on the computational complexities of various kind of problems related to the underlying language, and present an example of a compact language. Our example is related to walks in directed graphs and we use this example later on to establish that problems efficiently solvable in the streaming model admit small extended formulations. In Section 5.3 we finally discuss some closure properties of compact languages.

In Chapter 6 we use the discussions from the previous chapter to prove that if a language is accepted by an online Turing machine (possibly nondeterministic) using only logarithmic space then it admits a polynomial size extended formulation. In Section 6.3 we discuss various applications of these results. We present lower bounds in the streaming model and upper bounds on extension complexity of some polytopes.

0.0.3 *Part three: Variations*

In this part we discuss some ways in which the notion of extension complexity can be generalized and applied in other settings. This part of the thesis refers to some results that the present author coauthored in [1, 3, 4, 30] and [44].

In Chapter 7 we consider the extension complexities of parameterized problems. In Section 7.1 we define the notion of parameterized extension complexity and in Section 7.2 we apply this definition to study the parameterized extension complexity of the independent set polytope parameterized by the size of the independent sets. We show that this polytope does not admit a fixed-parameter polynomial extended formulation. Finally, in Section 7.3 we given examples of two general classes of polytopes that do admit fixed-parameter polynomial extension complexity: polytopes of assignments for first order logic over graphs of bounded expansion, and polytopes of assignments for monadic second order logic over graphs of bounded treewidth.

In Chapter 8 we consider the following: suppose we identify a subset of facets of a polytope P and show that the extension complexity of the corresponding inequalities with respect to the vertices of P is large. What can be said about the extension complexity of the polytope formed by removing these inequalities from the description of P? This is motivated by practical considerations. Often for hard polytopes – such as the TSP – one can optimize efficiently over a subset of facets and various cutting plane algorithm exploit this in search for violated inequalities. In Section 8.1 we define this notion precisely. In Section 8.2 we provide strong lower bounds for polytopes related to various NP-hard matching polytopes even if we ignore the odd-set

inequalities of Edmonds. In Section 8.3 we discuss similar results for the TSP polytope with respect to a general class of comb inequalities.

In Chapter 9 we consider a weaker form of extended formulations that we call Weak Extended Formulation (WEF). Instead of requiring that the LP formulation of a problem project to the entire feasible set, we only put milder constraints on the formulation. The most important restriction being that the polytope may have some bad vertices, but it must have the right vertices along the right directions. The motivation for this is that under a polyhedral representation of a problem, we are not always interested in optimizing along arbitrary directions but only some. This hold specially true for decision problems. In Section 9.1 we motivate the reader and present the definition of a WEF in Section 9.2. Finally in Section 9.3 we discuss how every problem in P/poly admits a polynomial sized WEF.

NOTATIONS AND CONVENTIONS

We will use the following conventions throughout this document except when specifically stated otherwise.

- Boldface small letters represent column vectors; boldface capital letters represent matrices. The i-th row and the j-th column of a matrix $\mathbf{M}$ will be denoted by $\mathbf{M}_i$ and $\mathbf{M}^j$ respectively.
- Set of vectors will be denoted by capital letters. A set V can also be written as a matrix $\mathbf{V}$. For example, let $V = \{\mathbf{v_1}, \mathbf{v_2}, \ldots\}$ be a set of vectors, then $\mathbf{V}$ is the corresponding matrix whose columns are vectors $\mathbf{v_i}$.
- Similar to previous convention, if $\mathbf{V}$ is an $n \times m$ matrix then V will denote the set of column vectors of $\mathbf{V}$.
- The number of rows of a matrix $\mathbf{V}$ will be denoted by $\text{numrows}(\mathbf{V})$ and the number of columns by $\text{numcols}(\mathbf{V})$.
- For matrices $\mathbf{A}$ and $\mathbf{B}$ we will denote the matrix obtained by concatenating the columns of the matrices by $\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}$. Similarly, the matrix obtained by concatenating the rows will be denoted by $\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}$.
- Capital letters such as P will be used for denoting polytopes; $\mathcal{P}$ will denote a family of polytopes, and $\mathbb{P}$ will denote a clan of polytopes (cf. Definition 3.1.10).

SOME LINEAR ALGEBRA

The following basic notion from Linear Algebra will be useful to us later.

Definition 0.0.1. A *linear inequality* is of the form $\mathbf{a}^\top \mathbf{x} \leqslant b$ where $\mathbf{a} \in \mathbb{R}^n$ is a vector and $b \in \mathbb{R}$.

Definition 0.0.2. A linear inequality $a^\top x \leqslant b$ defines the *halfspace* $h(a, b) := \{x \in \mathbb{R}^n \mid a^\top x \leqslant b\}$. A *hyperplane* is the boundary $\{x \in \mathbb{R}^n \mid a^\top x = b\}$ associated with the halfspace $h(a, b)$.

At times – if it is clear from the context – we will not make any distinction between a linear inequality and the associated halfspace.

Definition 0.0.3. Let $V = \{v_1, v_2, \ldots, v_m\}$ be a set of m vectors in $\mathbb{R}^n$. A point $x \in \mathbb{R}^n$ is said to be a *convex combination* of the vectors in V if there exists a vector $\lambda \in \mathbb{R}^m$ such that

$$\begin{aligned} V\lambda &= x, && (1)\\ 1^\top\lambda &= 1, && (2)\\ \lambda &\geqslant 0 && (3) \end{aligned}$$

where 1 and 0 are column vectors of all ones and zeroes respectively.

Dropping the requirement that the sum of λ_i's be one, we get the notion of affine combination.

Definition 0.0.4. The *affine hull* of a set of vectors $V = \{v_1, v_2, \ldots, v_m\}$ – denoted by $\mathrm{aff}(V)$ – is defined to be the set of vectors that are affine combinations of v_i's. That is,

$$\mathrm{aff}(V) := \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{aligned} V\lambda &= x, \\ \lambda &\geqslant 0 \end{aligned} \right\}$$

Finally, dropping the nonnegativity requirement, we get the notion of linear combination.

Definition 0.0.5. The *linear hull* of a set of vectors $V = \{v_1, v_2, \ldots, v_m\}$ – denoted by $\mathrm{lin}(V)$ – is defined to be the set of vectors that are linear combinations of v_i's. That is,

$$\mathrm{aff}(V) := \left\{ x \in \mathbb{R}^n \;\middle|\; V\lambda = x \right\}$$

For a vector $v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$ we say that the dimension of v is n. For a set of vectors $V = \{v_1, v_2, \ldots, v_m\}$ with $v_i \in \mathbb{R}^n$ the *ambient dimension* is n, while the dimension – denoted by $\dim(V)$ – is equal to the dimension of their affine hull. That is, the vectors in the set V live in a $\dim(\mathrm{aff}(V))$ dimensional subspace of $\mathbb{R}^n$.

Part I

INGREDIENTS

Whenever Gutei Oshõ was asked about Zen, he simply raised his finger. Once a visitor asked Gutei's boy attendant, "What does your master teach?"

The boy too raised his finger.

Hearing of this, Gutei cut off the boy's finger with a knife. The boy, screaming with pain, began to run away. Gutei called to him, and when he turned around, Gutei raised his finger.

The boy suddenly became enlightened.

— The Gateless Gate: Case 3 [38]

1 POLYTOPES

Polytopes are generalizations of polygons to higher dimensional Euclidean spaces. Whereas polygons are relatively simple objects, their higher dimensional analogs have a much richer structure. In this chapter we collect some basic notions related to polytopes that will be relevant to us. For more details we refer the reader to the excellent textbooks by Grünbaum [35] and Ziegler [65].

1.1 BASIC FACTS ABOUT POLYTOPES

Definition 1.1.1. Let $V = \{v_1, \ldots, v_m\}$ be a subset of $\mathbb{R}^n$. The *convex hull* of V – denoted by $\operatorname{conv}(V)$ – is defined as

$$\operatorname{conv}(V) := \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{rcl} V\lambda & = & x, \\ \mathbf{1}^\top \lambda & = & 1, \\ \lambda & \geqslant & 0 \end{array} \right\}$$

Definition 1.1.2. An *$\mathcal{H}$-polytope* in $\mathbb{R}^n$ is a bounded subset of $\mathbb{R}^n$ that is defined by the intersection of a finite number of halfspaces. A *$\mathcal{V}$-polytope* in $\mathbb{R}^n$ is the convex hull of a finite subset of $\mathbb{R}^n$.

Note that the intersection of a finite number of halfspaces need not always be bounded. However we are only interested in the ones that are. Therefore, we will always assume boundedness unless explicitely stated otherwise.

Example 1.1.3. The same octagon described as the intersection of some halfspaces, and as the convex hull of some points.

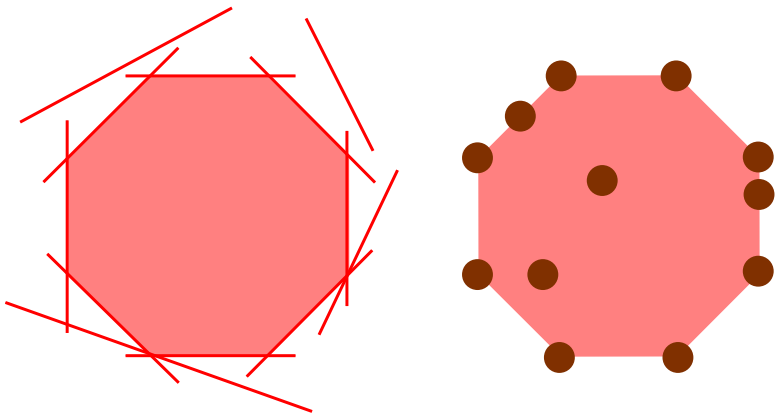


Figure 1: An octagon as an $\mathcal{H}$-polytope (left) and as a $\mathcal{V}$-polytope (right)

An $\mathcal{H}$-polytope $P := \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ will be often written as $P(A, b)$. Similarly, a $\mathcal{V}$-polytope $P := \operatorname{conv}(V)$ will be written as $P(V)$.

Example 1.1.4. Let $\mathbf{V}$ be an $n \times m$ real matrix. Then,

$$\begin{aligned} P(\mathbf{V}) &= \text{conv}(V), \\ P(\mathbf{V}, \mathbf{1}) &= \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{V}\mathbf{x} \leqslant \mathbf{1}\}, \\ P(\mathbf{V}^\top, \mathbf{1}) &= \left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{V}^\top \mathbf{x} \leqslant \mathbf{1}\right\}. \end{aligned}$$

Note again, that $P(\mathbf{V}, \mathbf{1})$ need not always a polytope. In fact, a precise characterization of boundedness of $P(\mathbf{V}, \mathbf{1})$ is possible in terms of the location of the origin with respect to $P(\mathbf{V}))$ and $\dim(\text{aff}(V))$ but the previous example is just to illustrate the notation.

Exercise 1.1.5. Let $\mathbf{V}$ be an $n \times m$ matrix. Show that $P(\mathbf{V}, \mathbf{1})$ is a polytope if and only if $\dim(\text{aff}(V)) = m$ and there exists strictly positive convex multipliers such that $\mathbf{0}$ is a convex combination of vectors in V (columns of $\mathbf{V}$).

Proposition 1.1.6. *P is an $\mathcal{H}$-polytope if and only if it is a $\mathcal{V}$-polytope.*

Proof. See [65], Theorem 1.1. □

Proposition 1.1.6 ensures that every polytope can be represented both by an intersection of a finite number of linear inequalities and as the convex hull of a finite number of points. So we can refer to a *polytope* instead of an $\mathcal{H}$- or a $\mathcal{V}$-polytope, and we can reserve the words $\mathcal{H}$-, $\mathcal{V}$-polytope to refer to a particular representation of a polytope. The shorthands such as $P(\mathbf{V})$ or $P(\mathbf{A}, \mathbf{b})$ will be referred to as a *description* of the associated polytope, with $P(\mathbf{V})$ being a $\mathcal{V}$-description and $P(\mathbf{A}, \mathbf{b})$ being an $\mathcal{H}$-description.

It should be immediately clear that neither $\mathcal{H}$- nor $\mathcal{V}$-descriptions of any polytope are unique, with the empty set and the convex hull of a single point being the only two exceptions for the $\mathcal{V}$-representation.

Definition 1.1.7. A $\mathcal{V}$-description, say $P(\mathbf{V})$, of a polytope is said to be *irredundant* if removing any column from $\mathbf{V}$ results in a different polytope. Otherwise, the description is said to be *redundant*.

An $\mathcal{H}$-description, say $P(\mathbf{A}, \mathbf{b})$, of a polytope P is said to be *irredundant* if removing any inequality from the system $\mathbf{A}\mathbf{x} \leqslant \mathbf{b}$ produces a set of feasible points that is different from P. Otherwise, the description is said to be *redundant*.

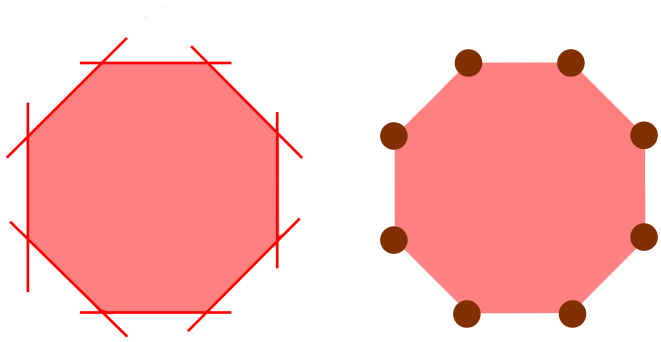


Figure 2: Irredundant $\mathcal{H}$- and $\mathcal{V}$-representation of an octagon

Definition 1.1.8. Let $P = P(\mathbf{V})$ be a polytope. We say that P is a d-polytope in $\mathbb{R}^n$ if $\dim(\text{aff}(V)) = d$ and $\text{numrows}(\mathbf{V}) = n$. Observe that $d \leqslant n$. We call P *full-dimensional* if $d = n$.

1.1.1 *Faces of a polytope*

Definition 1.1.9. An inequality $\mathbf{a}^\top \mathbf{x} \leqslant b$ is said to be valid for a polytope P if $P = P \cap \{\mathbf{x} \mid \mathbf{a}^\top \mathbf{x} \leqslant b\}$.

Definition 1.1.10. $F \subseteq P$ is called a *face* of polytope P if $F = P \cap \{\mathbf{x} \mid \mathbf{a}^\top \mathbf{x} = b\}$ for some $\mathbf{a}, b$ such that $\mathbf{a}^\top \mathbf{x} \leqslant b$ is valid for P.

We will often say that $\mathbf{a}^\top \mathbf{x} \leqslant b$ defines the face F. Observe that for any polytope P, both $\emptyset$ and P are its faces: pick $\mathbf{a} = \mathbf{0}, b = -1$ or $\mathbf{a} = \mathbf{0}, b = 1$. These two faces are called the *trivial* faces, and unless explicitly stated we will use the word "face" to refer only to non-trivial faces. Also, any face of a polytope is itself a polytope: a fact easily proven by observing that any polytope is an $\mathcal{H}$-polytope as well and that adding linear inequalities does not destroy boundedness.

Definition 1.1.11. Let $P = P(\mathbf{V})$ be a d-polytope in $\mathbb{R}^n$. We say that F is an i-face of P if F is an i-polytope and a face of P. Observe that $0 \leqslant i \leqslant d-1$. The 0-faces of a polytope are called the *vertices* and the $(d-1)$-faces are called the *facets*.

Exercise 1.1.12. Let P be a d-polytope in $\mathbb{R}^n$. Let $\mathbf{a}_1^\top \mathbf{x} \leqslant b_1$ and $\mathbf{a}_2^\top \mathbf{x} \leqslant b_2$ define the same facet. Show that there exists $\boldsymbol{\alpha} \in \mathbb{R}^n$ and scalars $\beta, \lambda_1, \lambda_2$ such that

1. $\boldsymbol{\alpha}^\top \mathbf{x} = \beta$ for all $\mathbf{x} \in P$, and
2. $\begin{pmatrix} \mathbf{a}_1 \\ b_1 \end{pmatrix} = \lambda_1 \begin{pmatrix} \mathbf{a}_2 \\ b_2 \end{pmatrix} + \lambda_2 \begin{pmatrix} \boldsymbol{\alpha} \\ \beta \end{pmatrix}$.

Exercise 1.1.13. Let P be a polytope, and let $\mathcal{F}$ be the set of facets of P. Show that, for any description $P(\mathbf{A}, \mathbf{b})$ of P $\text{numrows}(\mathbf{A}) \geqslant |\mathcal{F}|$.

Exercise 1.1.14. Let P be a full-dimensional polytope, and let $\mathcal{F}$ be the set of facets of P. Show that, there exists a description $P(\mathbf{A}, \mathbf{b})$ of P such that $\text{numrows}(\mathbf{A}) = |\mathcal{F}|$.

1.1.2 *Size of a polytope*

Definition 1.1.15. Let P be a k-polytope in $\mathbb{R}^d$. Let $\mathcal{F}$ be the set of facets of P. We define the *size* of P – denoted by $\text{size}(P)$ – as the number $|\mathcal{F}|$.

Exercises 1.1.13 and 1.1.14 are meant for the readers to convince themselves that the above definition makes sense: it assigns a unique number to every polytope and it is possible to describe a full-dimensional polytope with $\text{size}(P)$ inequalities. More precisely,

$$\text{size}(P) = \min_{P = P(\mathbf{A}, \mathbf{b})} \text{numrows}(\mathbf{A}).$$

Proposition 1.1.16. *Let* P *be a* d*-polytope in* $\mathbb{R}^n$*. Then, for every irredundant description* $P(\mathbf{A}, \mathbf{b})$ *of* P *we have that*

$$\text{numrows}(\mathbf{A}) = \text{size}(P) + 2(n - d).$$

In particular, the inequalities in any irredundant description of P can be split in two groups: $\text{size}(P)$ many of the inequalities have a bijection with the facets of P and $2(n-d)$ inequalities correspond to $(n-d)$ equations describing the affine hull of P. For full-dimensional polytopes, the irredundant $\mathcal{H}$- and $\mathcal{V}$-descriptions are unique up to a scaling of inequalities.

Proposition 1.1.17. *Let* P *be a full-dimensional polytope with* $P(\mathbf{V}_1)$, $P(\mathbf{V}_2)$, $P(\mathbf{A}_1, \mathbf{b}_1)$, *and* $P(\mathbf{A}_2, b_2)$ *its irredundant descriptions. Then,* $\mathbf{V}_1 = \mathbf{V}_2$ *up to permutation of columns and* $\begin{bmatrix}\mathbf{A}_1 & \mathbf{b}_1\end{bmatrix} = \mathbf{\Lambda}\begin{bmatrix}\mathbf{A}_2 & \mathbf{b}_2\end{bmatrix}$ *up to permutation of rows, for some diagonal matrix* $\mathbf{\Lambda}$ *with positive diagonal entries.*

1.1.3 *Slack Matrix*

Let $\mathbf{V}$ and $\mathbf{A}$ be matrices and let $\mathbf{b} \in \mathbb{R}^n$ with $\text{numrows}(\mathbf{V}) = \text{numcols}(\mathbf{A}) = n$, such that $P(\mathbf{V}) \subseteq P(\mathbf{A}, \mathbf{b})$. That is, $\mathbf{A}_i\mathbf{V}^j \leqslant \mathbf{b}_i$ for all $i \in [\text{numrows}(\mathbf{A})]$, $j \in [\text{numcols}(\mathbf{V})]$.

Definition 1.1.18. The *slack* of the polytope $P(\mathbf{V})$ with respect to the the polytope $P(\mathbf{A}, \mathbf{b})$ – denoted by $\mathbf{S}(\mathbf{A}, \mathbf{b}, \mathbf{V})$ – is defined to be the $\text{numrows}(\mathbf{A}) \times \text{numcols}(\mathbf{V})$ matrix $\mathbf{S}$ with $\mathbf{S}_{ij} = \mathbf{b}_i - \mathbf{A}_i\mathbf{V}^j$.

When $P(\mathbf{V})$ and $P(\mathbf{A}, \mathbf{b})$ describe the same polytope – say P – then $\mathbf{S}(\mathbf{A}, \mathbf{b}, \mathbf{V})$ is called a slack matrix of P. When $P(\mathbf{A}, \mathbf{b})$ and $P(\mathbf{V})$ are irredundant descriptions, we will say that $\mathbf{S}(\mathbf{A}, \mathbf{b}, \mathbf{V})$ is an irredundant slack matrix of P.

Exercise 1.1.19. Let P be a d-polytope in $\mathbb{R}^n$ with $d \geqslant 1$. Show that no slack matrix of P contains a zero column.

Proposition 1.1.20. *Let* P *be a* d*-polytope in* $\mathbb{R}^n$*. Let* $\mathbf{S}(\mathbf{A}_1, \mathbf{b}_1, \mathbf{V}_1)$ *and* $\mathbf{S}(\mathbf{A}_2, \mathbf{b}_2, \mathbf{V}_2)$ *be irredundant slack matrices of* P*. Then,* $\mathbf{V}_1 = \mathbf{V}_2$ *up to permuting rows, and* $\mathbf{S}(\mathbf{A}_1, \mathbf{b}_1, \mathbf{V}_1) = \mathbf{S}(\mathbf{A}_2, \mathbf{b}_2, \mathbf{V}_2)$ *up to permuting rows and columns and scaling each row by some positive factor.*

Definition 1.1.21. A matrix $\mathbf{S}_2$ is called *slack-equivalent* to a matrix $\mathbf{S}_1$ if $\mathbf{S}_2$ can be obtained from $\mathbf{S}_1$ by any combination of the following operations (in any order):

- Adding or removing convexly dependent rows or columns,
- Adding or removing zero rows or columns,
- Multiplying each row and column by (possibly different) positive scalar values, and
- Applying a permutation of rows and columns.

Proposition 1.1.22. *Let* P *be a polytope. Any two slack matrices of* P *are slack-equivalent to each other.*

Proof. This is easy to prove by noting that any slack matrix of a polytope can be brought to an irredundant form by applying previously described operations that preserve slack-equivalence, and by Proposition 1.1.20 any two irredundant slack matrices of a polytope are slack-equivalent. □

1.2 THE ROLE OF EMBEDDING

Let P be a d-polytope. Whenever we consider a particular description of P, we implicitly impose a specific embedding of P into the Euclidean space $\mathbb{R}^n$ for some $n \geqslant d$. This is the ambient space where P "lives" and in the terminology that we have so far, we say that P is a d-polytope in $\mathbb{R}^n$.

Example 1.2.1. Consider the three dimensional hypercube defined by the inequalities $0 \leqslant x_i \leqslant 1,\ i \in [3]$. The facet $x_1 = 0, 0 \leqslant x_i \leqslant 1,\ i \in \{2,3\}$ is a 2-polytope in $\mathbb{R}^3$. This facet is in fact a square.

Example 1.2.2. Imagine the octagon in Example 1.1.3 as a section of a three dimensional object. The following figure illustrates this.

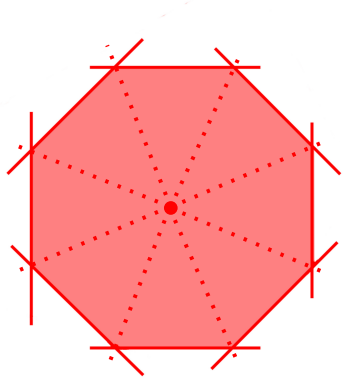


Figure 3: An octagon as a slice of 3-dimensional cone (viewed from top).

To be able to say that two polytopes are the same despite seemingly very different embedding and description – such as in the previous example – we consider various transformations that do not alter the slack matrices of a polytope too much.

1.2.1 *Simple lift*

Definition 1.2.3. Let P be a d-polytope in $\mathbb{R}^n$. A *simple lift* of P into $\mathbb{R}^{n+1}$ is obtained by embedding P into the hyperplane $\{x_{n+1} = 1\}$.

Proposition 1.2.4. *Let P' be obtained by a simple lift of P. Let $\mathbf{S}$ be an irredundant slack matrix of P and $\mathbf{S}'$ be an irredundant slack matrix of P'. Then, $\mathbf{S}'$ and $\mathbf{S}$ are slack equivalent.*

Proof. This follows from the fact that $\mathbf{S}'$ can be obtained from $\mathbf{S}$ by appending two zero rows, scaling each row by some positive factor, and applying some permutation of rows and columns. □

1.2.2 *Variable elimination*

Let P be a d-polytope in $\mathbb{R}^n$. If $d < n$ then any irredundant $\mathcal{H}$-description of P contains $\text{size}(P)$ inequalities and $(n-d)$ equations (cf. Proposition 1.1.16). One can use any of these equations to eliminate one variable resulting in a polytope P' that is an embedding of P in $\mathbb{R}^{n-1}$. This operation can be used to undo a simple lift.

Proposition 1.2.5. *Let P' be obtained from P by reducing the dimension, and let $\mathbf{S}'$ and $\mathbf{S}$ be irredundant slack matrices of P' and P respectively. Then, $\mathbf{S}'$ and $\mathbf{S}$ are slack equivalent.*

Proof. $\mathbf{S}'$ can be obtained from $\mathbf{S}$ by dropping two zero rows, scaling each row by a positive factor, and applying some permutation of rows and columns. □

1.2.3 *Non-degenerate affine transforms*

Definition 1.2.6. Let P be a d-polytope in $\mathbb{R}^n$. A *non-degenerate affine transform* of P is the set $\{\mathbf{T}\mathbf{x} + \mathbf{c} \mid \mathbf{x} \in P\}$ for some invertible matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$ and some vector $\mathbf{c} \in \mathbb{R}^n$.

Exercise 1.2.7. Prove that the image of any non-degenerate affine transform of a polytope is again a polytope with the same number of vertices and facets.

Proposition 1.2.8. *Let P' be obtained from P by a non-degenerate affine transform, and let $\mathbf{S}'$ and $\mathbf{S}$ be slack matrices of P' and P respectively. Then, $\mathbf{S}$ and $\mathbf{S}'$ are slack equivalent.*

Proof. This follows from the fact that $\mathbf{S}$ and $\mathbf{S}'$ are the same up to scaling each row by some positive factor, and permuting rows and columns. □

1.2.4 *Projective scaling*

Definition 1.2.9. Let P be a d-polytope in $\mathbb{R}^n$ with $d < n$. Furthermore, suppose that the origin is not contained in the affine hull $\text{aff}(P)$. A *projective scaling* of P defines a new polytope $P' = \text{conv}(\mathbf{V}')$ where $\mathbf{V}'$ contains each vertex of P scaled by some positive factor such that $\dim(\text{aff}(P')) = \dim(\text{aff}(P))$.

We call this operation a projective scaling because a simple lift of a full-dimensional polytope P followed by a projective scaling creates a simple lift of a polytope projectively isomorphic to P (cf. Subsection 3.2.1).

Proposition 1.2.10. *Let P be a d-polytope in $\mathbb{R}^n$ such that $\text{aff}(P)$ does not contain the origin, and let P' be obtained by a projective scaling of P. If $\mathbf{S}$ and $\mathbf{S}'$ are irredundant slack matrices of P and P' respectively, then $\mathbf{S}'$ is slack-equivalent to $\mathbf{S}$.*

Proof. This follows from the fact that a projective scaling is obtained by a sequence of simple lift, non-degenerate affine transform, and variable elimination. □

1.2.5 *The canonical slack matrix*

When we start discussing extended formulations in Chapter 3 we will see that we are only concerned with slack matrices of a polytope. Propositions 1.2.4, 1.2.5, 1.2.8, and 1.2.10 make it clear that the irredundant slack matrices of a polytope are all the same up to a few zero rows, scaling of rows and columns, and permutation of rows and columns regardless of the choice of the ambient space and the coordinate axes. In fact, the following is true.

Proposition 1.2.11. *Let* P *be a full-dimensional polytope with any irredundant slack matrix* $\mathbf{S}(P)$*. Let* P' *be the polytope obtained by applying any combination of the transformations described in subsections 1.2.1, 1.2.2, 1.2.3, and 1.2.4 and let* $\mathbf{S}'$ *be any slack matrix of* P'*. Then,* $\mathbf{S}'$ *is slack-equivalent to* $\mathbf{S}(P)$*.*

Proof. Follows from Propositions 1.2.4, 1.2.5, 1.2.8, and 1.2.10. □

For our purposes, there will be no distinction between two polytopes that can be obtained from each other via any of the operations described in subsections 1.2.1–1.2.4. That is, a three-dimensional hypercube – for us – remains the same polytope whether embedded in dimension three or thirty; regardless of the position of the origin and the orientation of the coordinate axes; and irrespective of any affine deformations.

We can associate a unique (up to positive scaling and permutation of rows) irredundant slack matrix $\mathbf{S}(P)$ with every polytope P by reducing the dimension of P until we get a full-dimensional polytope P'. Then any two irredundant slack matrices of P' differ only up to reordering and positive scaling (cf. Proposition 1.1.20) and so any irredundant slack matrix of P' is then defined to be $\mathbf{S}(P)$ and is simply referred to as *the* slack matrix of P.

1.3 SOME COMMON OPERATIONS

1.3.1 *Polar duality*

Definition 1.3.1. Let $C \subseteq \mathbb{R}^n$ be a convex set. The polar of C – denoted by C^Δ – is defined as

$$C^\Delta := \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{y}^\top \mathbf{x} \leqslant 1 \text{ for all } \mathbf{y} \in C \right\}.$$

Let P be a full-dimensional polytope containing origin in its interior. Without loss of generality we can assume that the irredundant descriptions of P are given by $P(\mathbf{V})$ and $P(\mathbf{A}, \mathbf{1})$ for some matrices $\mathbf{V}$ and $\mathbf{A}$. In this case, the polar is also a polytope and the irredundant description of P^Δ is closely related to that of P.

Proposition 1.3.2. $P^\Delta = P(\mathbf{A}^\top) = P(\mathbf{V}^\top, \mathbf{1})$.

1.3.2 *Intersection and Union*

Definition 1.3.3. The *intersection* of two polytopes P_1 and P_2 is the set of common points and is denoted by $P_1 \cap P_2$. That is,

$$P_1 \cap P_2 := \{\mathbf{x} \mid \mathbf{x} \in P_1 \wedge \mathbf{x} \in P_2\}.$$

Proposition 1.3.4. *$P_1 \cap P_2$ is a polytope. Furthermore, if $P_1 = P(\mathbf{A}_1, \mathbf{b}_1)$ and $P_2 = P(\mathbf{A}_2, \mathbf{b}_2)$ then,*

$$P_1 \cap P_2 = P\left(\begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}, \begin{pmatrix}\mathbf{b}_1\\ \mathbf{b}_2\end{pmatrix}\right).$$

A simple consequence of this is the following.

Proposition 1.3.5. $\text{size}(P_1 \cap P_2) \leqslant \text{size}(P_1) + \text{size}(P_2)$.

Similarly one may define the union of two polytopes but then the result is not necessarily convex and one may need to take the convex hull of the resulting set to obtain a polytope. We denote this operation by $\uplus$. That is,

$$P_1 \uplus P_2 := \text{conv}\left(\{\mathbf{x} \mid \mathbf{x} \in P_1 \vee \mathbf{x} \in P_2\}\right).$$

For full-dimensional polytopes containing origin in the interior this operation is the polar dual of intersection.

Proposition 1.3.6. *Let P_1 and P_2 be full-dimensional polytopes containing origin in the interior. Then, $(P_1 \cap P_2)^\Delta = P_1^\Delta \uplus P_2^\Delta$ and $(P_1 \uplus P_2)^\Delta = P_1^\Delta \cap P_2^\Delta$.*

1.3.3 *Join and product*

Definition 1.3.7. The *join* of polytopes P_1 and P_2 – denoted by $P_1 * P_2$ – is obtained by embedding them in $\mathbb{R}^d$ for some d such that the affine subspaces $\text{aff}(P_1)$ and $\text{aff}(P_2)$ are skew, and then taking the convex hull of the union.

Any particular choice of the skew subspaces is not very important in this definition since the resulting polytopes are affinely isomorphic (cf. [65]). For polytopes $P(\mathbf{V}_1)$ and $P(\mathbf{V}_2)$ we will take the following canonical embedding to be our definition of the join.

$$P(\mathbf{V}_1) * P(\mathbf{V}_2) := P\left(\begin{bmatrix}\mathbf{V}_1 & \mathbb{O}\\ \mathbb{O} & \mathbf{V}_2\\ -\mathbf{1}^\top & \mathbf{1}^\top\end{bmatrix}\right).$$

Proposition 1.3.8. *Let $P_1 = P(\mathbf{A}_1, \mathbf{b}_1)$ and $P_2 = P(\mathbf{A}_2, \mathbf{b}_2)$ be two polytopes. Then, $\dim(P_1 * P_2) = \dim(P_1) + \dim(P_2) + 1$. Furthermore,*

$$P_1 * P_2 = P\left(\begin{bmatrix}2\mathbf{A}_1 & \mathbb{O} & \mathbf{b}_1\\ \mathbb{O} & 2\mathbf{A}_2 & -\mathbf{b}_2\end{bmatrix}, \begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\end{bmatrix}\right).$$

Once we have the description of the facets of the join of two polytopes, it is a matter of simple substitution of values to relate the canonical slack matrix of the join $\mathbf{S}(P_1 * P_2)$ with the canonical slack matrices of the component polytopes $\mathbf{S}(P_1)$ and $\mathbf{S}(P_2)$.

Proposition 1.3.9. $\mathbf{S}(P_1 * P_2) = \begin{bmatrix} 2\mathbf{S}(P_1) & \mathbf{0} \\ \mathbf{0} & 2\mathbf{S}(P_2) \end{bmatrix}$.

Definition 1.3.10. Let P_1 and P_2 be two polytopes. The *product* – denoted by $P_1 \times P_2$ – is defined as

$$P_1 \times P_2 := \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \middle| x \in P_1, y \in P_2 \right\}.$$

Proposition 1.3.11. *Let* $P_1 = P(\mathbf{A}_1, \mathbf{b}_1)$ *and* $P_2 = P(\mathbf{A}_2, \mathbf{b}_2)$ *be two polytopes. Then,* $\dim(P_1 \times P_2) = \dim(P_1) + \dim(P_2)$. *Furthermore,*

$$P_1 \times P_2 = P\left(\begin{bmatrix} \mathbf{A}_1 & \mathbb{O} \\ \mathbb{O} & \mathbf{A}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \right).$$

1.3.4 *Glued-product*

Definition 1.3.12. Let P_1, P_2 be polytopes with $P_1 \subseteq \mathbb{R}^{n_1+d}$ and $P_2 \subseteq \mathbb{R}^{n_2+d}$. The *glued product* of P_1 and P_2 *over the last* d *coordinates* – denoted by $P_1 \otimes_d P_2$ – is defined as

$$P_1 \otimes_d P_2 := \operatorname{conv}\left\{ \begin{pmatrix} x \\ z \\ y \end{pmatrix} \in \mathbb{R}^{n_1+d+n_2} \middle| \begin{array}{l} \begin{pmatrix} x \\ z \end{pmatrix} \in \operatorname{vert}(P_1) \\ \\ \begin{pmatrix} y \\ z \end{pmatrix} \in \operatorname{vert}(P_2) \end{array} \right\}.$$

We also call these special coordinates the glued coordinates.

Example 1.3.13. Let $\mathbf{V}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and let $\mathbf{V}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Then

$$P(\mathbf{V}_1) \otimes_1 P(\mathbf{V}_2) = P\left(\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \right).$$

It can be shown that the glued product has a very simple description if the glued coordinates have some nice structure. In particular, we have the following.

Proposition 1.3.14. *Let* $P_1 \subseteq \mathbb{R}^{n_1+d}, P_2 \subseteq \mathbb{R}^{n_2+d}$ *be polytopes with descriptions* $\mathbf{A}_1 x + \mathbf{B}_1 z \leqslant \mathbf{c}_1$ *and* $\mathbf{A}_2 y + \mathbf{B}_2 w \leqslant \mathbf{c}_2$ *respectively. Suppose that for every vertex* $(u^\top, z^\top)^\top$ *of* P_1 *or* P_2, z *is a* $0/1$ *vector with at most one* 1. *Then,*

$$P_1 \otimes_d P_2 := \left\{ \begin{pmatrix} x \\ z \\ y \end{pmatrix} \in \mathbb{R}^{n_1+d+n_2} \middle| \begin{array}{rcl} \mathbf{A}_1 x + \mathbf{B}_1 z & \leqslant & \mathbf{c}_1 \\ \mathbf{A}_2 y + \mathbf{B}_2 z & \leqslant & \mathbf{c}_2 \end{array} \right\}.$$

Proof. See [20] Theorem 1. □

2

COMMUNICATION COMPLEXITY

In this chapter we collect some relevant facts about communication complexity. Our focus will be rather narrow and we refer the reader to the excellent text by Kushilevitz and Nisan [45].

Let X, Y, and Z be arbitrary finite sets with $Z \subseteq \mathbb{R}_+$, and let $f : X \times Y \to Z$ be a function. Suppose that there are two players Alice and Bob who wish to compute $f(x, y)$ for some inputs $x \in X$ and $y \in Y$. The players have unlimited computational power. However, Alice knows only x and Bob knows only y. They must therefore exchange information to be able to compute $f(x, y)$.

They could tell each other the inputs that they are holding and thus compute the value $f(x, y)$ but this may not be needed for every function.

Example 2.0.1. Let $f(x, y) = (x + y) \mod 2$. It suffices for Alice and Bob to send one bit each to the other party indicating whether their input is an odd number or not.

Given an ordering $x_1, \ldots, x_m$ of the elements of X, and $y_1, \ldots, y_n$ of the elements of Y, we can visualize the function $f : X \times Y \to Z$ as a $m \times n$ nonnegative matrix $\mathbf{F} = \mathbf{F}(f)$ such that $\mathbf{F}_{ij} = f(x_i, y_j)$ for all $(i, j) \in [m] \times [n]$. The matrix $\mathbf{F}$ is called the *communication matrix* of f. As is natural, we will not always make a distinction between a function and its communication matrix. In fact, for the remainder of the chapter we will use the same notation for a function as for a matrix. For example, if $\mathbf{F}$ denotes a function then both $\mathbf{F}(x, y)$ and $\mathbf{F}_{xy}$ represent the value of the function on input (x, y). Note that this is the same value as the entry in the communication matrix at the row corresponding to x and column corresponding to y.

What features of the communication matrix are relevant for Alice and Bob if they wish to minimize the number of bits that they have to exchange? Surely, we must make precise what is meant when we say that Alice and Bob wish to "compute a function $\mathbf{F}$". We will first discuss a property of matrices – called the nonnegative rank – which will play a crucial role in our discussions and then attempt to take a view of the communication between Alice and Bob in such a way that various notions of computing a function can be handled without requiring much modification.

2.1 NONNEGATIVE RANK

Definition 2.1.1. A *rank-r nonnegative factorization* of a matrix $\mathbf{S}$ is an expression of $\mathbf{S}$ as a product $\mathbf{S} = \mathbf{AB}$ where $\mathbf{A}$ and $\mathbf{B}$ are nonnegative matrices with $\text{numcols}(\mathbf{A}) = \text{numrows}(\mathbf{B}) = r$. The *nonnegative rank* of $\mathbf{S}$, denoted by $\text{rank}_+(\mathbf{S})$, is the minimum nonnegative integer r such that $\mathbf{S}$ admits a rank-r nonnegative factorization.

The nonnegative rank of a matrix S is finite if and only if S is a nonnegative matrix. This is of little consequence for us since we are, in fact, only interested in nonnegative matrices.

Proposition 2.1.2. *The nonnegative rank of a matrix S is the minimum nonnegative integer r such that S is the sum of r nonnegative rank-1 matrices.*

Proof. If $S = AB$, then $S = \sum_{i=1}^{\text{numcols}(A)} A^i B_i$. □

The following is an easy observation which turns out to be very useful when proving lower bounds on the nonnegative matrices.

Proposition 2.1.3. *Let S' be a submatrix of S. Then,* $\text{rank}_+(S) \geqslant \text{rank}_+(S')$.

2.1.1 *Modifying matrices: effect on nonnegative rank*

Now we will see some simple properties of nonnegative rank that will be specially useful for us. Most of the matrices whose nonnegative rank we would like to bound from below will be slack matrices of polytopes. As we will see next, the choice of a canonical slack matrix as done in subsection 1.2.5 was not an ad-hoc choice.

Proposition 2.1.4. *Let F, G be $m \times n$ matrices, then*

1. $\text{rank}_+(F + G) \leqslant \text{rank}_+(F) + \text{rank}_+(G)$
2. $\text{rank}_+(F \circ G) \leqslant \text{rank}_+(F) \cdot \text{rank}_+(G)$

One immediate consequence of this is that the nonnegative rank of a matrix remains unchanged if each row and column of a matrix is scaled independently by a positive factor. One can show something stronger.

Proposition 2.1.5. *Let S', S be matrices such that S' is slack-equivalent to S. Then* $\text{rank}_+(S') = \text{rank}_+(S)$.

Proof. Suppose S' is obtained from S by appending a zero column. Then $\text{rank}_+(S') = \text{rank}_+(S)$ since $[AB\ 0] = A[B\ 0]$. Similarly for appending a zero row.

Suppose S' is obtained from S by appending a convexly dependent column. Then $\text{rank}_+(S') = \text{rank}_+(S)$ since $[AB\ \sum_{i=1}^{\text{numcols}(S)} \lambda_i (AB)^i] = A[B\ \sum_{i=1}^{\text{numcols}(B)} \lambda_i (B)^i]$. Similarly for appending a convexly dependent row.

Finally, suppose S' is obtained by scaling the (i, j)-th entry by $\alpha_i \beta_j$. Let F, G be defined by $F_{ij} = \alpha_i \beta_j$ and $G_{ij} = 1/(\alpha_i \beta_j)$. Clearly, $\text{rank}_+(F) = \text{rank}_+(G) = 1$. Moreover, $S' = S \circ F$ and $S = S' \circ G$. Therefore, by Proposition 2.1.4 we have that $\text{rank}_+(S') \leqslant \text{rank}_+(S)$ and $\text{rank}_+(S) \leqslant \text{rank}_+(S')$. □

Due to the fact that any slack matrix of a polytope P is slack-equivalent to its canonical slack matrix $S(P)$ (Prop. 1.2.11), we get that all slack matrices of a polytope have the same nonnegative rank.

Proposition 2.1.6. *Let* P *be a polytope and* $\mathbf{S}$ *be any slack matrix of* P*. Then,* $\operatorname{rank}_+(\mathbf{S}) = \operatorname{rank}_+(\mathbf{S}(P))$.

At last we can convince ourselves that we do not need to fret over a particular choice of description of a polytope if we are only interested in the nonnegative rank. The nonnegative rank of any slack matrix of a polytope – to a large extent – depends only on the inner geometry and not a particular perspective.

We now describe a combinatorial argument that can sometimes be used to give lower bounds on the nonnegative rank of some matrices. We illustrate the argument by applying it to the slack matrices of joins of polytopes (cf. Subsection 1.3.3). Then in the next subsection we present a related technique that is often used for lower bounding the nonnegative rank of a matrix.

Proposition 2.1.7. *Let* P_1 *and* P_2 *be polytopes with the canonical slack matrices* $\mathbf{S}(P_1)$ *and* $\mathbf{S}(P_2)$*. Let* $\mathbf{S}(P_1 * P_2)$ *be the canonical slack matrix of the join* $P_1 * P_2$*. Then,* $\operatorname{rank}_+(\mathbf{S}(P_1 * P_2)) = \operatorname{rank}_+(\mathbf{S}(P_1)) + \operatorname{rank}_+(\mathbf{S}(P_2))$.

Proof. Due to Proposition 1.3.9 we known that

$$\mathbf{S}(P_1 * P_2) = \begin{bmatrix} 2\mathbf{S}(P_1) & 0 \\ 0 & 2\mathbf{S}(P_2) \end{bmatrix}.$$

Let $\operatorname{numrows}(\mathbf{S}_1) = m_1$, $\operatorname{numrows}(\mathbf{S}_2) = m_2$, $\operatorname{numcols}(\mathbf{S}_1) = n_1$, and $\operatorname{numcols}(\mathbf{S}_2) = n_2$. Also, let $\mathbf{S}(P_1 * P_2) = \mathbf{AB}$ be a rank-r nonnegative factorization with smallest possible r.

We observe that any column of $\mathbf{A}$ cannot contain nonzero entries among the first m_1 rows as well as the last m_2 rows. To see this, let $1 \leqslant k_1 \leqslant m_1$ and $m_1 + 1 \leqslant k_2 \leqslant m_2$ be any two rows. For any column l of $\mathbf{A}$ if $AA_{k_1 l} \neq 0$ then $\mathbf{B}_{ls} = 0$ for all $n_1 + 1 \leqslant s \leqslant n_2$ and if $AA_{k_2 l} \neq 0$ then $\mathbf{B}_{ls} = 0$ for all $1 \leqslant s \leqslant n_1$. Therefore, having nonzero entries within the first n_1 rows and the last m_2 rows of any column of $\mathbf{A}$ would make $\mathbf{B}$ to contain a zero row.

Therefore every column of $\mathbf{A}$ contains zeroes either for all first m_1 rows or for all last m_2 rows. Rearrange columns of $\mathbf{A}$ so that $AA = \begin{bmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{bmatrix}$. Arrange the rows of $\mathbf{B}$ accordingly to $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_3 & \mathbf{B}_4 \end{bmatrix}$ such that their product remains unchanged. Then, $\mathbf{S}_1 = 2\mathbf{A}_1\mathbf{B}_1$ and $\mathbf{S}_2 = 2\mathbf{A}_2\mathbf{B}_4$. Since $\operatorname{numcols}(\mathbf{A}_1) \geqslant \operatorname{rank}_+(\mathbf{S}_1)$ and $\operatorname{numcols}(\mathbf{A}_2) \geqslant \operatorname{rank}_+(\mathbf{S}_2)$ we have that

$$r = \operatorname{numcols}(\mathbf{A}) \geqslant \operatorname{rank}_+(\mathbf{S}(P_1)) + \operatorname{rank}_+(\mathbf{S}(P_2)).$$

Also, $\operatorname{rank}_+(\mathbf{S}(P_1 * P_2)) \leqslant \operatorname{rank}_+(\mathbf{S}(P_1)) + \operatorname{rank}_+(\mathbf{S}(P_2))$. Therefore, equality follows. □

2.1.2 *Rectangle covering bound*

Let $\mathbf{S}$ be an $m \times n$ matrix all whose entries are either zero or one. Such a matrix is often called a 0/1-matrix.

Definition 2.1.8. A combinatorial rectangle (or simply a rectangle) R is a subset of $[m] \times [n]$ such that $R = A \times B$ for some $A \subseteq [m]$ and $B \subseteq [n]$.

Definition 2.1.9. A rectangle R is called a 1-rectangle if for all $(x, y) \in R$ we have that $S_{xy} = 1$. A 0-rectangle is defined similarly. Finally, R is called monochromatic if it is either a 1-rectangle or a 0-rectangle.

Definition 2.1.10. A set of monochromatic rectangles $\mathcal{R}$ is said to cover S if for every $(x, y) \in [m] \times [n]$ there exists a rectangle $R \in \mathcal{R}$ such that $(x, y) \in R$. The rectangle covering number of S, denoted by $\mathrm{rc}(S)$, is the size of smallest $\mathcal{R}$ that covers S.

Let $\mathrm{suppmat}(S)$ be the binary support matrix of S. That is,

$$\mathrm{suppmat}(S)_{ab} = \begin{cases} 1 & \text{if } S_{ab} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 2.1.11. $\mathrm{rank}_+(S) \geqslant \mathrm{rc}(\mathrm{suppmat}(S))$.

Proof. See Theorem 4 in [27] (Appendix A). □

This provides a useful way of proving lower bounds on the nonnegative rank of matrices by means of combinatorial arguments. We illustrate this by an example that will play an important role later on.

2.1.3 *Unique Disjointness*

Consider the following $2^n \times 2^n$ matrix $U = U(n)$ with rows and columns indexed by n-bit strings a and b, and real nonnegative entries:

$$U_{ab} := (a^\top b - 1)^2.$$

An entry U_{ab} of this matrix is zero if and only if the strings a and b are different except at some unique index. A simple combinatorial argument shows the following.

Proposition 2.1.12. $\mathrm{rc}\left(\mathrm{suppmat}(U)\right) \geqslant \left(\frac{3}{2}\right)^n$.

Proof. See [41], Theorem 1. □

Combining Proposition 2.1.12 with Proposition 2.1.11 we get the following theorem.

Proposition 2.1.13. $\mathrm{rank}_+(U) \geqslant \left(\frac{3}{2}\right)^n$.

2.2 COMMUNICATION PROTOCOLS

However Alice and Bob may choose to define what it means to compute a function together, their communication is carried out as a protocol that is agreed upon beforehand by them, on the sole basis of the function f. At the beginning of an *execution* of the protocol Alice and Bob receive their inputs x and y respectively. At each step of the protocol, one of the players has the token. Whoever has the token sends

a bit to the other player. At any point, one of the players outputs a value and the execution stops. The correctness of the protocol is determined by a previously specified relation between the output value and the value of $f(x, y)$.

A protocol can be viewed as a rooted binary tree where each node is marked either Alice or Bob. The leaves have vectors associated with them. An execution of the protocol on a particular input is a path in the tree starting at the root. At a node owned by Alice, following the path to the left subtree corresponds to Alice sending a zero to Bob and taking the right subtree corresponds to Alice sending a one to Bob; and similarly for nodes owned by Bob.

Let X and Y be finite sets and let $f : X \times Y \to \mathbb{R}_+$ be a function that Alice and Bob wish to compute [1].

Definition 2.2.1. A *communication protocol (with private random bits and* nonnegative *outputs)* is a rooted binary tree with some extra information attached to its nodes. Each node of the tree has a *type*, which is either X or Y. To each node v of type X are attached two function $p_{0v}, p_{1v} : X \to [0, 1]$; to each node v of type Y are attached two functions $q_{0v}, q_{1v} : Y \to [0, 1]$; and to each leaf v is attached a nonnegative vector Λ_v that is a column vector of size $|X|$ for leaves of type X and a row vector of size $|Y|$ for leaves of type Y. The functions p_{iv} and q_{jv} define *transition probabilities*, and we assume that $p_{0v}(x) + p_{1v}(x) \leqslant 1$ and $q_{0v}(y) + q_{1v}(y) \leqslant 1$.

Figure 4 shows an example of a protocol.

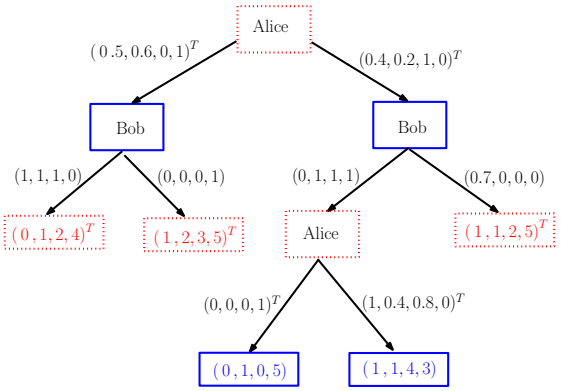


Figure 4: Example of a protocol viewed as a tree

Definition 2.2.2. An *execution* of the protocol on input $(x, y) \in X \times Y$ is a random path that starts at the root and descends to the left child of an internal node v with probability $p_{0v}(x)$ if v is of type X and $q_{0v}(y)$ if v is of type Y, and to the right child of v with probability $p_{1v}(x)$ if v is of type X and $q_{1v}(y)$ if v is of type Y. With probability $1 - p_{0v}(x) - p_{1v}(x)$ and $1 - q_{0v}(y) - q_{1v}(y)$ respectively, the execution stops at v.

1 For now, let us not worry about the precise meaning of computing a function.

Definition 2.2.3. The *value* of an execution on the input pair x, y – denoted by $\mathrm{val}(x, y)$ – is defined as follows. For an execution stopping at leaf v with vector $\mathbf{\Lambda}_v$, $\mathrm{val}(x, y)$ is defined as the entry of $\mathbf{\Lambda}_v$ that corresponds to input $x \in X$ if v is of type X, and $y \in Y$ if v is of type Y. For an execution stopping at an internal node, $\mathrm{val}(x, y)$ is defined to be 0.

Definition 2.2.4. The *complexity* of a protocol Π is measured by one of the two parameters. The *depth* of the protocol – denoted by $\mathrm{depth}(\Pi)$ – is the depth of the corresponding protocol tree, and the *size* of the protocol – denoted by $\mathrm{size}(\Pi)$ – is the number of leaves of the corresponding protocol tree.

When presenting a protocol, we shall often say that one of the two players sends an integer k rather than a binary value. This should be interpreted as the player sending the binary encoding of k or, as a (sub)tree of depth $\lceil \lg k \rceil$, or of size k. Finally, our definitions are such that the depth of a protocol equals the number of bits exchanged by Alice and Bob in the worst case.

Exercise 2.2.5. What is the relation between the depth and the size of a protocol, if the protocol tree is balanced?

With every node v of a communication protocol we can associate a nonnegative matrix $\mathbf{P}_v$ that specifies the probability of visiting that node in an execution. Let $v_1, \ldots, v_k$ denote the nodes of type X on the unique path from the root to the parent of v, and let $w_1, \ldots, w_\ell$ denote the nodes of type Y on this path. Then we have

$$\mathbf{P}_v(x, y) = \prod_{i=1}^{k} \mathbf{p}_{\alpha_i v_i}(x) \cdot \prod_{j=1}^{\ell} \mathbf{q}_{\beta_j w_j}(y),$$

where α_i is either 0 or 1 depending on if the path goes the left or right subtree at v_i, and similarly for β_j. Observe that $\mathbf{P}_v$ is a matrix of nonnegative rank one for each node v of the protocol tree as required.

2.2.1 *The expected value of a protocol*

For each input pair (x, y) given to Alice and Bob, $\mathrm{val}(x, y)$ is a random variable whose distribution depends on the transition probabilities at the nodes of the protocol tree. One may therefore talk about the expected value $\mathbb{E}[\mathrm{val}(x, y)]$.

Let L_X and L_Y be the set of all leaves of the protocol that are of type X and Y respectively and let $\mathbf{\Lambda}_v$ denote the (column or row) vector of values at a leaf $v \in L_X \cup L_Y$. We have

$$\mathbb{E}[\mathrm{val}(x, y)] \quad = \quad \sum_{v \in L_X} \mathbf{\Lambda}_v(x) \mathbf{P}_v(x, y) + \sum_{w \in L_Y} \mathbf{P}_w(x, y) \mathbf{\Lambda}_w(y).$$

Regardless of what Alice and Bob may think that they are computing using a protocol Π, we may associate a function $\mathbf{E}_\Pi : X \times Y \to \mathbb{R}_+$ with Π defined by $\mathbf{E}_\Pi(x, y) = \mathbb{E}[[\mathrm{val}(x, y)]]$. Therefore,

$$\mathbf{E}_\Pi = \sum_{v \in L_X} (\mathbf{\Lambda_v} \circ \mathbf{a}_v) \mathbf{b}_v + \sum_{v \in L_Y} \mathbf{a}_w (\mathbf{b}_w \circ \mathbf{\Lambda}_w).$$

Proposition 2.2.6. *Let S be a nonnegative matrix. Then,*

$$\operatorname{rank}_+(S) = \min\{\operatorname{size}(T) \mid T_T = S\}.$$

Proof. Implicit in the proof of Theorem 2 in [24]. □

2.3 COMPLEXITY OF COMPUTING A FUNCTION

To relate a communication protocol to a function f, it remains to establish the relation between $\operatorname{val}(x, y)$ and $f(x, y)$. Each of the following models specify in different way what it means to compute a function.

Definition 2.3.1. The *communication complexity* of a function f is defined to be the minimum depth among all communication protocols that compute f.

A natural reason to define the communication complexity in this way is to imagine Alice and Bob communicating with each other with the goal of selecting a particular leaf in the protocol tree, so that they can output a value that "computes" the function f without any further communication. To reach any leaf it suffices to send one bit at each node on the unique path from the root to the particular leaf indicating if the next node is the left or the right child.

2.3.1 *Classical deterministic model*

In classical deterministic models of communication complexity, the transition probabilities at each node of the protocol tree can take values either zero or one. Thus, $\operatorname{val}(x, y)$ can take only one possible value for each pair (x, y). A protocol is said to compute a function f if and only if $\operatorname{val}(x, y) = f(x, y)$ for all pairs (x, y) of inputs that Alice and Bob may receive.

2.3.2 *Classical randomized model*

In classical randomized models of communication complexity, the transition probabilities at each node of the protocol tree can take values between zero and one. Thus, for each fixed input $(x, y) \in X \times Y$, $\operatorname{val}(x, y)$ is a random variable and can take one of multiple possible values. A protocol is said to compute a function f if and only if $\operatorname{val}(x, y)$ is close to $f(x, y)$ for all pairs x, y of inputs that Alice and Bob may receive. One may further specify whether this happens with high probability for an execution, or for all executions.

We will not clarify the ambiguities in the previous paragraph since this model is not relevant to us. The interested reader may read Chapter 3 of [45]. We leave the discussion with the following food for thought.

Exercise 2.3.2. How does the variance of the random variable $\operatorname{val}(x, y)$ influence the communication complexity in classical randomized models?

2.3.3 *EF model*

As discussed in Section 2.2.1, for each fixed input $(x, y) \in X \times Y$, the value of an execution on input (x, y) is a random variable and one can define the function $\mathbf{E}_\Pi$ of the expected output of the protocol on input (x, y). In the EF model, we say that the protocol computes a function f if $f = \mathbf{E}_\Pi$. As a shorthand we will refer to a communication protocol with the EF model of computation as an *EF-protocol*. For example saying that f is computed by an EF-protocol Π should be understood as: Π is a communication protocol and $\mathbf{E}_\Pi = f$.

Computing a function only in expectation allows us to assume many things about the "smallest" communication protocol available for any given function.

Proposition 2.3.3. *If f can be computed by an EF-protocol of size r, then f can be computed by an EF-protocol of depth $\lceil \log r \rceil$.*

Proof. See [24], Theorem 2 (Appendix B). □

This allows us to measure the communication complexity either in terms of the depth or the size of the smallest protocol computing f. For our purposes, we will measure the communication complexity in the EF model by the smallest size of any EF-protocol for f.

3

EXTENDED FORMULATIONS

Let $P \subset \mathbb{R}^n$ and $Q \subset \mathbb{R}^{n+r}$ be polytopes.

Definition 3.0.1. Q is called an extended formulation (EF) of P if there exists a linear map $\pi : \mathbb{R}^{n+r} \to \mathbb{R}^n$ such that $P = \pi(Q)$.

The map π in the previous definition *projects* Q to P. With a change of basis one can always assume that this projection map just amounts to dropping r coordinates of Q.

Exercise 3.0.2. Let Q be an EF of P. Show that there exists an EF Q' of P such that $\text{size}(Q') = \text{size}(Q)$ and $P = \pi(Q)$ where the map π is defined by $\pi(z) = x$ if $z^\top = (x^\top, y^\top)$.

When the projection map is not specified, we will assume it to be the canonical orthogonal projection: drop-r-coordinates.

Example 3.0.3. A regular octagon can be seen as a projection of a deformed three-dimensional cube.

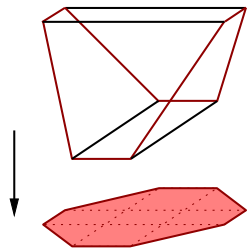


Figure 5: A deformed hypercube projects to a regular octagon.

An extended formulation can also be defined in terms of a certain equivalence in optimization, as follows.

Proposition 3.0.4. *Q is an EF of P if and only if there exists $t \in \mathbb{R}^n$ and an $(n+r) \times n$ matrix R such that*

$$\max_{x \in P} c^\top x = \max_{z \in Q} (Rc)^\top z + c^\top t$$

for all $c \in \mathbb{R}^n$.

Proof. Suppose Q is an EF of P. Then, there exists a linear map $\pi : \mathbb{R}^{n+r} \to \mathbb{R}^n$ such that $\pi(Q) = P$. Let π be defined as $\pi((x^\top, y^\top)^\top) = R^\top (x^\top, y^\top)^\top + t$ where R is an $(n+r) \times n$ matrix and $t \in \mathbb{R}^n$ is a vector. R and t satisfy the requirement of the lemma.

For the other direction, notice that $\alpha^\top x \leqslant \beta$ is valid for P if and only if $(x^\top, y^\top) R\alpha \leqslant \beta - \alpha^\top t$ is valid for Q. Therefore, we can define the linear map $\pi(z) = R^\top (x^\top, y^\top)^\top + t$, if $z^\top = (x^\top, y^\top)$ with $x \in \mathbb{R}^n, y \in \mathbb{R}^r$. For this map we have that $\pi(Q) = P$. □

3.1 EXTENSION COMPLEXITY

Definition 3.1.1. The *extension complexity* of a polytope P – denoted by $\text{xc}(P)$ – is defined to be the size of an extended formulation requiring the fewest number of inequalities. That is,

$$\text{xc}(P) := \min_{Q \text{ is EF of } P} \text{size}(Q)$$

Most often we are interested in the extension complexity of a polytope in terms of the ambient dimension. The ambient dimension, in turn, is often polynomially related to the dimension of the polytope.

Example 3.1.2. The convex hull of the characteristic vectors of all perfect matchings of the complete graph K_n lives in the ambient dimension $\binom{n}{2}$. This polytope, however, is not full-dimension and has dimension $\binom{n}{2} - n$. We may measure the extension complexity of this polytope either in terms of the ambient dimension $\binom{n}{2}$ or the actual dimension $\binom{n}{2} - n$ or n. In all these cases, the expression we will get are essentially equivalent to each other in terms of whether the extension complexity is polynomially bounded or not.

Sometimes we may be interested in the extension complexity of a polytope in terms of other things as well.

Example 3.1.3. Consider the polytope $\text{STAB}_k(G)$ defined as the convex hull of the characteristic vectors of all independent sets of G that are of size k. Are there constant c and function f such that for all graphs G on n vertices, $\text{xc}(\text{STAB}_k(G)) \leqslant f(k) \cdot n^c$?

To talk about such questions succinctly, we may think of k as a parameter of the polytope $\text{STAB}_k(G)$ and talk about its parametrized extension complexity.

Definition 3.1.4. Let $P \subset \mathbb{R}^n$ be a polytope and κ be a fixed number (somehow associated with P). The *parametrized extension complexity* of P is the extension complexity of P expressed as a function of κ and n. The number κ is called a *parameter*.

The above definition does not make a lot of sense since for any fixed polytope P the numbers n, κ, and $\text{xc}(P)$ are fixed numbers. However this definition does make sense for a set of polytopes. This is very convenient since usually we are not interested in the extension complexity of one fixed polytope but a set of related polytopes.

3.1.1 *Extension complexity of multiple polytopes*

Definition 3.1.5. Let $\mathcal{P}$ be a set of polytopes. Let $\mathbf{n}, \kappa : \mathcal{P} \to \mathbb{N}$ be two functions. We say that the extension complexity of $\mathcal{P}$ – denoted by $\text{xc}(\mathcal{P})$ – is a function $\mathbf{f} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ if for every $P \in \mathcal{P}$ we have that $\text{xc}(P) = \mathbf{f}(\mathbf{n}(P), \kappa(P))$. When not specified, the parameter κ is chosen to be $\kappa(P) = 1$ for all $P \in \mathcal{P}$.

The sets of polytopes that are of interest to us, will usually be implicitly described using objects such as graphs, a set of numbers, etc. In such cases, $\mathbf{n}(P)$ will usually be the "size" of these objects and will be polynomially related to the ambient dimension of P. The parameter $\kappa(P)$ will usually be a parameter related to the underlying object used to define P. Let us illustrate this with an example.

Example 3.1.6. For any graph G, consider $\text{CUT}^{\square}(G)$, the convex hull of the characteristic vectors of the edge cuts of G. A natural choice for $\mathbf{n}(\text{CUT}^{\square}(G))$ is the number of vertices of G. A natural parameter $\kappa(\text{CUT}^{\square}(G)$ can the the treewidth of G.

Remark 3.1.7. When the choice of the functions $\mathbf{n}$ and κ is clear and does not create ambiguities, we may say that the extension complexity of the set $\mathcal{P}$ is $\mathbf{f}(n,k)$ for some function $\mathbf{f}$. When κ is not mentioned, we may say that the extension complexity of the set is $\mathbf{g}(n)$ for some function $\mathbf{g}$.

Example 3.1.8. Suppose $\mathcal{MAGIC}$ is a set of polytopes somehow related to graphs. That is, each polytope in this set is defined using a uniquely associated graph. Saying that "$\text{xc}(\mathcal{MAGIC}) = 2^{\tau}n$ where τ is the treewidth and n is the number of vertices of the underlying graph" should be understood to mean that for every graph G with n vertices and treewidth τ the corresponding polytope in the set has extension complexity $2^{\tau}n$.

Example 3.1.9. Let $\mathcal{P}_{\square}$ be the set of full-dimensional hypercubes. Then $\text{xc}(\mathcal{P}_{\square}) = 2n$ where n is the dimension.

Now we will describe two special kinds of sets of polytopes that will help us deduce the parameters $\mathbf{n}$ and κ from the context.

Definition 3.1.10. A *clan* of polytopes is a set of related polytopes. The relation between polytopes will usually be clear from the description. For example, the convex hull of all satisfying assignments of 3CNF formulae defines a clan.

A *family* of polytopes is a countable ordered set $\{P_1, P_2, \ldots\}$ with $P_n \subseteq \mathbb{R}^n$.

Example 3.1.11. A polytope $\text{EP}(G)$ can be defined for every graph G as the convex hull of all perfect matchings of the graph G. For each natural number m, define Edmonds' polytope $\text{EP}(m)$ as the convex hull of characteristic vectors of the perfect matchings of K_n where $m = \binom{n}{2}$.

The set $\{\text{EP}(G) \mid G \text{ is a graph }\}$ defines the perfect matching clan $\mathbb{EP}$ while $\mathcal{EP} = \{\text{EP}(n) | n \in \mathbb{N}\}$ defines a particular family of this clan.

Proposition 3.1.12. $\text{EP}(m)$ *is empty if there is no* n *with* $m = \binom{n}{2}$ *or if such an* n *exists but is odd.*

Proposition 3.1.13 (Rothvoß)**.** *There exists a constant* $0 < c < 1/2$*, such that for every even* $n \in \mathbb{N}$ *we have that* $\text{xc}(\text{EP}(\binom{n}{2})) \geqslant 2^{cn}$.

Proof. See [55], Theorem 1. □

Proposition 3.1.14. *For every* $\epsilon > 0$ *there exists* $n_0 \in \mathbb{N}$ *such that for all* $n \geqslant n_0$ *we have that* $\mathrm{xc}(\mathrm{EP}(\binom{n}{2})) \leqslant 2^{(1/2+\epsilon)n}$.

Proof. See [24], Proposition 3 (Appendix B). □

For a family of polytopes the choice of the parameter $\mathbf{n}$ will most often be the ambient dimension. That is, if $\mathcal{P}$ is a family of polytopes then $\mathbf{n}(P_n) = n$ for $P_n \in \mathcal{P}$. Since a family contains exactly one polytope $P_n \subset \mathbb{R}^n$, the meaning of $\mathrm{xc}(\mathcal{P})$ is clear and a statement such $\mathrm{xc}(\mathcal{P}) = n^3$ is unambiguous with this convention. Note that the polytopes P_n in a family are not required to be full-dimensional (or even non-empty).

Definition 3.1.15. Extension complexity of a clan $\mathbb{P}$ is also denoted by $\mathrm{xc}(\mathbb{P})$ and is defined to be the extension complexity of the family $\mathcal{P} \in \mathbb{P}$ obtained by picking the polytopes with largest extension complexity for each dimension.

More precisely, given a clan $\mathbb{P}$, let $\mathcal{P}_{\max}$ be a family of polytopes such that if $P_n \subset \mathbb{R}^n$ belongs to $\mathcal{P}_{\max}$ then $\mathrm{xc}(P_n) \geqslant \mathrm{xc}(P'_n)$ for all $P'_n \in \mathbb{P}$ with $P'_n \subseteq \mathbb{R}^n$. Moreover, for every n exactly one $P_n \subset \mathbb{R}^n$ belongs to $\mathcal{P}_{\max}$. The extension complexity of clan $\mathbb{P}$ is defined to be equal to $\mathrm{xc}(\mathcal{P})$.

For different values of n, the corresponding polytopes in a family $\mathcal{P}$ may have extension complexities that are not well described by a simple function. Even if exact bounds are known for each polytope in a family of polytopes, it will simplify our lives if we use asymptotic notation to describe the extension complexity of the family. In fact, for a family (or clan) of polytopes, the asymptotic behavior of their extension complexity is what we generally care about. If it is a polynomial function then – at least in principle – the polytopes can be efficiently represented. If the extension complexity of the family (or clan) grows superpolynomially then at least some of the polytopes require large descriptions.

Let $\mathcal{P} = \{P_1, P_2, \ldots\}$ be a family of polytopes with $P_n \subseteq \mathbb{R}^n$. We will say that $\mathrm{xc}(\mathcal{P}) = \mathcal{O}(\mathbf{f})$ to mean that there exists a constant $c > 0$ and a natural number n_0 such that for every polytope $P_n \in \mathcal{P}$ with $n \geqslant n_0$ we have that $\mathrm{xc}(P_n) \leqslant c\mathbf{f}(n)$.

We will say that $\mathrm{xc}(\mathcal{P}) = \Omega(\mathbf{f})$ to mean that there exists a constant $c > 0$ and such that for every natural number n_0 there exists an $n \geqslant n_0$ such that $\mathrm{xc}(P_n) \geqslant c\mathbf{f}(n)$. Note the slight difference from the usual Ω notation used in asymptotic analysis of algorithms [1]. The intent here is to be able to say that $\mathcal{P}$ contains infinitely many polytopes that have high extension complexity.

Finally, we will say that $\mathrm{xc}(\mathcal{P}) = \Theta(\mathbf{f})$ to mean that $\mathrm{xc}(\mathcal{P}) = \mathcal{O}(\mathbf{f})$ as well as $\mathrm{xc}(\mathcal{P}) = \Omega(\mathbf{f})$.

Example 3.1.16. Proposition 3.1.13 can be translated in our setting to the following.

Proposition 3.1.17. $\mathrm{xc}(\mathcal{EP}) = \Omega(c^{\sqrt{n}})$ *for some* $c > 1$.

1 This usage, however, is common among number theorists

One can extend the above notation to provide more information by being able to use functions described in asysmptotic notation as well. We will not go into the details of this point except to present an example that should clarify the point.

Example 3.1.18. Combining Propositions 3.1.13 and 3.1.14 one could say that $\mathrm{xc}(\mathbb{EIP}) = \mathrm{xc}(\mathcal{EP}) = 2^{\Theta(\sqrt{n})}$.

3.1.2 *Bounding extension complexity: some tools*

Before we mention stronger results connecting extension complexity with nonnegative rank, we would like to list few simple facts that follow from the above definition and basic polyhedral properties.

Proposition 3.1.19. *If* P *is the convex hull of* m *points then* $\mathrm{xc}(P) \leqslant m$.

Proof. If $P = P(\mathbf{V})$, then by definition P is a projection of the polytope

$$\left\{ \begin{pmatrix} x \\ \lambda \end{pmatrix} \middle| \begin{array}{rcl} \mathbf{V}\lambda & = & x \\ \mathbf{1}^\top \lambda & = & 1 \\ \lambda & \geqslant & 0 \end{array} \right\}$$

□

Definition 3.1.20. Let Q be a polytope and h be a hyperplane. $Q \cap h$ defines A *slice* of Q.

Proposition 3.1.21. *If* P *is a slice of* Q*, then* $\mathrm{xc}(P) \leqslant \mathrm{xc}(Q)$.

In particular, noting that a polytope is a trivial slice of itself and every face of a polytope P is also a slice of P we get the following simple but important cases.

Proposition 3.1.22. *If* Q *is an* EF *of* P*, then* $\mathrm{xc}(P) \leqslant \mathrm{xc}(Q)$.

Proposition 3.1.23. *If* P *is a face of* Q*, then* $\mathrm{xc}(P) \leqslant \mathrm{xc}(Q)$.

3.1.3 *Yannakakis' characterization of Extension Complexity*

Proposition 3.1.24. *Let* P *be a polytope and* $\mathbf{S}$ *be any slack matrix of* P*. Then,* $\mathrm{xc}(P) = \mathrm{rank}_+(\mathbf{S})$.

Proof. See [24], Theorem 1 (Appendix B). □

Combining Propositions 2.2.6, 2.3.3 and 3.1.24 we get the following.

Proposition 3.1.25. *Let* P *be a polytope. Then, the following are equivalent.*

1. $\mathrm{xc}(P) \leqslant 2^r$.
2. $\mathrm{rank}_+(\mathbf{S}(P)) \leqslant 2^r$.
3. *There exists an EF protocol* Π *with* $\mathbf{E}_\Pi = \mathbf{S}(P)$ *and* $\mathrm{size}(\Pi) \leqslant 2^r$.
4. *There exists an EF protocol* Π *with* $\mathbf{E}_\Pi = \mathbf{S}(P)$ *and* $\mathrm{depth}(\Pi) \leqslant r$.

Already with the discussion so far, the reader should be able to prove bounds on extension complexities of a number of polytopes: some by simply referring to facts already established in previous chapters.

Example 3.1.26. Let $P_{ij} = \left\{x \in \{0,1\}^{n+1} \mid x_{n+1} = x_i \oplus x_j\right\}$ for $1 \leqslant i < j \leqslant n$. Then,

$$P_{ij} = \square_{n-2} \times P\left(\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}\right)$$

upto relabeling of coordinates. Therefore, $\mathrm{xc}(P_{ij}) \leqslant 2n$ (cf. Proposition 3.2.3).

3.1.4 *Combinatorially isomorphic polytopes with different extension complexity*

Before discussing how robust extension complexity is as a measure of intrinsic complexity of representing a polytope, we would like to remark that combinatorial isomorphism of two polytopes is not enough to ensure same extension complexity. This is seen by considering polygons.

Definition 3.1.27. Two polytopes are said to be *combinatorially isomorphic* if the posets of their faces (including trivial ones) ordered by inclusion are isomorphic.

Exercise 3.1.28. Any two n-gons are combinatorially isomorphic.

Proposition 3.1.29. *Let P_n be a regular n-gon. Then $\mathrm{xc}(P_n) \leqslant 2\log n$.*

Proof. See [26], Theorem 2 (Appendix C). □

Proposition 3.1.30. *For every $n \in \mathbb{N}$ there exists an n-gon with extension complexity $\Omega(\sqrt{n})$.*

Proof. See [26], Theorem 3 (Appendix C). □

3.2 EFFECTS OF COMMON OPERATIONS

The extension complexity of a polytope is a fairly robust measure of the inherent complexity of describing a polytope. It does not depend on the ambient space and the choice of a particular coordinate axes. In fact, the extension complexity remains unchanged if the polytope is distorted by a projective transform. Before describing projective transforms formally, we provide a more geometric picture from the excellent textbook "Lectures on Polytopes" by Ziegler.

3.2.1 *Projective transforms*

Let P be a full-dimensional polytope in $\mathbb{R}^n$. Embed this polytope into an affine hyperplane $H \subseteq \mathbb{R}^{n+1}$ and construct the homogenization

of P: $\mathrm{cone}(\{\mathbf{x} \mid \mathbf{x} \in P\})$. Cut this cone with any hyperplane K that intersects all its extreme rays and identify K with $\mathbb{R}^n$. This defines a projective transform of P.

Definition 3.2.1. Let P be a d-polytope in $\mathbb{R}^n$. A *projective transform* of P is defined by a matrix

$$\begin{bmatrix} \mathbf{B} & \mathbf{c} \\ \mathbf{a}^\top & a_{n+1} \end{bmatrix}$$

and a vector $\mathbf{c}'$ with the following conditions:

1. $\det \begin{pmatrix} \mathbf{B} & \mathbf{c} \\ \mathbf{a}^\top & a_{n+1} \end{pmatrix} \neq 0$
2. $\mathbf{a}^\top \mathbf{x} + a_{n+1} > 0$ for all $\mathbf{x} \in P$.

The polytope P' obtained from P via this projective transformation is defined by

$$P' = \left\{ \frac{\mathbf{B}\mathbf{x} + \mathbf{c}}{\mathbf{a}^\top \mathbf{x} + a_{n+1}} + \mathbf{c}' \,\middle|\, \mathbf{x} \in P \right\}.$$

Proposition 3.2.2. *Let P_1 and P_2 be two polytopes that are isomorphic under projective transformations. Then,*

$$\mathrm{xc}(P_1) = \mathrm{xc}(P_2).$$

Proof. See [33], Proposition 2.9. □

3.2.2 *Join, product, and free-sum*

Proposition 3.2.3. *Let P_1, P_2 be polytopes. Then,*

1. $\mathrm{xc}(P_1 * P_2) = \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$.
2. $\mathrm{xc}(P_1 \times P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$.
3. $\mathrm{xc}(P_1 \oplus P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$.

Proof. The first bound follows immediately from Proposition 2.1.7 and Proposition 3.1.24. The second follows from the fact that $P_1 \times P_2$ is a slice of $P_1 * P_2$ (cf. Proposition 3.1.21), while the third follows from the fact that $P_1 \oplus P_2$ is a projection of $P_1 * P_2$ (cf. Proposition 3.1.22). □

3.2.3 *Glued Product*

Proposition 3.2.4 (Margot)**.** *Let $P_1 \subseteq \mathbb{R}^{n_1+d}$ and $P_2 \subseteq \mathbb{R}^{n_2+d}$ be polytopes such that the last d coordinates of any vertex of either polytope is a zero-one vector with at most one 1. Then $\mathrm{xc}(P_1 \otimes_d P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$.*

Proof. See [20], Theorem 1 (cf. [44], Lemma 1, Appendix G). □

3.2.4 *Union*

Proposition 3.2.5. $\mathrm{xc}(P_1 \uplus P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$.

Proof. Let $P_1 = P(\mathbf{V}_1)$ and $P_2 = P(\mathbf{V}_2)$. Consider the projective transform given by the matrix

$$\begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{O} & \mathbf{I} & \mathbf{0} \\ \mathbf{0}^\top & \mathbf{0}^\top & 1 \end{bmatrix},$$

where $\mathbf{I}$ is the identity matrix of appropriate size, $\mathbf{O}$ is the matrix of all zeroes, and $\mathbf{0}$ is the zero vector.

This transforms the join $P_1 * P_2$ to

$$P\left(\begin{bmatrix} \mathbf{V}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{V}_2 \\ -\mathbf{1}^\top & \mathbf{1}^\top \end{bmatrix}\right) \to P\left(\begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \\ \mathbf{O} & \mathbf{V}_2 \\ -\mathbf{1}^\top & \mathbf{1}^\top \end{bmatrix}\right).$$

The later is easily seen to be an EF of $P_1 \uplus P_2$. Combining Propositions 3.2.3, 3.2.2 and 3.1.22 we get that $\mathrm{xc}(P_1 \uplus P_2) \leqslant \mathrm{xc}(P_1 * P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$. □

3.2.5 *Intersection*

Proposition 3.2.6. $\mathrm{xc}(P_1 \cap P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$.

Proof. Let $P_1 = \{\mathbf{x} \mid \mathbf{A}_1\mathbf{x} \leqslant \mathbf{b}_1\}$ and $P_2 = \{\mathbf{x} \mid \mathbf{A}_2\mathbf{x} \leqslant \mathbf{b}_1\}$. Let $Q_1 = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} \middle| \mathbf{E}_1\mathbf{x} + \mathbf{F}_1\mathbf{z} \leqslant \mathbf{g}_1 \right\}$ and $Q_2 = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} \middle| \mathbf{E}_2\mathbf{x} + \mathbf{F}_2\mathbf{w} \leqslant \mathbf{g}_2 \right\}$ be EFs of P_1 and P_2 respectively. Then,

$$R = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \\ \mathbf{y} \\ \mathbf{w} \end{pmatrix} \middle| \begin{array}{rcl} \mathbf{E}_1\mathbf{x} + \mathbf{F}_1\mathbf{z} & \leqslant & \mathbf{g}_1 \\ \mathbf{E}_2\mathbf{y} + \mathbf{F}_2\mathbf{w} & \leqslant & \mathbf{g}_2 \\ \mathbf{x} & = & \mathbf{y} \end{array} \right\}$$

is an EF of $P_1 \cap P_2$. □

3.3 SOME CANONICAL POLYTOPE FAMILIES

There are obvious clans – such as the one consisting of all polytopes – that have extension complexity unbounded by any function. Considering the clan $\mathbb{POLYGONS}$ of all polygons embedded in all dimensions, by Proposition 3.1.30 we already have that $\mathrm{xc}(\mathbb{POLYGONS})$ is not bounded by any function. One feature of such easily-produced high complexity clans is that describing the members may require unbounded precision. In any case we will be mostly interested in 0/1-polytopes which cannot have arbitrarily high extension complexity.

Proposition 3.3.1. *Let* $\mathbb{ZERO{-}ONE}$ *be the clan of all* $0/1$*-polytopes. Then,* $2^{\frac{n}{2}(1-o(1))} \leqslant xc(\mathbb{ZERO{-}ONE}) \leqslant 2^n$

Proof. The upper bound follows from Proposition 3.1.19 since any $0/1$ polytope in $\mathbb{R}^n$ has at most 2^n vertices. The lower bound follows from the fact that there are $0/1$ polytopes of such extension complexity [54]. □

So we see that the clan of all $0/1$ polytopes has extension complexity $2^{\Theta(n)}$. A family with such complexity can essentially by picked by selecting random polytopes for each dimension. It may be quite intuitive that this will result in a family of large extension complexity but proving it requires some very precise argument controlling the number of bits required to encode any extended formulation. Rothvoss was able to do exactly this and together with an elegant double counting argument was able to show the lower bound.

Such examples may be unsatisfactory because we do not get an explicit family of polytopes that has high extension complexity. A clan with high extension complexity becomes more interesting when we can describe the clan members and a specially hard family in the clan rather succintly. This is what was first done by Fiorini et al. [27] with polytopes related to the maxcut problem and later extended by various authors. We will see some of these clans in Chapter 4. Now we describe three canonical clans of polytopes that will play an important role later: the polytopes of cut vectors of graphs; the polytopes of satisfying assignments of CNF formulae; and the polytopes of non-satisfying assignments of CNF formulae.

3.3.1 *The* $\mathbb{CUT}$ *clan*

An important clan of polytopes that has high extension complexity is the that of Cut polytopes. These polytopes are naturally associated with the familiar NP-hard MAXCUT problem and have a rich history. We direct the reader to the textbook "Geometry of Cuts" by Deza and Laurant [21].

Definition 3.3.2. For a graph G the cut polytope of G – denoted by $CUT^{\square}(G)$ – is defined to be the convex hull of characteristic vectors of all cuts in G. Any polytope P such that $P = CUT^{\square}(G)$ for some graph G is called a cut polytope. The clan $\mathbb{CUT}$ is defined to be the set of all cut polytopes.

Proposition 3.3.3.

$$CUT^{\square}(K_{n+1}) = \left\{ x \in \{0,1\}^{\binom{n}{2}+n} \,\middle|\, x_{ij} = x_i \oplus x_j, i < j \right\}$$

Proof. See [2], proof of Theorem 5 (Appendix D). □

If we take $\mathcal{CUT}$ to be any family of cut polytopes that contains $CUT^{\square}(K_n)$ for each $n \in \mathbb{N}$ then this family has high extension complexity. This family was the first explicit family of polytopes that

was shown to have superpolynomial extension complexity. Here we present a combinatior of ideas that appeared in [2, 27, 41].

The crucial fact used for showing that the $\mathbb{CUT}$ clan has large extension complexity is that a certain matrix $\mathbf{U} = (\mathbf{a}^\top \mathbf{b} - 1)^2$ has large nonnegative rank (cf. Proposition 2.1.13). The next step is to show that this matrix is actually a submatrix of some slack matrix of $\mathrm{CUT}^{\square}(K_n)$. Then by Proposition 2.1.3 and Proposition 3.1.24 we get the desired lower bound on the extension complexity of the $\mathbb{CUT}$ clan.

The first step in embedding $\mathbf{U}(n-1)$ in a slack matrix of $\mathrm{CUT}^{\square}(K_n)$ is to identify some valid inequalities that produce the desired slack. The following lemma describes a set of such inequalities.

Lemma 3.3.4. *For any $n \geqslant 2$, let $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n$ be any set of n integers. The following inequality is valid for $\mathrm{CUT}^{\square}(K_n)$:*

$$\sum_{1 \leqslant i < j \leqslant n} \mathbf{b}_i \mathbf{b}_j x_{ij} \leqslant \left\lfloor \frac{(\sum_{i=1}^{n} b_i)^2}{4} \right\rfloor \tag{4}$$

Proof. See [2], Lemma 1 (Appendix D). □

The inequality (4) is called *hypermetric* (respectively, of *negative type*) if the integers b_i can be partitioned into two subsets whose sum differs by one (respectively, zero). A simple example of hypermetric inequalities are the triangle inequalities, obtained by setting three of the b_i to be +/- 1 and the rest to be zero. The most basic negative type inequality is non-negativity, obtained by setting one b_i to 1, another one to -1, and the others to zero. We note in passing that Deza and Laurent (see Section 6.1 of [21]) showed that each negative type inequality could be written as a convex combination of hypermetric inequalities, so that none of them are facet inducing for $\mathrm{CUT}^{\square}(K_n)$.

Let $n \geqslant 2$ be an integer. Let $S \subseteq [n-1]$. This defines a cut $\delta(S)$ of K_n and each cut in K_n has such a subset of vertices defining it. Define a vector $\mathbf{b}$ with

$$\mathbf{b}_i = \begin{cases} 1, & \text{if } i \in S \\ 0, & n \neq i \notin S \\ 3 - |S| & \text{i=n} \end{cases}$$

Observe that $|S| = \mathbf{1}^\top \mathbf{b}$. Inequality (4) for this $\mathbf{b}$-vector is easily seen to be of negative type and can be written as

$$\sum_{1 \leqslant i < j \leqslant n-1} \mathbf{b}_i \mathbf{b}_j x_{ij} \leqslant 1 + (\mathbf{1}^\top \mathbf{b} - 3) \sum_{i=1}^{n-1} \mathbf{b}_i x_{in}. \tag{5}$$

Let $C \subseteq [n-1]$ and accordingly $\delta(C)$ be a cut of K_n. Define the vector $\mathbf{a}$ with

$$\mathbf{a}_i = \begin{cases} 1, & \text{if } i \in C \\ 0, & i \notin S \end{cases}$$

Proposition 3.3.5. *Let C and S be subsets of $[n-1]$. Then the slack of the cut $\delta(C)$ with respect to (5) is $(\mathbf{a}^\top \mathbf{b} - 1)^2$ with $\mathbf{a}, \mathbf{b}$ as defined previously.*

Proof.

$$\begin{aligned}
&1 + (\mathbf{1}^\top \mathbf{b} - 3) \sum_{i=1}^{n-1} \mathbf{b}_i \delta(C)_{in} - \sum_{1 \leqslant i < j \leqslant n-1} \mathbf{b}_i \mathbf{b}_j \delta(C)_{ij} \\
&= 1 + (\mathbf{1}^\top \mathbf{b} - 3)\mathbf{a}^\top \mathbf{b} - \mathbf{a}^\top \mathbf{b}(\mathbf{1}^\top \mathbf{b} - \mathbf{a}^\top \mathbf{b} - 1) \\
&= (\mathbf{a}^\top \mathbf{b})^2 - 2\mathbf{a}^\top \mathbf{b} + 1.
\end{aligned}$$

□

This implies that the extension complexity of $\mathrm{CUT}^{\square}_{(}K_n)$ is at least as large as $\mathrm{rank}_+(\mathbf{U}(n-1))$. That is,

Proposition 3.3.6. $\mathrm{xc}(\mathrm{CUT}^{\square}(K_n)) \geqslant 2^{\Omega(n)}$.

This in turn implies that the extension complexity of the $\mathbb{CUT}$ clan is exponential in the ambient dimension.

Proposition 3.3.7. $\mathrm{xc}(\mathbb{CUT}) \geqslant 2^{\Theta(\sqrt{n})}$. *In particular,* $\mathrm{xc}(\mathcal{CUT}) \geqslant 2^{\Theta(\sqrt{n})}$.

It is not a coincidence that the high lower bound of $\mathbb{CUT}$ is obtained by taking the family $\mathcal{CUT}$ of cut polytopes corresponding to the complete graphs. The specific family of complete graphs is in some sense the most general family of graphs for defining a hard family of cut polytopes as evidenced by the following.

Proposition 3.3.8. *Let* G *be any graph on* n *vertices. Then* $\mathrm{CUT}^{\square}(G)$ *is a projection of* $\mathrm{CUT}^{\square}(K_n)$.

Proof. If an edge (i, j) is not present in G, project it out. □

3.3.2 *The $\mathcal{CNF}$-$\mathcal{CERT}$ family*

For any given boolean formula φ with n variables define the polytope $\mathrm{SAT}(\varphi)$ as the convex hull of all satisfying assignments. That is,

$$\mathrm{SAT}(\varphi) := \mathrm{conv}(\{\mathbf{x} \in \{0,1\}^n \mid \varphi(\mathbf{x}) = 1\})$$

Since every 0/1 polytope is trivially the SAT polytope for some CNF formula, the corresponding clan of polytopes is just the clan $\mathbb{ZERO-ONE}$ which has high extension complexity as pointed out in Proposition 3.3.1. We will now show the existence of an easy to construct family of SAT polytopes that has superpolynomial extension complexity. The polytopes in this family will correspond to CNF formulae encoding the cuts of K_n as their satisfying assignments. This family of polytopes will be called the $\mathcal{CNF}$-$\mathcal{CERT}$ family, and will turn out to be a canonical family of polytopes with high extension complexity that will be used in Chapter 4 to give lower bounds for other families.

Let $n \in \mathbb{N}$ and $m = n^2$. For the complete graph K_n define a 3SAT boolean formula φ_m such that $\mathrm{CUT}^{\square}(K_n)$ is a projection of $\mathrm{SAT}(\varphi_m)$. Consider the relation $x_{ij} = x_{ii} \oplus x_{jj}$, where $\oplus$ is the xor operator. The boolean formula

$$(x_{ii} \vee \overline{x}_{jj} \vee x_{ij}) \wedge (\overline{x}_{ii} \vee x_{jj} \vee x_{ij}) \wedge (x_{ii} \vee x_{jj} \vee \overline{x}_{ij}) \wedge (\overline{x}_{ii} \vee \overline{x}_{jj} \vee \overline{x}_{ij})$$

is true if and only if $x_{ij} = x_{ii} \oplus x_{jj}$ for any assignment of the variables x_{ii}, x_{jj} and x_{ij}.

Therefore we define φ_m (with $m = n^2$) as

$$\varphi_m := \bigwedge_{\substack{i,j \in [n] \\ i \neq j}} \begin{bmatrix} (x_{ii} \vee \overline{x}_{jj} \vee x_{ij}) \wedge (\overline{x}_{ii} \vee x_{jj} \vee x_{ij}) \wedge \\ (x_{ii} \vee x_{jj} \vee \overline{x}_{ij}) \wedge (\overline{x}_{ii} \vee \overline{x}_{jj} \vee \overline{x}_{ij}) \end{bmatrix}. \tag{6}$$

This ensures that the the satisfying assignments of φ_m when restricted to the variables x_{ij} with $i \neq j$ are exactly the cut vectors of K_n and every cut vector of K_n can be extended to a satisfying assignment of φ. Consequently, we have that.

Proposition 3.3.9. $\mathrm{SAT}(\varphi_m)$ *is an* EF *of* $\mathrm{CUT}^{\square}(K_{n+1})$ *for each natural number* $m = n^2$.

Proof. This follows from Proposition 3.3.3. In fact, $\mathrm{SAT}(\varphi_m)$ is actually $\mathrm{CUT}^{\square}(K_{n+1})$. □

Definition 3.3.10. The $\mathcal{CNF}$-$\mathcal{CERT}$ family of polytopes is defined by the polytopes $P_m = \mathrm{SAT}(\varphi_m)$ with φ_m described previously by equation 6. For values of m with no corresponding φ_m we have $P_m = \emptyset$.

Note that $\mathrm{SAT}(\varphi_m)$ has $m = n^2$ variables and $4(m-n)$ clauses. Since $\mathrm{CUT}^{\square}(K_n)$ is a projection of $\mathrm{SAT}(\varphi_m)$, we can conclude that $\mathrm{xc}(\mathrm{SAT}(\varphi_m)) \geqslant \mathrm{xc}(\mathrm{CUT}^{\square}_{}) \geqslant 2^{\Omega(n)}$ (cf. Prop. 3.1.22), and thus,

Proposition 3.3.11. $\mathrm{xc}(\mathcal{CNF}\text{-}\mathcal{CERT}) = 2^{\Omega(\sqrt{n})}$.

Proof. This follows from Proposition 3.3.9. □

3.3.3 *The $\mathcal{DNF}$-$\mathcal{CERT}$ family*

Finally, we describe an interesting family of polytopes that has polynomial extension complexity. Similar to, and yet in contrast with the $\mathcal{CNF}$-$\mathcal{CERT}$ family, this family corresponds to the satisfying assignments of DNF formulae. Notice that whereas deciding satisfiability of a CNF formula is an NP-hard problem, deciding the same for a DNF formulae is trivial.

Let $\Phi = \{\varphi_1, \ldots\}$ be a family of DNF formulae where φ_n has n variables and $\mathrm{poly}(n)$ clauses. We will call the family $\{\mathrm{SAT}(\varphi_1), \ldots\}$ of polytopes a $\mathcal{DNF}$-$\mathcal{CERT}$ family. It is not important to pick a canonical representative of this family because as we will see next, the certificates of a polynomial sized DNF formula have polynomial extension complexity.

Proposition 3.3.12. *Let* φ *be a DNF formula with* n *variables and* m *clauses. Then* $\mathrm{xc}(\mathrm{SAT}(\varphi)) \leqslant 2mn$.

Proof. If φ consists of a single clause then it is just a conjunction of some literals. In this case $\mathrm{SAT}(\varphi)$ is a face of the n-hypercube and has $\mathrm{xc}(\mathrm{SAT}(\varphi)) \leqslant 2n$. Furthermore, for DNF formulae φ_1, φ_2 we have that $\mathrm{SAT}(\varphi_1 \vee \varphi_2) = \mathrm{SAT}(\varphi_1) \uplus \mathrm{SAT}(\varphi_2))$. Therefore, using Proposition 3.2.5 repeatedly we obtain that for a DNF formula φ with n variables and m clauses $\mathrm{SAT}(\varphi) \leqslant 2mn$. □

As a consequence we obtain the following.

Proposition 3.3.13. *Let* $\mathcal{DNF}\text{-}\mathcal{CERT} = \{P_1, P_2, \ldots\}$ *be a family of polytopes where* $P_n = \mathrm{SAT}(\boldsymbol{\varphi_n})$ *and* $\boldsymbol{\varphi_n}$ *a* DNF *formula with* n *varibles and* $\mathrm{poly}(n)$ *clauses. Then,* $\mathrm{xc}(\mathcal{DNF}\text{-}\mathcal{CERT}) = \mathrm{poly}(n)$.

Part II

RECIPES

會則事同一家　*With realization, things make one family;*
不會萬別千差　*Without realization, things are separated in a thousand ways.*
不會事同一家　*Without realization, things make one family;*
會則萬別千差　*With realization, things are separated in a thousand ways.*

— The Gateless Gate: Case 16 [38]

4

TURING REDUCTIONS

In Section 3.1 we saw some simple observations that make it possible to translate bounds on extension complexity of a polytope P to that of another polytope Q simply by demonstrating that Q essentially contains P (cf. Propositions 3.1.21, 3.1.22, and 3.1.23). Now we will see actual examples where these observations are put to use.

4.1 RELATIVES OF CUT POLYTOPES

4.1.1 *Cut polytope for minors of a graph*

Definition 4.1.1. Let $G = (V, E)$ be a graph. A graph $H = (V', E')$ is called a *minor* of G if an isomorphic copy of H can be obtained from G by a sequence of the following operations.

- *Vertex deletion* : $V' = V \setminus \{v\}$ and $E' = E \setminus \{e \in E \mid v \in e\}$ for some vertex $v \in V$.
- *Edge deletetion* : $V' = V$ and $E' = E \setminus \{e\}$ for some edge $e \in E$.
- *Edge contraction* : $V' = (V \setminus \{u, v\}) \cup \{w\}$ and $E' = E \setminus \{e \in E | u \in e \vee v \in e\} \cup \{(w, z) | (x, z) \in E, x \in \{u, v\}, z \notin \{u, v\}\}$ for some $u, v \in V$ and $w \notin V$.

It turns out that extension complexity is a monotone property under taking minors.

Proposition 4.1.2. *Let G be a graph and let H be a minor of G, then some face of $\mathrm{CUT}^{\square}(G)$ is an EF of $\mathrm{CUT}^{\square}(H)$. In particular,*

$$\mathrm{xc}(\mathrm{CUT}^{\square}(G)) \geqslant \mathrm{xc}(\mathrm{CUT}^{\square}(H)).$$

Proof. See [2], Theorem 12 (Appendix D). □

Using this together with Proposition 3.3.6 we can conclude the following.

Proposition 4.1.3. *The extension complexity of $\mathrm{CUT}^{\square}(G)$ for a graph G with a K_n minor is at least $2^{\Omega(n)}$.*

The cut polytope of the tripartite graph $K_{1,n,n}$ is called the Bell inequality polytope and plays an important role in Quantum Physics for the study of quantum entanglement [5]. We can conclude that this polytope cannot be represented by a polynomial number of inequalities even if we allow extra variables.

Proposition 4.1.4. $\mathrm{xc}(\mathrm{CUT}^{\square}(K_{1,n,n})) = 2^{\Omega(n)}$.

Proof. Pick any matching of size n between the vertices in each of the two parts of cardinality n. Contracting the edges in this matching yields K_{n+1} and the result follows. □

So we see that a large clique as a minor is sufficient for the cut polytope of a graph to have high extension complexity. Is it also necessary? Before we answer this question (in the negative) we discuss a polytope whose relation to cut polytope will become clear in Subsection 4.1.3.

4.1.2 *Stable set for cubic planar graphs*

Definition 4.1.5. Let $G = (V, E)$ be a graph. The convex hull of characteristic vectors of the independent sets in G is called the *stable set polytope* of G and is denoted by $\mathrm{STAB}(G)$. Any polytope P such that $P = \mathrm{STAB}(G)$ for some graph G is called a STAB polytope.

Since finding the largest independent set in arbitrary graphs is an NP-hard problem, it would be very surprising if the clan $\mathbb{STAB}$ of STAB polytopes had polynomial extension complexity. Indeed this is not the case and this clan was also one of the first to be shown to have superpolynomial extension complexity.

Proposition 4.1.6. $\mathrm{xc}(\mathbb{STAB}) = 2^{\Omega(\sqrt{n})}$.

Proof. See [27], Theorem 10 (Appendix A). □

In fact a family of STAB polytopes of cubic planar graphs can already has superpolynomial extension complexity.

Proposition 4.1.7. *There exists a family $\mathcal{STAB}$ of* STAB *polytopes of cubic planar graphs such that* $\mathrm{xc}(\mathcal{STAB}) = 2^{\Omega(\sqrt[4]{n})}$.

Proof. See [2], Corollary 5 (Appendix D). □

4.1.3 *Cut Polytope for K_6 minor-free graphs*

In Subsection 4.1.1 we saw that a large clique as a minor is sufficient for the cut polytope of a graph to have high extension complexity. Now we will see that a large clique minor is not necessary for high extension complexity of the cut polytope. In particular, there are K_6-minor free graphs whose cut polytopes have large extension complexity. Note that if a n-vertex graph G has no K_5-minor then $\mathrm{CUT}^{\square}(G)$ has $\mathcal{O}(n^3)$ extension complexity [21]. Contrast this with the fact that the MAXCUT problem is solvable in polynomial time on K_5 minor-free graphs but becomes NP-hard on K_6 minor-free graphs.

Definition 4.1.8. Let $G = (V, E)$ be any graph with $V = \{1, \ldots, n\}$. The *suspension* G' of G is obtained by adding an extra vertex labeled 0 with edges to all vertices V.

The operation of creating suspension of a graph is actually what relates the CUT and the STAB polytopes with each other.

Proposition 4.1.9. *Let* $G = (V, E)$ *be a graph and let* G' *be a suspension over* G. *Then* $\mathrm{STAB}(G)$ *is the projection of a face of* $\mathrm{CUT}^{\square}(G')$.

Proof. See [2], Theorem 13 (Appendix D). □

By Proposition 4.1.6 we we have a family of cubic planar graphs whose STAB polytopes give a family with superpolynomial extension complexity. Planar graphs do not contain K_5 as a minor and so the suspension of any planar graph is K_6 minor-free. This gives us a family of K_6-minor graphs whose cut polytopes have high extension complexity.

Proposition 4.1.10. *There exists a family* $\mathcal{CUT}'$ *of* CUT *polytopes of* K_6 *minor-free graphs such that* $\mathrm{xc}(\mathcal{CUT}') = 2^{\Omega(\sqrt[4]{n})}$.

This provides a sharp contrast for the complexity of the cut polytope for graphs in terms of their minors. As noted earlier, for any K_5 minor-free graph G with n vertices $\mathrm{CUT}^{\square}(G)$ has an extension of size $\mathcal{O}(n^3)$ whereas the above result shows that there are K_6 minor-free graphs whose cut polytope has superpolynomial extension complexity.

4.2 EMBEDDING ARGUMENTS FROM TURING REDUCTIONS

The central technique used in the previous section was to argue that a polytope P can be obtained as a projection of some face of polytope Q and then using the fact that $\mathrm{xc}(P)$ is large to argue that $\mathrm{xc}(Q)$ must be large too. How difficult is it to come up with a reduction that shows such an embedding?

Surprisingly it is quite common that for a polytope family related to an NP-hard problem the standard NP-hardness reduction also gives the desired embedding of one polytope family into another. In fact, the proofs of Propositions 4.1.7 and 4.1.9 are based on standard NP-hardness reductions for the associated problems. Now we present some more examples where the standard reductions suffice.

4.2.1 *Traveling Salesman*

Definition 4.2.1. Let G be a graph. The *traveling salesman polytope* of G – denoted by $\mathrm{TSP}(G)$ – is the convex hull of all Hamiltonian cycles of G. Any polytope P such that $P = \mathrm{TSP}(G)$ for some graph G will be referred to as a TSP polytope. The clan of all TSP polytopes is denoted by $\mathbb{TSP}$.

Similar to the $\mathbb{CUT}$ clan, complete graphs a canonical hard class of graphs for the extension complexity of the $\mathbb{TSP}$ clan. This is because of the following.

Proposition 4.2.2. *Let* G *be a graph on* n *vertices. Then* $\mathrm{TSP}(G)$ *is a face of* $\mathrm{TSP}(K_n)$.

Proof. For any edge (i, j) missing in G, restrict to the face of $\mathrm{TSP}(K_n)$ defined by the valid inequality $x_{ij} = 0$. □

The TSP problem asks whether a given graph contains a tour visiting every vertex exactly once and is known to be NP-hard. In fact, the standard NP-hardness reduction can be used to show superpolynomial lower bound on the extension complexity of the $\mathbb{TSP}$ clan.

Proposition 4.2.3. $\mathrm{xc}(\mathbb{TSP}) = 2^{\Omega(\sqrt[4]{n})}$.

Proof. See [27], Theorem 12 (Appendix A). □

Rothvoß [55] has improved the above bound to show that in fact $\mathrm{xc}(\mathbb{TSP}) = 2^{\Omega(\sqrt{n})}$. His bound again uses a standard embedding argument from perfect matching to TSP, but his lower bound for the perfect matching polytope uses new tools which are out of scope for us.

4.2.2 *Subset sum*

Definition 4.2.4. Given n integers $a^\top = (a_1, \ldots, a_n)$ and another integer b, the subset sum problems asks whether any subset of the set $\{a_1, \ldots, a_n\}$ sums exactly to b. Define the subset sum polytope $\mathrm{SUBSETSUM}(a, b)$ as the convex hull of all characteristic vectors of the subsets of a whose sum is exactly b.

$$\mathrm{SUBSETSUM}(a, b) := \mathrm{conv}\left(\left\{x \in \{0,1\}^n \mid a^\top x = b\right\}\right)$$

A polytope P which is $\mathrm{SUBSETSUM}(a, b)$ for some integers a, b will be a SUBSETSUM polytope.

The subset sum problem is, then, equivalent to asking whether the SUBSETSUM polytope for a given integer vector a and integer b is empty.

A related knapsack polytope can be defined as

$$\mathrm{KNAPSACK}(a, b) := \mathrm{conv}\left(\left\{x \in \{0,1\}^n \mid a^\top x \leqslant b\right\}\right)$$

Using the standard NP-hardness reduction for the subset sum problem one can show the following.

Proposition 4.2.5. *For every 3SAT formula φ with n variables and m clauses, there exists an integer vector $a(\varphi)^\top = (a_1, \ldots, a_{2n+2m})$ and integer $b(\varphi)$ such that $\mathrm{SAT}(\varphi)$ is a projection of $\mathrm{SUBSETSUM}(a, b)$.*

Proof. See [2], Theorem 6 (Appendix D). □

It immediately follows using Proposition 3.3.11 that there is a family of SUBSETSUM polytopes with high extension complexity.

Proposition 4.2.6. *Let $\mathbb{SUBSETSUM}$ be the clan of all SUBSETSUM polytopes. Then, $\mathrm{xc}(\mathbb{SUBSETSUM}) = 2^{\Omega(\sqrt{n})}$.*

Since the polytope $\mathrm{SUBSETSUM}(a, b)$ is a face of $\mathrm{KNAPSACK}(a, b)$, we have the following.

Proposition 4.2.7. *Let $\mathbb{KNAPSACK}$ be the clan of of KNAPSACK polytopes. Then, $\mathrm{xc}(\mathbb{KNAPSACK}) = 2^{\Omega(\sqrt{n})}$.*

4.2.3 *3d-matching*

Definition 4.2.8. Consider a hypergraph $G = ([n], E)$, where E contains triples (i, j, k) for some distinct $i, j, k \in [n]$. A subset $E' \subseteq E$ is said to be a 3-dimensional matching if all the triples in E' are disjoint. The 3d-matching polytope $3DM(G)$ is defined as the convex hull of the characteristic vectors of every 3d-matching of G. That is,

$$3DM(G) := \text{conv}(\{\chi(E') \mid E' \subseteq E \quad \text{is a 3d-matching}\})$$

It is often customary to consider only hypergraphs defined over three disjoint set of vertices X, Y, Z such that the hyperedges are subsets of $X \times Y \times Z$. Observe that any hypergraph G can be converted into a hypergraph H in such a form by making three copies of the vertex set V, V', V'' and using a hyperedge (i, j', k'') in H if and only if (i, j, k) is a hyperedge in G.

Exercise 4.2.9. Show that $xc(3DM(G)) = \Theta(xc(3DM(H)))$.

Definition 4.2.10. A polytope P is said to be a 3DM polytope if $P = 3DM(G)$ for some hypergraph G.

The 3d-matching problem asks: given a hypergraph G, does there exist a 3d-matching that covers all vertices? This problem is known to be NP-complete and was one of Karp's 21 problems proved to be NP-complete [31, 42]. This problem can be solved by linear optimization over the polytope $3DM(G)$ and therefore it is to be expected that $3DM(G)$ would not have a polynomial size EF for every hypergraph G.

Proposition 4.2.11. *For the clan* $3\mathbb{DM}$ *of* 3DM *polytopes we have that* $xc(3\mathbb{DM}) = 2^{\Omega(\sqrt[4]{n})}$.

This follows from the following Proposition whose proof relies on the standard NP-hardness reduction for the 3d-matching problem.

Proposition 4.2.12. *Let* φ *be a CNF formula with* n *variables and* m *clauses. Then there exists a hypergraph* $H = (V, E)$ *with* $|V| = \mathcal{O}(nm)$ *and* $|E| = \mathcal{O}(nm)$ *such that* $SAT(\varphi)$ *is the projection of a face of* $3DM(H)$.

Proof. See [2], Corollary 3 (Appendix D). □

4.2.4 *Induced matchings*

Definition 4.2.13. A matching in a graph $G = (V, E)$ is called induced if there is no edge in G between any pair of matching edges.

Stockmeyer and Vazirani [58] and Cameron [15] proved that the problem of finding a maximum cardinality induced matching is NP-hard.

Definition 4.2.14. Let G be a graph. The convex hull of all induced matchings G is called the *induced matching polytope* of G and is denoted by $\text{IndMatch}(G)$. A polytope P is said to be an IndMatch polytope if there exists a graph G such that $P = \text{IndMatch}(G)$.

Using the reduction in [15] one can show the following.

Proposition 4.2.15. *For every n there exists a bipartite graph G_n with $\mathcal{O}(n)$ edges and vertices such that* $\mathrm{xc}(\mathrm{IndMatch}(G)) = 2^{\Omega(\sqrt[4]{n})}$.

Proof. See [1], Theorem 1 (Appendix H). □

This implies the existence of a family with high extension complexity.

Proposition 4.2.16. *There exists a family $\mathcal{INDMATCH}$ of* IndMatch *polytopes such that* $\mathrm{xc}(\mathcal{INDMATCH}) = 2^{\Omega(\sqrt[4]{n})}$.

4.2.5 *Maximal matchings*

Definition 4.2.17. A matching in a graph $G = (V, E)$ is called *maximal* if its edge set is not included in a larger matching.

It is known that finding the minimum maximal matching is NP-hard [64].

Definition 4.2.18. Let G be a graph. The convex hull of all maximal matchings of G is called the maximal matching polytope of G and is denote it by $\mathrm{MaxMatch}(G)$. Accordingly a polytope P is called a MaxMatch polytope if $P = \mathrm{MaxMatch}(G)$ for some graph G.

Since the perfect matching polytope of K_{2n} has extension complexity $2^{\Omega(n)}$, we clearly have that the clan of MaxMatch polytopes has high extension complexity. However one can use the standard proof of NP-hardness of minimum maximal matching to show superpolynomial bound as well.

The only hurdle in using the reduction of [64] is that the reduction is from CNF formulae of special kind, namely formulae with at most one nonnegated literal and at most two negated literals in each clause. High extension complexity for the SAT polytopes of formulae of this specific kind can be shown by a fairly simple reduction.

Proposition 4.2.19. *For every n there exists a 3-CNF formula φ_n with $\mathcal{O}(n)$ variables and clauses such* $\mathrm{xc}(\mathrm{SAT}(\varphi_n)) \neq \mathrm{poly}(n)$. *Furthermore, in every clause of φ_n every variable appears at most twice non-negated and at most once negated.*

Proof. See [1], Theorem 2 (Appendix H). □

The reduction of [64] then gives the following.

Proposition 4.2.20. *For every n there exists a bipartite graph $G = (V_1 \cup V_2, E)$ with $\mathcal{O}(n)$ vertices and edges, such that* $\mathrm{xc}(\mathrm{MaxMatch}(G)) \neq \mathrm{poly}(n)$.

Proof. See [1], Theorem 3 (Appendix H). □

Thus we have the following.

Proposition 4.2.21. *Let $\mathbb{MAXMATCH}$ be the clan of* MaxMatch *polytopes. Then,* $\mathrm{xc}(\mathbb{MAXMATCH}) \neq \mathrm{poly}(n)$.

4.2.6 *Edge disjoint matching and perfect matching*

Given a bipartite graph $G(V_1 \cup V_2, E)$ and a natural number k, it is NP-hard to decide whether G contains a perfect matching M and a matching M' of size k such that M and M' do not share an edge [52].

For a given graph G with n vertices and m edges consider an encoding of a perfect matching and a matching using variables $x_1, \ldots, x_m$, $y_1, \ldots, y_m$ as follows. For a subset of edges encoding a perfect matching M and a matching M' of size k we construct a vector with

$$x_i = \begin{cases} 1, & \text{if } e_i \in M \\ 0, & \text{if } e_i \notin M \end{cases}, \quad y_i = \begin{cases} 1, & \text{if } e_i \in M' \\ 0, & \text{if } e_i \notin M' \end{cases}$$

Definition 4.2.22. Let $G = (V, E)$ be a graph. Define the polytope $\mathrm{MPM}(G, k)$ to be the convex hull of all the vectors encoding an edge disjoint perfect matching and a matching of size at least k. As usual, a polytope P such that $P = \mathrm{MPM}(G, k)$ for some graph G and natural number k is called an MPM polytope.

We would like to remark that one can also define a "natural" polytope here without using separate variables for a matching and a perfect matching and instead using the characteristic vectors of all subsets of edges that are an edge-disjoint union of a matching and a perfect matching. However, the formulation that we consider allows different cost functions to be applied to the matching and the perfect matching.

Using the reduction in [52], one can show that for every n there exists a bipartite graph G_n with $\mathcal{O}(n)$ vertices and a constant $0 < c < \frac{1}{2}$ such that $\mathrm{MPM}(G, cn)$ has extension complexity super polynomial in n. Again the reduction in [52] is from MAX-2-SAT, so we first need to prove a super polynomial lower bound for the SAT polytopes of 2-CNF formulas.

Proposition 4.2.23. *For every n there exists a 2-SAT formula φ_n with n variables such that* $\mathrm{xc}(\mathrm{SAT}(\varphi_n)) = 2^{\Omega(\sqrt[4]{n})}$.

Proof. See [1], Theorem 4 (Appendix H). □

As a side remark, the 2-SAT instances required in the above theorem are always satisfiable.

Proposition 4.2.24. *For every n there exists a bipartite graph G_n on n vertices and a constant $0 < c < \frac{1}{2}$ such that* $\mathrm{xc}(\mathrm{MPM}(G, cn)) \neq \mathrm{poly}(n)$.

Proof. See [1], Theorem 5 (Appendix H). □

Thus, we have the following.

Proposition 4.2.25. *There exists a family $\mathcal{MPM}$ of* MPM *polytopes such that* $\mathrm{xc}(\mathcal{MPM}) \neq \mathrm{poly}(n)$.

4.3 DIFFICULITIES IN HANDLING GENERAL REDUCTIONS

Seeing so many NP-hardness reductions yield superpolynomial extension complexity lower bounds for the associated polytopes, it was suspected[1] that it may be possible to prove a meta result. A result similar to: "A problem is in PTIME if and only if the associated polytope has polynomial extension complexity". Notwithstanding what "associated polytope of a problem" meant, this was shown to be impossible in a remarkable paper [55] that showed that the perfect matching polytope has exponential extension complexity (See Proposition 3.1.13).

It should be noted that it may still be possible to have the polytopes associated with NP-complete problems always have superpolynomial extension complexity. However a more likely scenario may be that the answer depends crucially on how one associates polytopes with problems, and for various choices of such associations the extension complexity may be trivially exponential while for others trivially polynomial, and for yet others very difficult to determine.

In any case, for all we know PTIME reductions that are allowed for proving NP-hardness may be as powerful as NP itself and so being able to translate superpolynomial lower bounds on extension complexity using arbitrary polynomial reductions may be too much to ask.

1 At least by the present author

5

COMPACT LANGUAGES

Let φ be a boolean formula. Consider the following languages:

$$\begin{aligned} L &= \{x \mid x \text{ encodes a satisfiable boolean formula}\} \\ L((\varphi)) &= \{x \mid \varphi(x) = 1\} \end{aligned}$$

The former language consists of all strings that encode[1] all satisfiable boolean formulae, while the later language consists of all satisfying assignments of a given boolean formula. Which of these represents the boolean satisfiability problem *more naturally*?

Reasonable people will agree that there is no correct choice of a natural polytope for a problem. One complication is that there various kinds of problems: decision, optimization, enumeration, etc, and very similar problems can have very different behaviour if the notion of problem changes.

Example 5.0.1. Checking whether a bipartite graph has a perfect matching can be solved by a simple polynomial time algorithm. A related problem where one wants to count the number of bipartite matchings is #P-hard.

Therefore one can pick any clan of polytopes, that they consider reasonable, as representing a given problem but it can be asked whether the extension complexities of that particular choice of polytopes reflect some underlying complexity measure of the problems. Often the most immediate choice of polytopes does not really correspond well to the computational complexity.

Example 5.0.2. The clan $\mathbb{EP}$ of perfect matching polytopes is a natural choice for the underlying decision problem: given a graph G, does it have a perfect matching? While the computational problem is polynomial time solvable, the extension complexity of Edmonds' polytopes is exponential.

One reason for such complication is that wildly different kinds of problems are defined over the same set of objects. For example, over the set of graphs and their perfect matchings, we can ask natural decision, optimization, and counting questions. The first two are polynomial time solvable while the last one is #P-hard.

Our perspective of the situation will be as follows. Algorithms will be identified with Turing machines with five tapes[2].

- A two-way read-only input tape.

1 Assume that some (arbitrary but fixed) encoding of boolean formulae as binary strings.

2 This does not make the Turing machines special.

- An auxiliary read-only input tape than can be read only from left to right.
- A two-way read-write work tape.
- A two-way write-only output tape.
- An auxiliary write-only output tape that can be written only from left to right.

Computational problems are then questions about existence of Turing machines of the above kind with various restrictions on the consumption of resources such as space consumed on the work tape or the overall time, and on the relation between the contents of the input tapes and the output tapes when the machine halts.

5.1 PROBLEMS AS LANGUAGES

For us computational problems will just be questions about an underlying language. Various natural problems can be modeled in this way. Usually a specific computational problem comes with an underlying language $\mathbf{L}$. The specific problem at hand is then some question about the language $\mathbf{L}$ (or about strings of this language).

5.1.1 *Membership Problem*

The membership problems asks whether a given string belongs to a particular language. We will denote such problems by $\text{mem}(\mathbf{L})$.

Example 5.1.1. Let $\mathbf{L} = \{x \mid x \text{ encodes a satisfiable boolean formula}\}$. The problem $\text{mem}(\mathbf{L})$ then is just the familiar boolean satisfiability problem where an encoding of the input has been agreed upon.

5.1.2 *Optimization Problem*

The optimization problem for a particular language $\mathbf{L}$ comes equipped with a function that assigns a real number to every string in the language and one is interested in finding the "best" string from $\mathbf{L}$.

A particularly important class of such problems is one of linear optimization. Given a cost vector $\mathbf{c} \in \mathbb{R}^n$ one is interested in a string $\mathbf{x}^* \in \mathbf{L}$ with $|\mathbf{x}^*| = n$ such that for every $\mathbf{x} \in \mathbf{L}$ with $|\mathbf{x}| = n$ we have that $\mathbf{c}^\top \mathbf{x}^* \geqslant \mathbf{c}^\top \mathbf{x}$.

We will denote by $\text{opt}(\mathbf{L})$ the problem of maximizing a linear function over all members of $\mathbf{L}$ that have length n.

Example 5.1.2. Given a linear cost function and a boolean formula φ, find a satisfying assignment (if any) of minimum cost.

5.1.3 *Enumeration Problem*

The canonical enumeration problem for a particular language $\mathbf{L}$ – denoted by $\text{enum}(\mathbf{L})$ – is as follows: given a number n enumerate all members of $\mathbf{L}$ that have length n.

Example 5.1.3. Enumerate all satisfying assignments of a given boolean formula.

5.1.4 *Sampling Problem*

The canonical sampling problem for a particular language $\mathbf{L}$ – denoted by sample($\mathbf{L}$) – is as follows: given a number n produce a length n member of $\mathbf{L}$ with a given probability distribution.

Example 5.1.4. Given a boolean formula, produce a satisfying assignment (if any) uniformly at random.

5.1.5 *Counting problem*

The canonical counting problem for a particular language $\mathbf{L}$ – denoted by count($\mathbf{L}$) – is as follows: given a number n how many members of $\mathbf{L}$ have length exactly n?

Example 5.1.5. How many satisfying assignents does a given boolean formula have?

5.2 COMPACT LANGUAGES

5.2.1 *Languages to Polytopes*

For every natural number n define the set $\mathbf{L}(n) := \{\mathbf{x} \in \{0,1\}^n \mid \mathbf{x} \in \mathbf{L}\}$. Viewing each string $\mathbf{x} \in \mathbf{L}(n)$ as a column vector, and ordering the strings lexicographically, we can view the set $\mathbf{L}(n)$ as a matrix of size $n \times |\mathbf{L}(n)|$. Thus we are in a position to naturally associate a family of polytopes with a given language and the extension complexity of these polytopes can serve as a natural measure of how hard is it to model these languages as Linear Programs.

That is, one can associate with $\mathbf{L}$, the family of polytopes $\mathcal{P}(\mathbf{L}) = \{P(\mathbf{L}(1)), P(\mathbf{L}(2)), \ldots\}$ and the extension complexity $\mathrm{xc}(\mathcal{P}(\mathbf{L}))$ is then an intrinsic measure of complexity of the language $\mathbf{L}$.

Example 5.2.1. Matching polytope of complete graphs with a canonical (say lexicographic) ordering on the edges. The associated language consists of the characteristic vectors of all perfect matchings of K_n for $n \in \mathbb{N}$.

Definition 5.2.2. The *extension complexity* of a language $\mathbf{L}$ – denoted by $\mathrm{xc}(\mathbf{L})$ – is defined by $\mathrm{xc}(\mathbf{L}) := \mathrm{xc}(\mathcal{P}(\mathbf{L}))$.

Now we are ready to define the class of languages that we are interested in: namely, the languages that have small extension complexities.

Definition 5.2.3. $\mathcal{CF}$ is the class of languages admitting $\mathcal{C}$ompact extended $\mathcal{F}$ormulations and is defined as

$$\mathcal{CF} \quad = \quad \{\mathbf{L} \subseteq \{0,1\}^* \mid \exists c > 0 \text{ s.t. } \mathrm{xc}(\mathbf{L}) \leqslant n^c\}$$

5.2.2 *Easy problems for Compact languages*

Let $\mathbf{L}$ be a compact language. That is for each $n \in \mathbb{N}$ the polytope $P(\mathbf{L}(n))$ has small extension complexity. What does that give us in terms of solving computational problems related to $\mathbf{L}$? Naturally, one would need such an extended formulation itself to be efficiently computable. It may very well happen that a polytope has a small extension but the actual numbers needed to represent any small sized extension require very large precision. In fact it is unknown whether small extension complexity using real numbers implies small extension complexity using rational numbers.

Notwithstanding the previous discussion, let us assume that language $\mathbf{L}$ has a small extension that is also efficiently constructible. By efficiently constructible, we mean that given n we can construct an extension of $P(\mathbf{L}(n))$ requiring $\text{poly}(n)$ bits to describe in time $\text{poly}(n)$. What do we gain in this case?

Proposition 5.2.4. *Let $\mathbf{L}$ have an efficiently constructible extended formulation of size $s(n)$. Then, mem$(\mathbf{L})$ and opt$(\mathbf{L})$ can be solved in time $\text{poly}(s(n) + n)$.*

Proof. Let $Q = \{(\mathbf{x}, \mathbf{y} \mid \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \leqslant \mathbf{c}\}$ be an EF of $P(\mathbf{L}(n)$. Checking whether a given $\mathbf{x}^*$ belongs to $\mathbf{L}$ or not can be done by checking the feasibility of $Q \cap \{\mathbf{x} = \mathbf{x}^*\}$. The problem opt$(()\mathbf{L})$ is also easily solved by standard Linear Programming. □

Proposition 5.2.5. *Let $\mathbf{L}$ have an efficiently constructible extended formulation of size $s(n)$. Then, enum$(\mathbf{L})$ and sample$(()\mathbf{L})$ can be solved in time $\text{poly}(s(n) + n + h(n))$ where $h(n)$ is the number of strings of length n.*

Proof. This follows from the fact that vertices of a zero-one polytope can be enumerated in strongly polynomial time [14]. For the method of Bussiek and Lübbecke to work, we only need to check whether a given face of the zero-one polytope is empty or not.

For the sampling problem one can just start the enumeration and select any of the output string uniformly at random on the fly. Note that one does not have to store the entire output to select an element with uniform distribution. □

Proposition 5.2.6. *The problem count$(\mathbf{L})$ may be #P-hard even if $P(\mathbf{L}(n))$ has small size.*

Proof. This follows from the fact that the polytope of perfect matchings of bipartite graphs has small description, but counting the number of perfect matchings in bipartite graphs is #P-hard. □

5.3 CLOSURE PROPERTIES

Now we discuss the closure properties of the class $\mathcal{CF}$ with respect to some common operations on formal languages. The operations that we consider are as follows.

- **Complement** : $\overline{L} = \{x \mid x \notin L\}$
- **Union** : $L_1 \cup L_2 = \{x \mid x \in L_1 \vee x \in L_2\}$
- **Intersection** : $L_1 \cap L_2 = \{x \mid x \in L_1 \wedge x \in L_2\}$
- **Set difference** : $L_1 \setminus L_2 = \{x \mid x \in L_1 \wedge x \notin L_2\}$
- **Concatenation** : $L_1 L_2 = \{xy \mid x \in L_1 \wedge y \in L_2\}$
- **Kleene star** : $L^* = L \cup LL \cup LLL \cup LLLL \cup \ldots$

Proposition 5.3.1. *$\mathcal{CF}$ is not closed under taking complement.*

Proof. Consider the family $\mathcal{CNF}$-$\mathcal{CERT}$ of polytopes discussed in Subsection 3.3.2. Let $\mathbf{\Phi}$ be the family of boolean formulae corresponding to this polytope family, and let L be the language consisting of all binary strings that correspond to a vertex of some polytope in $\mathcal{CNF}$-$\mathcal{CERT}$. That is, L is the language of all cut vectors of K_n for $n \in \mathbb{N}$, viewed as binary strings.

Now consider the family of DNF formulae $\overline{\mathbf{\Phi}} = \{\overline{\varphi} \mid \varphi \in \mathbf{\Phi}\}$. Notice that the language of all certifying assignments of formulae in $\overline{\mathbf{\Phi}}$ is precisely $\overline{L}$.

We see that $\mathcal{P}(\overline{L}) = \mathcal{DNF}$-$\mathcal{CERT}$ while $\mathcal{P}(L) = \mathcal{CNF}$-$\mathcal{CERT}$. Propositions 3.3.11 and 3.3.13 state that $\text{xc}(\mathcal{DNF}\text{-}\mathcal{CERT}) = \text{poly}(n)$ while $\text{xc}(\mathcal{CNF}\text{-}\mathcal{CERT}) \neq \text{poly}(n)$. Therefore $\overline{L} \in \mathcal{CF}$ and $L \notin \mathcal{CF}$. □

Proposition 5.3.2. *$\mathcal{CF}$ is closed under taking union.*

Proof. Let L_1 and L_2 be two languages. Then, $\text{xc}(L_1 \cup L_2) \leqslant \text{xc}(L_1) + \text{xc}(L_2)$ (cf. Proposition 3.2.5). □

Proposition 5.3.3. *$\mathcal{CF}$ is not closed under taking intersection.*

Proof. Let L_1 be a language such that a string $x \in L_1$ if and only if it satisfies the following properties.

- $|x| = (n+1)\binom{n}{2}$ for some natural number n, and
- $x_{ij(n+1)} = x_{iji} \oplus x_{ijj}$ if the characters are indexed as x_{ijk} with $1 \leqslant i < j \leqslant n$, $1 \leqslant k \leqslant n+1$.

We claim that $\text{xc}(L_1) = \mathcal{O}(n^3)$. Indeed $P\left(L_1\left((n+1)\cdot\binom{n}{2}\right)\right)$ is the product of polytopes

$$P_{ij} = \left\{x \in \{0,1\}^{n+1} \mid x_{n+1} = x_i \oplus x_j\right\}$$

for $1 \leqslant i < j \leqslant n$ and $\text{xc}(P_{ij}) = \mathcal{O}(n)$ (cf. Example 3.1.26).

Now let L_2 be a language such that a string $x \in L_2$ if and only if it satisfies the following properties.

- $|x| = (n+1)\binom{n}{2}$ for some natural number n, and
- $x_{i_1 j_1 k} = x_{i_2 j_2 k}$ for all $k \in [n], i \neq j \in [n]$

Each polytope $P\left(L_1\left((n+1)\cdot\binom{n}{2}\right)\right)$ is just an embedding of $\square_{n+\binom{n}{2}}$ in $\mathbb{R}^{(n+1)\binom{n}{2}}$ and therefore, $\text{xc}(L_2) = \mathcal{O}(n^2)$.

Finally, observe that for $m = (n+1)\binom{n}{2}$ the polytope $P((L_1 \cap L_2)(m))$ when projected to the coordinates labelled $x_{ij(n+1)}$ is just the polytope $\text{CUT}^{\square}_n$ (cf. Proposition 3.3.3). Therefore, $\text{xc}(L_1 \cap L_2) = 2^{\Omega(n)}$ and even though $L_1, L_2 \in \mathcal{CF}$, the intersection $L_1 \cap L_2 \notin \mathcal{CF}$. □

Proposition 5.3.4. *$\mathcal{CF}$ is not closed under taking set difference.*

Proof. The complete language $\{0,1\}^*$ clearly belongs to $\mathcal{CF}$. For any language $\mathbf{L}$ we have $\overline{\mathbf{L}} = \{0,1\}^* \setminus \mathbf{L}$. If $\mathcal{CF}$ were closed under taking set-difference, it would also be closed under taking complements. But as pointed out in Proposition 5.3.1, it is not. □

Proposition 5.3.5. *$\mathcal{CF}$ is closed under concatenation.*

Proof. $P(\mathbf{L}_1\mathbf{L}_2(n))$ is the union of the polytopes $P(\mathbf{L}_1(i)) \times P(\mathbf{L}_2(n-i))$ for $i \in [n]$. Therefore, using Propositions 3.2.3 and 3.2.5 we have that $\mathrm{xc}(\mathbf{L}_1\mathbf{L}_2) \leqslant n(\mathrm{xc}(\mathbf{L}_1) + \mathrm{xc}(\mathbf{L}_2))$. □

Proposition 5.3.6. *$\mathcal{CF}$ is closed under taking Kleene star.*

Proof. Let $\mathbf{L} \in \mathcal{CF}$. For $0 \leqslant k \leqslant n$, consider the polytope P_k defined as

$$P_k := \mathrm{conv}\left(\left\{ \begin{pmatrix} e^{n+1}_{i+1} \\ 0^i \\ x \\ 0^{n-i-k} \\ e^{n+1}_{i+|x|+1} \end{pmatrix} \in \{0,1\}^{3n+2} \;\middle|\; \begin{array}{ll} & x \in \mathbf{L} \\ \wedge & |x| = k \\ \wedge & 0 \leqslant i \leqslant n-k \end{array} \right\}\right)$$

Define $P := \cup_{j=0}^{n} P_j$. Then, $\mathrm{xc}(P) \leqslant \sum_{k=0}^{n} \mathrm{xc}(P_k) \leqslant \sum_{k=0}^{n} (n\,\mathrm{xc}(P(\mathbf{L}(k)))) \leqslant \mathcal{O}(n^2\,\mathrm{xc}(\mathbf{L}))$.

Let S_0 be the face of P defined by the first n coordinates being 0 and the $(n+1)$-th coordinate being 1. Construct S_{i+1} by taking the glued product of S_i with P over the last $n+1$ coordinates of S_i and the first $n+1$ coordinates of Q.

Take the face R of S_n defined by the last n coordinates being 0 and the $(n+1)$-th penultimate coordinate being 1. Then, R is an EF for $P(\mathbf{L}^*(n))$. Moreover, $\mathrm{xc}(R) \leqslant \mathrm{xc}(S_n) \leqslant (n+1)\,\mathrm{xc}(P) \leqslant \mathcal{O}(n^3\,\mathrm{xc}(\mathbf{L}))$.

Therefore, $\mathrm{xc}(\mathbf{L}^*) = \mathcal{O}(n^3\,\mathrm{xc}(\mathbf{L}))$ and $\mathbf{L}^* \in \mathcal{CF}$. □

6 ONE-PASS LANGUAGES

6.1 ONLINE TURING MACHINES

An online Turing machine is a two tape Turing machine where one of the tapes stores the input and can be read only from left to right. The second tape is the work tape and the machine can read and write freely on it and the head is free to move in any direction. The space consumed on the worktape in the worst case is the measure of space complexity.

6.1.1 *History*

Online Turing machines that require only logarithmic space on the work tape were considered by Hartmanis, Immerman, and Mahaney [36] to restrict the power of reductions between two problems. Traditionally, for establishing equivalence of problems arbitrary polynomial time reduction between them is allowed. In the light of the fact that we do not know whether PTIME is different from NP, such reductions may be misleading. In fact, even the possibly smaller class of LOGSPACE problems are not known to be different from the class NP and so even logspace reductions between problems may be misleading about their true complexity.

It is known that one-pass logspace Turing machines can only accept regular languages [57, 59] and therefore if two problems are reducible to each other using only one-pass logspace reduction, they are equivalent in a stronger sense.

6.1.2 *Determinism vs. Non-determinism*

For online Turing machines requiring at least logarithmic space, nondeterminism allows provably stronger machines. Non-regular languages can be accepted by non-deterministic machines using logarithmic space while any one-pass deterministic logspace Turing machine can only accept regular languages [59].

6.2 EXTENSION COMPLEXITY OF ONE-PASS LANGUAGES

Definition 6.2.1. The complexity class $k\text{-NSPACE}(s(n))$ is the class of languages accepted by a k-pass non-deterministic Turing machines using space $s(n)$. Similarly, the complexity class $k\text{-DSPACE}(s(n))$ is the class of languages accepted by a k-pass deterministic Turing machine using space $s(n)$.

What is the extension complexity of any language in this class? Before we answer this we note the following.

Proposition 6.2.2. $\mathbf{L} \in \text{k-NSPACE}(s(n)) \implies \mathbf{L} \in \text{1-NSPACE}(ks(n))$.

Proof. Let M_n be the Turing machine that accepts strings of length n. We will simulate M_n using a multi-tape single pass nondeterministic Turing machine called the simulator S. S is supplied with $p(n)$ work tapes. S starts by guessing the initial work state of M_n at the start of i-th pass and writing them on the i-th work tape. S then simulates (using extra space on each work tape) each of the passes independently starting from their respective initial configuration. Once the entire input has been scanned, the simulator verifies that the work space of M_n on the i-th tape at the end of the pass matches the guess for the initial content for the $(i+1)$-th tape. S will accept only if the last tape is in an accepting state.

To store the content of work tape and the current state, S needs $s(n) + o(s(n))$ space for each pass. Thus S uses a single pass and total space of $p(n)s(n)(1 + o(1))$. By Proposition 6.2.8 the extension complexity of the strings accepted by M_n is then $2^{\mathcal{O}(p(n)s(n))}n$. □

Thus for our purposes it suffices to restrict our attention to single pass TMs. In the next subsection we describe the polytope associated with walks in a directed graph, that will help us bound the extension complexity of such languages.

6.2.1 *Walks in directed graphs*

Definition 6.2.3. Let $D = (V, A)$ be a directed graph with every edge labeled either zero or one. Consider two nodes $u, v \in V$ and a walk ω of length n from u to v. The *signature* of ω – denoted by σ_ω – is the sequence of edge labels along the walk ω. The node u is called the *source* of the walk and the node v the *destination*.

Definition 6.2.4. Consider the convex hull of all zero-one vectors of the form (u, σ, v) where u and v are indices of two nodes in D and σ is the signature of some walk of length n from u to v. This polytope – denoted by $P_{\text{markov}}(D, n)$ – is called the *Markovian polytope of* D.

Proposition 6.2.5. *Let $D = (V, A)$ be directed graph (possibly with self-loops and multiple edges) with every edge labeled either zero or one. Then, $P_{\text{markov}}(D, n)$ has extension complexity at most $2|V| + |A| \cdot n$.*

Proof. Let us encode every vertex of D with a zero-one vector of length V such that the unit vector e_i represents vertex i.

Define polytope $P_{\text{trans}} \subset \{0,1\}^{|V|+1+|V|}$ with $(a, z, b) \in \{0,1\}^{|V|+1+|V|}$ a vertex of P_{trans} if and only if it encodes a possible transition in D. That is, a and b encode vertices of V, and the coordinate z represents the label of the edge following which one can move from a to b. Since P_{trans} has at most $|E|$ vertices $\text{xc}(P_{\text{trans}}) \leqslant |E|$ (cf: Proposition 3.1.19).

Let P_0 be the convex hull of (i, e_i) for $i \in V$ and P_f be the convex hull of (e_i, i) for $i \in V$. Observe that the two polytopes are the same except for relabeling of coordinates. Also, $\text{xc}(P_0) = \text{xc}(P_f) \leqslant |V|$.

Let $P_1 = P_{\text{trans}}$. For $2 \leqslant i \leqslant n$, construct the polytope P_i by glueing the last $|V|$ coordinates of P_{i-1} with the first $|V|$ coordinates of P_{trans}. By Proposition 3.2.4 we have that $\text{xc}(P_n) \leqslant |E| \cdot n$.

Finally, let P be the polytope obtained by glueing last $|V|$ coordinates of P_0 with the first $|V|$ coordinates of P_n, and then glueing the last $|V|$ vertices of the result with the first $|V|$ coordinates of P_f. Note that $xc(P) \leqslant 2|V| + |E| \cdot n$.

To complete the proof, notice that P is an extended formulation for $P_{markov}(D, n)$. In particular, projecting out every coordinate except the ones corresponding to the source node in P_0, the ones corresponding to the destination node in P_f, and ones that correspond to the z coordinates in all the copies of P_{trans} produces exactly the vertices of $P_{markov}(D, n)$. The z-coordinate corresponding to the i-th copy of P_{trans} corresponds to the i-th index of signatures in the vectors in $P_{markov}(D, n)$. □

6.2.2 *Extension complexity of single-pass machines*

Definition 6.2.6. The *configuration graph* for input of length n for a given one-pass Turing machine (deterministic or non-determinisitic) is constructed as follows. For each fixed n, consider the directed graph whose nodes are marked with a label consisting of $s(n) + \lceil \log(s(n)) \rceil$ characters. The labels encode the complete configuration of the Turing machine: the content of the worktape and head position on the worktape. We make directed edges between two nodes u and v if the machine can reach from configuration u to configuration v by a sequence of transitions with exactly one input bit read in between. The directed edge is labeled by the input bit read during this sequence of transition.

Finally, we add two special nodes: a start node with a directed edge to each possible starting configuration of the machine, and a finish node with a directed edge from each possible accepting configuration. Each of these directed edges are labeled by zero.

Proposition 6.2.7. *The configuration graph for input of length n for a one-pass Turing machine has $\mathcal{O}(2^{s(n)}s(n))$ nodes. If the Turing machine is non-deterministic, this graph has $\mathcal{O}(4^{s(n)}(s(n))^2)$ edges. If the Turing machine is deterministic then this graph has $O(2^{s(n)}s(n))$ edges.*

Proof. The bound for number of nodes is clear from the construction of the configuration graph. We can have at most two transition edges between any two (possibly non-distinct) nodes: one corresponding to reading a zero on the input tape, and one corresponding to reading a one. Therefore, asymptotically the configuration graph can have at most square of the number of nodes.

For deterministic Turing machine, each node in the configuration graph has exactly two outgoing edges (possibly to the same node). Therefore the number of edges is asymptotically the same as the number of vertices. □

Now Proposition 6.2.5 can be used to bound the extension complexity of language accepted by one-pass machines.

Proposition 6.2.8. *Let $L \in$ 1-NSPACE$(s(n))$. Then,*

$$xc(L) = \mathcal{O}(4^{s(n)}(s(n))^2 \cdot n).$$

Proof. Let $\mathbf{L} \in$ 1-NSPACE$(s(n))$ be a language. That is, there exists a Turing machine that when supplied with a string on the one-way input tape uses at most $s(n)$ cells on the worktape, makes a single pass over the input and then accepts or rejects the input. If the input string is in $\mathbf{L}$, some sequence of non-deterministic choices lead the machine to an accepting state, otherwise the machine always rejects.

The length-n strings that are accepted by such a Turing machine correspond exactly to the signatures of length $n+2$ walks on the corresponding configuration graph D. The first and the last character of these strings is always zero. Therefore, an extended formulation for $P(\mathbf{L}(n))$ is obtained by taking the face of $P_{\text{markov}}(D, n+2)$ corresponding to walks that start and the start node and finish at the finish node. By Proposition 6.2.5 $P_{\text{markov}}(D, n+2)$ has extension complexity $\mathcal{O}(4^{s(n)}(s(n))^2 \cdot n$, and by Proposition 3.1.23 so does the desired face. □

If $\mathbf{L}$ is accepted by a one-pass deterministic TM then one can do better because the configuration graph has fewer edges.

Proposition 6.2.9. *Let* $\mathbf{L} \in$ 1-DSPACE$(s(n))$. *Then,*

$$\text{xc}(\mathbf{L}) = \mathcal{O}(2^{s(n)}s(n) \cdot n).$$

6.2.3 *Extensions for multiple-pass machines*

Proposition 6.2.10. *Let* $\mathbf{L} \in$ p-NSPACE$(s(n))$. *Then,*

$$\text{xc}(\mathbf{L}) = 2^{\mathcal{O}(p(n)s(n))}n.$$

Proof. This follows immediately from Propositions 6.2.2 and 6.2.8. □

Proposition 6.2.11. *Let* $\mathcal{M}$ *be a (not necessarily uniform) family of deterministic online Turing machines. Let the number of passes and the space used by the family be bounded by functions,* $p(n), s(n)$ *respectively. Let* $\mathbf{L}(M)$ *be the language accepted by* M. *Then,* $\text{xc}(L(\mathcal{M})) \leqslant 2^{\mathcal{O}(p(n)s(n))}n$.

Proposition 6.2.12. *If* $\mathbf{L}$ *is accepted by a fixed-pass non-deterministic logspace Turing machine then* $\mathbf{L} \in \mathcal{CF}$.

We end this section with the following remark. For a language to be compact (that is, to have polynomial extension complexity), it is sufficient to be accepted by an online Turing machine (deterministic or not) that requires only logarithmic space. However, this requirement is clearly not necessary. This can be proved by contradiction: Suppose that the condition is necessary. Then the class of compact languages must be closed under taking intersection. (Simply chain the two accepting machines and accept only if both do). Since we have already established (cf. Proposition 5.3.3) that the class of compact languages is not closed under taking intersection, we have a contradiction.

6.3 APPLICATIONS

6.3.1 *Streaming lower bounds*

Reading Proposition 6.2.10 in reverse readily yields lower bounds in the streaming model of computation. We illustrate this by an example.

Example 6.3.1. We know that the perfect matching polytope of the complete graph K_n has extension complexity $2^{\Omega(n)}$. Any $p(n)$-pass algorithm requiring space $s(n)$, that correctly determines whether a given stream of $\binom{n}{2}$ is the characteristic vector of a perfect matching in K_n, must have $p(n)s(n) = \Omega(n)$. This bound applies even to nondeterministic algorithms.

In fact Proposition 6.2.5 provides an even stronger lower bound.

Definition 6.3.2. Let $\mathbf{L} \subseteq \{0,1\}^n$ be a language. $\mathbf{L}$ is said to be online μ-magic if there exists a Turing machine T that accepts $\mathbf{L}$ with the following oracle access. On an input of length n on the one-way input tape, the machine T scans the input only once. T may prepare its working tape to describe any well-formed function $f : \{0,1\}^{\mu(n)} \to \{0,1\}^{m u(n)}$ and a particular input x and invoke the oracle that changes the contents of the work-tape to $f(x)$. The machine must always reject strings not in $\mathbf{L}$. For strings in $\mathbf{L}$ there must be some possible execution resulting in accept.

Notice that even the working of such a machine can be encoded in terms of the configuration graph where the transitions may depend arbitrarily but in a well-formed way on the contents of the work-tape.

Proposition 6.3.3. *If the set of characteristic vectors of perfect matchings in K_n are accepted by an online μ-magic Turing machine, then $\mu(n) = \Omega(n)$.*

Thus we see that extension complexity lower bounds highlight deep limitations of the streaming model: even powerful oracles do not help solve in sublinear space problems that are LOGSPACE solvable if the one-way restriction on the input is removed.

6.3.2 *Upper bounds from online algorithms*

Parity Polytope

As an example, consider the language containing strings where the last bit indicates the parity of the previous bits. This language can be accepted by a deterministic logspace turing machine requiring a single pass over the input and a single bit of space. Therefore, the parity polytope has extension complexity $O(n)$.

The parity polytope is known to have extension complexity at most $4n - 4$ [16].

Integer Partition Polytope

For non-negative integer n the Integer Partition Polytope, P_n, is defined as

$$P_n := \text{conv}\{x \in \mathbb{Z}_+^n | \sum_{k=1}^{n} kx_k = n\}.$$

It is known that $\text{xc}(P_n) = \mathcal{O}(n^3)$ [50].

Consider the polytope in $\mathbb{R}^{\lceil \log n \rceil \times n}$ that encodes each x_i as a binary string. For example, for $n = 4$ the vector $(2, 1, 0, 0)$ is encoded as $(1, 0, 0, 1, 0, 0, 0, 0)$. This polytope is clearly an extended formulation of the Integer Partition Polytope. Call this polytope BIPP_n. The following single pass deterministic algorithm accepts a string $(x_1, x_2, \ldots, x_n) \in \{0, 1\}^{\lceil \log n \rceil \times n}$ if and only if the string represents a vertex of BIPP_n.

```
Data : Binary string of length n⌈log n⌉
Result : Accept if the input encodes a vertex of the BIPP_n
s = 0; i = 0; l = 0;
while i < n do
    b =read_next_bit;
    if (s + (i + 1)2^l b) > n then
        reject;
    else
        s = (s + (i + 1)2^l b);
        l = (l + 1)%⌈log n⌉;
        if l == 0 then
            i++;
        end
    end
end
if s == n then
    accept;
else
    reject;
end
```

Algorithmus 1 : One pass algorithm for accepting vertices of BIPP_n.

The above algorithm together with Proposition 6.2.9 shows that $\text{xc}(\text{IPP}_n) \leqslant \text{xc}(\text{BIPP}_n) \leqslant \mathcal{O}(n^3 \log^2 n)$.

Knapsack Polytopes

Let $(a, b) = (a_1, a_2, \ldots, a_n, b)$ be a given sequence of (non-negative) integers. The Knapsack polytope $\text{KS}(a, b)$ is defined as

$$\text{KS}(a, b) := \{x \in \{0, 1\}^n | \sum_{i=1}^{n} a_i x_i \leqslant b\}.$$

The Knapsack polytope is known to have extension complexity super-polynomial in n. However, optimizing over $\text{KS}(a, b)$ can be done via dynamic programming in time $O(nW)$ where W is the largest number among $a_1, \ldots, a_n, b$.

Suppose the integers a_i, b are arriving in a stream with a bit in between indicating whether $x_i = 0$ or $x_i = 1$. With a space of W bits, an online Turing machine can store and update $\sum_{i=1}^{n} a_i x_i$. At the end, it can subtract b and accept or reject depending on whether the result is 0 or not. Any overflow during intermediate steps can be

used to safely reject the input. Therefore, the extension complexity of the Knapsack polytope is $O(nW \log W)$. Note however the extension obtained this way is actually an extended formulation of a polytope encoding all the instances together with their solutions.

*Languages in co-*DLIN

Let $\mathbf{L}$ be a language generated by a determinisitic linear grammar [37]. The following result was proved by Babu, Limaye, and Varma [6].

Proposition 6.3.4 (BLV)**.** *Let* $\mathbf{L} \in$ DLIN*. Then there exists a probabilistic one-pass streaming algorithm using* $\mathcal{O}(\log n)$ *space that accepts every string in* $\mathbf{L}$ *and rejects every other string with probability at least* $1/n^c$.

Using the above algorithm together with Proposition 6.2.10 we get the following.

Proposition 6.3.5. *If* $\mathbf{L} \in$ DLIN*, then* $\overline{\mathbf{L}} \in \mathcal{CF}$.

Part III

VARIATIONS

A novice was trying to fix a broken Lisp machine by turning the power off and on.

Knight, seeing what the student was doing, spoke sternly: "You cannot fix a machine by just power-cycling it with no understanding of what is going wrong."

Knight turned the machine off and on. The machine worked.

— Tom Knight and the Lisp Machine [53]

7

FPT EXTENDED FORMULATIONS

7.1 PARAMETERIZED EXTENSION COMPLEXITY

Most of the bounds seen so far (specially in Chapter 4) were only in terms of the ambient dimension n. For example the perfect matching polytope $EP(K_n)$ for the complete graph K_n was seen to have extension complexity $2^{\Theta(n)}$. What about other graphs on n vertices? It was shown by Barahona [7] that for planar graphs the perfect matching polytope has polynomial extension complexity. Gerard [32] showed that if G is a n-vertex graph of genus g, then $xc(EP(G)) \leqslant n^{\mathcal{O}(g)}$.

In this chapter we shall see results of similar type. We have already seen some results on parametrized extension complexity although it was not made explicit so far.

Example 7.1.1. Let each 0/1 polytope $P \in \mathbb{R}^n$ be parameterized by the minimum number of clauses in any DNF formula φ such that $P = SAT(\varphi)$. Denoting this parameter by μ, one can use Proposition 3.3.13 to conclude that $xc(\mathbb{ZERO}-\mathbb{ONE}) = \mathcal{O}(\mu n)$.

In other words, the clan of 0/1 polytopes has polynomial extension complexity when *parameterized* by the size of the smalled DNF formula describing the vertex set of the polytopes.

Definition 3.1.5 can be used to formally speak about parametrized extension complexity of a family (and therefore a clan) of polytopes. Recall that for a family of polytopes there is exactly[1] one polytope $P_n \subset \mathbb{R}^n$ in the family for each $n \in \mathbb{N}$.

Definition 7.1.2. Let $\mathbb{P}$ be a clan of polytopes and $\kappa : \mathbb{P} \to \mathbb{N}$ be a parameter. Let $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a function. We will say that the *parameterized extension complexity* of $\mathbb{P}$ is $g(\kappa, n)$ if for every polytope $P \in \mathbb{P}$ such that $P \subseteq \mathbb{R}^n$ we have that $xc(P) = g(\kappa(P), n)$.

We will say that extension complexity of $\mathbb{P}$ or $xc(\mathbb{P})$ is FPT if there exists $f : \mathbb{N} \to \mathbb{N}$ and a constant c such that $xc(\mathbb{P}) \leqslant f(\kappa)n^c$, and if no such function or constant exist then we will say that the extension complexity of $\mathbb{P}$ is not FPT.

As described in Subsection 3.1.1, we will mostly use asymptotic descriptions of functions whenever we use the above definition.

Example 7.1.3. The extension complexity of the clan $\mathbb{ZERO}-\mathbb{ONE}$ parameterized by the size of the smallest DNF formula describing the vertices is FPT.

1 In case there is no such polytope explicitely present, the empty polytope can play the desired role.

7.2 THE INDEPENDENT SET POLYTOPE

The k-independent set problem asks one to decide whether a graph has independent set of size at most k. When parameterized by k this problem is W[1]—hard but is fixed-parameter tractable for graphs of bounded expansion. The corresponding polytope has analogous behavior with respect to the extension complexity [30].

7.2.1 *The k-independent set polytope*

Definition 7.2.1. Let $G = (V, E)$ be a graph on n vertices. The $k^{\leqslant}$-*independent set polytope of* G – denoted by $\mathrm{STAB}_{k^{\leqslant}}(G)$ – is defined to be the convex hull of the independent sets of G that have size at most k.

Alternatively, one could define the $k^{=}$-*independent set polytope of* G – denoted by $\mathrm{STAB}_{k^{=}}(G)$ – to be the convex hull of all independent sets of size exactly equal to k.

As far as extension complexity is concerned, either definition can be used to define the clan of stable set polytopes parameterized by the size of the independent set. This is because the extension complexities of the two polytopes defined above are within a polynomial factor of each other.

Proposition 7.2.2.

$$\mathrm{xc}(\mathrm{STAB}_{k^{=}}(G)) \leqslant \mathrm{xc}(\mathrm{STAB}_{k^{\leqslant}}(G)) \leqslant \sum_{i=0}^{k} \mathrm{xc}(\mathrm{STAB}_{i^{=}}(G)).$$

Proof. Clearly, $\mathrm{STAB}_{k^{=}}(G)$ is a face of $\mathrm{STAB}_{k^{\leqslant}}(G)$. Therefore, we have that $\mathrm{xc}(\mathrm{STAB}_{k^{=}}(G)) \leqslant \mathrm{xc}(\mathrm{STAB}_{k^{\leqslant}}(G))$ (cf. Proposition 3.1.23).

On the other hand, $\mathrm{STAB}_{k^{\leqslant}}(G) = \mathrm{conv}(\bigcup_{i=1}^{k} \mathrm{STAB}_{i^{=}}(G))$, and therefore $\mathrm{xc}(\mathrm{STAB}_{k^{\leqslant}}(G)) \leqslant \sum_{i=0}^{k} \mathrm{xc}(\mathrm{STAB}_{i^{=}}(G))$ by Proposition 3.2.5. □

Therefore any bounds (whether lower or upper) that are valid for $\mathrm{xc}(\mathrm{STAB}_{k^{=}}(G))$ are also asymptotically valid for $\mathrm{xc}(\mathrm{STAB}_{k^{\leqslant}}(G))$. To simplify our notations, instead of either $\mathrm{STAB}_{k^{\leqslant}}(G)$ or $\mathrm{STAB}_{k^{=}}(G)$, we will use $\mathrm{STAB}_{k}(G)$. In the rest of the chapter $\mathrm{STAB}_{k}(G)$ represents $\mathrm{STAB}_{k^{=}}(G)$ but with minor adjustments the same arguments can be made using $\mathrm{STAB}_{k^{\leqslant}}(G)$.

Buchanan [13] showed that the extension complexity of the stable set polytopes parametrized by the treewidth τ of the underlying graph is $2^{\mathcal{O}(\tau)}n$.

Proposition 7.2.3. *Let* $\mathbb{STAB}$ *be the clan of stable set polytopes parametrized by the treewidth of the underlying graph. That is, let* $\tau : \mathbb{STAB} \to \mathbb{N}$ *be defined as* $\tau(P) = \min\{\tau(G) \mid P = \mathrm{STAB}(G)\}$, *where* $\tau(G)$ *is the treewidth of graph* G. *Then,* $\mathrm{xc}(\mathbb{STAB}) \leqslant 2^{\mathcal{O}(\tau)} \cdot n$.

Proof. See [13], Theorem 3. □

Buchanan also asked if the extension complexity of the stable set polytope parametrized by the size of the independent set is FPT. We next present the answer to this question (in the negative). The proof relies on encoding cuts of the complete graph on roughly $k \log n$ vertices as independent sets of size k^2 in another graph whose size is not too big. The result then follows from the fact that the cut polytope of the initial graph has size $\Omega(n^k)$.

7.2.2 *Paired Local-Cut Graphs*

Given positive integers k and n, we define a graph called a *Paired Local-Cut Graph* and denoted by $\mathrm{PLC}(k, n)$.

First we create $k2^{\lfloor \log n \rfloor}$ vertices labeled with tuples (i, S) for $i \in [k]$ and $S \subseteq [\lfloor \log n \rfloor]$. These vertices will be called *cut vertices*. Then we create $2\binom{k}{2}2^{2\lfloor \log n \rfloor}$ vertices labeled with tuples (i, j, S_1, S_2) where $1 \leqslant i \neq j \leqslant k$ and $S_1, S_1 \subseteq [\lfloor \log n \rfloor]$. These vertices will be called *pairing vertices*.

We add edges to these vertices of $\mathrm{PLC}(k, n)$ as follows. For each fixed $i \in [k]$ we add the edges between all cut nodes that have labels (i, S). Furthermore, for each fixed pair $i, j \in [k]$ we add the edges between all pairing nodes that have labels (i, j, S_1, S_2). Finally, let u be a cut vertex labeled (i, S) and let v be a pairing vertex labeled (j_1, j_2, S_1, S_2). If $i = j_1$ but $S \neq S_1$ we add edge uv. Symmetrically, if $i = j_2$ but $S \neq S_2$ we add edge uv.

For ease of exposition we will identify vertices of $\mathrm{PLC}(k, n)$ with their labels whenever convenient.

Proposition 7.2.4. *The number of vertices of the graph* $\mathrm{PLC}(k, n)$ *equals* $2\binom{k}{2}2^{2\lfloor \log n \rfloor} + k2^{\lfloor \log n \rfloor} \leqslant (kn)^2$.

Proposition 7.2.5. *Let* (i, S) *and* (j_1, j_2, S_1, S_2) *be two vertices of* $\mathrm{PLC}(k, n)$ *that are not joined by an edge. If* $i = j_1$ *then* $S = S_1$*, and if* $i = j_2$ *then* $S = S_2$.

This together with the next proposition will ensure that in any independent set I of $\mathrm{PLC}(k, n)$ that has size k^2, every index $i \in [k]$ can be uniquely associated with a subset $S_i \subseteq \lfloor \log n \rfloor$.

Proposition 7.2.6. *Let* I *be an independent set in* $\mathrm{PLC}(k, n)$*. Then,* $|I| \leqslant k^2$*. Moreover, an equality holds if and only if* I *contains exactly one cut vertex for each* $1 \leqslant i \leqslant k$ *and exactly one pairing vertex for each* $1 \leqslant i \neq j \leqslant k$.

Proof. Clearly, the set I can contain at most k cut vertices – at most one vertex (i, S_i) for each $1 \leqslant i \leqslant k$. Also, set I can contain at most $2\binom{k}{2} = k^2 - k$ pairing vertices – at most one vertex (i, j, S_i, S_j) for each ordered pair $1 \leqslant i, j \leqslant k$. □

The vertices of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$ are related to the vertices of the *polytope* $\mathrm{CUT}^{\square}(K_r)$ where $r = k\lfloor \log n \rfloor$, in the following way. Denote the vertices and edges of K_r by V_r and E_r respectively, and group the vertices of K_r into k groups, each of size $\lfloor \log n \rfloor$. Label the vertices v_j^i where $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant \lfloor \log n \rfloor$. Finally, order the vertices lexicographically according to their labels.

A cut vector of K_r – corresponding to a cut C – is a $0/1$ vector of length $\binom{r}{2}$ whose coordinates correspond to whether an edge of K_r is in the cut C or not. The edges of K_r are labeled with pairs (i_1, j_1, i_2, j_2) where $1 \leqslant i_1, i_2 \leqslant k$; $1 \leqslant j_1, j_2 \leqslant \lfloor \log n \rfloor$, and $(i_1, j_1) \leqslant (i_2, j_2)$ lexicographically. So, if z is a cut vector corresponding to a given cut $C \subset E_r$, then $z_{i_1,j_1,i_2,j_2} = 1$ if and only if the edge (i_1, j_1, i_2, j_2) is in C. $\mathrm{CUT}^{\square}(K_r)$ is the convex hull of all such cut vectors.

Similarly, an independent-set vector of $\mathrm{PLC}(k, n)$ – corresponding to an independent set I – is a $0/1$ vector of length $2\binom{k}{2}2^{2\lfloor \log n \rfloor} + k2^{\lfloor \log n \rfloor}$ (cf. Prop. 7.2.4) whose coordinates correspond to whether the corresponding vertex is in I or not. Recall that the cut vertices of $\mathrm{PLC}(k, n)$ are labeled with a pair consisting of an index from $[k]$, and a subset of $[\lfloor \log n \rfloor]$. Also, the pairing vertices of $\mathrm{PLC}(k, n)$ are labeled with a tuple consisting of two indices from $[k]$ and two subsets of $[\lfloor \log n \rfloor]$.

Let $\mathcal{C}$ be the set of all cuts in K_r, and let $\mathbf{I}$ be the set of all independent sets of size k^2 in $\mathrm{PLC}(k, n)$. Any cut $C \in \mathcal{C}$ creates a bipartition $(S, \overline{S})$ of the vertices of K_r. Recall that the vertices of K_r have been split in k groups. The partition $(S, \overline{S})$ thus induces a partition $(S_i, \overline{S}_i)$ within each of these groups.

Proposition 7.2.7. *For every pair of natural numbers (k, n) and $r = k\lfloor \log n \rfloor$ it holds that $\mathrm{CUT}^{\square}(K_r)$ is a projection of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$.*

Proof. See [30], Lemma 3.4 (Appendix F). □

A lower bound on the extension complexity of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$ immediately follows.

Proposition 7.2.8. *There exists a constant $c' > 0$ such that for $k, n \in \mathbb{N}$,*

$$\mathrm{xc}\left(\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))\right) \geqslant n^{c'k}.$$

Proof. By Proposition 7.2.7, $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$ is an extended formulation of $\mathrm{CUT}^{\square}(K_r)$ with $r = k\lfloor \log n \rfloor$. So any extended formulation of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$ is also an extended formulation of $\mathrm{CUT}^{\square}(K_r)$. By proposition 3.3.6, $\mathrm{xc}\left(\mathrm{CUT}^{\square}(K_r)\right) \geqslant 2^{\Omega(r)}$. Therefore,

$$\mathrm{xc}\left(\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))\right) \geqslant \mathrm{xc}\left(\mathrm{CUT}^{\square}(K_r)\right) \geqslant 2^{\Omega(r)} \geqslant n^{c'k}$$

for some constant $c' > 0$. □

We can now easily conclude that the parameterized extension complexity of $\mathbb{STAB}$ parameterized by the size of the independent sets is not FPT.

Proposition 7.2.9. *There does not exist any function $f : \mathbb{N} \to \mathbb{R}$ such that $\mathrm{xc}(\mathrm{STAB}_k(G)) \leqslant f(k) \cdot n^{O(1)}$ for all natural numbers k and all graphs G on n vertices.*

Proof. Suppose, on the contrary, that such a function f exists. That is, there is a constant c such that for every pair of natural numbers (ℓ, m) and for all m-vertex graphs G it holds that $\mathrm{xc}(\mathrm{STAB}_\ell(G)) \leqslant f(\ell) \cdot m^c$.

Given a pair (k, n) of natural numbers consider the graph $\mathrm{PLC}(k, n)$. By Proposition 7.2.8, we have that $\mathrm{xc}(\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))) \geqslant n^{c'k}$ for some constant $c' > 0$. On the other hand, from our assumption for $\ell = k^2$ and $m \leqslant (kn)^2$ we have that $\mathrm{xc}(\mathrm{STAB}_{k^2}(G)) \leqslant f(k^2) \cdot (kn)^{2c}$. Therefore, $n^{c'k} \leqslant f(k^2) \cdot (kn)^{2c}$ and so $c'k \log n \leqslant \log f(k^2) + 2c(\log k + \log n)$. This in turn implies that $\log n \leqslant \frac{\log f(k^2) + 2c \log k}{c'k - 2c}$ which clearly cannot be true for any fixed k and arbitrary n, and hence no such function f exists. □

7.3 FPT UPPER BOUNDS

Now we will see that FPT upper bounds exist for a large number of interesting polytopes.

7.3.1 *MSO Polytopes parametrized by Treewidth*

In most cases, we stick to standard notation as given by Libkin [47] and by Downey and Fellows [22]. We use the standard approach and view every graph $G = (V, E)$ as a labeled graph $I(G) = (V_I, E_I, L_V, L_E)$, called the *incidence graph* of G, where $V_I = V \cup E$, $E_I = \{\{v, e\} \mid v \in e, e \in E\}$, $L_V = V$ and $L_E = E$; this way, every MSO_2 formula about the original graph G can be turned into an MSO formula about $I(G)$. Since the treewidth of the incidence graph $I(G)$ is at most $tw(G) + 1$ [43], this does not pose any limitation.

Also, for simplicity, we will work with a version of MSO that has only set variables and a special predicate s of arity one to emulate element variables (for every graph $G = (V, E)$ and every $X \subseteq V \cup E$, $s(X)$ is true in G if and only if $|X| = 1$); it is easy to see that this syntactical restriction does not mean any restriction in the expressive power. All results can be extended to general finite structures where the restriction on treewidth applies to the treewidth of their Gaifman graph [22].

Formally, the set of MSO formulae is defined recursively as follows. We assume an infinite supply of set variables $X, Y, X_1, \ldots$. For every two variables X and Y, $s(X)$, $ver(X)$, $edg(X)$, $inc(X, Y)$, $X \subseteq Y$ and $X = Y$ are formulae, namely *atomic* formulae. For a given graph G, $ver(X)$ or $edg(X)$ is true, if $X \subseteq L_V$ or $X \subseteq L_E$, resp.; $inc(X, Y)$ is true if and only if $s(X)$, $s(Y)$ are true and $\{x, y\} \in E_I$ where x is the only element in X and y is the only element in Y. If φ, ψ_1 and ψ_2 are formulae then $\neg\varphi$, $\psi_1 \wedge \psi_2$ and $\exists X \varphi(X)$ are formulae.

A variable X is *free* in φ if it does not appear in any quantification in φ. If $\vec{X}$ is the tuple of all free variables in φ, we write $\varphi(\vec{X})$. A variable X is *bound* in φ if it is not free. By $qr(\varphi)$ we denote the *quantifier rank* of φ which is the number of quantifiers of φ when transformed into the prenex form (i.e., all quantifiers are in the front of the formula).

For a given MSO formula $\varphi(\vec{X})$ with m free set variables $X_1, \ldots, X_m$, we define a polytope of satisfying assignments on a given graph G with n vertices in a natural way. We encode any assignment of vertices of G to the sets $X_1, \ldots, X_m$ as follows. For each X_i in φ and each

v in G, we introduce a binary variable y_i^v. We set y_i^v to be one if $v \in X_i$ and zero otherwise. For a given 0/1 vector y, we say that y *satisfies* φ if interpreting the coordinates of y as described above yields a satisfying assignment for φ. The polytope of satisfying assignments, also called the *MSO polytope*, is defined as

$$\mathrm{MSO}_\varphi(G) = \mathrm{conv}\left(\{y \in \{0,1\}^{nm} \mid y \text{ satisfies } \varphi\}\right) .$$

Proposition 7.3.1. *For every graph G on n vertices with $\tau(G) = \tau$ and for every $\varphi \in MSO$, $\mathrm{xc}(MSO_\varphi(G)) = f(|\varphi|, \tau) \cdot n$ where f is some computable function.*

Proof. See [44], Theorem 2 (Appendix G). □

In fact the extended formulation can be efficiently constructed.

Proposition 7.3.2. *Let G be a graph of treewidth τ and let φ be an MSO formula. An extended formulation for $MSO_\varphi(G)$ (with size bounded as mentioned in Proposition 7.3.1) can be constructed in time $f'(|\varphi|, \tau) \cdot n$, for some computable function f'.*

Proof. See [44], Theorem 3 (Appendix G). □

In the language that we have adopted so far it means that the extension complexity of MSO polytopes, when parameterized by the treewidth of the underlying graph and the size of the MSO formula, is FPT.

7.3.2 *FO Polytopes parameterized by Expansion*

The *first-order logic of graphs* (abbreviated as FO) applies the standard language of first-order logic to a graph G viewed as a relational structure with the domain $V(G)$ and the single binary (symmetric) relation $E(G)$. For example, the formula $\iota(x_1, \ldots, x_k) \equiv \bigwedge_{i \neq j} \left(\neg \mathrm{edge}(x_i, x_j) \wedge x_i \neq x_j\right)$ asserts that $\{x_1, \ldots, x_k\}$ is an independent set of size exactly k. A slightly more involved example describes a vertex cover tuple as $\gamma(x_1, \ldots, x_k) \equiv \forall y,z\big(\mathrm{edge}(y,z) \to \bigvee_{i=1}^{k}(y = x_i \vee z = x_i)\big)$.

To any FO formula $\phi(x_1, \ldots, x_k)$ and a graph, one can assign a polytope in the following way. For an ordered k-tuple of vertices $W = (w_1, \ldots, w_k) \in V(G)^k$ we thus define its characteristic vector χ^W of length $k|V(G)|$ by

$$\chi^W_{v,i} = \begin{cases} 1 & \text{if } v = w_i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that χ^W always satisfies $\sum_{v \in V(G)} \chi^W_{v,i} = 1$ for each $i = 1, \ldots, k$, by the definition.

If $W = (w_1, \ldots, w_k) \in V(G)^k$ is such that $\phi(w_1, \ldots, w_k)$ holds true in G, we write $G \models \phi(w_1, \ldots, w_k)$. We can now give the following definition:

Definition 7.3.3 (FO polytope)**.** Let $\phi(x_1, \ldots, x_k)$ be an FO formula with k free variables. The *(first-order) ϕ-polytope of* G, denoted by

$\mathrm{FOP}_\phi(G)$, is defined to be the convex hull of the characteristic vectors of every k-tuple of vertices of G such that $\phi(w_1,\dots,w_k)$ holds true in G. That is,

$$\mathrm{FOP}_\phi(G) = \mathrm{conv}\left(\left\{\chi^W \in \{0,1\}^n \;\middle|\; \begin{array}{rcl} W & = & (w_1,\dots,w_k) \in V(G)^k, \\ G & \models & \phi(w_1,\dots,w_k) \end{array}\right\}\right).$$

FO polytopes are quite general. For example, the polytopes $\mathrm{STAB}_k(G)$ defined earlier is easily seen to be an instance of an FO polytope.

Proposition 7.3.4. *Let* $\iota(x_1,\dots,x_k) \equiv \bigwedge_{i\neq j}\left(\neg \mathtt{edge}(x_i,x_j) \wedge x_i \neq x_j\right)$ *(the* k*-independent set formula). For every graph* G*, the* ι*-polytope* $\mathrm{FOP}_\iota(G)$ *is an extension of* $\mathrm{STAB}_k(G)$*.*

Proof. If G has n vertices then

$$\mathrm{STAB}_k(G) = \left\{ y \in \mathbb{R}^n \;\middle|\; y_v = \sum_{i=1}^{k} \chi^W_{v,i},\ \chi^W \in \mathrm{FOP}_\iota(G) \right\}.$$

Therefore, $\mathrm{STAB}_k(G)$ is a projection of $\mathrm{FOP}_\iota(G)$ given by the projection map described by $y_v = \sum_{i=1}^{k} \chi^W_{v,i}$ for all vertices v of G. □

We say that an FO formula $\phi(x_1,\dots,x_k)$ is *existential FO* if it can be written as $\phi(x_1,\dots,x_k) \equiv \exists y_1 \dots y_\ell\, \psi(x_1,\dots,x_k,y_1,\dots,y_\ell)$, where ψ is quantifier-free. The number ℓ of quantified variables in ϕ is called the *quantifier rank of* ϕ.

Proposition 7.3.5. *Let* $\phi(x_1,\dots,x_k)$ *be an existential FO formula with* k *free variables and quantifier rank* ℓ*. Also, let* $\mathcal{G}$ *be any graph class of bounded expansion. Then there exists a computable function* $f : \mathbb{N} \to \mathbb{N}$*, depending on the expansion function of* $\mathcal{G}$*, such that*

$$\mathrm{xc}\big(\mathrm{FOP}_\phi(G)\big) \leqslant f(k+\ell)\cdot n$$

holds for every integer n *and every* n*-vertex graph* $G \in \mathcal{G}$*. Furthermore, an explicit extension of* $\mathrm{FOP}_\phi(G)$ *of size at most* $f(k+\ell)\cdot n$ *can be found in linear time for fixed* k, ℓ *and* $\mathcal{G}$*.*

Proof. See [30], Theorem 20 (Appendix F). □

8

$\mathcal{H}$-FREE EXTENDED FORMULATIONS

Since linear programming is in P we will not be able to solve an NP-hard problem X in polynomial time by linear programming unless $P = NP$. On the other hand, since linear programming is P-complete, we will not be able to prove a super-polynomial lower bound on solving X by a linear program (LP) without showing that $P \neq NP$. One way to make progress on this problem is to consider restricted versions of linear programming which have two properties:

PROPERTY (1): Problems in P will still be solvable in polynomial time even in the restricted version of linear programming.

PROPERTY (2): Known NP-hard problems with natural LP formulations will have provable super-polynomial lower bounds under the restricted version of linear programming.

Note that results of type (1) and (2) will still be true, independently of whether or not $P = NP$.

Here we propose a stronger version of extension complexity which satisfies property (1). We also exhibit some NP-hard problems that satisfy property (2). In the proposed model we concentrate on the separation problem rather than the polynomial time equivalent optimization problem.

8.1 $\mathcal{H}$-FREE EXTENSIONS

Definition 8.1.1. Let $P = P(\mathbf{A}, \mathbf{b})$ be a polytope and let H be a set of valid inequalities for P. We delete from $\mathbf{Ax} \leqslant \mathbf{b}$ all inequalities that are redundant with respect to H and call the resulting (possibly empty) polyhedron P_H. The *$\mathcal{H}$-free extension complexity of* P *with respect to the inequalities* H is defined to be the extension complexity of P_H.

Let X be some computational problem that can be solved by an LP over a polytope Q. For the applications considered in this chapter, it is convenient to consider the case where Q is given by an implicit description of its vertices. So for the matching problem, Q is the convex hull of all 0/1 matching vectors, and for the TSP problem it is the convex hull of all 0/1 incidence vectors of Hamiltonian circuits.

Let $H = H(Q)$ be a possibly super-polynomial size set of valid inequalities for Q equipped with an H-separation oracle. We can solve the separation problem for Q for a point x by first solving it for H and then, if necessary, for Q_H. Suppose x is not in Q. If x is not in H we get a violated inequality by the oracle. Otherwise x must violate a facet of Q_H. We will allow separation for Q_H to be performed using any extension Q'_H of Q_H by explicitly checking the facets of Q'_H for

the lifting of x. We call Q'_H an *$\mathcal{H}$-free EF for Q*. Using this separation algorithm and the ellipsoid method we have a way to solve LPs over Q. We call such a restricted method of solving LPs an *$\mathcal{H}$-free LP for Q*.

Definition 8.1.2. Let Q be a polytope and H be a set of inequalities valid for Q. We say that an *$\mathcal{H}$-free EF for Q has polynomial size* if:

(a) The H-separation oracle runs in polynomial time and

(b) $xc(Q_H)$ is polynomial in the input size of X.

In this case we also have an $\mathcal{H}$-free LP for Q that runs in polynomial time. On the other hand, if for given H, $xc(Q_H)$ is super-polynomial in the size of X then we say that all $\mathcal{H}$-free LPs for X run in super-polynomial time. Note that this statement is independent of whether or not $P = NP$. When H is empty all of the above definitions reduce to standard definitions for EFs and extension complexity.

Example 8.1.3. For the matching problem if H is the set of odd-set inequalities then Q_H is empty. In this case we have an $\mathcal{H}$-free EF for matching of poly-size even though matching has exponential extension complexity.

This example generalizes to show that every problem X in P has a poly-size $\mathcal{H}$-free EF for some H. Indeed, since LP is P-complete, X can be solved by optimizing over a polytope Q. Let H be the entire facet list $F(Q)$ so that Q_H is again empty. Optimization over Q can be performed in polynomial time so, by the equivalence of optimization and separation, separation over H can be performed in polynomial time also. Therefore (a) and (b) are satisfied as required.

For the TSP, let H be the sub-tour constraints. In this case Q_H is non-empty and in fact one can show (cf. next section) that it has exponential extension complexity. Therefore $\mathcal{H}$-free LPs for the TSP require exponential time, extending the existing extension complexity result for this problem.

We remark that H is an essential parameter here. Matching, for example, has poly-size $\mathcal{H}$-free extension complexity when H are the odd set inequalities, but not when H is empty. Nevertheless, any problem with poly-size $\mathcal{H}$-free extension complexity for some H can of course be solved in polynomial time. For a given hard problem, one gets stronger hardness results by letting H be larger and larger sets of poly-size separable inequalities, as long as one can still prove that Q_H has super-polynomial extension complexity. We give some examples to illustrate this in subsequent sections.

8.2 MATCHING PROBLEMS

Recall that Edmonds' polytope has the following halfspace representation[23]:

$$\sum_{e \in S} x_e \leqslant (|S|-1)/2, \quad S \subseteq V, \ |S| \text{ is odd} \tag{7}$$

$$0 \leqslant x_e \leqslant 1, \quad e \in E. \tag{8}$$

Let H be this half-space representation of Q. Since optimization over Q can be performed in polynomial time by Edmonds' algorithm there is a polynomial time separation algorithm for H. It follows that the matching problem has a poly-size $\mathcal{H}$-free EF.

In the next three subsections we give NP-hard generalizations of the matching problem which have super-polynomial lower bounds on their $\mathcal{H}$-free extension complexity, where H are the odd set inequalities (7).

8.2.1 *Induced matchings*

Let Q be the convex hull of the incidence vectors of all induced matchings in G. Let H be the odd set inequalities (7). Clearly H are valid for Q, and as remarked above, they admit a polynomial time separation oracle. It can be shown that $\text{xc}_H(Q)$ is super-polynomial.

Proposition 8.2.1. $\text{xc}_H(Q)$ *is super-polynomial.*

Proof. Since Proposition 4.2.15 applies to bipartite graphs G, each of the odd set inequalities (7) is redundant for the induced matching polytope of G. Therefore the $\mathcal{H}$-free extension complexity of the induced matching polytope is super polynomial in the worst case. □

Although this example offers an example of $\mathcal{H}$-free extension complexity, it suffers from one obvious weakness. For every graph, all of the inequalities in H are redundant with respect to Q even for non-bipartite graphs! A graph is called *hypomatchable* if the deletion of any vertex yields a graph with a perfect matching. Pulleyblank proved in 1973 (see [48]) that facet-inducing inequalities in (7) correspond to subsets S that span 2-connected hypomatchable subgraphs of G. Let x be the incidence vector for any matching M in G that satisfies such an inequality as an equation. Since S spans a 2-connected subgraph, M cannot be an induced matching.

In order to avoid such trivial cases it is desirable that most, if not all, inequalities of H define facets for at least one polytope Q that corresponds to some instance of the given problem.

8.2.2 *Maximal matchings*

Again, since the graphs G in Proposition 4.2.20 are bipartite, each of the odd set inequalities (7) is redundant for the maximal matching polytope of G. Therefore the $\mathcal{H}$-free extension complexity of the induced matching polytope is super polynomial in the worst case.

This example differs from the example in the previous subsection in that (7) are facet defining for maximum matching polytopes of non-bipartite graphs. To see this, fix a graph G and odd-set S of its vertices. Pulleyblanks's characterisation [48] states that (7) is facet defining for the matching polytope of G whenever S spans a 2-connected hypomatchable subgraph. The only matchings in G that lie on this facet have precisely $(|S|-1)/2$ edges from the set S and are therefore maximal on S. Each of these matchings can be extended to a maximal

matching in G which appears as a vertex of $MM(G)$. Therefore, provided these extensions do not lie in a lower dimensional subspace and $MM(G)$ is full dimensional, (7) is also facet inducing for $MM(G)$ for the given set S. For example, the odd cycles C_{2k+1}, $k \geqslant 3$ with the addition of a chord cutting off a triangle are a family of such graphs.

8.2.3 *Edge disjoint matching and perfect matching*

Note that for every pair of odd subsets S_1, S_2 of G two odd set inequalities can be written: one corresponding to the odd set inequalities for perfect matching polytope on variables x_i, and the other corresponding to the odd set inequalities for matching polytope on variables y_i. For a subset of vertices S, let $\delta(S)$ denote the subset of edges with exactly one endpoint in S. The two sets of inequalities are:

$$\sum_{e \in \delta(S_1)} x_e \geqslant 1, \qquad S_1 \subseteq V, \ |S_1| \text{ is odd} \tag{9}$$

$$\sum_{e \in S_2} y_e \leqslant \frac{|S_2|-1}{2}, \quad S_2 \subseteq V, \ |S_2| \text{ is odd} \tag{10}$$

Again the graphs G in the Proposition 4.2.24 are bipartite so each of the odd set inequalities (9,10) is redundant for $MPM(G)$. Therefore taking H to be the set of these inequalities we have that the $\mathcal{H}$-free extension complexity of the these polytopes is super polynomial in the worst case.

8.3 THE TSP POLYTOPE

Recall that an undirected TSP instance X is defined by a set of integer weights $w_{ij}, 1 \leqslant i < j \leqslant n$, for each edge of the complete graph K_n. A tour is a Hamiltonian cycle in K_n defined by a permutation of its vertices. It is required to compute a tour of minimum weight. We define the polytope Q to be the convex hull of the 0/1 incidence vectors $x = (x_{ij} : 1 \leqslant i < j \leqslant n\}$ of the tours. It is known that $xc(Q) = 2^{\Omega(n)}$ [55].

We define H to be the set of *subtour elimination* constraints:

$$\sum_{i,j \in S, i \neq j} x_{ij} \leqslant |S|-1, \quad S \subseteq \{1,2,...,n-1\}, \ |S| \geqslant 2. \tag{11}$$

$$x_{ij} \geqslant 0, \quad 1 \leqslant i < j \leqslant n \tag{12}$$

It is well known that the subtour elimination constraints can be polynomial time separated by using network flows. These constraints by themselves define the convex hull of all forests in K_{n-1} and Martin [49] has given an EF for them that has size $O(n^3)$.

Therefore, $xc(Q_H) = 2^{\Omega(n)}$, otherwise together with Martin's result and Proposition 3.2.6), it would imply an upper bound of $2^{o(n)}$ for the travelling salesman polytope. It follows that every $\mathcal{H}$-free LP for the TSP runs in exponential time, where H are the subtour inequalities.

8.3.1 *Comb inequalities for TSP*

Definition 8.3.1. For a graph $G = (V, E)$, a comb is defined by a subset of vertices H called the handle and a set of subsets of vertices $T_i, 1 \leqslant i \leqslant k$ where k is an odd number at least three. The sets T_i are called the teeth. The handle and the teeth satisfy the following properties:

$$H \cap T_i \neq \emptyset, \tag{13}$$

$$T_i \cap T_j = \emptyset, \qquad \forall i \neq j \tag{14}$$

$$H \setminus \bigcup_{i=1}^{k} T_i \neq \emptyset \tag{15}$$

The following inequality is valid for the TSP polytope of G and is called the comb inequality for the comb defined by handle H and teeth T_i as above.

$$x(\delta(H)) + \sum_{i=1}^{k} x(\delta(T_i)) \geqslant 3k + 1$$

Grötschel and Padberg [34] showed that every comb inequality defines a facet of TSP_n for each $n \geqslant 6$. It is not known whether separating over comb inequalities is NP-hard, neither is a polynomial time algorithm known.

For a given comb C and a TSP tour T of G, the slack between the corresponding comb inequality and T is denoted by $sl_{comb}(C, T)$.

8.3.2 *2-matching inequalities*

Definition 8.3.2. A comb inequality corresponding to a handle H and k teeth T_i is called a 2-matching inequality if each tooth T_i has size exactly two.

In particular this means that $|H \cap T_i| = 1$ and $|T_i \setminus H| = 1$ for each $1 \leqslant i \leqslant k$. These inequalities are sometimes also referred to as blossom inequalities. Padberg and Rao [51] gave a polynomial time algorithm to separate over the 2-matching inequalities.

8.3.2.1 *Simple comb inequalities*

Definition 8.3.3. A comb inequality corresponding to a handle H and k teeth T_i is called a simple comb inequality if $|H \cap T_i| = 1$ or $|T_i \setminus H| = 1$ for each $1 \leqslant i \leqslant k$.

Simple comb inequalities contain all the 2-matching inequalities. It is not known whether one can separate over them in polynomial time.

8.3.2.2 (h, t)-*uniform comb inequalities*

Let us define a subclass of comb inequalities called (h, t)-*uniform comb inequalities* associated with what we will call (h, t)-*uniform combs* for arbitrary $1 \leqslant h < t$.

Definition 8.3.4. A comb, with handle H and k teeth T_i, is said be (h,t)-uniform if $|T_i| = t$ and $H \cap T_i = h$, for all $1 \leqslant i \leqslant k$.

8.3.3 *Odd set inequalities for perfect matching*

Definition 8.3.5. Let V denote the vertex set of K_n. For every odd set $U \subseteq V$ the following inequality is valid for the perfect matching polytope PM_n and is called an odd set inequality.

$$x(\delta(U)) \geqslant 1$$

For a given odd set S and a perfect matching M of K_n, the slack between the corresponding odd set inequality and M is denoted by $sl_{odd}(S, M)$.

8.3.4 *t-subdivided prisms of a graph*

Definition 8.3.6. A prism over a graph G is obtained by taking two copies of G and connecting corresponding vertices.

It is helpful to visualise this as stacking the two copies one over the other and then connecting corresponding vertices in the two copies by a vertical edge. A t-subdivided prism is then obtained by subdividing the vertical edges by putting $t-2$ extra vertices on them. See Figure 6 for an example.

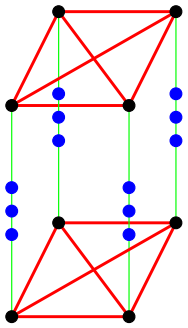


Figure 6: A 5-subdivided prism over K_4.

Let G be the t-subdivided prism of K_n. Let the vertices of the two copies be labeled $u_1^1, \ldots, u_n^1$ and $u_1^t, \ldots, u_n^t$. As a shorthand we will denote the path $u_i^1, u_i^2, \ldots, u_i^t$ as $u_i^1 \rightsquigarrow u_i^t$. Similarily, $u_i^t \rightsquigarrow u_i^1$ will denote $u_i^t, \ldots, u_i^2, u_i^1$.

The graph G has path $u_i^1 \rightsquigarrow u_i^t$ for all $i \in [n]$ and $(u_i^1, u_j^1), (u_i^t, u_j^t)$ for all $i \neq j, i, j \in [n]$. Thus G has tn vertices and $2\binom{n}{2} + (t-1)n$ edges.

8.3.5 *Motivation*

The motivation for looking at t-subdivided prisms stems from a simple observation which we state in the form of a proof of the following proposition:

Proposition 8.3.7. *Let* $2\mathrm{MP}(n)$ *be the convex hull of the incidence vectors of all 2-matchings of the complete graph* K_n*. Then,* $\mathrm{xc}(2\mathrm{MP}(n)) \geqslant 2^{\Omega(n)}$.

Proof. Let G be a graph with n vertices and m edges and let G' be the 3-subdivided prism of G. G' has $3n$ vertices and $2m + 2n$ edges. Any 2-matching in G' contains all the vertical edges and thus when restricted to a single copy – say the bottom one – of G gives a matching in G. Conversely, any matching in G can be extended to a (not necessarily unique) 2-matching in G'.

Taking G as K_n we obtain a G' that is a subgraph of K_{3n}. The 2-matching polytope of G' lies on a face of the 2-matching polytope of the complete graph on $3n$ vertices (corresponding to all missing edges having value 0). Therefore, the extension complexity of the 2-matching polytope $2\mathrm{MP}(n)$ is at least as large as that of the perfect matching polytope. That is, $\mathrm{xc}(2\mathrm{MP}(n)) \geqslant 2^{\Omega(n)}$. □

The above generalizes to P-matching polytopes for arbitrary P in the obvious way, and is probably part of folklore[1].

The generalization of the 3-subdivided prism to larger subdivisions allows us to be able to argue not only about the 2-matching inequalities – which are the facet-defining inequalities for the 2-matching polytope – but also about comb inequalities by using the vertical paths as teeth for constructing combs.

8.3.6 *Uniform combs of odd sets*

Let n and t be positive integers. In the following we will assume that n is a multiple of t. Since we are interested in asymptotic statements only, this does not result in any loss of generality. Let G be the t-subdivided prism of $K_{n/t}$ for some $t \geqslant 2$. Given an odd set S and a perfect matching M in $K_{n/t}$, and arbitrary $1 \leqslant h < t$, we are interested in constructing a comb C and a TSP tour T in K_n such that the following conditions hold:

(C1): C is a (h, t)-uniform comb.

(C2): C depends only on S and 2 edges of M.

(C3): T depends only on M.

(C4): $\mathrm{sl}_{\mathrm{comb}}(C, T) = \mathrm{sl}_{\mathrm{odd}}(S, M)$.

If such a pair (C, T) of a comb and a TSP tour is shown to exist for every pair (S, M) of an odd set and a perfect matching, then we can show that any EF-protocol for computing the slack $\mathrm{sl}_{\mathrm{comb}}(C, T)$ can be used to construct an EF-protocol for computing $\mathrm{sl}_{\mathrm{odd}}(S, M)$ due to condition (C4). Furthermore, due to conditions (C2) and (C3) the number of bits required for the later protocol will not be much larger than the number of bits required for the former, as C can be locally constructed from S after an exchange of two edges, and T can be locally constructed from M.

1 W. Cook (private communication) attributes the same argument to T. Rothvoß

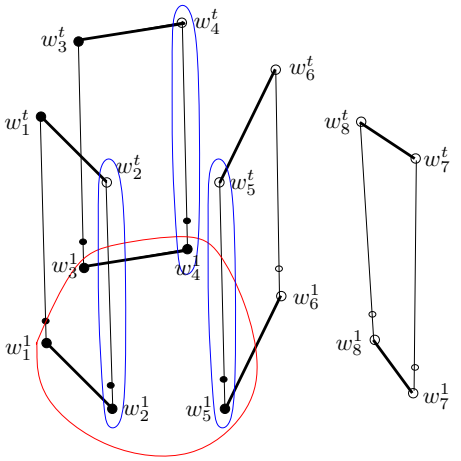


Figure 7: Construction of a comb from given odd set: The odd set consists of 5 vertices displayed as big filled circles in the bottom copy. The corresponding handle consists of all vertices represented by filled circles. The teeth are represented by the vertical ellipsoidal enclosures.
The big circles represent vertices of the original graph and their top copies. The small circles represent the h-th copy, while the other copies have been omitted here. Bold edges at the bottom are matching edges. All other edges displayed are just for illustration of the relationship of various copies of vertices.

Now we show that such a pair does exist if at least two edges of M are contained in S and $|S| \geqslant 5$.

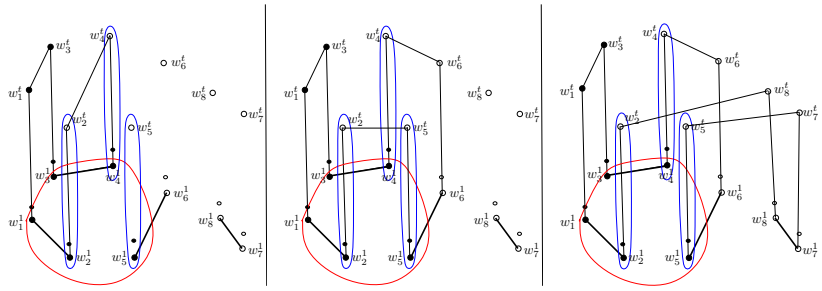
Proposition 8.3.8. *Let (S, M) be a pair of an odd set and a perfect matching in $K_{n/t}$, and let $1 \leqslant h < t$. Suppose that $|S| \geqslant 5$, and let $w_1, w_2, w_3, w_4 \in S$ be distinct with (w_1, w_2) and (w_3, w_4) in M. Then, there exists a pair (C, T) of a comb C and a TSP tour T in K_n satisfying the four conditions (C1)–(C4).*

Proof. Let $|S| = s$. For simplicity of exposition, we assume that the vertices of S are labeled $w_1, \ldots, w_s$. By w_i^j, we denote the copy of w_i in the j-th layer of the t-subdivided prism over $K_{n/t}$.

The comb C is constructed as follows. The handle H is obtained by taking all vertices in S and the copies $w_1^2, \ldots, w_1^t$ and $w_3^2, \ldots, w_3^t$. For every other vertex $w \in S$ the vertices $w^2, \ldots, w^h$ are also added to H. The teeth T_i are formed by pairing each vertex v in $S \setminus \{w_1, w_3\}$ with its copies $v^2, \ldots, v^t$ producing $s-2$ teeth. See Figure 7 for an illustration. Since $s \geqslant 5$ is odd, the number of teeth is odd and at least 3. Thus, the constructed comb is (h, t)-uniform satisfying conditions (C1) and (C2), and the corresponding comb inequality is

$$x(\delta(H)) + \sum_{i=1}^{s-2} x(\delta(T_i)) \geqslant 3(s-2) + 1. \tag{16}$$

To construct a tour T from the given perfect matching M such that conditions (C3) and (C4) are satisfied, we start with a subtour $(w_1^1 \rightsquigarrow w_1^t, w_3^t \rightsquigarrow w_3^1, w_4^1 \rightsquigarrow w_4^t, w_2^t \rightsquigarrow w_2^1, w_1^1)$. At each stage we maintain a subtour that contains all matching edges on the induced vertices in the lower copy, the edge (w_1^t, w_3^t), and at least one top edge different from (w_1^t, w_3^t). Clearly the starting subtour satisfies these requirements. As long as we have some matching edges in M that are not in our subtour, we pick an arbitrary edge (w_a, w_b) in M and extend our subtour as follows. Select a top edge (w_q^t, w_r^t) different from (w_1^t, w_3^t), remove the edge and add the path $(w_q^t, w_a^t \rightsquigarrow w_a^1, w_b^1 \rightsquigarrow w_b^t, w_r^t)$. The new subtour contains the selected perfect matching edge (w_a^1, w_b^1), the paths $w_a^1 \rightsquigarrow w_a^t$ and $w_b^1 \rightsquigarrow w_b^t$ and has one more top edge distinct from (w_1^t, w_3^t) than in the previous subtour. See Figure 8 for an example.



(a) The initial tour going through $w_1^1, w_1^t, w_3^t, w_3^1, w_4^1, w_4^t, w_2^t, w_2^1, w_1^t$

(b) Adding a new matching edge (w_5, w_6) by removing (w_2^t, w_4^t)

(c) The final tour

Figure 8: Constructing a TSP tour from a perfect matching.

At the completion of the procedure, we have a TSP tour that satisfies the following properties:

1. Each edge of M is used in the tour.
2. Each vertical path $w_i^1 \rightsquigarrow w_i^t$ for all $i \in [n]$ is used in the tour.
3. Edge (w_1^t, w_3^t) is used in the tour.

From the construction, edges in $|\delta(H) \cap T|$ are precisely the edges in $|\delta(S) \cap M|$ together with $s-2$ other edges exiting the comb: one through each of the $s-2$ teeth. Therefore, $|\delta(H) \cap T| = |\delta(S) \cap M| + s - 2$. Also, the tour T enters and exits each teeth precisely once so $|\delta(T_i) \cap T| = 2$ for each of the $s-2$ teeth. Substituting these values in the inequality 16, we obtain the slack $\mathrm{sl}_{\mathrm{comb}}(C, T) = |\delta(S) \cap M| + (s-2) + 2(s-2) - 3(s-2) - 1 = \mathrm{sl}_{\mathrm{odd}}(S, M)$. This completes the proof because the pair (C, T) satisfies conditions (C1)–(C4).

□

Using the existence of the pair (C, T) as described earlier and the fact that any EF-protocol for the perfect matching polytope requires an exchange of a linear number of bits, we will lower bound the number of bit exchanged by any EF-protocol computing the slack of (h, t)-uniform comb inequalities with respect to TSP tours. In the next section we will use the following proposition multiple times by fixing different values for the parameters h and t.

Proposition 8.3.9. *Any EF-protocol computing the slack of (h, t)-uniform comb inequalities with respect to the TSP tours of K_n, requires an exchange of $\Omega(n/t)$ bits. Equivalently, the extension complexity of the polytope of (h, t)-uniform comb inequalities is $2^{\Omega(n/t)}$.*

Proof. Due to Proposition 3.1.13 and 3.1.25, it suffices to show if such a protocol uses r bits, then an EF-protocol for the perfect matching polytope for $K_{n/t}$ can be constructed, that uses $r + \mathcal{O}(\log(n/t))$ bits. The protocol for computing the slack of an odd set inequality with respect to a perfect matching in $K_{n/t}$ works as follows.

Suppose Alice has an odd set S in $K_{n/t}$, with $|S| = s$, and Bob has a matching M in $K_{n/t}$. The slack of the odd-set inequality corresponding to S with respect to matching M in the perfect matching polytope for $K_{n/t}$ is $|\delta(S) \cap M| - 1$.

We assume that $s \geqslant 5$. Otherwise, Alice can send the identity of the entire set S with at most $4 \log(n/t)$ bits and Bob can output the slack exactly.

Alice first sends an arbitrary vertex $w_1 \in S$, to Bob. Bob replies with the matching vertex of w_1, say w_2. Alice then sends another arbitrary vertex $w_3 \in S, w_3 \neq w_2$ to Bob who again replies with the matching vertex for w_3, say w_4. So far the number of bits exchanged is $4 \lceil \log(n/t) \rceil$.

Now there are two possibilities: either at least one of the vertices w_2, w_4 is not in S, or both w_2, w_4 are in S. Alice sends one bit to communicate which of the possibilities has occurred and accordingly they switch to one of the two protocols as described next.

In the former case, Alice has identified an edge, say e, in $\delta(S) \cap M$. Now Bob selects an edge e' of his matching uniformly at random (i.e. with probability $2/n$) and sends it to Alice. If e' is in $\delta(S) \setminus \{e\}$, Alice outputs $n/2$. Otherwise, Alice outputs zero. The expected contribution by edges in $(\delta(S) \cap M) \setminus \{e\}$ is then exactly one while the expected contribution of all other edges is zero. Therefore the expected output is $|\delta(S) \cap M| - 1$, and the number of bits exchanged for this step is $\lceil \log m \rceil$ where m is the number of edges in $K_{n/t}$. Thus the total cost in this case is $O(\log(n/t))$ bits.

In the latter case, the matching edges (w_1, w_2) and (w_3, w_4) lie inside S. Alice constructs a comb C in the t-subdivided prism of $K_{n/t}$, and Bob a TSP tour T in the t-subdivided prism of $K_{n/t}$ such that (C, T) satisfies conditions (C1)–(C4). By Proposition 8.3.8 they can do this without exchanging any more bits. Since $\mathrm{sl}_{\mathrm{comb}}(C, T) = \mathrm{sl}_{\mathrm{odd}}(S, M)$, they proceed to compute the corresponding slack with the new inequality and tour, exchanging r bits. The total number of bits exchanged in this case is $r + 4 \lceil \log(n/t) \rceil + 1 = r + O(\log(n/t))$. □

8.3.7 *lower bounds*

In this section we consider the extension complexity of the polytope of comb inequalities and $\mathcal{H}$-free extension complexity of the TSP polytope when $\mathcal{H}$ is the set of simple comb inequalities. As we will see, the results in this section are obtained by instantiating Proposition 8.3.9 with different values of the parameters h and t.

8.3.8 *Extension complexity of Comb inequalities*

We show that the polytope defined by the Comb inequalities has high extension complexity.

Proposition 8.3.10. *Let $COMB(n)$ be the polytope defined by the intersection of all comb inequalities for TSP_n. Then $\mathrm{xc}(COMB(n)) \geqslant 2^{\Omega(n)}$.*

Proof. Suppose there exists an EF-protocol that computes the slack of $\mathrm{COMB}(n)$ that uses r bits. Since $(1,2)$-uniform comb inequalities are valid for TSP_n we can use the given protocol to compute the slack of these inequalities with respect to the TSP tours of K_n using r bits. Then, using Proposition 8.3.9, the slack matrix of the perfect matching polytope for $K_{n/2}$ can be computed using $r + \mathcal{O}(\log n)$ bits. By Proposition 3.1.13 and 3.1.25, this must be $\Omega(n)$. Finally, by Proposition 3.1.25 this implies that $\mathrm{xc}(\mathrm{COMB}(n)) \geqslant 2^{\Omega(n)}$. □

8.3.9 $\mathcal{H}$-free extension complexity

Let $\mathcal{C}_{h,t}$ be the set of (h,t)-uniform comb inequalities for fixed values of h and t. Observe that, since at least three teeth are required to define a comb and the handle must contain some vertex not in any teeth, for (h,t)-uniform combs on n vertices we must have $t \leqslant \left\lfloor \frac{n-1}{3} \right\rfloor$. So for any values of $1 \leqslant h < t \leqslant \left\lfloor \frac{n-1}{3} \right\rfloor$, the set $\mathcal{C}_{h,t}$ is a nonempty set of facet-defining inequalities for TSP_n, and for any other values of h and t the set $\mathcal{C}_{h,t}$ is empty.

Proposition 8.3.11. *If $\mathcal{H}$ is a set of inequalities valid for the polytope TSP_n, such that $\mathcal{H} \cap \mathcal{C}_{h,t} = \emptyset$ for some nonempty $\mathcal{C}_{h,t}$, then the $\mathcal{H}$-free extension complexity of TSP_n is at least $2^{\Omega(n/t)}$.*

Proof. Let $1 \leqslant h < t$ be integers such that $\mathcal{H} \cap \mathcal{C}_{h,t} = \emptyset$. That is, the set $\mathcal{H}$ does not contain any (h,t)-uniform comb inequalities. Let P be the polytope formed from TSP_n by throwing away any facet-defining inequalities that are in $\mathcal{H}$. Then, any EF-protocol computing the slack matrix of P correctly must use $\Omega(n/t)$ bits due to Proposition 8.3.9. The claim then follows from Proposition 3.1.25. □

The previous proposition shows that for every set $\mathcal{H}$ of valid inequalities of TSP_n, if the extension complexity of the TSP polytope becomes polynomial after removing the inequalities in $\mathcal{H}$, then $\mathcal{H}$ must contain some inequalities from every (h,t)-uniform comb inequality class, for all $t = o(n/\log n)$.

Exercise 8.3.12. Show that the statement can be made stronger by replacing the requirement $\mathcal{H} \cap \mathcal{C}_{h,t} = \emptyset$ with $|\mathcal{H} \cap \mathcal{C}_{h,t}| \leqslant \mathrm{poly}(n)$. (**Hint:** See the discussion in Section 8.3).

We can use the previous proposition to give lower bounds for $\mathcal{H}$-free extension complexity of the TSP polytope with respect to important classes of valid inequalities by simply demonstrating some class of (h,t)-uniform comb inequalities that has been missed.

2-matching inequalities

Proposition 8.3.13. *Let P be the polytope obtained by removing the 2-matching inequalities from the TSP polytope. Then, $\mathrm{xc}(P) = 2^{\Omega(n)}$.*

Proof. The 2-matching inequalities are defined by combs for which each tooth has size exactly two. Therefore the set of $(1,3)$-uniform combs are not 2-matching inequalities, and Proposition 8.3.11 applies. □

Simple comb inequalities

Proposition 8.3.14. *Let* P *is the polytope obtained by removing the set of simple comb inequalities from the TSP polytope. Then,* $\mathrm{xc}(P) = 2^{\Omega(n)}$.

Proof. Recall that a comb is called simple if $|H \cap T_i| = 1$ or $|T_i \setminus H| = 1$ for all $1 \leqslant i \leqslant k$ where k is the (odd) number of teeth in the comb and H is the handle. Clearly, $(2,4)$-uniform combs are not simple and Proposition 8.3.11 applies. □

As mentioned before, simple comb inequalities define a superclass of 2-matching inequalities and a polynomial time separation algorithm is known for 2-matching inequalities. Althought a similar result was claimed for simple comb inequalities, the proof was apparently incorrect, as pointed out by Fleischer et al. [29]. This latter paper includes a polynomial time separation algorithm for the wider class of simple domino-parity inequalities that we do not consider here.

It remains unknown whether there exists a polynomial time separation algorithm for the (h,t)-uniform comb inequalities.

9

WEAK EXTENDED FORMULATIONS

9.1 P-COMPLETENESS OF LINEAR PROGRAMMING

It is well established that Linear Programming is P-complete with respect to logspace reductions. That is, any problem in P can be reduced to a Linear Programming problem using a logspace reduction. On the other hand it is now also known that the perfect matching polytope has exponential extension complexity. That is any polytope that projects to the perfect matching polytope of the complete graph K_{2n} requires at least $2^{\Omega(n)}$ inequalities to describe. These two facts may appear contradictory at first sight. How come the perfect matching problem is solvable in polynomial time and yet the polytope requires exponential size?

The previous conundrum is easily resolved if one notices subtle differences between decision and optimization problems, and between "reduction to a Linear Program" and "extension complexity of the perfect matching polytope". Extended formulations require the feasible region of the LP formulations to project exactly to the perfect matching polytope, while a reduction to a Linear Programming problem may produce other polytopes as a feasible region. Based on the particular objective function (that is, a particular instance) the reduction may produce different polytopes. At the heart of this issue is the fact that logspace reductions can do fairly non-trivial computation with the objective function. Indeed, it is not even known whether LOGSPACE $\neq$ NP. So for all we know, the reduction may as well solve the perfect matching problem and produce a trivial LP instance.

One may still obtain reasonable and interesting statements if one were to ask the following: Can we obtain a small Linear Program for perfect matching even if the input instance is allowed to be modified only in very restricted ways? The answer obviously depeneds on how this question is formulated in a precise way. We will give one interpretation and obtain small Linear Programs for problems in P/poly. Before we formalize anything we start with an example.

Definition 9.1.1. Let n be an even integer and let x be a binary vector of length $\binom{n}{2}$. We let $G(x) = (V, E)$ denote the graph with edge incidence vector given by x, let n be the number of its vertices and $m = \mathbf{1}^\top x$ the number of its edges. Furthermore, let $w_x = 1$ if $G(x)$ has a perfect matching and zero otherwise. We define the polytope PM_n as:

$$PM_n = \operatorname{conv}\left\{ \begin{pmatrix} x \\ w_x \end{pmatrix} \middle| \, x \in \{0,1\}^{\binom{n}{2}} \right\} \tag{17}$$

PM_n may be visualized by starting with a hypercube in dimension $\binom{n}{2}$ and embedding it in one higher dimension with extra coordi-

nate w. For vertices of the cube corresponding to graphs with perfect matchings $w = 1$ else $w = 0$. It is easy to see that PM_n has precisely $2^{\binom{n}{2}}$ vertices. EP_n is closely related to PM_n, in fact it forms a face.

Proposition 9.1.2. *EP_n is a face of PM_n and can be defined by*

$$EP_n = \left\{ x \;\middle|\; \begin{pmatrix} x \\ w \end{pmatrix} \in PM_n \cap \left\{ \mathbf{1}^\top x + (1-w)n^2 = \frac{n}{2} \right\} \right\} \tag{18}$$

Proof. We first show that the inequality

$$\mathbf{1}^\top x + (1-w)n^2 \geqslant \frac{n}{2} \tag{19}$$

is valid for PM_n. We need only verify it for the extreme points (x, w_x) given in (17). If $w_x = 0$, (19) holds since $\mathbf{1}^\top x + n^2 \geqslant \frac{n}{2}$. Otherwise $w_x = 1$, x is the incidence vector of graph containing a perfect matching, so $\mathbf{1}^\top x \geqslant n/2$. The vectors x with $w_x = 1$ and $\mathbf{1}^\top x = n/2$ are the incidence vectors of perfect matchings of K_n and are precisely those used to define EP_n. □

For a given input graph $G(\bar{x}) = (V, E)$ we define the vector $\mathbf{c}$ by:

$$\mathbf{c}_{ij} = 1 \quad ij \in E \qquad \mathbf{c}_{ij} = -1 \quad ij \notin E \qquad 1 \leqslant i < j \leqslant n \tag{20}$$

and let d be a constant such that $0 < d \leqslant 1/2$. We construct the LP:

$$\begin{aligned} z^* = \max\ z &= \mathbf{c}^\top x + dw \\ \begin{pmatrix} x \\ w \end{pmatrix} &\in PM_n \end{aligned} \tag{21}$$

Proposition 9.1.3. *For any edge incidence vector $\bar{x} \in [0,1]^{\binom{n}{2}}$ let $m = \mathbf{1}^\top \bar{x}$. The optimum solution to (21) is unique, $z^* = m + d$ if $G(\bar{x})$ has a perfect matching, and $z^* = m$ otherwise.*

Proof. See [4], Proposition 2 (Appendix J). □

9.2 WEAK EXTENDED FORMULATIONS

Let X denote a poly-time decision problem defined on binary input vectors $x = (x_1, \ldots, x_q)$, and an additional bit w_x, where $w_x = 1$ if x results in a "yes" answer and $w_x = 0$ otherwise. We define the polytope P as:

$$P = \operatorname{conv}\left\{ \begin{pmatrix} x \\ w_x \end{pmatrix} \middle| \, x \in \{0,1\}^q \right\} \tag{22}$$

For a given binary input vector $\bar{x}$ we define the vector $\mathbf{c}$ by:

$$\mathbf{c}_j = 1 \quad \bar{x}_j = 1 \quad \text{and} \quad \mathbf{c}_j = -1 \quad \bar{x}_j = 0 \qquad 1 \leqslant j \leqslant q \tag{23}$$

and let d be a constant such that $0 < d \leqslant 1/2$. As before we construct an LP:

$$\begin{aligned} z^* = \max\ z &= \mathbf{c}^\top x + dw \\ \begin{pmatrix} x \\ w \end{pmatrix} &\in P \end{aligned} \tag{24}$$

The following proposition can be proved in an identical way to Proposition 9.1.3.

Proposition 9.2.1. *For any $\bar{x} \in [0,1]^q$ let $m = \mathbf{1}^\top \bar{x}$. The optimum solution to (24) is unique, $z^* = m + d$ if $\bar{x}$ has a "yes" answer and $z^* = m$ otherwise.*

Definition 9.2.2. Let Q be a polytope which is a subset of the $(q+t)$-cube with variables labeled $x_1, ..., x_q, y_1, ..., y_t$. We say that Q has the *x-o/1 property* if each of the 2^q ways of assigning $0/1$ to the x variables uniquely extends to a vertex $(x^\top, y^\top)^\top$ of Q and, furthermore, y is $0/1$ valued. Q may have additional fractional vertices.

In polyhedral terms, this says that the intersection of Q with the hyperplanes $x_j = e_j, j = 1, ..., q$ is a $0/1$ vertex, for each assignment of zero or one to the e_j's. We can show that we can solve a polytime decision problem X by replacing P in (21) by a polytope Q of polynomial size, while maintaining the same objective functions. We call Q a *weak extended formulation* as it does not necessarily project onto P.

Definition 9.2.3. A polytope

$$Q = \left\{ \begin{pmatrix} x \\ w \\ s \end{pmatrix} \in [0,1]^{q+1+r} \;\middle|\; Ax + bw + Cs \leqslant h \right\}$$

is a *weak extended formulation (WEF)* of P if

- Q has the x-$0/1$ property.
- For any binary vector $\bar{x} \in [0,1]^q$ let $m = \mathbf{1}^\top \bar{x}$. Let c be defined by (23) and let $0 < d \leqslant 1/2$. The optimum solution

$$z^* = \max \left\{ c^\top x + dw \;\middle|\; \begin{pmatrix} x \\ w \\ s \end{pmatrix} \in Q \right\}$$

  is unique and takes the value $z^* = m + d$ if $\bar{x}$ has a "yes" answer. Otherwise $z^* < m + d$ and for all sufficiently small d, $z^* = m$ and is unique.

For example, let X be the perfect matching problem so that $P = PM_n$. Let $Q = Q_n$ be a WEF as given by this definition. It follows from Proposition 9.1.3 that we can determine whether an input graph G has a perfect matching by solving an LP over either PM_n or Q_n using the same objective function which is derived directly from the edge adjacency vector of G.

Example 9.2.4. Consider $n = 2$ giving $PM_2 = \text{conv}\{(0,0)^\top, (1,1)^\top\}$. A WEF, for example, is given by:

$$Q_2 = \text{conv}\{(0,0,0)^\top, (1,1,1)^\top, (1/4,1,1/2)^\top\}$$

Initially let $d = 1/2$. When $G(\bar{x})$ is an edge, $m = 1$, $c_{12} = 1$ and $z = c^\top x + dw$ obtains the same optimum solution of $z^* = 3/2 = m + d$

over both PM_2 and Q_2. When $G(\bar{x})$ is a non-edge, $m = 0$, $c_{12} = -1$ and $z = c^\top x + dw$ obtains the optimum solution of $z^* = 0 = m$ over PM_2 and $z^* = 1/4 < 1/2 = m + d$ over Q_2, at the fractional vertex $(1/4, 1, 1/2)^\top$. However, if $0 < d < 1/4$ then $z = \mathbf{c}^\top \mathbf{x} + dw$ obtains the unique optimum solution of $z^* = 0 = m$ over both PM_2 and Q_2. We see that Q_2 projects onto a triangle in the (x, w)-space, whereas PM_2 is a line segment.

9.3 WEAK EXTENSION FOR P/poly

In order to show that Linear Programming is **P**-complete, Valiant [61] gave a construction to transform boolean circuits into a linear sized set of linear inequalities with the x-o/1 property (where x_i are the variables corresponding to the inputs of the circuit); a similar construction was used by Yannakakis [63] in the context of the Hamiltonian Circuit problem. One can show that Valiant's construction implies the following.

Proposition 9.3.1. *Every decision problem* X *in* P/poly *admits a weak extended formulation* Q *of polynomial size.*

Valiant's point of view is slightly different from ours in that he explicitly fixes the values of the input variables before solving an LP-feasibility problem (as opposed to using different objective functions with a fixed set of inequalities). Showing that the result of this fixing is a 0/1-vertex is precisely our x-o/1 property.

We begin with a standard definition[1]:

Definition 9.3.2. A *(boolean) circuit* with q inputs $x = (x_1, x_2, \ldots, x_q)$ is a directed acyclic graph in which each of its t nodes, called *gates*, is either an AND($\wedge$) gate, an OR($\vee$) or a NOT($\neg$) gate. We label each gate by its output bit. One of these gates is designated as the *output gate* and gives output bit w. The *size* of a circuit is the number of gates it contains and its *depth* is the maximal length of a path from an input gate to the output gate.

For example, the circuit shown in Figure 9 can be used to compute whether or not a graph on 4 nodes has a perfect matching. The input is the binary edge-vector of the graph and the output is $w = 1$ if the graph has a matching (e.g. G_1) or $w = 0$ if it does not (e.g. G_2). If the graph has a perfect matching, exactly one of y_{12}, y_{13} or y_{14} is one, defining the matching. For each gate we have labeled the output bit by a new variable. We will construct a polytope from the circuit by constructing a system of inequalities on the same variables.

From an AND gate, say $y_{12} = x_{12} \wedge x_{34}$, we generate the inequalities:

$$\begin{aligned} x_{12} + x_{34} - y_{12} &\leqslant 1 \\ -x_{12} \qquad + y_{12} &\leqslant 0 \\ -x_{34} + y_{12} &\leqslant 0 \\ y_{12} &\geqslant 0 \end{aligned} \tag{25}$$

1 See, e.g., the text by Savage [56]

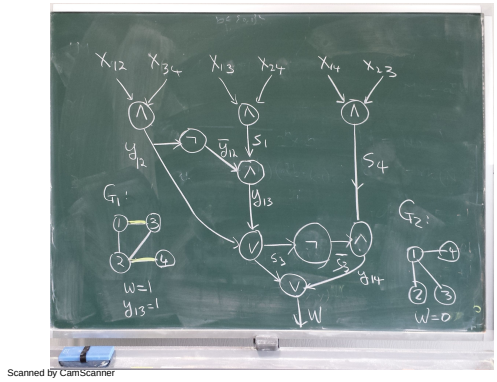


Figure 9: A circuit to compute whether a 4 node graph has a perfect matching

The system (25) defines a polytope in three variables whose 4 vertices represent the truth table for the AND gate:

x_{12}	x_{34}	y_{12}
0	0	0
0	1	0
1	0	0
1	1	1

Note that the variables x_{12}, x_{34} define a 2-cube and so the polytope is an extension of the 2-cube. In the terminology of the last section, it has the $\{x_{12}, x_{34}\}$-0/1 property.

From an OR gate, say $s_3 = y_{12} \vee y_{13}$, we generate the inequalities:

$$\begin{aligned} -y_{12} - y_{13} + s_3 &\leqslant 0 \\ y_{12} - s_3 &\leqslant 0 \\ y_{13} - s_3 &\leqslant 0 \\ s_3 &\leqslant 1 \end{aligned} \tag{26}$$

The system (26) defines a polytope in three variables whose 4 vertices represent the truth table for the OR gate, as can easily be checked. Indeed, this polytope has the $\{y_{12}, y_{13}\}$-0/1 property.

From a NOT gate, say $\bar{y}_{12} = \neg y_{12}$, we could generate the equation

$$\bar{y}_{12} = 1 - y_{12} \tag{27}$$

However it is equivalent to just replace all instances of $\bar{y}_{12}$ by $1 - y_{12}$ in the inequality system, and this is what we will do in the sequel.

The circuit in Figure 9 contains 5 AND gates and 2 OR gates. By suitably replacing variables in (25) and (26) we obtain a system of 28 inequalities in 13 variables. As just mentioned, the NOT gates are handled by variable substitution rather than explicit equations. Let Q_4 denote the corresponding polytope. It will follow by the general argument below that Q_4 is a weak extended formulation (WEF) of PM_4.

It can be shown that the above construction can be applied to any boolean circuit C to obtain a polytope Q which has the 0/1 property with respect to the inputs of C.

Proposition 9.3.3 ([61]). *Let* C *be a boolean circuit with* q *input bits* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_q)$, t *gates labeled by their output bits* $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t)$ *and with circuit output bit* $w = \mathbf{y}_t$. *Construct the polytope* Q *with* $4t$ *inequalities and* $q + t$ *variables using the systems (25) and (26) respectively.* Q *has the the* $\mathbf{x}$*-0/1 property and for every input* $\mathbf{x}$ *the value of* w *computed by* C *corresponds to the value of* $\mathbf{y}_t$ *in the unique extension* $(\mathbf{x}^\top, \mathbf{y}^\top)^\top \in Q$ *of* $\mathbf{x}$.

Proof. See [4], Lemma 1 (Appendix J). □

This allows one to construct a WEF for any problem in P/poly.

Proposition 9.3.4. *Let* C *be a circuit that solves a decision problem* X *with* q *input bits* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_q)$ *and has associated polytope* P *as defined in (22). The polytope* Q *constructed in Proposition 9.3.3 is a WEF for* P.

Proof. See [4], Lemma 2 (Appendix J). □

Since each gate in the circuit gives rise to 4 inequalities and one new variable, we have the following.

Proposition 9.3.5. *Let* X *be a decision problem with corresponding polytope* P *defined by (22). A set of circuits for* X *with size* $p(n)$ *generate a WEF* Q *for* P *with* $4p(n)$ *inequalities and variables.*

BIBLIOGRAPHY

[1] David Avis and Hans Raj Tiwary. "A generalization of extension complexity that captures P." In: *Information Processing Letters* 115.6-8 (2015), pp. 588–593. DOI: 10.1016/j.ipl.2015.02.005.

[2] David Avis and Hans Raj Tiwary. "On the extension complexity of combinatorial polytopes." In: *Mathematical Programming* 153.1 (2015), pp. 95–115. DOI: 10.1007/s10107-014-0764-2.

[3] David Avis and Hans Raj Tiwary. "On the $\mathcal{H}$-free extension complexity of the TSP." In: *Optimization Letters* (2016), pp. 1–11. ISSN: 1862-4480. DOI: 10.1007/s11590-016-1029-1.

[4] David Avis, David Bremner, Hans Raj Tiwary, and Osamu Watanabe. "Polynomial size linear programs for non-bipartite matching problems and other problems in P." In: *CoRR* abs/1408.0807 (2014). eprint: arXiv:1408.0807.

[5] David Avis, Hiroshi Imai, Tsuyoshi Ito, and Yuuya Sasaki. "Two-party Bell inequalities derived from combinatorics via triangular elimination." In: *Journal of Physics A: Mathematical and General* 38.50 (2005), p. 10971. DOI: 10.1088/0305-4470/38/50/007.

[6] Ajesh Babu, Nutan Limaye, Jaikumar Radhakrishnan, and Girish Varma. "Streaming algorithms for language recognition problems." In: *Theoretical Computer Science* 494 (2013), pp. 13–23. DOI: 10.1016/j.tcs.2012.12.028.

[7] Francisco Barahona. "On cuts and matchings in planar graphs." In: *Mathematical Programming* 60 (1993), pp. 53–68. DOI: 10.1007/BF01580600.

[8] Gábor Braun and Sebastian Pokutta. "Common information and unique disjointness." In: *Proc. FOCS*. 2013. DOI: 10.1109/FOCS.2013.79.

[9] Gábor Braun, Samuel Fiorini, Sebastian Pokutta, and David Steurer. "Approximation Limits of Linear Programs (Beyond Hierarchies)." In: *Mathematics of Operations Research* 40.3 (2015), pp. 756–772. DOI: 10.1287/moor.2014.0694.

[10] Gábor Braun, Rahul Jain, Troy Lee, and Sebastian Pokutta. "Information theoretic approximations of the nonnegative rank." In: *Electronic Colloquium on Computational Complexity (ECCC)* 20 (2013), p. 158. URL: http://eccc.hpi-web.de/report/2013/158.

[11] Mark Braverman and Ankur Moitra. "An information complexity approach to extended formulations." In: *Proc. STOC*. 2013, pp. 161–170. DOI: 10.1145/2488608.2488629.

[12] Jop Briët, Daniel Dadush, and Sebastian Pokutta. "On the existence of 0/1 polytopes with high semidefinite extension complexity." In: *Mathematical Programming* 153.1 (2015), pp. 179–199. DOI: 10.1007/s10107-014-0785-x.

[13] Austin Buchanan and Segiy Butenko. *Tight extended formulations for independent set*. Available on Optimization Online. 2014. URL: http://www.optimization-online.org/DB_HTML/2014/09/4540.html.

[14] Michael R. Bussieck and Marco E. Lübbecke. "The vertex set of a 0/1-polytope is strongly P-enumerable." In: *Computational Geometry* 11.2 (1998), pp. 103–109. DOI: 10.1016/S0925-7721(98)00021-2.

[15] Kathie Cameron. "Induced matchings." In: *Discrete Applied Mathematics* 24.1-3 (1989), pp. 97–102. DOI: 10.1016/0166-218X(92)90275-F.

[16] Robert D. Carr and Goran Konjevod. "Polyhedral Combinatorics." In: *Tutorials on Emerging Methodologies and Applications in Operations Research*. Vol. 76. International Series in Operations Research & Management Science. 2005, pp. 2–1–2–46. ISBN: 978-0-387-22826-6.

[17] Lewis Carroll. *Through the Looking Glass*. Penguin Books, 1871.

[18] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. "Approximate Constraint Satisfaction Requires Large LP Relaxations." In: *Proc. FOCS*. 2013, pp. 350–359. DOI: 10.1109/FOCS.2013.45.

[19] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. "Extended formulations in combinatorial optimization." In: *4OR* 8.1 (2010), pp. 1–48. DOI: 10.1007/s10288-010-0122-z.

[20] Michele Conforti and Kanstantsin Pashkovich. "The projected faces property and polyhedral relations." In: *Mathematical Programming* 156.1-2 (2016), pp. 331–342. DOI: 10.1007/s10107-015-0882-5.

[21] Michel Marie Deza and Monique Laurent. *Geometry of cuts and metrics*. Vol. 15. Algorithms and Combinatorics. Springer-Verlag, 1997, pp. xii, 587. DOI: 10.1007/978-3-642-04295-9.

[22] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013, I–SSS, 3–707. ISBN: 978-1-4471-5558-4; 978-1-4471-5559-1.

[23] Jack Edmonds. "Maximum Matching and a Polyhedron with 0, 1 Vertices." In: *Journal of Research of the National Bureau of Standards* 69 B (1965), pp. 125–130.

[24] Yuri Faenza, Samuel Fiorini, Roland Grappe, and Hans Raj Tiwary. "Extended formulations, nonnegative factorizations, and randomized communication protocols." In: *Mathematical Programming* 153.1 (2015), pp. 75–94. DOI: 10.1007/s10107-014-0755-3.

[25] Hamza Fawzi and Pablo A. Parrilo. "Exponential lower bounds on fixed-size psd rank and semidefinite extension complexity." In: *CoRR* abs/1311.2571 (2013). URL: http://arxiv.org/abs/1311.2571.

[26] Samuel Fiorini, Thomas Rothvoß, and Hans Raj Tiwary. "Extended Formulations for Polygons." In: *Discrete & Computational Geometry* 48.3 (2012), pp. 658–668. DOI: 10.1007/s00454-012-9421-9.

[27] Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. "Exponential Lower Bounds for Polytopes in Combinatorial Optimization." In: *Journal of the ACM* 62.2 (2015), p. 17. DOI: 10.1145/2716307.

[28] Samuel Fiorini, Serge Massar, Manas K Patra, and Hans Raj Tiwary. "Generalized probabilistic theories and conic extensions of polytopes." In: *Journal of Physics A: Mathematical and Theoretical* 48.2 (2015), p. 025302. DOI: 10.1145/2716307.

[29] Lisa Fleischer, Adam N. Letchford, and Andrea Lodi. "Polynomial-Time Separation of a Superclass of Simple Comb Inequalities." In: *Mathematics of Operations Research* 31.4 (2006), pp. 696–713. DOI: 10.1287/moor.1060.0214.

[30] Jakub Gajarský, Petr Hliněný, and Hans Raj Tiwary. "Parameterized Extension Complexity of Independent Set and Related Problems." In: *CoRR* abs/1511.08841 (2015). eprint: arXiv:1511.08841.

[31] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990. ISBN: 0716710455.

[32] Albertus M. H. Gerards. "Compact Systems for T-join and Perfect Matching Polyhedra of Graphs with Bounded Genus." In: *Operations Research Letters* 10.7 (Oct. 1991), pp. 377–382. ISSN: 0167-6377. DOI: 10.1016/0167-6377(91)90038-Q.

[33] João Gouveia, Pablo A. Parrilo, and Rekha R. Thomas. "Lifts of Convex Sets and Cone Factorizations." In: *Mathematics of Operations Research* 38.2 (2013), pp. 248–264. DOI: 10.1287/moor.1120.0575.

[34] Martin Grötschel and Manfred Padberg. "On the symmetric travelling salesman problem II: Lifting theorems and facets." In: *Mathematical Programming* 16.1 (1979), pp. 281–302. DOI: 10.1007/BF01582117.

[35] Branko Grünbaum. *Convex polytopes*. Graduate texts in mathematics. Springer, 2003. ISBN: 0-387-00424-6.

[36] Juris Hartmanis, Neil Immerman, and Stephen R. Mahaney. "One-Way Log-Tape Reductions." In: *Proc. FOCS*. 1978, pp. 65–72. DOI: 10.1109/SFCS.1978.31.

[37] Colin de la Higuera and José Oncina. "Inferring Deterministic Linear Languages." In: *Proc. COLT*. 2002, pp. 185–200. DOI: 10.1007/3-540-45435-7_13.

[38] Hui-kai, Yuan-wu, Huikai, Yuanwu, Katsuki Sekida, and A. V. Grimstone. *Two Zen classics : Mumonkan and Hekiganroku / translated with commentaries by Katsuki Sekida ; edited and introduced by A.V. Grimstone*. English. 1st ed. Weatherhill New York, 1977, 413 p. ; ISBN: 0834801310 0834801302.

[39] Volker Kaibel. *Extended Formulations in Combinatorial Optimization*. Optima 85. 14 pages. 2011. URL: http://www.mathopt.org/Optima-Issues/optima85.pdf.

[40] Volker Kaibel, Kanstantsin Pashkovich, and Dirk Oliver Theis. "Symmetry Matters for Sizes of Extended Formulations." In: *SIAM Journal on Discrete Mathematics* 26.3 (2012), pp. 1361–1382. DOI: 10.1137/110839813.

[41] Volker Kaibel and Stefan Weltge. "A Short Proof that the Extension Complexity of the Correlation Polytope Grows Exponentially." In: *Discrete & Computational Geometry* 53.2 (2015), pp. 397–401. DOI: 10.1007/s00454-014-9655-9.

[42] Richard M. Karp. "Reducibility Among Combinatorial Problems." In: *Proc. Symposium on the Complexity of Computer Computations*. 1972, pp. 85–103. URL: http://www.cs.berkeley.edu/~luca/cs172/karp.pdf.

[43] Phokion G. Kolaitis and Moshe Y. Vardi. "Conjunctive-Query Containment and Constraint Satisfaction." In: *Journal of Computer and System Sciences* 61.2 (2000), pp. 302–332. DOI: 10.1006/jcss.2000.1713.

[44] Petr Kolman, Martin Koutecký, and Hans Raj Tiwary. "Extension Complexity, MSO Logic, and Treewidth." In: *Proceedings of the 15th SWAT* To appear (2016). eprint: arXiv:1507.04907.

[45] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[46] Troy Lee and Dirk Oliver Theis. *Support-based lower bounds for the positive semidefinite rank of a nonnegative matrix*. arXiv:1203.3961. 2012.

[47] Leonid Libkin. *Elements of Finite Model Theory*. Berlin: Springer-Verlag, 2004. ISBN: 3-540-21202-7.

[48] László Lovász and Michael D. Plummer. *Matching theory*. North-Holland mathematics studies. Includes indexes. Amsterdam, New York: North-Holland, 1986. ISBN: 0-444-87916-1.

[49] R. Kipp Martin. "Using Separation Algorithms to Generate Mixed Integer Model Reformulations." In: *Operations Research Letters* 10.3 (Apr. 1991), pp. 119–128.

[50] Shmuel Onn and Vladimir A. Shlyk. "Some efficiently solvable problems over integer partition polytopes." In: *Discrete Applied Mathematics* 180 (2015), pp. 135–140. DOI: 10.1016/j.dam.2014.08.015.

[51] Manfred W. Padberg and M. R. Rao. "Odd Minimum Cut-Sets and *b*-Matchings." In: *Mathematics of Operations Research* 7.1 (1982), pp. 67–80. DOI: 10.1287/moor.7.1.67.

[52] Dömötör Pálvölgyi. *Partitioning to three matchings of given size is NP-complete for bipartite graphs*. Tech. rep. QP-2013-01. Egerváry Research Group, Budapest, 2013.

[53] Eric Raymond. *The new hacker's dictionary*. MIT Press, 1991.

[54] Thomas Rothvoß. "Some 0/1 polytopes need exponential size extended formulations." In: *Mathematical Programming* 142.1-2 (2013), pp. 255–268. DOI: 10.1007/s10107-012-0574-3.

[55] Thomas Rothvoß. "The matching polytope has exponential extension complexity." In: *Proc. STOC*. 2014, pp. 263–272. DOI: 10.1145/2591796.2591834.

[56] John E. Savage. *Models of Computation: Exploring the Power of Computation*. http://cs.brown.edu/~jes/book, 2015.

[57] Richard Edwin Stearns, Juris Hartmanis, and Philip M. Lewis II. "Hierarchies of memory limited computations." In: *Proc. Symposium on Switching Circuit Theory and Logical Design*. 1965, pp. 179–190. DOI: 10.1109/FOCS.1965.11.

[58] Larry J. Stockmeyer and Vijay V. Vazirani. "NP-Completeness of Some Generalizations of the Maximum Matching Problem." In: *Information Processing Letters* 15.1 (1982), pp. 14–19. DOI: 10.1016/0020-0190(82)90077-1.

[59] Andrzej Szepietowski. "Weak and Strong One-Way Space Complexity Classes." In: *Information Processing Letters* 68.6 (1998), pp. 299–302. DOI: 10.1016/S0020-0190(98)00176-8.

[60] Hans Raj Tiwary. "Extension Complexity of Formal Languages." In: *ArXiv e-prints* (Mar. 2016). arXiv: 1603.07786 [cs.CC].

[61] Leslie G. Valiant. "Reducibility by algebraic projections." In: *Enseignement Mathématique (2)* 28.3-4 (1982), pp. 253–268. ISSN: 0013-8584.

[62] Laurence A. Wolsey. "Using extended formulations in practice." In: (2011). 14 pages. URL: http://www.mathopt.org/Optima-Issues/optima85.pdf.

[63] Mihalis Yannakakis. "Expressing Combinatorial Optimization Problems by Linear Programs." In: *Journal of Computer and System Sciences* 43.3 (1991), pp. 441–466. DOI: 10.1016/0022-0000(91)90024-Y.

[64] Mihalis Yannakakis and Fanica Gavril. "Edge dominating sets in graphs." In: *SIAM Journal of Applield Mathematics* 38 (1980), pp. 364–372. DOI: 10.1137/0138030.

[65] Günter M. Ziegler. *Lectures on polytopes*. Vol. 152. Graduate Texts in Mathematics. Springer-Verlag, 1995, pp. ix+370.

Part IV

APPENDIX

A

EXPONENTIAL LOWER BOUNDS FOR POLYTOPES IN COMBINATORIAL OPTIMIZATION

The following article has appeared in *Journal of the ACM* and is included here as an appendix for completeness.

Exponential Lower Bounds for Polytopes in Combinatorial Optimization

SAMUEL FIORINI and SERGE MASSAR, Université libre de Bruxelles
SEBASTIAN POKUTTA, Georgia Institute of Technology
HANS RAJ TIWARY, Université libre de Bruxelles
RONALD DE WOLF, CWI and University of Amsterdam

We solve a 20-year old problem posed by Yannakakis and prove that no polynomial-size linear program (LP) exists whose associated polytope projects to the traveling salesman polytope, even if the LP is not required to be symmetric. Moreover, we prove that this holds also for the cut polytope and the stable set polytope. These results were discovered through a new connection that we make between one-way quantum communication protocols and semidefinite programming reformulations of LPs.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.0 [**Discrete Mathematics**]: General

General Terms: Theory

Additional Key Words and Phrases: Combinatorial optimization, linear programming, communication complexity, semidefinite programming, quantum communication complexity

ACM Reference Format:
Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. 2015. Exponential lower bounds for polytopes in combinatorial optimization. J. ACM 62, 2, Article 17 (April 2015), 23 pages.
DOI: http://dx.doi.org/10.1145/2716307

1. INTRODUCTION

Since the advent of the simplex method [Dantzig 1951], linear programming has become a prominent tool for solving optimization problems in practice. On the theoretical side, LPs can be solved in polynomial time via either the ellipsoid method [Khachiyan 1979] or interior point methods [Karmarkar 1984].

In 1986–1987, there were attempts [Swart 1987] to prove P = NP by giving a polynomial-size LP that would solve the traveling salesman problem (TSP). Due to

---

The author list is alphabetical.

S. Fiorini acknowledges support from the *Actions de Recherche Concertées* (ARC) fund of the French community of Belgium. S. Massar acknowledges support from the European Commission under the projects QCS (Grant No. 255961) and QALGO (Grant No. 600700). H. R. Tiwary was a postdoctoral researcher of the *Fonds National de la Recherche Scientifique* (F.R.S.–FNRS). R. de Wolf was partially supported by a Vidi grant from the Netherlands Organization for Scientific Research (NWO), by ERC Consolidator grant QPROGRESS, and by the European Commission under the projects QCS (Grant No. 255961) and QALGO (Grant No. 600700).

Authors' present addresses: S. Fiorini, Université libre de Bruxelles, Département de Mathématique, Algebra and Combinatorics (CP 216), Boulevard du Triomphe, B-1050 Brussels, Belgium; S. Massar, Université libre de Bruxelles, Laboratoire d'information quantique (CP 255), Boulevard du Triomphe, B-1050 Brussels, Belgium; S. Pokutta, Georgia Institute of Technology, Groseclose 0205, Ofice 338, 765 Ferst Drive, Atlanta, GA 30332; H. R. Tiwary, Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, 118 00 Prague 1, Czech Republic; R. de Wolf, CWI, P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands. Correspondence email: hansrajt@gmail.com.

0004-5411/2015/04-ART17 $15.00
DOI: http://dx.doi.org/10.1145/2716307

the large size and complicated structure of the proposed LP for the TSP, it was difficult to show directly that the LP was erroneous. In a groundbreaking effort to refute all such attempts, Yannakakis [1988] proved that every symmetric LP for the TSP has exponential size (see Yannakakis [1991] for the journal version). Here, an LP is called *symmetric* if every permutation of the cities can be extended to a permutation of all the variables of the LP that preserves the constraints of the LP. Because the proposed LP for the TSP was symmetric, it could not possibly be correct.

In his paper, Yannakakis left as a main open problem the question of proving that the TSP admits no polynomial-size LP, symmetric or not. We solve this question by proving a super-polynomial lower bound on the number of inequalities in every LP for the TSP. We also prove such unconditional super-polynomial lower bounds for the maximum cut and maximum stable set problems. Therefore, it is impossible to prove P = NP by means of a polynomial-size LP that expresses any of these problems. Our approach is inspired by a close connection between semidefinite programming reformulations of LPs and one-way quantum communication protocols that we introduce here.

1.1. State of the Art

From Problems to Polytopes. For combinatorial optimization problems such as the TSP, the feasible solutions can be encoded as points in a set $X \subseteq \{0,1\}^d$ in such a way that solving an instance of the problem amounts to optimizing a linear objective function over X, with coefficients given by the instance. By taking the convex hull of X, one obtains a polytope $P := \text{conv}(X)$ (see Appendix A for background on polytopes). Optimizing any linear function $f(x)$ over X is equivalent to optimizing this function $f(x)$ over $P = \text{conv}(X)$.

For example, for the TSP, we have a set $X \subseteq \{0,1\}^{\binom{n}{2}}$ of 0/1-points that correspond to a Hamiltonian cycle in the complete n-vertex graph K_n. The convex hull of these points is the *TSP polytope* $\text{TSP}(n) = \text{conv}(X)$. An instance of the TSP is given by the set of edge-weights w_{ij}. Solving this instance amounts to minimizing $f(x) := \sum_{i<j} w_{ij}x_{ij}$ over all $x \in \text{TSP}(n)$. This minimum is attained at a vertex of the polytope, that is, at a point $x \in X$.

The idea of representing the set of feasible solutions of a problem by a polytope forms the basis of a standard and powerful methodology in combinatorial optimization, see, for example, Schrijver [2003].

Extended Formulations and Extensions. Resuming the previous discussion (and assuming that the problem is a minimization problem), we have $\min\{f(x) \mid x \in X\} = \min\{f(x) \mid x \in P\} = \min\{f(x) \mid Ax \leqslant b\}$, where $Ax \leqslant b$ is any linear description of P. This turns any given instance of the combinatorial optimization problem into an LP, however, over an implicit system of constraints, the LP is potentially large since it has at least one inequality per facet of P. In fact, even for polynomially solvable problems, the associated polytope P may have an exponential number of facets.

By working in an extended space, that is, considering extra variables $y \in \mathbb{R}^k$ besides the original variables $x \in \mathbb{R}^d$, it is often possible to decrease the number of constraints. In some cases, a polynomial increase in dimension can be traded for an exponential decrease in the number of constraints. This is the idea underlying extended formulations.

Formally, an *extended formulation* (EF) of a polytope $P \subseteq \mathbb{R}^d$ is a linear system

$$E^{=}x + F^{=}y = g^{=},\ E^{\leqslant}x + F^{\leqslant}y \leqslant g^{\leqslant} \tag{1}$$

in variables $(x,y) \in \mathbb{R}^{d+k}$, where $E^{=}, F^{=}, E^{\leqslant}, F^{\leqslant}$ are real matrices with d, k, d, k columns respectively, and $g^{=}, g^{\leqslant}$ are column vectors, such that $x \in P$ if and only if there exists y such that (1) holds.

The *size* of an EF is defined as the number of *inequalities* in the system. Another possible definition of size would be the sum of the number of variables and total number of constraints (equalities plus inequalities) defining the EF. This would make little difference because if a polytope $P \subseteq \mathbb{R}^d$ has an EF with r inequalities, then it has an EF with $d+r$ variables, r inequalities and at most $d+r$ equalities (see Remark 3.1 for a proof). If we assume that P is full-dimensional (otherwise, one may cheat and make d artificially high), then $d \leqslant r$ and thus the two measures of size are within a constant of each other.

Notice that optimizing any (not necessarily linear) objective function $f(x)$ over all $x \in P$ amounts to optimizing $f(x)$ over all $(x, y) \in \mathbb{R}^{d+k}$ satisfying (1), provided (1) defines an EF of P.

Here, we often restrict to EFs in slack form, that is, containing only equalities and one nonnegativity inequality per additional variable:

$$Ex + Fy = g,\ y \geqslant \mathbf{0}. \tag{2}$$

The proof of the factorization theorem (Theorem 3) shows that this can be done without loss of generality, see Remark 3.1. In the following, we put EFs in slack form to ease the generalization to arbitrary cones. Notice that the size of an EF in slack form can equivalently be defined as the number of additional variables since the only inequalities are from $y \geqslant \mathbf{0}$.

An *extension* of the polytope P is another polytope $Q \subseteq \mathbb{R}^e$ such that P is the image of Q under a linear map. We define the *size* of an extension Q as the number of facets of Q. If P has an extension of size r, then it has an EF of size r. Conversely, it is known that if P has an EF of size r, then it has an extension of size at most r (see Theorem 3). In this sense, the concepts of EF and extension are equivalent.

The Impact of Extended Formulations. EFs have pervaded the areas of discrete optimization and approximation algorithms for a long time. For instance, Balas' disjunctive programming [Balas 1985], the Sherali-Adams hierarchy [Sherali and Adams 1990], the Lovász-Schrijver closures [Lovász and Schrijver 1991], lift-and-project [Balas et al. 1993], and configuration LPs are all based on the idea of working in an extended space. Recent surveys on EFs in the context of combinatorial optimization and integer programming are Conforti et al. [2010], Vanderbeck and Wolsey [2010], Kaibel [2011], and Wolsey [2011].

Symmetry Matters. Yannakakis [1991] proved a $2^{\Omega(n)}$ lower bound on the size of any *symmetric* EF of the TSP polytope TSP(n) (defined previously and in Section 3.4). Although he remarked that he did "not think that asymmetry helps much", it was recently shown by Kaibel et al. [2010] (see also Pashkovich [2009]) that symmetry is a restriction in the sense that there exist polytopes that have polynomial-size EFs but no polynomial-size symmetric EF. This revived Yannakakis's tantalizing question about unconditional lower bounds. That is, bounds which apply to the *extension complexity* of a polytope P, defined as the minimum size of an EF of P (irrespective of any symmetry assumption).

0/1-Polytopes with Large Extension Complexity. The strongest unconditional lower bounds so far were obtained by Rothvoss [2011]. By an elegant counting argument inspired by Shannon's theorem [Shannon 1949], it was proved that there exist 0/1-polytopes in $\mathbb{R}^d$ whose extension complexity is at least $2^{d/2-o(d)}$. However, Rothvoß's counting technique does not provide *explicit* 0/1-polytopes with an exponential extension complexity.

The Factorization Theorem. Yannakakis [1991] discovered that the extension complexity of a polytope P is determined by certain factorizations of an associated matrix,

called the *slack matrix* of P, that records for each pair (F, v) of a facet F and vertex v, the algebraic distance of v to a valid hyperplane supporting F. Defining the *nonnegative rank* of a matrix M as the smallest natural number r such that M can be expressed as $M = TU$ where T and U are nonnegative matrices (i.e., matrices whose elements are all nonnegative) with r columns (in case of T) and r rows (in case of U), respectively, it turns out that the extension complexity of every polytope P is exactly the nonnegative rank of its slack matrix.

We point out that this result generalizes to *any* slack matrix of the polytope, which may contain additional rows corresponding to faces F of P which are not facets and/or additional columns corresponding to points v of P that are not vertices. This fact is used in the proof of our lower bounds on extension complexity, starting with Theorem 7.

This *factorization theorem* led Yannakakis to explore connections between EFs and communication complexity. Let S denote the slack matrix of the polytope P. He proved that: (i) every deterministic communication protocol of complexity k computing S gives rise to an EF of P of size at most 2^k; (ii) the nondeterministic communication complexity of the *support matrix* of S (i.e., the binary matrix that has 0-entries exactly where S is 0) yields a lower bound on (the base-2 logarithm[1] of) the extension complexity of P, or more generally, the nondeterministic communication complexity of the support matrix of every nonnegative matrix M yields a lower bound on (the base-2 logarithm of) the nonnegative rank of M.[2]

Tighter Communication Complexity Connection. Faenza et al. [2011] proved that the base-2 logarithm of the nonnegative rank of a matrix equals, up to a small additive constant, the minimum complexity of a randomized communication protocol *with nonnegative outputs* that computes the matrix *in expectation*. In particular, every EF of size r can be regarded as such a protocol of complexity $\log r + O(1)$ bits that computes a slack matrix in expectation.

The Clique vs. Stable Set Problem. When P is the stable set polytope STAB(G) of a graph G (see Section 3.3), the slack matrix of P contains an interesting row-induced 0/1-submatrix that is the communication matrix of the *clique vs. stable set problem* (also known as the *clique vs. independent set problem*): its rows correspond to the cliques and its columns to the stable sets (or independent sets) of graph G. The entry for a clique K and stable set S equals $1 - |K \cap S|$. Yannakakis [1991] gave an $O(\log^2 n)$ deterministic protocol for the clique vs. stable set problem, where n denotes the number of vertices of G. This gives a $2^{O(\log^2 n)} = n^{O(\log n)}$ size EF for STAB(G) whenever the whole slack matrix is 0/1, that is, whenever G is a perfect graph.

A notoriously hard open question is to determine the communication complexity (in the deterministic or nondeterministic sense) of the clique vs. stable set problem. (For recent results that explain why this question is hard, see Kushilevitz and Weinreb [2009a, 2009b].) The best lower bound to this day is due to Huang and Sudakov [2012]: they obtained a $\frac{6}{5}\log n - O(1)$ lower bound. Furthermore, they state a graph-theoretical conjecture that, if true, would imply a $\Omega(\log^2 n)$ lower bound, and hence settle the communication complexity of the clique vs. stable set problem. Moreover it would give a worst-case $n^{\Omega(\log n)}$ lower bound on the extension complexity of stable set polytopes. However, a solution to the Huang-Sudakov conjecture seems far away.

[1]All logarithms in this article are in base 2.

[2]The classical nondeterministic communication complexity of a binary communication matrix is defined as $\lceil \log B \rceil$, where B is the minimum number of monochromatic 1-rectangles that cover the matrix, see Kushilevitz and Nisan [1997]. This last quantity is also known as the *rectangle covering bound*. It is easy to see that the rectangle covering bound of the support matrix of any matrix M lower bounds the nonnegative rank of M (see Theorem 4).

Factorization Theorem for General Cones. Gouveia et al. [2013] generalized Yannakakis's factorization theorem to other convex cones. There, the question is to know which polytopes $P \subseteq \mathbb{R}^d$ can be described via a *conic extended formulation*

$$Ex + Fy = g,\ y \in C \tag{3}$$

for some given closed, convex cone $C \subseteq \mathbb{R}^k$. Cone C is said to be nice if $C^* + F^\perp$ is closed for all faces F of C, where C^* is the *dual cone* of C. It is known that the nonnegative orthants and the PSD cones are nice. Gouveia et al. [2013] prove that, in case C is nice and P has dimension at least 1, such a conic EF exists if and only if the slack matrix S of P admits a factorization $S = TU$ where (the transpose of) each row of T is in C^* and each column of U is in C. This implies the following factorization theorem for *semidefinite EFs*: the *semidefinite extension complexity* of every polytope P equals the *PSD rank* of its slack matrix S (see Theorem 13).

1.2. Our Contribution

Our contribution in this article is twofold.

—First, inspired by earlier work [de Wolf 2003], we define a $2^n \times 2^n$ matrix $M = M(n)$ and show that the nonnegative rank of M is $2^{\Omega(n)}$ because the nondeterministic communication complexity of its support matrix is $\Omega(n)$. The latter was proved by de Wolf [2003] using the well-known disjointness lower bound of Razborov [1992]. We use the matrix M to prove a $2^{\Omega(n)}$ lower bound on the extension complexity of the cut polytope CUT(n) (Section 3.2). That is, we prove that every EF of the cut polytope has an exponential number of inequalities. Via reductions, we infer from this: (i) an infinite family of graphs G such that the extension complexity of the corresponding stable set polytope STAB(G) is $2^{\Omega(\sqrt{n})}$, where n denotes the number of vertices of G (Section 3.3); (ii) that the extension complexity of the TSP polytope TSP(n) is $2^{\Omega(\sqrt{n})}$ (Section 3.4).

In addition to simultaneously settling the previously mentioned open problems of Yannakakis [1991] and Rothvoss [2011], our results provide a lower bound on the extension complexity of stable set polytopes that goes much beyond what is implied by the Huang-Sudakov conjecture (thanks to the fact that we consider a different part of the slack matrix). Although our lower bounds are strong, unconditional, and apply to explicit polytopes that are well known in combinatorial optimization, they have very accessible proofs.

—Second, we generalize the tight connection between linear[3] EFs and classical communication complexity found by Faenza et al. [2011] to a tight connection between semidefinite EFs and quantum communication complexity.[4] We show that any *rank-r PSD factorization* of a (nonnegative) matrix M gives rise to a one-way quantum protocol computing M in expectation that uses $\log r + O(1)$ qubits and, conversely, that any one-way quantum protocol computing M in expectation that uses q qubits results in a PSD factorization of M of rank 2^q. Via the semidefinite factorization theorem, this yields a characterization of the semidefinite extension complexity of a polytope in terms of the minimum complexity of (one-way) quantum protocols that compute the corresponding slack matrix in expectation.

[3]In this paragraph, and later in Section 4, an EF (in the sense of the previous section) is called a *linear* EF. The use of adjectives such as "linear," "semidefinite," or "conic" will help distinguishing the different types of EFs.

[4]After a first version of this article appeared, Jain et al. [2013, Theorem 2] have used this notion of PSD rank to characterize the number of qubits of communication between Alice and Bob needed to generate a shared probability distribution.

Then, we give a complexity $\log r + O(1)$ quantum protocol for computing a nonnegative matrix M in expectation, whenever there exists a rank-r matrix N such that M is the entry-wise square of N. This implies in particular that every d-dimensional polytope with 0/1 slacks has a semidefinite EF of size $O(d)$.

Finally, we obtain an exponential separation between classical and quantum protocols that compute our specific matrix $M = M(n)$ in expectation. On the one hand, our quantum protocol gives a rank-$O(n)$ PSD factorization of M. On the other hand, the nonnegative rank of M is $2^{\Omega(n)}$ because the nondeterministic communication complexity of the support matrix of M is $\Omega(n)$. Thus, we also obtain an exponential separation between PSD rank and nonnegative rank.

We would like to point out that the lower bounds on the extension complexity of polytopes established in Section 3 were obtained by first finding an efficient PSD factorization or, equivalently, an efficient one-way quantum communication protocol for the matrix $M = M(n)$. In this sense, our classical lower bounds stem from quantum considerations somewhat similar in style to Kerenidis and de Wolf [2004], Aaronson [2006], and Aharonov and Regev [2004]. See Drucker and de Wolf [2011] for a survey of this line of work.

We would also like to point out that the fact that a matrix M with a rank-r entrywise square-root has a PSD-rank at most $r + O(1)$, which follows from Theorem 16, was also obtained by Gouveia et al. [2013], independently (since their results were not publicly available at the time we performed our research) and in a different context. Also, after a preprint of our paper had appeared, we learned that Klauck et al. [2011] had independently found a matrix (similar but not quite the same as ours) with an exponential separation between PSD rank and nonnegative rank.

1.3. Other Related and Subsequent Work

Yannakakis's paper has deeply influenced the TCS community. In addition to the works cited previously, it has inspired a whole series of papers on the quality of restricted *approximate* EFs, such as those defined by the Sherali-Adams hierarchies and Lovász-Schrijver closures starting with Arora et al. [2002] (Arora et al. [2006] for the journal version), see, for example, Buresh-Oppenheim et al. [2006], Schoenebeck et al. [2007], Fernandez de la Vega and Mathieu [2007], Charikar et al. [2009], Georgiou et al. [2009, 2010], and Benabbas and Magen [2010].

After the conference version of our article appeared, there has been a lot of follow-up work, including on approximations. Braun et al. [2012] developed a general framework for studying the power of approximate EFs, independent of specific hierarchies. In particular, via lower bounds on the extension complexity of approximations of the cut polytope, they showed that linear programs for approximating Max-Clique to within a factor $n^{1/2-\epsilon}$ need size at least $2^{\Omega(n^{\epsilon})}$. Similarly, they show the existence of a spectrahedron of small size that cannot be approximated by any LP with a polynomial number of inequalities within a factor of $n^{1/2-\epsilon}$. Braverman and Moitra [2013] used methods from information complexity to show the same size lower bound even for approximation factor $n^{1-\epsilon}$; Braun and Pokutta [2013] subsequently simplified and generalized their result and Braun et al. [2013b] show that the amortized log nonnegative rank is characterized by information. Such inapproximability results should be contrasted with Håstad's famous result [Håstad 1999] that it is hard to approximate Max-Clique to within a factor $n^{1-\epsilon}$: Håstad's result gives is a lower bound for all algorithms approximating Max-Clique and is conditional on the unproven assumption that RP $\neq$ NP, while the results of Braun et al. [2012] and Braverman and Moitra [2013] are geometric statements about the nonexistence of polynomial-size extended formulations.

Braun et al. [2013a] analyze the average-case polyhedral complexity of the maximum stable set problem showing that the extension complexity of the stable set polytope is high for almost all graphs. Pokutta and Van Vyve [2013] proved lower bounds on extension complexity for the knapsack problem, and Avis and Tiwary [2013] proved lower bounds for the subset-sum and three-dimensional matching problems, as well as others. Kaibel and Weltge [2013] gave a more direct proof of the lower bound for the cut polytope, via bounding the measure of the largest rectangle in the slack matrix, under the condition that this rectangle is contained in the support. However, they still use the same set of 2^n valid constraints that we use here (Lemma 6).

Chan et al. [2013] prove super-polynomial lower bounds on approximate EFs for MAX CSPs (constraint satisfaction problems). In particular, they prove that every $(2-\varepsilon)$-approximate (linear) EF for Max-Cut has $n^{\Omega(\frac{\log n}{\log\log n})}$ size. This is striking because the celebrated approximation algorithm of Goemans and Williamson [1995] is based on a $\Theta(n)$-size semidefinite EF with an approximation factor of at most 1.14. Again, the result of Chan et al. [2013] on Max-Cut matches the algorithmic hardness of the problem Khot et al. [2007], which assumes the Unique Games Conjecture.

Rothvoss [2014] proves that the matching polytope has extension complexity $2^{\Omega(n)}$, solving the second part of the main open problem in Yannakakis [1991]. This is the first time such a strong bound is obtained for a polytope over which one can optimize in polynomial time. Rothvoß's groundbreaking result implies in particular that the extension complexity of the TSP polytope is $2^{\Omega(n)}$, thus going beyond our $2^{\Omega(\sqrt{n})}$ lower bound.

Not much is known yet about lower bounds on *semidefinite* EFs. Extending the work of Rothvoss [2011] and Briët et al. [2013] show that most 0/1 polytopes (i.e., polytopes that are the convex hull of a random subset of $\{0,1\}^d$) need exponentially large semidefinite EFs. Fawzi and Parrilo [2013] give exponential lower bounds on the size of semidefinite EFs of explicit polytopes in a restricted setting, where the underlying cone is not the full PSD cone but rather a product of fixed-dimensional PSD cones. Lee and Theis [2012] obtain polynomial lower bounds based on the support pattern of slack matrices.

Finally, Fiorini et al. [2013] use the notion of conic extensions and its relation to communication complexity to study generalized probabilistic theories, which are different from the usual classical or quantum-mechanical theories, and show that all polynomially-definable 0/1-polytopes have small extension complexity with respect to the completely positive cone.

1.4. Organization

The discovery of our lower bounds on extension complexity crucially relied on finding the right matrix M and the right polytope whose slack matrix contains M. In our case, we found these through a connection with quantum communication. However, these quantum aspects are not strictly necessary for the resulting lower bound proof itself. Hence, in order to make the main results more accessible to those without background or interest in quantum computing, we start by giving a purely classical presentation of those lower bounds.

In Section 2, we define our matrix M and lower bound the nondeterministic communication complexity of its support matrix. In Section 3, we embed M in the slack matrix of the cut polytope in order to lower bound its extension complexity; further reductions then give lower bounds on the extension complexities of the stable set, and TSP polytopes. In Section 4, we establish the equivalence of PSD factorizations of a (nonnegative) matrix M and one-way quantum protocols that compute M in expectation, and give an efficient quantum protocol in the case where

some entry-wise square root of M has small rank. This is then used to provide an exponential separation between quantum and classical protocols for computing a matrix in expectation (equivalently, an exponential separation between nonnegative rank and PSD rank). Concluding remarks are given in Section 5.

2. A SIMPLE MATRIX WITH LARGE RECTANGLE COVERING BOUND

In this section we consider the following $2^n \times 2^n$ matrix $M = M(n)$ with rows and columns indexed by n-bit strings a and b, and real nonnegative entries:

$$M_{ab} := (1 - a^\intercal b)^2.$$

Note for later reference that M_{ab} can also be written as

$$M_{ab} = 1 - \langle 2\,\mathrm{diag}(a) - aa^\intercal, bb^\intercal \rangle, \tag{4}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product[5] and $\mathrm{diag}(a)$ is the $n \times n$ diagonal matrix with the entries of a on its diagonal. Let us verify this identity, using $a, b \in \{0,1\}^n$:

$$\begin{aligned} 1 - &\langle 2\,\mathrm{diag}(a) - aa^\intercal, bb^\intercal \rangle \\ &= 1 - 2\langle \mathrm{diag}(a), bb^\intercal \rangle + \langle aa^\intercal, bb^\intercal \rangle \\ &= 1 - 2a^\intercal b + (a^\intercal b)^2 = (1 - a^\intercal b)^2. \end{aligned}$$

Let $\mathrm{suppmat}(M)$ be the binary support matrix of M, so

$$\mathrm{suppmat}(M)_{ab} = \begin{cases} 1 & \text{if } M_{ab} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

For a given matrix, a rectangle is the Cartesian product of a set of row indices and a set of column indices. In de Wolf [2003] it was shown that an exponential number of (monochromatic) rectangles are needed to cover all the 1-entries of the support matrix of M. Equivalently, the corresponding function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ has nondeterministic communication complexity of $\Omega(n)$ bits. For the sake of completeness, we repeat the proof here.

THEOREM 1 [DE WOLF 2003]. *Every 1-monochromatic rectangle cover of* $\mathrm{suppmat}(M)$ *has size* $2^{\Omega(n)}$.

PROOF. Let $R_1, \ldots, R_k$ be a 1-cover for f, that is, a set of (possibly overlapping) 1-monochromatic rectangles in the matrix $\mathrm{suppmat}(M)$ that together cover all 1-entries in $\mathrm{suppmat}(M)$.

We use the following result from Kushilevitz and Nisan [1997, Example 3.22 and Section 4.6], which is essentially due to Razborov [1992].

> There exist sets $A, B \subseteq \{0,1\}^n \times \{0,1\}^n$ and probability distribution μ on $\{0,1\}^n \times \{0,1\}^n$ such that all $(a,b) \in A$ have $a^\intercal b = 0$, all $(a,b) \in B$ have $a^\intercal b = 1$, $\mu(A) = 3/4$, and there are constants $\alpha, \delta > 0$ (independent of n) such that for all rectangles R,
>
> $$\mu(R \cap B) \geqslant \alpha \cdot \mu(R \cap A) - 2^{-\delta n}.$$
>
> (For sufficiently large n, $\alpha = 1/135$ and $\delta = 0.017$ will do.)

[5]The *Frobenius inner product* is the component-wise inner product of two matrices. For matrices X and Y of the same dimensions, this equals $\mathrm{Tr}[X^\intercal Y]$. When X is symmetric, this can also be written $\mathrm{Tr}[XY]$.

Since the R_i are 1-rectangles, they cannot contain elements from B. Hence, $\mu(R_i \cap B) = 0$ and $\mu(R_i \cap A) \leqslant 2^{-\delta n}/\alpha$. However, since all elements of A are covered by the R_i, we have

$$\frac{3}{4} = \mu(A) = \mu\left(\bigcup_{i=1}^{k}(R_i \cap A)\right) \leqslant \sum_{i=1}^{k} \mu(R_i \cap A) \leqslant k \cdot \frac{2^{-\delta n}}{\alpha}.$$

Hence, $k \geqslant 2^{\Omega(n)}$. □

3. STRONG LOWER BOUNDS ON EXTENSION COMPLEXITY

Here we use the matrix $M = M(n)$ defined in the previous section to prove that the (linear) extension complexity of the cut polytope of the n-vertex complete graph is $2^{\Omega(n)}$, that is, every (linear) EF of this polytope has an exponential number of inequalities. Then, via reductions, we prove super-polynomial lower bounds for the stable set polytopes and the TSP polytopes. To start, let us define more precisely the slack matrix of a polytope. For a matrix A, let A_i denote the ith row of A and let A^j denote the jth column of A.

Let $P = \{x \in \mathbb{R}^d \mid Ax \leqslant b\} = \text{conv}(V)$ be a polytope, with $A \in \mathbb{R}^{m\times d}$, $b \in \mathbb{R}^m$ and $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$. Then $S \in \mathbb{R}_+^{m\times n}$ defined as $S_{ij} := b_i - A_i v_j$ with $i \in [m] := \{1, \ldots, m\}$ and $j \in [n] := \{1, \ldots, n\}$ is the *slack matrix* of P with respect to $Ax \leqslant b$ and V. We sometimes refer to the submatrix of the slack matrix induced by rows corresponding to facets and columns corresponding to vertices simply as *the* slack matrix of P, denoted by $S(P)$.

Recall that

(1) an *extended formulation* (EF) of P is a linear system in variables (x, y) such that $x \in P$ if and only if there exists y satisfying the system;
(2) an *extension* of P is a polytope $Q \subseteq \mathbb{R}^e$ such that there is a linear map $\pi : \mathbb{R}^e \to \mathbb{R}^d$ with $\pi(Q) = P$;
(3) the *extension complexity* of P is the minimum size (i.e., number of inequalities) of an EF of P.

We denote the extension complexity of P by $\text{xc}(P)$.

3.1. The Factorization Theorem

A *rank-r nonnegative factorization* of a (nonnegative) matrix M is a factorization $M = TU$ where T and U are nonnegative matrices with r columns (in case of T) and r rows (in case of U), respectively. The nonnegative rank of M, denoted by $\text{rank}_+(M)$, is thus simply the minimum rank among all nonnegative factorizations of M. Note that $\text{rank}_+(M)$ is also the minimum r such that M is the sum of r nonnegative rank-1 matrices. In particular, the nonnegative rank of a matrix M is at least the nonnegative rank of any submatrix of M.

The following factorization theorem was proved by Yannakakis (see also Fiorini et al. [2011]). It can be stated succinctly as: $\text{xc}(P) = \text{rank}_+(S)$ whenever P is a polytope and S a slack matrix of P. We include a sketch of the proof for completeness and we will use the following lemma which follows easily from Farkas's Lemma [Schrijver 2003; Ziegler 1995] by first showing that $\mathbf{0}^\intercal x \leqslant 1$ can be derived from the system.

LEMMA 2. *Let $P = \{x \in \mathbb{R}^d \mid Ax \leqslant b\}$ be a (possibly unbounded) polyhedron that admits a direction $u \in \mathbb{R}^d$ with $-\infty < \min\{u^\intercal x \mid x \in P\} < \max\{u^\intercal x \mid x \in P\} < +\infty$, and $c^\intercal x \leqslant \delta$ a valid inequality for P. Then there exist nonnegative multipliers $\lambda \in \mathbb{R}^d$ such that $\lambda^\intercal A = c^\intercal$ and $\lambda^\intercal b = \delta$, that is, $c^\intercal x \leqslant \delta$ can be derived as a nonnegative combination from $Ax \leqslant b$. In particular, this holds whenever P is a polytope of dimension at least 1 or*

whenever P is an unbounded polyhedron that linearly projects to a polytope of dimension at least 1.

We are ready to state Yannakakis's factorization theorem.

THEOREM 3 [YANNAKAKIS 1991]. *Let $P = \{x \in \mathbb{R}^d \mid Ax \leqslant b\} = \text{conv}(V)$ be a polytope with $\dim(P) \geqslant 1$, and let S denote the slack matrix of P with respect to $Ax \leqslant b$ and V. Then the following are equivalent for all positive integers r:*

(i) S has nonnegative rank at most r;
(ii) P has an extension of size at most r (i.e., with at most r facets);
(iii) P has an EF of size at most r (i.e., with at most r inequalities).

PROOF. It should be clear that (ii) implies (iii). We prove that (i) implies (ii), and then that (iii) implies (i).

First, consider a rank-r^* nonnegative factorization $S = TU$ of the slack matrix of P, where $r^* \leqslant r$. Notice that we may assume that no column of T is zero, because otherwise r^* can be decreased. We claim that P is the image of

$$Q := \{(x, y) \in \mathbb{R}^{d+r^*} \mid Ax + Ty = b,\ y \geqslant \mathbf{0}\}$$

under the projection $\pi_x : (x, y) \mapsto x$ onto the x-space. We see immediately that $\pi_x(Q) \subseteq P$ since $Ty \geqslant \mathbf{0}$. To prove the inclusion $P \subseteq \pi_x(Q)$, it suffices to remark that for each point $v_j \in V$ the point (v_j, U^j) is in Q since

$$Av_j + TU^j = Av_j + b - Av_j = b \text{ and } U^j \geqslant \mathbf{0}.$$

Since no column of T is zero, Q is a polytope. Moreover, Q has at most $r^* \leqslant r$ facets, and is thus an extension of P of size at most r. This proves that (i) implies (ii).

Second, suppose that the system

$$E^{=}x + F^{=}y = g^{=},\ E^{\leqslant}x + F^{\leqslant}y \leqslant g^{\leqslant}$$

with $(x, y) \in \mathbb{R}^{d+k}$ defines an EF of P with at most r inequalities. Let $Q \subseteq \mathbb{R}^{d+k}$ denote the set of solutions to this system. Thus, Q is a (not necessarily bounded) polyhedron. For each point $v_j \in V$, pick $w_j \in \mathbb{R}^k$ such that $(v_j, w_j) \in Q$. Because

$$Ax \leqslant b \iff \exists y : E^{=}x + F^{=}y = g^{=},\ E^{\leqslant}x + F^{\leqslant}y \leqslant g^{\leqslant},$$

each inequality in $Ax \leqslant b$ is valid for all points of Q. Let S_Q be the nonnegative matrix that records the slacks of the points (v_j, w_j) with respect to the inequalities of $E^{\leqslant}x + F^{\leqslant}y \leqslant g^{\leqslant}$, and then of $Ax \leqslant b$. By construction, the submatrix obtained from S_Q by deleting the rows corresponding to the inequalities of $E^{\leqslant}x + F^{\leqslant}y \leqslant g^{\leqslant}$ and leaving only those corresponding to the inequalities of $Ax \leqslant b$ is exactly S, thus $\text{rank}_+(S) \leqslant \text{rank}_+(S_Q)$. Furthermore, by Lemma 2, any valid inequality $c^\intercal x \leqslant \delta$ is a nonnegative combination of inequalities of the system $Ax \leqslant b$ and thus every row of S_Q is a nonnegative combination of the first r rows of S_Q. Thus, $\text{rank}_+(S_Q) \leqslant r$. Therefore, $\text{rank}_+(S) \leqslant r$. Hence, (iii) implies (i). □

Remark 3.1. By the factorization theorem, if polytope $P \subseteq \mathbb{R}^d$ has an EF of size r, then its slack matrix S has a nonnegative factorization $S = TU$ of rank r. But then $Ax + Ty = b,\ y \geqslant \mathbf{0}$ is an EF of P in slack form with $d+r$ variables, r inequalities and m equalities, where m is the number of rows in the linear description $Ax \leqslant b$ of P. Notice that if $m > d + r$ some of these equalities will be redundant, and that there always exists a subset of at most $d + r$ equalities defining the same subspace. By removing redundant equalities from the EF, we can assume that there are at most $d+r$ equalities in the EF.

We would like to emphasize that we will not restrict the slack matrix to have rows corresponding only to the facet-defining inequalities. This is not an issue since appending rows corresponding to redundant[6] inequalities does not change the nonnegative rank of the slack matrix. This fact was already used in the second part of the previous proof.

Theorem 3 shows in particular that we can lower bound the extension complexity of P by lower bounding the nonnegative rank of its slack matrix S; in fact, it suffices to lower bound the nonnegative rank of any submatrix of the slack matrix S corresponding to an implied system of inequalities. To that end, Yannakakis made the following connection with nondeterministic communication complexity. Again, we include the (easy) proof for completeness.

THEOREM 4 [YANNAKAKIS 1991]. *Let M be any matrix with nonnegative real entries and* suppmat(M) *its support matrix. Then* $\text{rank}_+(M)$ *is lower bounded by the rectangle covering bound for* suppmat(M).

PROOF. If $M = TU$ is a rank-r nonnegative factorization of M, then S can be written as the sum of r nonnegative rank-1 matrices:

$$S = \sum_{k=1}^{r} T^k U_k.$$

Taking the support on each side, we find

$$\begin{aligned} \text{supp}(S) &= \bigcup_{k=1}^{r} \text{supp}(T^k U_k) \\ &= \bigcup_{k=1}^{r} \text{supp}(T^k) \times \text{supp}(U_k). \end{aligned}$$

Therefore, suppmat(M) has a 1-monochromatic rectangle cover with r rectangles. □

3.2. Cut and Correlation Polytopes

Let $K_n = (V_n, E_n)$ denote the n-vertex complete graph. For a set X of vertices of K_n, we let $\delta(X)$ denote the set of edges of K_n with one endpoint in X and the other in its complement $\bar{X}$. This set $\delta(X)$ is known as the *cut* defined by X. For a subset F of edges of K_n, we let $\chi^F \in \mathbb{R}^{E_n}$ denote the *characteristic vector* of F, with $\chi_e^F = 1$ if $e \in F$ and $\chi_e^F = 0$ otherwise. The *cut polytope* CUT(n) is defined as the convex hull of the characteristic vectors of all cuts in the complete graph $K_n = (V_n, E_n)$. That is,

$$\text{CUT}(n) := \text{conv}\left\{\chi^{\delta(X)} \in \mathbb{R}^{E_n} \mid X \subseteq V_n\right\}.$$

We will not deal with the cut polytopes directly, but rather with 0/1-polytopes that are linearly isomorphic to them. Two polytopes are called *linearly isomorphic* if one can be obtained from the other by applying an invertible linear map. It is easy to check that, if P_1 and P_2 are linearly isomorphic, then they have same number of vertices and facets. Furthermore, any extended formulation for one can be converted to an extended formulation of the other using the same transformation. So any bound on the extension complexity of one polytope applies to any other polytope that is linearly isomorphic to it. The *correlation polytope* (or *Boolean quadric polytope*) COR(n) is defined as the convex

[6]An inequality of a linear system is called *redundant* if removing the inequality from the system does not change the set of solutions.

hull of all the rank-1 binary symmetric matrices of size $n \times n$. In other words,

$$\mathrm{COR}(n) := \mathrm{conv}\{bb^\intercal \in \mathbb{R}^{n\times n} \mid b \in \{0,1\}^n\}.$$

We use the following known result.

THEOREM 5 [DE SIMONE 1990]. *For all n,* COR(n) *is linearly isomorphic to* CUT(n+1).

Consider the matrix M defined in Section 2. Because M is nonnegative, Eq. (4) gives us a linear inequality that is satisfied by all vertices $bb^\intercal$ of COR(n), and hence (by convexity) is satisfied by all points of COR(n):

LEMMA 6. *For all $a \in \{0,1\}^n$, the inequality*

$$\langle 2\,\mathrm{diag}(a) - aa^\intercal, x\rangle \leqslant 1 \tag{5}$$

is valid for COR(n). *Moreover, the slack of vertex $x = bb^\intercal$ with respect to* (5) *is precisely M_{ab}.*

We remark that (5) is weaker than the *hypermetric inequality* [Deza and Laurent 1997] $\langle \mathrm{diag}(a) - aa^\intercal, x\rangle \leqslant 0$, in the sense that the face defined by the former is strictly contained in the face defined by the latter. Nevertheless, we persist in using (5). Now, we prove the main result of this section.

THEOREM 7. *There exists some constant $C > 0$ such that, for all n,*

$$\mathrm{xc}(\mathrm{CUT}(n+1)) = \mathrm{xc}(\mathrm{COR}(n)) \geqslant 2^{Cn}\,.$$

In particular, the extension complexity of CUT(n) *is* $2^{\Omega(n)}$.

PROOF. The equality is implied by Theorem 5. Now, consider any system of linear inequalities describing COR(n) starting with the 2^n inequalities (5), and a slack matrix S with respect to this system and $\{bb^\intercal \mid b \in \{0,1\}^n\}$. Next delete from this slack matrix all rows except the 2^n first rows. By Lemma 6, the resulting $2^n \times 2^n$ matrix is M. Using Theorems 3, 4, and 1, and the fact that the nonnegative rank of a matrix is at least the nonnegative rank of any of its submatrices, we have

$$\begin{aligned}\mathrm{xc}(\mathrm{COR}(n)) &= \mathrm{rank}_+(S)\\ &\geqslant \mathrm{rank}_+(M)\\ &\geqslant 2^{Cn}\end{aligned}$$

for some positive constant C. □

In their follow-up work, Kaibel and Weltge [2013] proved that one can take $C = \log(3/2) \approx 0.58$.

3.3. Stable Set Polytopes

A stable set S (also called an independent set) of a graph $G = (V, E)$ is a subset $S \subseteq V$ of the vertices such that no two of them are adjacent. For a subset $S \subseteq V$, we let $\chi^S \in \mathbb{R}^V$ denote the *characteristic vector* of S, with $\chi^S_v = 1$ if $v \in S$ and $\chi^S_v = 0$ otherwise. The stable set polytope, denoted STAB(G), is the convex hull of the characteristic vectors of all stable sets in G, that is,

$$\mathrm{STAB}(G) := \mathrm{conv}\{\chi^S \in \mathbb{R}^V \mid S \text{ stable set of } G\}.$$

Recall that a polytope Q is an extension of a polytope P if P is the image of Q under a linear projection.

LEMMA 8. *For each n, there exists a graph H_n with $O(n^2)$ vertices such that* STAB(H_n) *contains a face that is an extension of* COR(n).

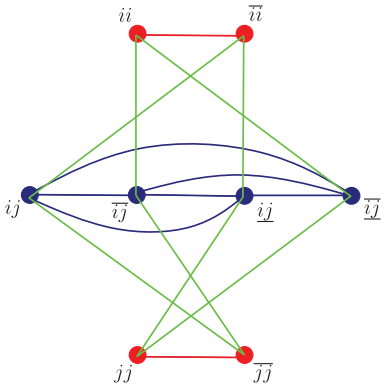


Fig. 1. The edges and vertices of H_n corresponding to vertices i, j and edge ij of K_n.

PROOF. Consider the complete graph K_n with vertex set $V_n := [n]$. For each vertex i of K_n, we create two vertices labeled $ii, \overline{ii}$ in H_n and an edge between them. Let us label the edges of K_n in the following way. The edge between vertices i and j with $i < j$ gets the label ij. Now, for each edge ij of K_n, we add to H_n four vertices labeled $ij, \overline{ij}, \underline{ij}, \underline{\overline{ij}}$ and all possible six edges between them. We further add the following eight edges to H_n:

$$\{ij, \overline{ii}\}, \{ij, \overline{jj}\}, \{\overline{ij}, ii\}, \{\overline{ij}, \overline{jj}\},$$
$$\{\underline{ij}, \overline{ii}\}, \{\underline{ij}, jj\}, \{\underline{\overline{ij}}, ii\}, \{\underline{\overline{ij}}, jj\}.$$

See Figure 1 for an illustration. The number of vertices in H_n is $2n + 4\binom{n}{2}$.

Thus, the vertices and edges of K_n are represented by cliques of size 2 and 4 respectively in H_n. We will refer to these as *vertex-cliques* and *edge-cliques*, respectively. Consider the face $F = F(n)$ of STAB(H_n) whose vertices correspond to the stable sets containing exactly one vertex in each vertex-clique and each edge-clique. (The vertices in this face correspond exactly to stable sets of H_n with maximum cardinality.)

Consider the linear map $\pi : \mathbb{R}^{V(H_n)} \to \mathbb{R}^{n \times n}$ mapping a point $x \in \mathbb{R}^{V(H_n)}$ to the point $y \in \mathbb{R}^{n \times n}$ such that $y_{ij} = y_{ji} = x_{ij}$ for $i \leqslant j$. In this equation, the subscripts in y_{ij} and y_{ji} refer to an ordered pair of elements in $[n]$, while the subscript in x_{ij} refers to a vertex of H_n that corresponds either to a vertex of K_n (if $i = j$) or to an edge of K_n (if $i \neq j$).

We claim that the image of F under π is COR(n), hence F is an extension of COR(n); observe that it suffices to consider 0/1 vertices as F is a 0/1 polytope and the projection is an orthogonal projection. Indeed, pick an arbitrary stable set S of H_n such that $x := \chi^S$ is on face F. Then, define $b \in \{0, 1\}^n$ by letting $b_i := 1$ if $ii \in S$ and $b_i := 0$ otherwise (i.e., $\overline{ii} \in S$). Notice that for the edge ij of K_n we have $ij \in S$ if and only if both vertices ii and jj belong to S. Hence, $\pi(x) = y = bb^\intercal$ is a vertex of COR(n). This proves $\pi(F) \subseteq$ COR(n). Now pick a vertex $y := bb^\intercal$ of COR(n) and consider the unique maximum stable set S that contains vertex ii if $b_i = 1$ and vertex $\overline{ii}$ if $b_i = 0$. Then, $x := \chi^S$ is a vertex of F with $\pi(x) = y$. Hence, $\pi(F) \supseteq$ COR(n). Thus, $\pi(F) =$ COR(n). This concludes the proof. □

Our next lemma establishes simple monotonicity properties of the extension complexity used in our reduction.

LEMMA 9. *Let P, Q, and F be polytopes. Then, the following hold:*

(*i*) *if F is an extension of P, then* $\text{xc}(F) \geqslant \text{xc}(P)$;
(*ii*) *if F is a face of Q, then* $\text{xc}(Q) \geqslant \text{xc}(F)$.

PROOF. The first part is obvious because every extension of F is in particular an extension of P. For the second part, notice that a slack matrix of F can be obtained from the (facet-vs-vertex) slack matrix of Q by deleting columns corresponding to vertices not in F. Now apply Theorem 3. □

Using previous results, we can prove the following result about the worst-case extension complexity of the stable set polytope.

THEOREM 10. *For all n, one can construct a graph G_n with n vertices such that the extension complexity of the stable set polytope* $\mathrm{STAB}(G_n)$ *is* $2^{\Omega(\sqrt{n})}$.

PROOF. Without loss of generality, we may assume $n \geqslant 18$. For an integer $p \geqslant 3$, let $f(p) := |V(H_p)| = 2p + 4\binom{p}{2}$. Given $n \geqslant 18$, we define p as the largest integer with $f(p) \leqslant n$. Now let G_n be obtained from H_p by adding $n - f(p)$ isolated vertices. Then $\mathrm{STAB}(H_p)$ is linearly isomorphic to a face of $\mathrm{STAB}(G_n)$. Using Theorem 7 in combination with Lemmas 8 and 9, we find that

$$\begin{aligned} \mathrm{xc}(\mathrm{STAB}(G_n)) &\geqslant \mathrm{xc}(\mathrm{STAB}(H_p)) \\ &\geqslant \mathrm{xc}(\mathrm{COR}(p)) \\ &= 2^{\Omega(p)} \\ &= 2^{\Omega(\sqrt{n})}. \quad \square \end{aligned}$$

3.4. TSP Polytopes

Recall that $\mathrm{TSP}(n)$, the *traveling salesman polytope* or *TSP polytope* of $K_n = (V_n, E_n)$, is defined as the convex hull of the characteristic vectors of all subsets $F \subseteq E_n$ that define a tour of K_n. That is,

$$\mathrm{TSP}(n) := \mathrm{conv}\{\chi^F \in \mathbb{R}^{E_n} \mid F \subseteq E_n \text{ is a tour of } K_n\}.$$

We now prove that the polytope $\mathrm{COR}(n)$ is the linear projection of a face of $\mathrm{TSP}(O(n^2))$, implying the following.

LEMMA 11. *For each n, there exists a positive integer $q = O(n^2)$ such that* $\mathrm{TSP}(q)$ *contains a face that is an extension of* $\mathrm{COR}(n)$.

PROOF. Recall that

$$\mathrm{COR}(n) = \mathrm{conv}\{bb^\intercal \in \mathbb{R}^{n\times n} \mid b \in \{0,1\}^n\}.$$

To prove the lemma, we start with constructing a graph G_n with $q = O(n^2)$ vertices such that the tours of G_n correspond to the $n \times n$ rank-1 binary symmetric matrices $bb^\intercal$, where $b \in \{0,1\}^n$. This is done in three steps:

(i) define a 3SAT formula ϕ_n with n^2 variables such that the satisfying assignments of ϕ_n bijectively correspond to the matrices $bb^\intercal$, where $b \in \{0,1\}^n$;
(ii) construct a directed graph D_n with $O(n^2)$ vertices such that each directed tour of D_n defines a satisfying assignment of ϕ_n, and conversely each satisfying assignment of ϕ_n has at least one corresponding directed tour in D_n;
(iii) modify the directed graph D_n into an undirected graph G_n in such a way that the tours of G_n bijectively correspond to the directed tours of D_n.

Step (i). For defining ϕ_n we use Boolean variables $C_{ij} \in \{0,1\}$ for $i, j \in [n]$ and let

$$\phi_n := \bigwedge_{\substack{i,j\in[n]\\ i\neq j}} [(C_{ii} \vee C_{jj} \vee \overline{C_{ij}}) \wedge (C_{ii} \vee \overline{C_{jj}} \vee \overline{C_{ij}}) \wedge (\overline{C_{ii}} \vee C_{jj} \vee \overline{C_{ij}}) \wedge (\overline{C_{ii}} \vee \overline{C_{jj}} \vee C_{ij})].$$

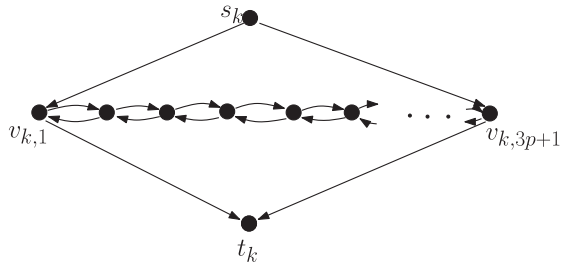


Fig. 2. Gadget for the kth variable occurring in p clauses.

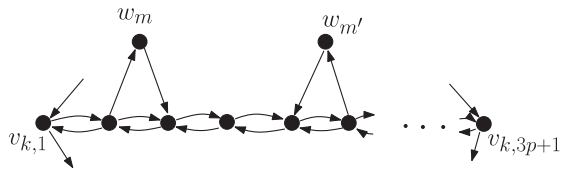


Fig. 3. Gadgets for clauses in case the kth variable appears negated in the mth clause and non-negated in the m'th clause.

The four clauses $(C_{ii} \vee C_{jj} \vee \overline{C_{ij}})$, $(C_{ii} \vee \overline{C_{jj}} \vee \overline{C_{ij}})$, $(\overline{C_{ii}} \vee C_{jj} \vee \overline{C_{ij}})$, and $(\overline{C_{ii}} \vee \overline{C_{jj}} \vee C_{ij})$ model the equation $C_{ij} = C_{ii} \wedge C_{jj}$. Hence, $C \in \{0,1\}^{n \times n}$ satisfies ϕ_n if and only if there exists $b \in \{0,1\}^n$ such that $C_{ij} = b_i \wedge b_j$ for all $i, j \in [n]$, or in matrix language, $C = bb^\intercal$.

Step (ii). To construct a directed graph D_n whose directed tours correspond to the satisfying assignments of ϕ_n we use the standard reduction from 3SAT to HAMPATH [Sipser 1996].

We order the variables of ϕ_n arbitrarily and construct a gadget for each variable as follows. Suppose that the kth variable occurs in p clauses. We create a chain of $3p+1$ nodes, labeled $v_{k,1}, \ldots, v_{k,3p+1}$, where each node $v_{k,\ell}$ with $\ell < 3p+1$ is connected to the next node $v_{k,\ell+1}$ with two opposite directed edges. Figure 2 illustrates this. Traversing this chain from left to right is interpreted as setting the kth variable to false and traversing it from right to left is interpreted as setting the kth variable to true. We also have two nodes s_k, t_k connected to this chain with directed edges $(s_k, v_{k,1})$, $(s_k, v_{k,3p+1})$, $(v_{k,1}, t_k)$ and $(v_{k,3p+1}, t_k)$ creating a diamond structure.

Next, we order the clauses of ϕ_n arbitrarily and create a node for each clause. The node for the mth clause is denoted by w_m. We connect these extra nodes to the gadgets for the variables as follows. Suppose, as before, that the kth variable appears in p clauses. Consider the ℓth of these clauses in which the kth variable appears, and let m be the index of that clause. If the kth variable appears negated in the mth clause then we add the path $v_{k,3\ell-1}, w_m, v_{k,3\ell}$. Otherwise, the kth variable appears nonnegated in the mth clause and we add the path $v_{k,3\ell}, w_m, v_{k,3\ell-1}$. Figure 3 illustrates this.

Next we connect the gadgets corresponding to the variables by identifying t_k with s_{k+1} for $1 \leqslant k < n^2$. Finally, we add a directed edge from t_{n^2} to s_1. Figure 4 illustrates the final directed graph obtained.

To see why the directed tours of the final directed graph D_n define satisfying assignments of our Boolean formula ϕ_n, observe that each directed tour of D_n encodes a truth assignment to the n^2 variables depending on which way the corresponding chains are traversed. Because a directed tour visits every node and because the node w_m corresponding to a clause can be visited only if we satisfy it, the truth assignment satisfies ϕ_n. Conversely, every satisfying assignment of ϕ_n yields at least one directed tour in D_n. (If the mth clause is satisfied by the value of more than one variable, we visit w_m only once, from the chain of the first variable whose value makes the clause satisfied.)

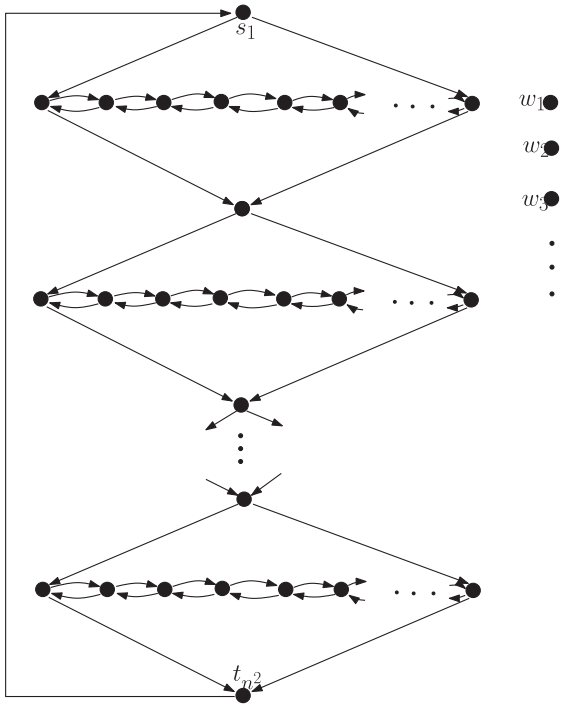


Fig. 4. Final graph. Directed edges incident to nodes w_m depend on the actual ordering of variables and clauses in Boolean formula ϕ_n.

Step (iii). For each node v of D_n we create a path $v_{\text{in}}, v_{\text{mid}}, v_{\text{out}}$ in the (undirected) graph G_n. For each directed edge (v, w) of D_n, we add to the graph G_n an edge between v_{out} and w_{in}. As is easily seen, the tours of G_n bijectively correspond to the directed tours of D_n. Note that G_n has $q := 3(n(n-1)\cdot 13 + n\cdot(3n-2) + n^2 + 4n(n-1)) = O(n^2)$ vertices.

Consider the face F of TSP(q) defined by setting to 0 all variables x_e corresponding to non-edges of G_n, so that the vertices of F are the characteristic vectors of the tours of G_n. To conclude the proof, we give a linear projection $\pi : x \mapsto y := \pi(x)$ mapping F to COR(n). For $x \in \mathbb{R}^{E_q}$ and $i, j \in [n]$, we let $y_{ij} = x_e$, where e is the edge $(v_{out,k,2}, v_{in,k,1})$ of G_n corresponding to the directed edge $(v_{k,2}, v_{k,1})$ and k is the index of the variable C_{ij} of ϕ_n. It follows from this discussion that π maps the face F of TSP(q) to COR(n). The lemma follows. □

The final theorem in this section follows from Theorem 7, Lemmas 9 and 11, using an argument similar to that used in the proof of Theorem 10.

THEOREM 12. *The extension complexity of the TSP polytope* TSP(n) *is* $2^{\Omega(\sqrt{n})}$.

4. QUANTUM COMMUNICATION AND PSD FACTORIZATIONS

In this section, we explain the connection with quantum communication. This yields results that are interesting in their own right, and also clarifies where the matrix M of Section 2 came from.

For a general introduction to quantum computation we refer to Nielsen and Chuang [2000] and to Mermin [2007], and for quantum communication complexity we refer to de Wolf [2002] and to Buhrman et al. [2010]. For our purposes, an r-dimensional *quantum state* ρ is an $r \times r$ PSD matrix of trace 1.[7] A *k-qubit state* is a state in dimension $r = 2^k$. If ρ has rank 1, it can be written as an outer product $|\phi\rangle\langle\phi|$ of some unit column

[7]For simplicity, we restrict to real rather than complex entries, which does not significantly affect the results.

vector $|\phi\rangle$ and its conjugate transpose $\langle\phi|$ (which is a row vector). This $|\phi\rangle$ is sometimes called a *pure state*. We use $|i\rangle$ to denote the pure state vector that has 1 at position i and 0s elsewhere. A quantum measurement (POVM) is described by a set of PSD matrices $\{E_\theta\}_{\theta\in\Theta}$, each labeled by a real number θ, and summing to the r-dimensional identity: $\sum_{\theta\in\Theta} E_\theta = I$. When measuring state ρ with this measurement, the probability of outcome θ equals $\mathrm{Tr}[E_\theta\rho]$. Note that if we define the PSD matrix $E := \sum_{\theta\in\Theta}\theta E_\theta$, then the *expected value* of the measurement outcome is $\sum_{\theta\in\Theta}\theta\,\mathrm{Tr}[E_\theta\rho] = \mathrm{Tr}[E\rho]$.

4.1. PSD Factorizations

Analogous to nonnegative factorizations and nonnegative rank, one can define PSD factorizations and PSD rank. A *rank-r PSD factorization* of an $m\times n$ matrix M is a collection of $r\times r$ symmetric positive semidefinite matrices $T_1,\ldots,T_m$ and $U^1,\ldots,U^n$ such that the Frobenius product $\langle T_i, U^j\rangle = \mathrm{Tr}[(T_i)^\intercal U^j] = \mathrm{Tr}[T_i U^j]$ equals M_{ij} for all $i\in[m]$, $j\in[n]$. The *PSD rank* of M is the minimum r such that M has a rank-r PSD factorization. We denote this by $\mathrm{rank}_{\mathrm{PSD}}(M)$.

Here, we show that $\mathrm{rank}_{\mathrm{PSD}}(M)$ can be expressed in terms of the amount of communication needed by a one-way quantum communication protocol for computing M in expectation (Corollary 15). Before doing so, we state the geometric interpretation of $\mathrm{rank}_{\mathrm{PSD}}(M)$ when M is a slack matrix.

For a positive integer r, we let $\mathbb{S}^r_+$ denote the cone of $r\times r$ symmetric positive semidefinite matrices embedded in $\mathbb{R}^{r(r+1)/2}$ in such a way that, for all $y,z\in\mathbb{S}^r_+$, the scalar product $z^\intercal y$ is the Frobenius product of the corresponding matrices. A *semidefinite EF of size r* is a conic EF with respect to $C=\mathbb{S}^r_+$, that is, a system $Ef+Fy=g$, $y\in\mathbb{S}^r_+$ such that $P=\{x\in\mathbb{R}^d \mid \exists y: Ef+Fy=g,\ y\in\mathbb{S}^r_+\}$. We call the set $Q=\{(x,y)\in\mathbb{R}^{d+r(r+1)/2} \mid Ex+Fy=g,\ y\in\mathbb{S}^r_+\}$ a *semidefinite extension of P*. The *semidefinite extension complexity* of polytope P, denoted by $\mathrm{xc}_{\mathrm{SDP}}(P)$, is the minimum r such that P has a semidefinite EF of size r. Observe that $(\mathbb{S}^r_+)^* = \mathbb{S}^r_+$.

The following result follows from Gouveia et al. [2013]:

THEOREM 13. *Let $P=\{x\in\mathbb{R}^d \mid Ax\leqslant b\}=\mathrm{conv}(V)$ be a polytope of dimension at least 1. Then the slack matrix S of P with respect to $Ax\leqslant b$ and V has a factorization $S=TU$ so that $(T_i)^\intercal, U^j\in\mathbb{S}^r_+$ if and only if there exists a semidefinite extension $Q=\{(x,y)\in\mathbb{R}^{d+r(r+1)/2} \mid Ex+Fy=g,\ y\in\mathbb{S}^r_+\}$ with $P=\pi_x(Q)$.*

4.2. Quantum Protocols

A *one-way quantum protocol with r-dimensional messages* can be described as follows. On input i, Alice sends Bob an r-dimensional state ρ_i. On input j, Bob measures the state he receives with a POVM $\{E^j_\theta\}$ for some *nonnegative* values θ, and outputs the result. We say that such a protocol *computes a matrix M in expectation*, if the expected value of the output on respective inputs i and j, equals the matrix entry M_{ij}. Analogous to the equivalence between classical protocols and nonnegative factorizations of M established by Faenza et al. [2011], such quantum protocols are essentially equivalent to PSD factorizations of S.

THEOREM 14. *Let $M\in\mathbb{R}^{m\times n}_+$ be a matrix. Then the following holds.*

(*i*) *A one-way quantum protocol with r-dimensional messages that computes M in expectation, gives a rank-r PSD factorization of M.*
(*ii*) *A rank-r PSD factorization of M gives a one-way quantum protocol with $(r+1)$-dimensional messages that computes M in expectation.*

PROOF. The first part is straightforward. Given a quantum protocol as above, define $E^j := \sum_{\theta\in\Theta} \theta E^j_\theta$. Clearly, on inputs i and j the expected value of the output is $\mathrm{Tr}[\rho_i E^j] = M_{ij}$.

For the second part, suppose we are given a PSD factorization of a matrix M, so we are given PSD matrices $T_1, \ldots, T_m$ and $U^1, \ldots, U^n$ satisfying $\mathrm{Tr}[T_i U^j] = M_{ij}$ for all i, j. In order to turn this into a quantum protocol, define $\tau = \max_i \mathrm{Tr}[T_i]$. Let ρ_i be the $(r+1)$-dimensional quantum state obtained by adding a $(r+1)$st row and column to T_i/τ, with $1 - \mathrm{Tr}[T_i]/\tau$ as $(r+1)$st diagonal entry, and 0s elsewhere. Note that ρ_i is indeed a PSD matrix of trace 1, so it is a well-defined quantum state. For input j, derive Bob's $(r+1)$-dimensional POVM from the PSD matrix U^j as follows. Let λ be the largest eigenvalue of U^j, and define $E^j_{\tau\lambda}$ to be U^j/λ, extended with a $(r+1)$st row and column of 0s. Let $E^j_0 = I - E^j_{\tau\lambda}$. This is positive semidefinite because the largest eigenvalue of $E^j_{\tau\lambda}$ is 1. Hence, the two operators $E^j_{\tau\lambda}$ and E^j_0 together form a well-defined POVM. The expected outcome (on inputs i, j) of the protocol induced by the states and POVMs that we just defined, is

$$\tau\lambda \,\mathrm{Tr}\left[E^j_{\tau\lambda}\rho_i\right] = \mathrm{Tr}[T_i U^j] = M_{ij},$$

so the protocol indeed computes M in expectation. □

We obtain the following corollary which summarizes the characterization of semidefinite EFs:

COROLLARY 15. *For a polytope P with slack matrix S, the following are equivalent:*

(*i*) *P has a semidefinite extension $Q = \{(x, y) \in \mathbb{R}^{d+r(r+1)/2} \mid Ex + Fy = g, y \in \mathbb{S}^r_+\}$;*
(*ii*) *the slack matrix S has a rank-r PSD factorization;*
(*iii*) *there exists a one-way quantum communication protocol with $(r+1)$-dimensional messages (i.e., using $\lceil \log(r+1) \rceil$ qubits) that computes S in expectation (for the converse we consider r-dimensional messages).*

4.3. A General Upper Bound on Quantum Communication

Now we provide a quantum protocol that efficiently computes a nonnegative matrix M in expectation, whenever there is a low rank matrix N whose entry-wise square is M.

THEOREM 16. *Let M be a matrix with nonnegative real entries, N be a rank-r matrix of the same dimensions such that $M_{ij} = N^2_{ij}$. Then there exists a one-way quantum protocol using $(r+1)$-dimensional pure-state messages that computes M in expectation.*

PROOF. By Corollary 15, it suffices to give a rank-r PSD factorization of M. To this end, let t_i, u_j be r-dimensional real vectors such that $N_{ij} = t_i^\intercal u_j$; such vectors exist because N has rank r. Define $r \times r$ PSD matrices $T_i := t_i t_i^\intercal$ and $U^j := u_j u_j^\intercal$. Then

$$\mathrm{Tr}[T_i U^j] = \left(t_i^\intercal u_j\right)^2 = N^2_{ij} = M_{ij},$$

hence we have a rank-r PSD factorization of M. □

Note that, if M is a 0/1-matrix, then we may take $N = M$, hence any low-rank 0/1-matrix can be computed in expectation by an efficient quantum protocol. If this M is the slack matrix for a polytope $P \subseteq \mathbb{R}^d$, then it is easy to see that its rank is at most $d+1$: the slack $M_{ij} = b_i - A_i v_j$ of a constraint $A_i x \leqslant b_i$ with respect to a point $v_j \in P$ can be written as the inner product between the two $(d+1)$-dimensional vectors $(b_i, -A_i)$ and $(1, v_j)$. We thus obtain the following corollary (implicit in Theorem 4.2 of Gouveia et al. [2010]) which also implies a compact (i.e., polynomial size) semidefinite EF for the stable set polytope of perfect graphs, reproving the previously known result by

Lovász [1979, 2003]. We point out that the result still holds when $\dim(P)+2$ is replaced by $\dim(P)+1$, see Gouveia et al. [2013]; this difference is due to normalization.

COROLLARY 17. *Let P be a polytope such that $S(P)$ is a 0/1 matrix. Then* $\mathrm{xc}_{\mathrm{SDP}}(P) \leqslant \dim(P)+2$.

4.4. Quantum vs Classical Communication, and PSD vs Nonnegative Factorizations

We now give an example of an exponential separation between quantum and classical communication in expectation, based on the matrix M of Section 2. This result actually preceded and inspired the results in Section 3.

THEOREM 18. *For each n, there exists a nonnegative matrix $M \in \mathbb{R}^{2^n \times 2^n}$ that can be computed in expectation by a quantum protocol using $\log n + O(1)$ qubits, while any classical randomized protocol needs $\Omega(n)$ bits to compute M in expectation.*

PROOF. Consider the matrix $N \in \mathbb{R}^{2^n \times 2^n}$ whose rows and columns are indexed by n-bit strings a and b, respectively, and whose entries are defined as $N_{ab} = 1 - a^\intercal b$. Define $M \in \mathbb{R}_+^{2^n \times 2^n}$ by $M_{ab} = N_{ab}^2$. This M is the matrix from Section 2. Note that N has rank $r \leqslant n+1$ because it can be written as the sum of $n+1$ rank-1 matrices. Hence, Theorem 16 immediately implies a quantum protocol with $(n+2)$-dimensional messages that computes M in expectation.

For the classical lower bound, note that a protocol that computes M in expectation has positive probability of giving a nonzero output on input a, b if and only if $M_{ab} > 0$. With a message m in this protocol we can associate a rectangle $R_m = A \times B$ where A consists of all inputs a for which Alice has positive probability of sending m, and B consists of all inputs b for which Bob, when he receives message m, has positive probability of giving a nonzero output. Together these rectangles will cover exactly the nonzero entries of M. Accordingly, a c-bit protocol that computes M in expectation induces a rectangle cover for the support matrix of M of size 2^c. Theorem 1 lower bounds the size of such a cover by $2^{\Omega(n)}$, hence $c = \Omega(n)$. □

Together with Theorem 14 and the equivalence of randomized communication complexity (in expectation) and nonnegative rank established in Faenza et al. [2011], we immediately obtain an exponential separation between nonnegative rank and PSD rank.

COROLLARY 19. *For each n, there exists $M \in \mathbb{R}_+^{2^n \times 2^n}$, with* $\mathrm{rank}_+(M) = 2^{\Omega(n)}$ *and* $\mathrm{rank}_{\mathrm{PSD}}(M) = O(n)$.

In fact a simple rank-$(n+1)$ PSD factorization of M is the following: let $T_a := \binom{1}{-a}\binom{1}{-a}^\intercal$ and $U^b := \binom{1}{b}\binom{1}{b}^\intercal$, then $\mathrm{Tr}[T_a U^b] = (1 - a^\intercal b)^2 = M_{ab}$.

5. CONCLUDING REMARKS

In addition to proving the first unconditional super-polynomial lower bounds on the size of linear EFs for the cut polytope, stable set polytope, and TSP polytope, we demonstrate that the rectangle covering bound can prove strong results in the context of EFs. In particular, it can be super-polynomial in the dimension and the logarithm of the number of vertices of the polytope, settling an open problem of Fiorini et al. [2011].

The exponential separation between nonnegative rank and PSD rank that we prove here (Theorem 18) actually implies more than a super-polynomial lower bound on the extension complexity of the cut polytope. As noted in Theorem 5, the polytopes $\mathrm{CUT}(n)$ and $\mathrm{COR}(n-1)$ are affinely isomorphic. Let $Q(n)$ denote the polyhedron isomorphic (under the same affine map) to the polyhedron defined by (5) for $a \in \{0,1\}^n$. Then (i) *every* polytope (or polyhedron) that contains $\mathrm{CUT}(n)$ and is contained in $Q(n)$ has

exponential extension complexity; (ii) there exists a low complexity spectrahedron that contains CUT(n) and is contained in $Q(n)$. (A *spectrahedron* is any projection of an affine slice of the positive semidefinite cone.) This was used in Braun et al. [2012] to establish the existence of a spectrahedron that cannot be well approximated by linear programs of polynomial size.

An important problem also left open in Yannakakis [1991] is whether the perfect matching polytope has a polynomial-size linear EF. Yannakakis proved that every *symmetric* EF of this polytope has exponential size, a striking result given the fact that the perfect matching problem is solvable in polynomial time. He conjectured that asymmetry also does not help in the case of the perfect matching polytope. Because it is based on the rectangle covering bound, our argument does not yield a super-polynomial lower bound on the extension complexity of the perfect matching polytope. This question was recently answered in the affirmative in Rothvoss [2014], showing that the extension complexity of the perfect matching polytope is $2^{\Omega(n)}$. This groundbreaking result is based on a general lower bound called the *hyperplane separation bound*, which was used implicitly, for example, in Braun et al. [2012].

As mentioned at the end of the introduction, the new connections developed have already inspired much follow-up research in particular about *approximate* EFs. Here are two concrete questions left open for future work: (i) find a *slack matrix* that has an exponential gap between nonnegative rank and PSD rank; (ii) prove that the cut polytope has no polynomial-size *semidefinite* EF (that would rule out SDP-based algorithms for optimizing over the cut polytope, in the same way that this article ruled out LP-based algorithms).

Our final remark concerns the famous *log-rank conjecture* [Lovász and Saks 1993]. It states that the deterministic communication complexity of a (finite) Boolean matrix M is upper bounded by a polynomial in the logarithm of its rank $\mathrm{rank}(M)$. On the one hand, this conjecture is equivalent to the following statement: $\log(\mathrm{rank}_+(M)) \leqslant \mathrm{polylog}(\mathrm{rank}(M))$ for all Boolean matrices M. On the other hand, we know that $\mathrm{rank}_{\mathrm{PSD}}(M) = O(\mathrm{rank}(M))$ for all Boolean matrices M by Theorem 16. Using the interpretation of the nonnegative and PSD rank of M in terms of classical and quantum communication protocols computing M in expectation (see Faenza et al. [2011] and Theorem 14), we see that the log-rank conjecture is *equivalent* to the conjecture that classical protocols computing M in expectation are at most polynomially less efficient than quantum protocols. Accordingly, one way to prove the log-rank conjecture would be to give an efficient classical simulation of such quantum protocols for Boolean M (for *non-Boolean* M, we already exhibited an exponential separation in this article).

APPENDIX

A. BACKGROUND ON POLYTOPES

A *(convex) polytope* is a set $P \subseteq \mathbb{R}^d$ that is the convex hull $\mathrm{conv}(V)$ of a finite set of points V. Equivalently, P is a polytope if and only if P is bounded and the intersection of a finite collection of closed halfspaces. This is equivalent to saying that P is bounded and the set of solutions of a finite system of linear inequalities and possibly equalities (each of which can be represented by a pair of inequalities).

Let $P \subseteq \mathbb{R}^d$ be a polytope. A closed halfspace H^+ that contains P is said to be *valid* for P. In this case, the hyperplane H that bounds H^+ is also said to be *valid* for P. A *face* of P is either P itself or the intersection of P with a valid hyperplane. Every face of a polytope is again a polytope. A face is called *proper* if it is not the polytope itself. A *vertex* is a minimal nonempty face. A *facet* is a maximal proper face. An inequality $c^\intercal x \leqslant \delta$ is said to be *valid* for P if it is satisfied by all points of P. The face it defines

is $F := \{x \in P \mid c^\intercal x = \delta\}$. The inequality is called *facet-defining* if F is a facet. The *dimension* of a polytope P is the dimension of its affine hull $\mathrm{aff}(P)$.

Every (finite or infinite) set V such that $P = \mathrm{conv}(V)$ contains all the vertices of P. Conversely, letting $\mathrm{vert}(P)$ denote the vertex set of P, we have $P = \mathrm{conv}(\mathrm{vert}(P))$. Suppose now that P is *full dimensional*, that is, $\dim(P) = d$. Then, every (finite) system $Ax \leqslant b$ such that $P = \{x \in \mathbb{R}^d \mid Ax \leqslant b\}$ contains all the facet-defining inequalities of P, up to scaling by positive numbers. Conversely, P is described by its facet-defining inequalities.

If P is not full dimensional, these statements have to be adapted as follows. Every (finite) system describing P contains all the facet-defining inequalities of P, up to scaling by positive numbers and adding an inequality that is satisfied with equality by *all* points of P. Conversely, a linear description of P can be obtained by picking one inequality per facet and adding a system of equalities describing $\mathrm{aff}(P)$.

A 0/1-*polytope* in $\mathbb{R}^d$ is simply the convex hull of a subset of $\{0,1\}^d$.

A (convex) *polyhedron* is a set $P \subseteq \mathbb{R}^d$ that is the intersection of a finite collection of closed halfspaces. A polyhedron P is a polytope if and only if it is bounded.

For more background on polytopes and polyhedra, see the standard reference [Ziegler 1995].

Note added in proof. Lee et al. [2015] have very recently proved super-polynomial lower bounds on the semidefinite extension complexity of the cut, TSP, and stable set polytopes, thereby answering one of the open questions raised in Section 5.

ACKNOWLEDGMENTS

We thank Kota Ishihara for carefully reading the manuscript and pointing out an error in a previous version of the text. We thank Monique Laurent for information about hypermetric inequalities, and the three anonymous STOC'12 referees as well as one JACM referee for suggesting improvements to the text. Sebastian Pokutta would like to thank Alexander Martin for the inspiring discussions and support. Ronald de Wolf thanks Giannicola Scarpa and Troy Lee for useful discussions.

REFERENCES

S. Aaronson. 2006. Lower bounds for local search by quantum arguments. *SIAM J. Comput.* 35, 4, 804–824. (Earlier version in *STOC'04*).

D. Aharonov and O. Regev. 2004. Lattice problems in NP ∩ coNP. In *Proceedings of FOCS 2004*. 362–371.

S. Arora, B. Bollobás, and L. Lovász. 2002. Proving integrality gaps without knowing the linear program. In *Proceedings of FOCS 2002*. 313–322.

S. Arora, B. Bollobás, L. Lovász, and I. Tourlakis. 2006. Proving integrality gaps without knowing the linear program. *Theory Comput.* 2, 19–51.

D. Avis and H. R. Tiwary. 2013. On the Extension Complexity of Combinatorial Polytopes. In *Proceedings of ICALP(1) 2013*. 57–68.

E. Balas. 1985. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM J. Algeb. Disc. Meth.* 6, 466–486.

E. Balas, S. Ceria, and G. Cornuéjols. 1993. A lift-and-project algorithm for mixed 0-1 programs. *Math. Prog.* 58, 295–324.

S. Benabbas and A. Magen. 2010. Extending SDP integrality gaps to Sherali-Adams with applications to quadratic programming and MaxCutGain. In *Proceedings of IPCO 2010*. 299–312.

G. Braun, S. Fiorini, S. Pokutta, and D. Steurer. 2012. Approximation limits of linear programs (beyond hierarchies). In *Proceedings of FOCS 2012*. 480–489.

G. Braun, S. Fiorini, and S. Pokutta. 2013a. Average case polyhedral complexity of the maximum stable set problem. arXiv:1311.4001.

G. Braun, R. Jain, T. Lee, and S. Pokutta. 2013b. Information-theoretic approximations of the nonnegative rank. *ECCC Report no.* 158 (2013).

G. Braun and S. Pokutta. 2013. Common information and unique disjointness. In *Proceedings of FOCS 2013*. 688–697.

M. Braverman and A. Moitra. 2013. An information complexity approach to extended formulations. In *Proceedings of STOC 2013*. 161–170.

J. Briët, D. Dadush, and S. Pokutta. 2013. On the existence of 0/1 polytopes with high semidefinite extension complexity. In *Algorithms ESA 2013*, Lecture Notes in Computer Science, vol. 8125, Springer, 217–228.

H. Buhrman, R. Cleve, S. Massar, and R. de Wolf. 2010. Nonlocality and communication complexity. *Rev. Modern Phys.* 82, 665.

J. Buresh-Oppenheim, N. Galesi, S. Hoory, A. Magen, and T. Pitassi. 2006. Rank bounds and integrality gaps for cutting planes procedures. *Theory Comput.* 2, 65–90.

S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer. 2013. Approximate Constraint Satisfaction Requires Large LP Relaxations. In *Proceedings of FOCS 2013*, 350–359.

M. Charikar, K. Makarychev, and Y. Makarychev. 2009. Integrality gaps for Sherali-Adams relaxations. In *Proceedings of STOC 2009*. 283–292.

M. Conforti, G. Cornuéjols, and G. Zambelli. 2010. Extended formulations in combinatorial optimization. *4OR* 8, 1–48.

G. B. Dantzig. 1951. Maximization of a linear function of variables subject to linear inequalities. In *Activity Analysis of Production and Allocation*, Cowles Commission Monograph No. 13, John Wiley & Sons Inc., New York, 339–347.

C. De Simone. 1990. The cut polytope and the Boolean quadric polytope. *Disc. Math.* 79, 71–75.

M. M. Deza and M. Laurent. 1997. *Geometry of Cuts and Metrics*. Algorithms and Combinatorics Series, vol. 15, Springer-Verlag.

A. Drucker and R. de Wolf. 2011. Quantum proofs for classical theorems. *Theory Comput. Graduate Surveys* 2.

Y. Faenza, S. Fiorini, R. Grappe, and H. R. Tiwary. 2011. Extended formulations, non-negative factorizations and randomized communication protocols. arXiv:1105.4127.

H. Fawzi and P. A. Parrilo. 2013. Exponential lower bounds on fixed-size PSD rank and semidefinite extension complexity. arXiv:1311.2571.

W. Fernandez de la Vega and C. Mathieu. 2007. Linear programming relaxation of Maxcut. In *Proceedings of SODA 2007*. 53–61.

S. Fiorini, V. Kaibel, K. Pashkovich, and D. O. Theis. 2011. Combinatorial bounds on nonnegative rank and extended formulations. arXiv:1111.0444.

S. Fiorini, S. Massar, M. K. Patra, and H. R. Tiwary. 2013. Generalised probabilistic theories and conic extensions of polytopes. CoRR abs/1310.4125.

K. Georgiou, A. Magen, T. Pitassi, and I. Tourlakis. 2010. Integrality gaps of $2 - o(1)$ for vertex cover SDPs in the Lovász-Schrijver hierarchy. *SIAM J. Comput.* 39, 3553–3570.

K. Georgiou, A. Magen, and M. Tulsiani. 2009. Optimal Sherali-Adams gaps from pairwise independence. In *Proceedings of APPROX-RANDOM 2009*. 125–139.

M. X. Goemans and D. P. Williamson. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* 42, 1115–1145.

J. Gouveia, P. A. Parrilo, and R. R. Thomas. 2010. Theta bodies for polynomial ideals. *SIAM J. Optim.* 20, 2097–2118.

J. Gouveia, P. A. Parrilo, and R. R. Thomas. 2013. Lifts of convex sets and cone factorizations. *Math. Oper. Res.* 38, 2, 248–264.

J. Håstad. 1999. Clique is Hard to Approximate within $n^{1-\epsilon}$. *Acta Math.* 182, 105–142. (Earlier version in Proceedings of *FOCS 1996*.)

H. Huang and B. Sudakov. 2012. A counterexample to the Alon-Saks-Seymour conjecture and related problems. *Combinatorica* 32, 2, 205–219.

R. Jain, Y. Shi, Z. Wei, and S. Zhang. 2013. Efficient protocols for generating bipartite classical distributions and quantum states. *IEEE Trans. Inf. Theory* 59, 8, 5171–5178.

V. Kaibel. 2011. Extended formulations in combinatorial optimization. *Optima* 85, 2–7.

V. Kaibel, K. Pashkovich, and D. O. Theis. 2010. Symmetry matters for the sizes of extended formulations. In *Proceedings of IPCO 2010*. 135–148.

V. Kaibel and S. Weltge. 2013. A short proof that the extension complexity of the correlation polytope grows exponentially. *Discrete Computa. Geom.* 53, 2, 396–401.

N. Karmarkar. 1984. A new polynomial time algorithm for linear programming. *Combinatorica* 4, 373–395.

I. Kerenidis and R. de Wolf. 2004. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Syst. Sci.* 69, 3, 395–420. (Earlier version in *STOC 2003*).

L. G. Khachiyan. 1979. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR* 244, 5, 1093–1096.

S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. 2007. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.* 37, 1, 319–357.

H. Klauck, T. Lee, and S. Zhang. 2011. An explicit and exponential separation between randomized and quantum correlation complexities. Unpublished manuscript from Oct/Nov 2011. Personal communication between Troy Lee and Ronald de Wolf, December 2011 at Centre for Quantum Technologies, Singapore.

E. Kushilevitz and N. Nisan. 1997. *Communication Complexity*. Cambridge University Press.

E. Kushilevitz and E. Weinreb. 2009a. The communication complexity of set-disjointness with small sets and 0-1 intersection. In *Proceedings of FOCS 2009*. 63–72.

E. Kushilevitz and E. Weinreb. 2009b. On the complexity of communication complexity. In *Proceedings of STOC 2009*. 465–474.

J. R. Lee, P. Raghavendra, and D. Steurer. 2015. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of STOC 2015*.

T. Lee and D. O. Theis. 2012. Support-based lower bounds for the positive semidefinite rank of a nonnegative matrix. arXiv:1203.3961.

L. Lovász. 1979. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory* 25, 1–7.

L. Lovász. 2003. Semidefinite programs and combinatorial optimization. In *Recent Advances in Algorithms and Combinatorics*. CMS Books Math./Ouvrages Math, SMC Series, vol. 11, Springer, 137–194.

L. Lovász and M. Saks. 1993. Communication complexity and combinatorial lattice theory. *J. Comput. Syst. Sci.* 47, 322–349.

L. Lovász and A. Schrijver. 1991. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.* 1, 166–190.

N. D. Mermin. 2007. *Quantum Computer Science: An Introduction*. Cambridge University Press.

M. A. Nielsen and I. L. Chuang. 2000. *Quantum Computation and Quantum Information*. Cambridge University Press.

K. Pashkovich. 2009. Symmetry in extended formulations of the permutahedron. arXiv:0912.3446.

S. Pokutta and M. Van Vyve. 2013. A note on the extension complexity of the knapsack polytope. *Oper. Res. Lett.* 41, 4, 347–350.

A. A. Razborov. 1992. On the distributional complexity of disjointness. *Theoret. Comput. Sci.* 106, 2, 385–390.

T. Rothvoss. 2011. Some 0/1 polytopes need exponential size extended formulations. arXiv:1105.0036.

T. Rothvoss. 2014. The matching polytope has exponential extension complexity. In *Proceedings of STOC 2014*. 263–272.

G. Schoenebeck, L. Trevisan, and M. Tulsiani. 2007. Tight integrality gaps for Lovasz-Schrijver LP relaxations of vertex cover and max cut. In *Proceedings of STOC 2007*. 302–310.

A. Schrijver. 2003. *Combinatorial Optimization. Polyhedra and Efficiency*. Springer-Verlag.

C. E. Shannon. 1949. The synthesis of two-terminal switching circuits. *Bell Syst. Tech. J.* 25, 59–98.

H. D. Sherali and W. P. Adams. 1990. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Disc. Math.* 3, 411–430.

M. Sipser. 1996. *Introduction to the Theory of Computation*. PWS, Boston, MA.

E. R. Swart. 1986; revision 1987. P = NP. Tech. rep., University of Guelph.

F. Vanderbeck and L. A. Wolsey. 2010. Reformulation and decomposition of integer programs. In *50 Years of Integer Programming 1958-2008*, M. Jünger, Th, M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds., Springer, 431–502.

R. de Wolf. 2002. Quantum communication and complexity. *Theoret. Comput. Sci.* 287, 337–353.

R. de Wolf. 2003. Nondeterministic quantum query and communication complexities. *SIAM J. Comput.* 32, 681–699.

L. A. Wolsey. 2011. Using extended formulations in practice. *Optima* 85, 7–9.

M. Yannakakis. 1988. Expressing combinatorial optimization problems by linear programs (extended abstract). In *Proceedings of STOC 1988*. 223–228.

M. Yannakakis. 1991. Expressing combinatorial optimization problems by linear programs. *J. Comput. System Sci.* 43, 3, 441–466.

G. M. Ziegler. 1995. *Lectures on Polytopes*. Graduate Texts in Mathematics, vol. 152. Springer-Verlag.

Received July 2012; revised November 2013 and September 2014; accepted January 2015

B

EXTENDED FORMULATIONS, NONNEGATIVE FACTORIZATIONS, AND RANDOMIZED COMMUNICATION PROTOCOLS

The following article has appeared in *Mathematical Programming* and is included here as an appendix for completeness.

Math. Program., Ser. B (2015) 153:75–94
DOI 10.1007/s10107-014-0755-3

FULL LENGTH PAPER

Extended formulations, nonnegative factorizations, and randomized communication protocols

Yuri Faenza · Samuel Fiorini · Roland Grappe · Hans Raj Tiwary

Received: 28 May 2013 / Accepted: 27 January 2014 / Published online: 19 February 2014

Abstract An extended formulation of a polyhedron P is a linear description of a polyhedron Q together with a linear map π such that $\pi(Q) = P$. These objects are of fundamental importance in polyhedral combinatorics and optimization theory, and the subject of a number of studies. Yannakakis' factorization theorem (Yannakakis in J Comput Syst Sci 43(3):441–466, 1991) provides a surprising connection between extended formulations and communication complexity, showing that the smallest size of an extended formulation of P equals the nonnegative rank of its slack matrix S. Moreover, Yannakakis also shows that the nonnegative rank of S is at most 2^c, where c is the complexity of any *deterministic* protocol computing S. In this paper,

A previous and reduced version of this paper appeared in the Proceedings of ISCO 2012.

H. R. Tiwary: Postdoctoral Researcher of the *Fonds National de la Recherche Scientifique* (F.R.S.–FNRS).

Y. Faenza
Institut de mathématiques d'analyse et applications, EPFL, Lausanne, Switzerland
e-mail: yuri.faenza@epfl.ch

S. Fiorini
Département de Mathématique, Université libre de Bruxelles,
CP 216, Boulevard du Triomphe, 1050 Brussels, Belgium
e-mail: sfiorini@ulb.ac.be

R. Grappe
Laboratoire d'Informatique de Paris-Nord, UMR CNRS 7030, Institut Galilée,
Université Paris-Nord, Avenue Jean-Baptiste Clément, 93430 Villetaneuse, France
e-mail: roland.grappe@lipn.univ-paris13.fr

H. R. Tiwary (✉)
Department of Applied Mathematics (KAM), Institute of Theoretical Computer Science (ITI),
Charles University, Malostranské nám. 25, 11800 Prague 1, Czech Republic
e-mail: hansraj@kam.mff.cuni.cz

we show that the latter result can be strengthened when we allow protocols to be *randomized*. In particular, we prove that the base-2 logarithm of the nonnegative rank of any nonnegative matrix equals the minimum complexity of a randomized communication protocol computing the matrix in expectation. Using Yannakakis' factorization theorem, this implies that the base-2 logarithm of the smallest size of an extended formulation of a polytope P equals the minimum complexity of a randomized communication protocol computing the slack matrix of P in expectation. We show that allowing randomization in the protocol can be crucial for obtaining small extended formulations. Specifically, we prove that for the spanning tree and perfect matching polytopes, small variance in the protocol forces large size in the extended formulation.

Mathematics Subject Classifcation 52B05

1 Introduction

Extended formulations are a powerful tool for minimizing linear or, more generally, convex functions over polyhedra (see, e.g., Ziegler [28] for background on polyhedra and polytopes). Consider a polyhedron P in $\mathbb{R}^d$ and a convex function $f : \mathbb{R}^d \to \mathbb{R}$, that has to be minimized over P. If a small size linear description of P is known, then minimizing f over P can be done efficiently using an interior point algorithm, or the simplex algorithm if f is linear and theoretical efficiency is not required.

However, P can potentially have many facets. Or worse: it can be that no explicit complete linear description of P is known. This does not necessarily make the given optimization problem difficult. A fundamental result of Grötschel, Lovász and Schrijver [11] states that if there exists an efficient algorithm solving the separation problem for P, then optimizing over P can be done efficiently. However, this result uses the ellipsoid algorithm, which is not very efficient in practice. Thus it is desirable to avoid using the ellipsoid algorithm.

Now suppose that there exists a polyhedron Q in a higher dimensional space $\mathbb{R}^e$ such that P is the image of Q under a linear projection $\pi : \mathbb{R}^e \to \mathbb{R}^d$. The polyhedron Q together with the projection π defines an *extension* of P, while we call *extended formulation* of P any description of Q by means of linear inequalities and equations, together with the map π. Minimizing f over P amounts to minimizing $f \circ \pi$ over Q. If Q has few facets, then we can resort to an interior point algorithm or the simplex algorithm to solve the optimization problem. Of course, one should also take into account the size of the coefficients in the linear description of Q and in the matrix of π. But this can essentially be ignored for 0/1-polytopes P [21].

The success of extended formulations is due to the fact that a moderate increase in dimension can result in a dramatic decrease in the number of facets. For instance, P can have exponentially many facets, while Q has only polynomially many. We will see examples of this phenomenon later in this paper. For more examples, and background, see the recent surveys by Conforti, Cornuéjols and Zambelli [4], Kaibel [13] and Wolsey [26].

Extensions provide an interesting measure of how "complex" a polyhedron is: define the *size* of an extension Q of P as the number of facets of Q and the *extension*

complexity of a polyhedron P as the minimum size of any extension of P. Following [9], we denote this number by $\mathrm{xc}(P)$. The *size* of an extended formulation of P is the number of inequalities of the linear system (hence, neither equations nor variables are taken into account). Note that the size of an extended formulation is at least the size of the associated extension, and any extension Q has an extended formulation describing Q with the same size.

This paper builds on Yannakakis' seminal paper [24]. We briefly review his contribution, postponing formal definitions to Sect. 2. Because we mainly consider polytopes, we assume from now on that P is bounded, that is, P is a polytope. (This is not a major restriction.) Yannakakis' *factorization theorem* (Theorem 1) states that to each size-r extension of a polytope P corresponds a *rank-r nonnegative factorization* of some matrix $S(P)$ associated to P, called the *slack matrix*, and conversely to each rank-r nonnegative factorization of $S(P)$ corresponds a size-r extension of P. In particular, the extension complexity $\mathrm{xc}(P)$ equals the smallest rank of a nonnegative factorization of $S(P)$, that is, the *nonnegative rank* of $S(P)$.

In [24], Yannakakis also shows that every $\lg r$-complexity *deterministic* protocol computing a nonnegative matrix M determines a rank-r nonnegative factorization of M.[1] By the aforementioned factorization theorem, this implies that one can produce extended formulations (and hence upper bounds to the extension complexity) via deterministic communication protocols. Yannakakis used this to obtain a quasi-polynomial $n^{O(\log n)}$-size extension for the stable set polytope of a n-vertex perfect graph.

Our contribution The main goal of this paper is to strengthen the connection between nonnegative rank of matrices (and hence, extension complexity of polytopes) and communication protocols. First we give a brief overview of our results and then provide more details along with an outline of the paper. Our contribution is threefold:

- We pinpoint the "right" model of communication protocol, that exactly corresponds to nonnegative factorizations. We remark that this was done independently by Zhang [27]. Proving such a correspondence is an important conceptual step since it gives a third equivalent way to think about extensions of polytopes, besides projections of polytopes and nonnegative factorizations. Communication protocols are very versatile and we hope that this paper will convince discrete optimizers to add this tool to their arsenal.
- We provide examples of already known extensions, seen as communication protocols, and also of new extensions obtained from communication protocols.
- We prove that the randomization allowed in our protocols is sometimes necessary for obtaining small size extensions. We give a general condition under which small variance in the protocol implies that the size of the corresponding extension is large, which in particular applies to the perfect matching polytope and spanning tree polytope. This indicates that Yannakakis' approach for the stable set polytope of a perfect graph *cannot work* for the perfect matching polytope or spanning tree polytope, since his protocol is deterministic and hence the corresponding variance zero.

[1] Throughout this paper, we use lg for binary logarithm.

More specifically, we define a new model of *randomized communication protocols computing the matrix in expectation*. This generalizes the one used by Yannakakis in [24] (see Sect. 3; our definition differs substantially from the usual notion of of random protocol computing a matrix *with high probability*, which can be found e.g. in [16]). Our protocols perfectly model the relation between the nonnegative factorization of a matrix and communication complexity: in fact, we show that the base-2 logarithm of the nonnegative rank of any nonnegative matrix (rounded up to the next integer) *equals* the minimum complexity of a randomized communication protocol computing the matrix in expectation (Theorem 2). By Yannakakis' factorization theorem, this implies a new characterization of the extension complexity of polytopes (Corollary 3).

We then provide evidence that these protocols are substantially more powerful than the deterministic ones used, e.g., by Yannakakis. In fact, one can associate to each protocol a *variance* (see Sect. 3.3) which, roughly speaking, indicates the "amount of randomness" of the protocol: protocols with variance zero are deterministic protocols. We show that no compact formulation for the spanning tree polytope arises from protocols with small variance (see Sect. 6.3), while we provide a randomized protocol that produces the $O(n^3)$ formulation for the spanning tree polytope of K^n due to Martin [19] (see Sect. 5.2).

We also investigate the existence of compact extended formulation for the matching polytope—a fundamental open problem in polyhedral combinatorics. Yannakakis [24] (see also [14]) proved that every *symmetric* extension of the perfect matching polytope of the complete graph K^n has exponential size (we do not formally define *symmetric* here, since we shall not need it; the interested reader may refer to [24]). We show that a negative result similar to the one of the spanning tree polytope holds true for matchings: no compact formulation for the matching polytope arises from protocols with small variance (see Sect. 6). Thus, in particular, deterministic protocols cannot be used to provide compact extended formulations for the perfect matching polytope. We also provide a randomized protocol that produces a $O(1.42^n)$ formulation for the matching polytope implicit in Kaibel, Pashkovich and Theis [14] (see Sect. 5.3). The negative results on both the spanning tree and the matching polytopes are obtained via a general technique that exploits known negative results on the communication complexity of the set disjointness problem.

We would like to remark that the results contained in this paper were, at a conceptual level, an important stepping stone for the strong lower bounds on the extension complexities of the cut, stable set and TSP polytopes of Fiorini, Massar, Pokutta, Tiwary and de Wolf [8].

2 Preliminary definitions and results

2.1 The factorization theorem and related concepts

Consider a polytope P in $\mathbb{R}^d$ with m facets and n vertices. Let $h_1, \ldots, h_m$ be m affine functions on $\mathbb{R}^d$ such that $h_1(x) \geqslant 0, \ldots, h_m(x) \geqslant 0$ are all the facet-defining inequalities of P. Let also $v_1, \ldots, v_n$ denote the vertices of P. The *slack matrix* of P is the nonnegative $m \times n$ matrix $S = S(P) = (s_{ij})$ with $s_{ij} = h_i(v_j)$. Also note that the

facet-defining inequalities can be defined up to any positive scaling factor. It should be clear that such a scaling does not alter the non-negative rank of a matrix. To see this let $S = AB$ and let S' be a matrix obtained by multiplying the i-th row of S by $\lambda > 0$. Then, $S' = A'B$ where A' is obtained by multiplying the i-th row of A by λ.

A *rank-r nonnegative factorization* of a nonnegative matrix S is an expression of S as a product $S = AB$ where A and B are nonnegative matrices with r columns and r rows, respectively. The *nonnegative rank* of S, denoted by $\text{rank}_+(S)$, is the minimum nonnegative integer r such that S admits a rank-r nonnegative factorization [3]. Observe that the nonnegative rank of S can also be defined as the minimum nonnegative integer r such that S is the sum of r nonnegative rank-1 matrices.

In a seminal paper, Yannakakis [24] proved, among other things, that the extension complexity of a polytope is precisely the nonnegative rank of its slack matrix (see also [9]).

Theorem 1 (Yannakakis' factorization theorem) *For all polytopes P that are neither empty or a point,*

$$\text{xc}(P) = \text{rank}_+(S(P)).$$

Before going on, we sketch the proof of half of the theorem. Assuming $P = \{x \in \mathbb{R}^d : Ex \leqslant g\}$, consider a rank-$r$ nonnegative factorization $S(P) = FV$ of the slack matrix of P. Then it can be shown that $Q := \{(x, y) \in \mathbb{R}^{d+r} : Ex + Fy = g,\ y \geqslant \mathbf{0}\}$ is an extension of P. Notice that Q has at most r facets, and r extra variables.[2] Taking $r = \text{rank}_+(S(P))$ implies $\text{xc}(P) \leqslant \text{rank}_+(S(P))$. Moreover, since P is a polytope, one can also assume that Q is bounded, as shown by the following lemma.

Lemma 1 *Let $P = \{x \in \mathbb{R}^d : Ex \leqslant g\}$ be a polytope, let $S(P) = FV$ be a rank-r nonnegative factorization of the slack matrix of P with $r := \text{rank}_+(S(P))$, and let $Q := \{(x, y) \in \mathbb{R}^{d+r} : Ex + Fy = g,\ y \geqslant \mathbf{0}\}$. Then Q is bounded.*

Proof The polyhedron Q is unbounded if and only if its recession cone $\text{rec}(Q) = \{(x, y) \in \mathbb{R}^{d+r} : Ex + Fy = \mathbf{0}, y \geqslant \mathbf{0}\}$ contains some nonzero vector. Since P is bounded and the image of Q under the projection $(x, y) \mapsto x$ is P, we have $x = \mathbf{0}$ for every point $(x, y) \in \text{rec}(Q)$. Therefore, Q is unbounded if and only if the system $Fy = \mathbf{0}$, $y \geqslant \mathbf{0}$ has a solution $y \neq \mathbf{0}$. But any such y represents $\mathbf{0}$ as a non-trivial conical combination of the column vectors of F. Since F is nonnegative, this is only possible if one of the columns of F is identically zero, which would contradict the minimality of r. □

2.2 Polytopes relevant to this work

Now we describe briefly various families of polytopes relevant to this paper. For a more detailed description of these polytopes, we refer the reader to Schrijver [22].

[2] The extended formulation for Q given above potentially has a large number of equalities, but recall we only consider the number of inequalities in the size of the extended formulation. The reasons for this are twofold: first, one can ignore most of the equalities after picking a small number of linearly independent equalities; and second, our concern in this paper is mainly the *existence* of certain extensions.

Let I be a finite ground set. The *characteristic vector* of a subset $J \subseteq I$ is the vector $\chi^J \in \mathbb{R}^I$ defined as

$$\chi_i^J = \begin{cases} 1 & \text{if } i \in J \\ 0 & \text{if } i \notin J \end{cases}$$

for $i \in I$. For $x \in \mathbb{R}^I$, we let $x(J) := \sum_{i \in J} x_i$.

Throughout this section, $G = (V, E)$ denotes a (finite, simple, undirected) graph. For a subset of vertices $U \subseteq V$, we denote the edges of the subgraph induced by U as $E[U]$. The *cut* defined by U, denoted as $\delta(U)$, is the set of edges of G exactly one of whose endpoints is in U. That is,

$$E[U] = \{uv \in E : u \in U, v \in U\}, \text{ and}$$
$$\delta(U) = \{uv \in E : u \in U, v \notin U\}.$$

Later in this paper, we will often take G to be the *complete graph* K^n with vertex set $V(K^n) = [n] := \{1, \ldots, n\}$ and edge set $E(K^n) = \{ij : i, j \in [n], i \neq j\}$.

2.2.1 Spanning tree polytope

A *spanning tree* of G is a tree $T = (V(T), E(T))$ (i.e., a connected graph without cycles) whose set of vertices and edges respectively satisfy $V(T) = V$ and $E(T) \subseteq E$. The *spanning tree polytope* of G is the convex hull of the characteristic vectors of the spanning trees of G, i.e.,

$$\mathrm{P}_{\text{spanning tree}}(G) = \text{conv}\{\chi^{E(T)} \in \mathbb{R}^E : T \text{ spanning tree of } G\}.$$

Edmonds [6] showed that this polytope admits the following linear description (see also [22, page 861]):

$$\begin{aligned} x(E[U]) &\leqslant |U| - 1 && \text{for nonempty } U \subsetneq V, \\ x(E) &= |V| - 1, && \\ x_e &\geqslant 0 && \text{for } e \in E. \end{aligned}$$

This follows, e.g., from the fact that the spanning tree polytope of G is the base polytope of the graphic matroid of G.

2.2.2 Perfect matching polytope

A *perfect matching* of G is set of edges $M \subseteq E$ such that every vertex of G is incident to exactly one edge in M. The *perfect matching polytope* of the graph G is the convex hull of the characteristic vectors of the perfect matchings of G, i.e.,

$$\mathrm{P}_{\text{perfect matching}}(G) = \text{conv}\{\chi^M \in \mathbb{R}^E : M \text{ perfect matching of } G\}.$$

Edmonds [5] showed that the perfect matching polytope of G is described by the following linear constraints (see also [22, page 438]):

$$\begin{aligned} x(\delta(U)) &\geqslant 1 \quad \text{for } U \subseteq V \text{ with } |U| \text{ odd}, \ |U| \geqslant 3 \\ x(\delta(\{v\})) &= 1 \quad \text{for } v \in V, \\ x_e &\geqslant 0 \quad \text{for } e \in E. \end{aligned}$$

2.2.3 Stable set polytope

A *stable set* S (often also called an *independent set*) of G is a subset of the vertices such that no two of them are adjacent. A *clique* K of G is a subset of the vertices such that every two of them are adjacent. The *stable set polytope* $\mathrm{STAB}(G)$ of a graph $G(V, E)$ is the convex hull of the characteristic vectors of the stable sets in G, i.e.,

$$\mathrm{STAB}(G) = \mathrm{conv}\{\chi^S \in \mathbb{R}^V : S \text{ stable set of } G\}.$$

No complete linear description of the stable set polytope for arbitrary graphs is known. It is, however, known that the following inequalities are valid for $\mathrm{STAB}(G)$ for any graph G:

$$x(K) \leqslant 1 \quad \text{for cliques } K \text{ of } G, \tag{1}$$

$$x_v \geqslant 0 \quad \text{for } v \in V. \tag{2}$$

Inequalities (1) are called the *clique inequalities*. See Schrijver [22] for details.

A graph G is called *perfect* if the chromatic number of every induced subgraph equals the size of the largest clique of that subgraph. It is known that G is perfect if and only if inequalities (1) and (2) completely describe $\mathrm{STAB}(G)$ [2].

3 Communication complexity

We start by an overview of the standard model of deterministic communication protocols, as described in detail in the book by Kushilevitz and Nisan [16]. We follow this with a detailed description of our notion of a randomized protocol (with private random bits and nonnegative outputs) computing a function *in expectation*. This differs significantly from the standard definition in the literature where randomized protocols usually compute a function exactly *with high probability*.

3.1 Deterministic protocols

Let X, Y, and Z be arbitrary finite sets with $Z \subseteq \mathbb{R}_+$, and let $f : X \times Y \to Z$ be a function. Suppose that there are two players Alice and Bob who wish to compute $f(x, y)$ for some inputs $x \in X$ and $y \in Y$. Alice knows only x and Bob knows only y. They must therefore exchange information to be able to compute $f(x, y)$. (We assume that each player possesses unlimited computational power.)

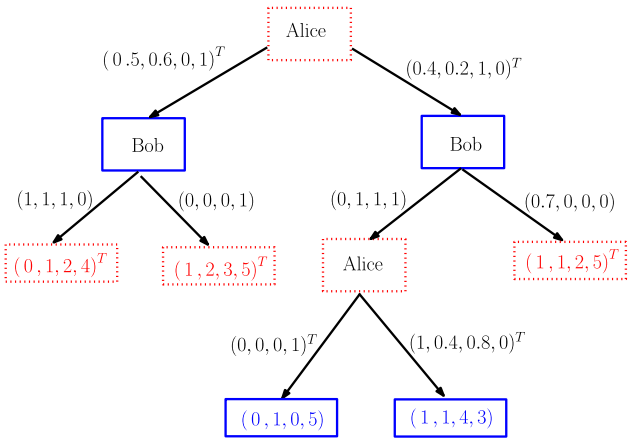
The communication is carried out as a protocol that is agreed upon beforehand by Alice and Bob, on the sole basis of the function f. At each step of the protocol, one of the players has the token. Whoever has the token sends a bit to the other player, that depends only on their input and on previously exchanged bits. This is repeated until the value of f on (x, y) is known to both players. The minimum number of bits exchanged between the players in the worst case to be able to evaluate f by any protocol is called the *communication complexity* of f.

3.2 Randomized protocols and computation in expectation

A protocol can be viewed as a rooted binary tree where each node is marked either Alice or Bob. The leaves have vectors associated with them. An execution of the protocol on a particular input is a path in the tree starting at the root. At a node owned by Alice, following the path to the left subtree corresponds to Alice sending a zero to Bob and taking the right subtree corresponds to Alice sending a one to Bob; and similarly for nodes owned by Bob.

More formally, we define a *randomized protocol (with private random bits and* nonnegative *outputs)* as a rooted binary tree with some extra information attached to its nodes. Let X and Y be finite sets, as above. Each node of the tree has a *type*, which is either X or Y. To each node v of type X are attached two function $p_{0,v}, p_{1,v} : X \to [0, 1]$; to each node v of type Y are attached two functions $q_{0,v}, q_{1,v} : Y \to [0, 1]$; and to each leaf v is attached a nonnegative vector Λ_v that is a column vector of size $|X|$ for leaves of type X and a row vector of size $|Y|$ for leaves of type Y. The functions $p_{i,v}$ and $q_{j,v}$ define *transition probabilities*, and we assume that $p_{0,v}(x) + p_{1,v}(x) \leqslant 1$ and $q_{0,v}(y) + q_{1,v}(y) \leqslant 1$. Figure 1 shows an example of a protocol.

An *execution* of the protocol on input $(x, y) \in X \times Y$ is a random path that starts at the root and descends to the left child of an internal node v with probability



X \ Y	y_1	y_2	y_3	y_4
x_1	0.28	0.40	1.60	1.70
x_2	0.74	0.68	0.92	1.44
x_3	1.40	0.80	3.20	2.40
x_4	4.00	4.00	4.00	5.00

(b)

Fig. 1 Illustration of a (non-optimal) randomized protocol computing a matrix in expectation, **a** protocol as a tree, **b** the associated communication matrix

$p_{0,v}(x)$ if v is of type X and $q_{0,v}(y)$ if v is of type Y, and to the right child of v with probability $p_{1,v}(x)$ if v is of type X and $q_{1,v}(y)$ if v is of type Y. With probability $1 - p_{0,v}(x) - p_{1,v}(x)$ and $1 - q_{0,v}(y) - q_{1,v}(y)$ respectively, the execution stops at v. For an execution stopping at leaf v with vector Λ_v, the *value* of the execution is defined as the entry of Λ_v that corresponds to input $x \in X$ if v is of type X, and $y \in Y$ if v is of type Y. For an execution stopping at an internal node, the *value* is defined to be 0.

For each fixed input $(x, y) \in X \times Y$, the value of an execution on input (x, y) is a random variable. If we let $Z \subseteq \mathbb{R}_+$ as before, we say that the protocol *computes* a function $f : X \times Y \to Z$ *in expectation* if the expectation of this random variable on each $(x, y) \in X \times Y$ is precisely $f(x, y)$.

The *complexity* of a protocol is the height of the corresponding tree.

Given an ordering $x_1, \ldots, x_m$ of the elements of X, and $y_1, \ldots, y_n$ of the elements of Y, we can visualize the function $f : X \times Y \to Z$ as a $m \times n$ nonnegative matrix $S = S(f)$ such that $S_{i,j} = f(x_i, y_j)$ for all $(i, j) \in [m] \times [n]$. The matrix S is called the *communication matrix* of f. Below, as is natural, we will not always make a distinction between a function and its communication matrix.

These formal definitions capture the informal ones given above. Observe that the nodes of type X are assigned to Alice, and those of type Y to Bob. Observe also that Alice and Bob have unlimited resources for performing their part of the computation. It is only the communication between the two players that is accounted for. When presenting a protocol, we shall often say that one of the two players sends an integer k rather than a binary value. This should be interpreted as the player sending the binary encoding of k or, as a (sub)tree of height $\lceil \lg k \rceil$. Finally, our definitions are such that the complexity of a protocol equals the number of bits exchanged by Alice and Bob.

3.3 Normalized variance

Since the output of a randomized protocol—as defined above—is a random variable, one can define its variance. However, we would like to refine the notion of variance so that protocols computing different scalings of the same matrix have the same variance. This is essential since the nonnegative rank of a matrix is an invariant under scaling and, as we will see in the next section, there is an equivalence between the nonnegative rank of a matrix S and the smallest complexity protocol computing S in expectation.

Let S be a nonnegative matrix and suppose there exists a protocol of complexity c computing S in expectation. Let $\xi_{i,j}$ denote the random variable corresponding to the output of the protocol on input $(x_i, y_j) \in X \times Y$. That is $\mathbb{E}[\xi_{i,j}] = S_{i,j}$. The *normalized variance* σ^2 of the protocol is defined as the maximum variance of the random variables $\xi'_{i,j} = \frac{\xi_{i,j}}{S_{i,j}}$ for the nonzero entries of S. That is

$$\sigma^2 = \max_{(i,j)|S_{i,j}\neq 0} \mathrm{Var}(\xi_{i,j}/S_{i,j})$$

4 Factorizations versus protocols

Theorem 2 *If there exists a randomized protocol of complexity c computing a matrix $S \in \mathbb{R}_+^{X\times Y}$ in expectation, then $\lg \operatorname{rank}_+(S) \leqslant c$. Conversely, if the nonnegative rank of matrix $S \in \mathbb{R}_+^{m\times n}$ is r, then there exists a randomized protocol computing S in expectation, whose complexity is at most $\lceil \lg r \rceil$. In other words, if $c_{\min}(S)$ denotes the minimum complexity of a randomized protocol computing S in expectation, we have*

$$c_{\min}(S) = \lceil \lg \operatorname{rank}_+(S) \rceil.$$

Proof Suppose there exists a protocol of complexity c computing S in expectation. Each node v of the protocol has a corresponding *traversal probability matrix* $P_v \in \mathbb{R}_+^{X\times Y}$ such that, for all inputs $(x, y) \in X \times Y$, the entry $P_v(x, y)$ is the probability that an execution on input (x, y) goes through node v.

Let $v_1, \ldots, v_k$ denote the nodes of type X on the unique path from the root to the parent of v, and let $w_1, \ldots, w_\ell$ denote the nodes of type Y on this path. Then we have

$$P_v(x, y) = \prod_{i=1}^{k} p_{\alpha_i, v_i}(x) \cdot \prod_{j=1}^{\ell} q_{\beta_j, w_j}(y),$$

where α_i is either 0 or 1 depending on if the path goes the left or right subtree at v_i, and similarly for β_j. We immediately see that P_v is a rank one matrix of the form $a_v b_v$ where a_v is a column vector of size $|X|$ and b_v is a row vector of size $|Y|$.

Finally, let L_X and L_Y be the set of all leaves of the protocol that are of type X and Y respectively and let Λ_v denote the (column or row) vector of values at a leaf $v \in L_X \cup L_Y$. Because the protocol computes S in expectation, for all inputs $(x, y) \in X \times Y$ we have $S(x, y) = \sum_{v\in L_X} \Lambda_v(x) P_v(x, y) + \sum_{w\in L_Y} P_w(x, y)\Lambda_w(y)$. Thus, $S = \sum_{v\in L_X} (\Lambda_v \circ a_v) b_v + \sum_{v\in L_Y} a_w (b_w \circ \Lambda_w)$, where $\circ$ denotes the Hadamard product. Therefore, it is possible to express S as a sum of at most $|L_X \cup L_Y| \leqslant 2^c$ nonnegative rank one matrices. Hence, $\operatorname{rank}_+(S) \leqslant 2^c$, that is, $\lg \operatorname{rank}_+(S) \leqslant c$.

To prove the other part of the theorem, let $A \in \mathbb{R}_+^{m\times r}$ and $B \in \mathbb{R}_+^{r\times n}$ be nonnegative matrices such that $S = AB$. By scaling, we can assume that the maximum row sum of A is 1. Otherwise, we replace A and B by $\Delta^{-1}A$ and ΔB respectively, where Δ denotes the maximum row sum of A.

The protocol is as follows: Alice knows a row index i, and Bob knows a column index j. Together they want to compute $S_{i,j}$ in expectation, by exchanging as few bits as possible. They proceed as follows. Let $\delta_i := \sum_k A_{i,k} \leqslant 1$. Alice selects a column index $k \in [r]$ according to the probabilities found in row i of matrix A, sends this index to Bob, and Bob outputs the entry of B in row k and column j. With probability $1 - \delta_i$ Alice does not send any index to Bob and the computation stops with implicit output zero (see Sect. 3.2).

This randomized protocol computes the matrix S in expectation. Indeed, the expected value on input (i, j) is $\sum_{k=1}^{r} A_{i,k} B_{k,j} = S_{i,j}$. Moreover, the complexity of the protocol is precisely $\lceil \lg(r) \rceil$. □

We would like to remark that our contruction of a factorization from a protocol is similar to the one used by Krause [17] to construct an approximate factorization from a protocol. However his discussion was limited to traditional definitions of a randomized protocol and hence could not produce exact factorizations.

The above theorem together with Theorem 1 gives us the following corollary:

Corollary 3 *Let P be a polytope with associated slack matrix $S = S(P)$, such that P is neither empty or a point. If there exists a randomized protocol of complexity c computing S in expectation, then $\mathrm{xc}(P) \leqslant 2^c$. Conversely, if $\mathrm{xc}(P) = r$, then there exists a randomized protocol computing S in expectation, whose complexity is at most $\lceil \lg r \rceil$. In other words, if $c_{\min}(S)$ denotes the minimum complexity of a randomized protocol computing S in expectation, we have*

$$c_{\min}(S(P)) = \lceil \lg \mathrm{xc}(P) \rceil.$$

The concrete polytopes considered in this paper have some facet-defining inequalities enforcing nonnegativity of the variables along with other facet-defining inequalities. The next lemma and its corollary will allow us to ignore the rows corresponding to nonnegativity inequalities, and focus on the non-trivial parts of the slack matrices.

Lemma 2 *Let S be a nonnegative matrix. Let R_1, R_2 be a partition of the rows of S defining partition of S into S_1 and S_2. If there exist randomized protocols computing S_1 and S_2 in expectation with complexity c_1 and c_2 respectively, then there exists a randomized protocol complexity computing S with complexity $1 + \max\{c_1, c_2\}$.*

Proof When Alice gets a row index of S she sends a bit to Bob to indicate whether the corresponding row lies in R_1 or R_2. Now that both Alice and Bob know whether they want to compute an entry in S_1 or S_2, they use the protocol for that particular submatrix. □

Corollary 4 *Let $P \subseteq \mathbb{R}^d_+$ be a polytope and let $S'(P)$ denote the submatrix of $S(P)$ obtained by deleting the rows corresponding to nonnegativity inequalities. If there is a complexity c randomized protocol for computing $S'(P)$ in expectation, then there is a complexity $1+\max\{c, \lceil \lg d \rceil\}$ randomized protocol for computing $S(P)$ in expectation.*

Proof For computing the part of $S(P)$ that is deleted in $S'(P)$, which corresponds to nonnegativity inequalities, we use the obvious protocol where Alice sends her row number to Bob and Bob computes the slack. Since at most d facets of P are defined by nonnegativity inequalities, this protocol has complexity $\lceil \lg d \rceil$. The corollary thus follows from Lemma 2. □

For the protocols constructed here, we will always have $c \geqslant \lceil \lg d \rceil$. Because of Corollary 4, we can thus ignore the nonnegativity inequalities without blowing up the size of any extension by more than a factor of 2. Moreover, in terms of lower bounds, it is always safe to ignore inequalities because the nonnegative rank of a matrix cannot increase when rows are deleted.

5 Examples

In this section, we give three illustrative examples of protocols defining nonnegative factorizations of various slack matrices, and thus (via Corollary 3) extensions of the corresponding polytopes. The first one gives a $O(n^3)$-size extension of the stable set polytope of a claw-free perfect graph. The second one is a reinterpretation of a well-known $O(n^3)$-size extended formulation for the spanning tree polytopes due to Martin [19]. Our interpretation allows for a more general result. In particular we prove new upper bounds for the spanning tree polytopes for minor-free graphs. The third one concerns the perfect matching polytopes and is implicit in Kaibel, Pashkovich and Theis [14].

5.1 The stable set polytope of a claw-free perfect graph

A graph G is called *claw-free* if no vertex has three pairwise non-adjacent neighbors. Even though the separation problem for $\text{STAB}(G)$ for claw-free graphs is polynomial-time solvable, no explicit description of all its facets is known (see, e.g., [22, page 1216]). Recently Faenza, Oriolo, and Stauffer [7] provided (non-compact) extended formulations for this polytope, while Galluccio et al. [10] gave a complete description of the facets for claw-free graphs with at least one stable set of size greater than or equal to four, and no clique-cutsets. Also, recall that for a perfect graph G the facets of $\text{STAB}(G)$ are defined by inequalities (1) and (2) (see Sect. 2.2.3).

Let G be a claw-free, perfect graph with n vertices. We give a deterministic protocol that computes the slack matrix of the stable set polytope $\text{STAB}(G)$ of G. Because G is perfect, the (non-trivial part of the) slack matrix of $\text{STAB}(G)$ has the following structure: it has one column per stable set S in G, and each one of its rows corresponds to a clique K in G. The entry for a pair (K, S) equals 0 if K and S intersect (in which case they intersect in exactly one vertex) and 1 if K and S are disjoint (note that we are ignoring the $|V|$ rows that correspond to nonnegativity inequalities (2). This can be done safely, see Corollary 4).

Consider the communication problem in which Alice is given a clique K of G, Bob is given a stable set S of G, and Alice and Bob together want to compute $1 - |K \cap S|$. Alice starts and sends the name of any vertex u of her clique K to Bob. Then Bob sends the names of all the vertices of his stable set S that are in $N(u) \cup \{u\}$ to Alice, where $N(u)$ denotes the neighborhood of u in G. Finally, Alice can compute $K \cap S$ because this intersection is contained in $N(u) \cup \{u\}$ and Alice knows all vertices of $S \cap (N(u) \cup \{u\})$. She outputs $1 - |K \cap S|$. Because G is claw-free, there are at most two vertices in $S \cap (N(u) \cup \{u\})$, thus at most $3 \lg n + O(1)$ bits are exchanged by Alice and Bob. It follows that there exists an extension (and hence, an extended formulation) of $\text{STAB}(G)$ of size $O(n^3)$. Notice that the normalized variance of our protocol is zero, because it is deterministic.

We obtain the following result.

Proposition 1 *For every perfect, claw-free graph G with n vertices,* $\text{STAB}(G)$ *has an extended formulation of size $O(n^3)$.*

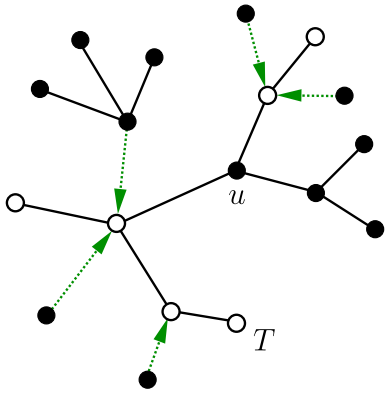


Fig. 2 Illustration of the protocol for the slack of MST polytope. The *black vertices* are those in U. The *green* directed edges are those for which Alice outputs a non-zero value. The number of such edges is the number of connected components of $T[U]$ minus one (color figure online)

5.2 The spanning tree polytope

Let $\mathrm{P}_{\text{spanning tree}}(G)$ denote the spanning tree polytope of a graph $G = (V, E)$ (see Sect. 2.2.1). The (non-trivial part of the) slack matrix of P has one column per spanning tree T and one row per proper nonempty subset U of vertices. The slack of T with respect to the inequality that corresponds to U is the number of connected components of the subgraph of T induced by U (denoted by $T[U]$ below) minus one.

In terms of the corresponding communication problem, Alice has a proper nonempty set U and Bob a spanning tree T. Together, they wish to compute the slack of the pair (U, T). Alice sends the name of some (arbitrarily chosen) vertex u in U. Then Bob picks an edge e of T uniformly at random and sends to Alice the endpoints v and w of e as an ordered pair of vertices (v, w), where the order is chosen in such a way that w is on the unique path from v to u in the tree. That is, he makes sure that the directed edge (v, w) "points" towards the root u. Then Alice checks that $v \in U$ and $w \notin U$, in which case she outputs $n - 1$; otherwise she outputs 0.

The resulting randomized protocol is clearly of complexity $\lg |V| + \lg |E| + O(1)$. Moreover, it computes the slack matrix in expectation because for each connected component of $T[U]$ distinct from that which contains u, there is exactly one directed edge (v, w) that will lead Alice to output a non-zero value, see Fig. 2 for an illustration. Since she outputs $(n-1)$ in this case, the expected value of the protocol on pair (U, T) is $(n-1) \cdot (k-1)/(n-1) = k-1$, where k is the number of connected components of $T[U]$. Therefore, we obtain the following result.

Proposition 2 *For every graph G with n vertices and m edges, $\mathrm{P}_{\text{spanning tree}}(G)$ has an extended formulation of size $O(mn)$.*

The above result is implicit in Martin [19], although the paper only states the following corollary. More specifically, variables $z_{i,j,k}$ such that ij is not an edge of G can be deleted from his $O(n^3)$-size extended formulation, so that the resulting formulation has size $O(mn)$.

Corollary 5 *$\mathrm{P}_{\text{spanning tree}}(K^n)$ has extended formulation of size $O(n^3)$, where K^n is the complete graph on n vertices.*

Corollary 6 *Let G be an H minor-free graph, where H is a graph with h vertices, then $\mathrm{P}_{\text{spanning tree}}(G)$ has extended formulation of size $O(n^2 h\sqrt{\lg h})$.*

Proof It is known that any H minor-free graph G with n vertices has at most $O(nh\sqrt{\lg h})$ edges, where h is the number of vertices of H [23]. The result follows. □

We remark that when G is planar, $P_{\text{spanning tree}}(G)$ has an extended formulation of size $O(n)$ [25]. It is natural to ask whether a linear size extended formulation also exists for general H minor-free graphs. So far, the best that seems to be known is the upper bound in Corollary 6.

Finally, it can be easily verified that the normalized variance of the protocol given above is $\sigma^2 = n - 2$, which is large compared to the previous protocol.

5.3 Perfect matching polytope

For the next example, we will need the fact that one can cover K^n with $k = O(2^{n/2}\text{poly}(n))$ balanced complete bipartite graphs $G_1,\ldots, G_k$ in such a way that every perfect matching of K^n is a perfect matching of at least one of the G_i's. We say that $X \subseteq [n]$ is an $(n/2)$-*subset* of $[n]$ if $|X| = n/2$. Given a matching M of K^n and a $(n/2)$-subset X of $[n]$, we say that X is *compatible* with M if all the edges of M have exactly one end in X.

Lemma 3 *Let n be an even positive integer. Then, there exists a collection of $k = O(2^{n/2}\sqrt{n}\ln n)$ $(n/2)$-subsets $X_1,\ldots, X_k$ of $[n]$ such that for every perfect matching M of K^n at least one of the subsets X_i is compatible with M.*

Proof Finding a minimum size such collection $X_1, \ldots, X_k$ amounts to solving a set covering instance that we formulate by an integer linear program. For each $(n/2)$-subset X, we define a variable binary variable $\lambda(X)$. For each perfect matching M, these variables have to satisfy the constraint $\sum\{\lambda(X) : X \text{ is compatible with } M\} \geqslant 1$. The goal is to minimize $\sum \lambda(X)$, the sum of all variables $\lambda(X)$.

A feasible fractional solution to this linear program is to let $\lambda^*(X) = 1/2^{n/2}$. This gives a feasible fractional solution because each perfect matching M is compatible with exactly $2^{n/2}$ $(n/2)$-subsets X, so $\sum\{\lambda^*(X) : X \text{ is compatible with } M\} = 2^{n/2}(1/2^{n/2}) = 1$. (By symmetry considerations, it is in fact possible to argue that this solution is actually optimal.) The cost of this fractional solution λ^* is

$$\sum \lambda^*(X) = \frac{1}{2^{n/2}}\binom{n}{n/2} \leqslant \frac{2^{n/2}}{\sqrt{n}},$$

for n sufficiently large. By Lovász's analysis of the greedy algorithm for the set covering problem [18], there exists a feasible integer solution λ of cost at most $(1 + \ln u)$ times the fractional optimum, where u is the number of elements to cover. By what precedes, this is at most

$$\left(1 + \ln \frac{n!}{2^{n/2}(n/2)!}\right)\frac{2^{n/2}}{\sqrt{n}} = O(2^{n/2}\sqrt{n}\lg n),$$

from which the result follows directly. □

Assume that n is even and let P denote the perfect matching polytope of the complete graph K^n with vertex set $[n]$, see Sect. 2.2.2. The (non-trivial part of the) slack matrix of P has one column per perfect matching M, and its rows correspond to odd sets $U \subseteq [n]$. The entry for a pair (U, M) is $|\delta(U) \cap M| - 1$ (recall that $\delta(U)$ denotes the set of edges that have one endpoint in U and the other endpoint in $\overline{U}$, the complement of U).

We describe a randomized protocol for computing the slack matrix in expectation, of complexity at most $(1/2 + \varepsilon)n$, where $\varepsilon > 0$ can be made as small as desired by taking n large. First, Bob finds an $(n/2)$-subset $X \subseteq [n]$ that is compatible with his matching M, and tells the name of this subset to Alice, see Lemma 3. Then Alice checks which of X and $\overline{X}$ contains the least number of vertices of her odd set U. Without loss of generality, assume it is X. If $U \cap X = \varnothing$ then, because $U \subseteq \bar{X}$ and X is compatible with M, Alice can correctly infer that the slack is $|U| - 1$, and outputs this number. Otherwise, she picks a vertex u of $U \cap X$ uniformly at random and send its name to Bob. He replies by sending the name of u', the mate of u in the matching M. Alice then checks whether u' is in U or not. If u' is not in U, then she outputs $|U| - 1$. Otherwise u' is in U, and she outputs $|U| - 1 - 2|U \cap X|$. Telling the name of X can be done in at most $n/2 + \lg \sqrt{n} + \lg \lg n + O(1)$ bits, see Lemma 3. The extra amount of communication is $2 \lg n + O(1)$ bits. In total, at most $(1/2 + \varepsilon)n$ bits are exchanged, for n sufficiently large ($\varepsilon > 0$ can be chosen arbitrarily).

Now, we check that the protocol correctly computes the slack matrix of the perfect matching polytope. Letting $E[U]$ denote the edges of the complete graph with both endpoints in U, the expected value output by Alice (in the case $U \cap X \neq \varnothing$) is

$$\begin{aligned}
&(|U| - 1)\frac{|U \cap X| - |E[U] \cap M|}{|U \cap X|} + (|U| - 1 - 2|U \cap X|)\frac{|E[U] \cap M|}{|U \cap X|} \\
&= |U| - 1 - 2|U \cap X|\frac{|E[U] \cap M|}{|U \cap X|} \\
&= |U| - 2|E[U] \cap M| - 1 \\
&= |\delta(U) \cap M| - 1.
\end{aligned}$$

We obtain the following result.

Proposition 3 *Let $\varepsilon > 0$. For every large enough even nonnegative integer n, the polytope $\mathrm{P}_{\text{perfect matching}}(K^n)$ has an extended formulation of size at most $2^{(1/2+\varepsilon)n}$.*

We remark that our extension has size at most $2^{(1/2+\varepsilon)n} \leqslant (1.42)^n$, whereas the main result of Yannakakis [24] gives a lower bound of $\binom{n}{n/4} \geqslant (1.74)^n$ for the size of any *symmetric* extension.

6 When low variance forces large size

We have seen that every extension of a polytope P corresponds to a randomized protocol computing its slack matrix $S = S(P)$ in expectation and vice-versa. Now we

show that if the set disjointness matrix can be embedded in a certain way in a matrix S (see below for definitions), then efficient protocols computing S in expectation necessarily have large variance. We prove that such an embedding can be found for the slack matrices of the perfect matching polytope and also, surprisingly, of the spanning tree polytope.

6.1 Embedding the set disjointness matrix

The *set disjointness problem* is the following communication problem: Alice and Bob each are given a subset of $[n]$. They wish to determine whether the two subsets intersect or not. In other words, Alice and Bob have to compute the *set disjointness matrix* DISJ defined by $\mathrm{DISJ}(A, B) = 1$ if A and B are disjoint subsets of $[n]$, and $\mathrm{DISJ}(A, B) = 0$ if A and B are non-disjoint subsets of $[n]$. The set disjointness problem plays a central role in communication complexity, comparable to the role played by the satisfiability problem in NP-completeness theory [1].

It is known that any randomized protocol that computes the disjointness function *with high probability* (that is, the probability that the value output by the protocol is correct is, for each input, bounded from below by a constant strictly greater than $1/2$) has $\Omega(n)$ complexity [15,20].

Consider a matrix $S \in \mathbb{R}_+^{X \times Y}$. An *embedding* of the set disjointness matrix on $[n]$ in S is defined by two maps $\alpha : 2^{[n]} \to X$ and $\beta : 2^{[n]} \to Y$ such that

$$\forall A, B \subseteq [n] : \mathrm{DISJ}(A, B) = 1 \iff S(\alpha(A), \beta(B)) = 0. \tag{3}$$

Notice that this kind of embedding could be called "negative" because zeros in the set disjointness matrix correspond to non-zeros in S.

We remark that "positive" embeddings of the set disjointness matrix force up the rank of S, because the rank of any matrix with the same support as the set disjointness matrix on $[n]$ is at least 2^n [12]. This is not desirable because the nonnegative rank of S is always at least its rank. Thus the lower bound on the nonnegative rank of S obtained from such a "positive" embedding would be useless in our context (the rank of the slack matrix $S(P)$ of polytope P equals $\dim(P) + 1$).

However, "positive" embeddings the *unique set disjointness matrix*, that is the restriction of the set disjointness matrix to pairs (A, B) such that $|A \cap B| \leqslant 1$, do not have this problem of forcing up the rank. Actually, "positive" embeddings of the unique set disjointness matrix led to the main result of Fiorini et al. [8].

Theorem 7 *Let $S \in \mathbb{R}_+^{X \times Y}$ be a matrix in which the set disjointness matrix on $[n]$ can be embedded. Consider a randomized protocol computing S in expectation. If the probability that the protocol outputs a non-zero value, given an input (x, y) with $S(x, y) > 0$, is at least $p = p(n)$, then the protocol has complexity $\Omega(np)$. In particular, by Chebyshev's inequality, the complexity is $\Omega(n(1 - \sigma^2))$, where σ^2 denotes the normalized variance of the protocol.*

Proof Let c be the complexity of the protocol computing S in expectation. From this protocol, we obtain a new protocol, this time for the set disjointness problem,

by mapping each input pair $(A, B) \in 2^{[n]} \times 2^{[n]}$ to the corresponding input pair $(\alpha(A), \beta(B)) \in X \times Y$ (Alice and Bob can do this independently of each other), running the original protocol $\lceil 1/p \rceil$ times, and outputting 0 if at least one of the executions led to a non-zero value or 1 otherwise.

The new protocol always outputs 1 for every disjoint pair (A, B) because of (3) (remember that our protocols have nonnegative outputs), and outputs 0 most of the times for non-disjoint pairs (A, B). More precisely, the probability of outputting 0 in case (A, B) is non-disjoint is at least $1 - (1 - p)^{\frac{1}{p}} \geqslant 1 - \mathrm{e}^{-1} > 1/2$, where e is Euler's number. The theorem follows then directly from the fact that the new protocol has complexity $O(c/p)$ and from the fact that the set disjointness problem has randomized communication complexity $\Omega(n)$. □

6.2 The perfect matching polytope

First, we construct an embedding of the set disjointness matrix in the slack matrix of the perfect matching polytope. Then, we discuss implications for extensions of the perfect matching polytope.

Lemma 4 *There exists an embedding of the set disjointness matrix on* $[n]$ *in the slack matrix of the perfect matching polytope for perfect matchings of* K^ℓ*, where* $\ell \leqslant 3n+14$.

Proof Let $k \leqslant n + 4$ denote the first multiple of 4 that is strictly greater than n, and let $\ell := 3k + 2 \leqslant 3n + 14$.

For two subsets A and B of $[n]$, we define an odd set $U := \alpha(A)$ and a perfect matching $M := \beta(B)$ as follows.

First, we add the dummy element $n + 1$ to B in case $|B|$ is odd, so that both B and $[k] - B$ contain an even number of elements. Note that this does not affect the intersection of A and B because A is contained in $[n]$. Then, we let $U := \{i : i \in A\} \cup \{i + k : i \in A\} \cup \{3k + 1\}$.

Second, we define M by adding matching edges to the partial matching $\{\{i, i+k\} : i \in [k] - B\} \cup \{\{i + k, i + 2k\} : i \in B\} \cup \{\{3k + 1, 3k + 2\}\}$ in such a way that each of the extra edges matches two consecutive unmatched vertices both in $\{i : i \in [k]\}$ or both in $\{i + 2k : i \in [k]\}$. See Fig. 3 for an example.

It can be easily verified that A and B are disjoint if and only if the slack for (U, M) is zero. Hence, the maps $\alpha : A \mapsto U$ and $\beta : B \mapsto M$ define the desired embedding of the set disjointness matrix. □

Let P denote the perfect matching polytope of K^n. Consider a size-r extension of P and a corresponding complexity-$\lceil \lg r \rceil$ protocol computing $S(P)$ in expectation (the existence of such a protocol is guaranteed by Theorems 1 and 2). Lemma 4 and Theorem 7 together imply that $r = 2^{\Omega(n(1-\sigma^2))}$, where σ^2 is the normalized variance of the protocol. For instance, deterministic protocols for computing the slack matrix of the perfect matching polytope give rise to exponential size extensions ($\sigma^2 = 0$ in this case). The same holds if σ^2 is a constant with $0 < \sigma^2 < 1$. When σ^2 is about $(n - 1)/n$ or more, the bound given by Theorem 7 becomes trivial.

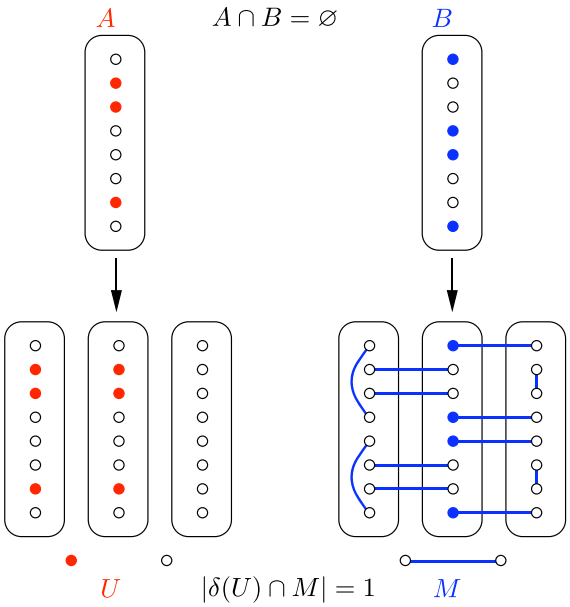


Fig. 3 Constructing an odd set and a perfect matching from a set disjointness instance

6.3 Spanning tree polytopes

We prove that similar results hold for the spanning tree polytope of K^n as well. This is surprising, because for this polytope an extension of size $O(n^3)$ exists.

Lemma 5 *There exists an embedding of the set disjointness matrix on* $[n]$ *in the slack matrix of the spanning tree polytope of* K^{2n+1}.

Proof Let $\ell := 2n+1$. Recall that the rows and columns of (the non-trivial part of) the slack matrix of the spanning tree polytope of K^ℓ respectively correspond to subsets U and spanning trees T. The entry for a pair (U, T) is zero iff the subgraph of T induced by U is connected.

Given an instance of the set disjointness problem with sets $A, B \subseteq [n]$, we define $U := \alpha(A)$ and $T := \beta(B)$ as follows. For every $i \in [n]$ add the edge $\{i, 2n+1\}$ to T. For every $i \in B$ add the edge $\{n+i, i\}$ to T and for every $i \in [n] - B$ add the edge $\{n+i, 2n+1\}$ to T. See Fig. 4 for an example.

Finally, we let $U := \{n+i : i \in A\} \cup \{2n+1\}$. As is easily seen, $T[U]$ is connected iff $A \cap B = \varnothing$. Indeed, if $i \in A \cap B$ then $n+i$ and $2n+1$ are in different connected components of $T[U]$. Moreover, if $A \cap B = \varnothing$ then $T[U]$ is a star with $2n+1$ as center. □

Therefore, the "low variance forces large size" phenomenon we exhibited for the perfect matching polytope also holds for the spanning tree polytope. Incidentally, the $O(n^3)$-size extension for the spanning tree polytope of K^n can be obtained via randomized protocols, but not via deterministic ones. This is because Lemma 5 and Theorem 7 implies that any extension for the spanning tree polytope that corresponds to a deterministic protocol must have exponential size. (Notice that the value of $p = p(n)$ for the protocol given in Sect. 5.2 is roughly $1/n$.)

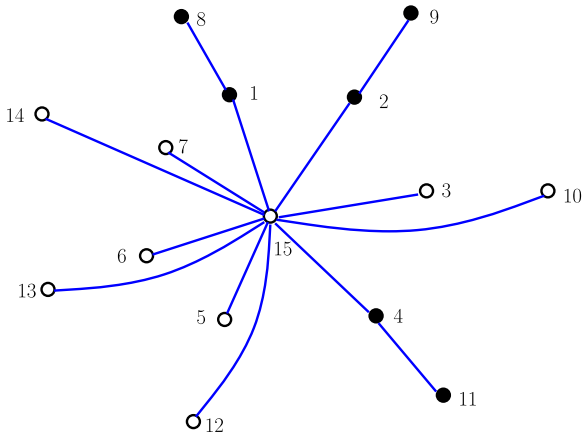


Fig. 4 The spanning tree T for $B = \{1, 2, 4\}$ and $n = 7$. *Black vertices* are those of the form i or $n + i$ where $i \in B$

7 Concluding remarks

Given a perfect matching M and an odd set U as above there is always an edge in $\delta(U) \cap M$. But it is not clear if such an edge can be found using a protocol with sublinear communication. Now we show that if such an edge can be found using few bits then the perfect matching polytope has an extension of small size. As one of the referees pointed out, this fact can be considered as folklore.

Theorem 8 *Suppose Alice is given an odd set $U \subseteq [n]$ and Bob is given a perfect matching M of K^n. Furthermore, suppose that Bob knows an edge $e \in \delta(U) \cap M$. Then, there exists a randomized protocol of complexity $2 \lg n + O(1)$ that computes the slack for the pair (U, M) in expectation.*

Proof The protocol works as follows. Bob picks an edge e' from $M \setminus \{e\}$ uniformly at random and sends it to Alice. She outputs $|M| - 1 = n/2 - 1$ if $e' \in \delta(U)$ and 0 otherwise. The expected value of the protocol is $(|M|-1)\cdot(|\delta(U)\cap M|-1)/(|M|-1) = |\delta(U) \cap M| - 1$, as required. Bob needs to send the endpoints of the edge e' to Alice and this requires $2 \lg n + O(1)$ bits. □

The theorem above implies that if an edge in $\delta(U) \cap M$ can be computed using a protocol requiring $o(n)$ bits, then there exists an extension for the perfect matching polytope of subexponential size. We leave it as an open question to settle the existence of such a protocol.

Acknowledgments The authors thank Sebastian Pokutta and Ronald de Wolf for their useful feedback. The research of Faenza was supported by the German Research Foundation (DFG) within the Priority Programme 1307 Algorithm Engineering. The research of Grappe was supported by the Progetto di Eccellenza 2008–2009 of the Fondazione Cassa di Risparmio di Padova e Rovigo. The research of Fiorini was partially supported by the *Actions de Recherche Concertées* (ARC) fund of the French community of Belgium. The research of Tiwary was supported by the *Fonds National de la Recherche Scientifique* (F.R.S.–FNRS). The authors would also like to thank the anonymous referees for their helpful comments.

References

1. Chattopadhyay, A., Pitassi, T.: The story of set disjointness. SIGACT News **41**(3), 59–85 (2010)
2. Chvátal, V.: On certain polytopes associated with graphs. J. Comb. Theory B **18**, 138–154 (1975)

3. Cohen, J.E., Rothblum, U.G.: Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. Linear Algebra Appl. **190**, 149–168 (1993)
4. Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. 4OR **8**(1), 1–48 (2010)
5. Edmonds, J.: Maximum matching and a polyhedron with 0, 1 vertices. J. Res. Nat. Bur. Stand. **69B**, 125–130 (1965)
6. Edmonds, J.: Matroids and the greedy algorithm. Math. Program. **1**, 127–136 (1971)
7. Faenza, Y., Oriolo, G., Stauffer, G.: Separating stable sets in claw-free graphs via Padberg-Rao and compact linear programs. In: Rabani, Y. (ed.) Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), pp. 1298–1308. SIAM, Japan (2012)
8. Fiorini, S., Massar, S., Pokutta, S., Tiwary, H.R., de Wolf, R.: Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In: Proceedings of the 44th ACM Symposium on Theory of Computing (STOC 2012), pp. 95–106 (2012)
9. Fiorini, S., Kaibel, V., Pashkovich, K., Theis, D.O.: Combinatorial bounds on nonnegative rank and extended formulations. Discret. Math. **313**(1), 67–83 (2013)
10. Galluccio, A., Gentile, C., Ventura, P.: The stable set polytope of claw-free graphs with large stability number. Electron. Notes Discrete Math. **36**, 1025–1032 (2010)
11. Grötschel, M., Lovász, L., Schrijver, A.: Geometric algorithms and combinatorial optimization, volume 2 of Algorithms and Combinatorics., 2nd edn. Springer, Berlin (1993)
12. Høyer, P., de Wolf, R.: Improved quantum communication complexity bounds for disjointness and equality. In Proceedings of STACS, pp. 299–310 (2002)
13. Kaibel, V.: Extended formulations in combinatorial optimization. Optima **85**, 2–7 (2011)
14. Kaibel, V., Pashkovich, K., Oliver, D.: Theis. Symmetry matters for the sizes of extended formulations. In: Proceedings of IPCO, pp. 135–148 (2010)
15. Kalyanasundaram, B., Schnitger, G.: The probabilistic communication complexity of set intersection. SIAM J. Discr. Math. **5**(4), 545–557 (1992)
16. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press, Cambridge (1997)
17. Krause, M.: Geometric arguments yield better bounds for threshold circuits and distributed computing. Theor. Comput. Sci. **156**(1&2), 99–117 (1996)
18. Lovász, L.: On the ratio of optimal integral and fractional covers. Discrete Math. **13**(4), 383–390 (1975)
19. Richard, K.M.: Using separation algorithms to generate mixed integer model reformulations. Oper. Res. Lett. **10**(3), 119–128 (1991)
20. Razborov, A.A.: On the distributional complexity of disjointness. Theor. Comput. Sci. **106**(2), 385–390 (1992)
21. Rothvoß, T.: Some 0/1 polytopes need exponential size extended formulations. arXiv:1105.0036 (2011)
22. Schrijver, A.: Combinatorial optimization. Polyhedra and efficiency. Vol. A and B, Volume 24 of Algorithms and Combinatorics. Springer, Berlin (2003)
23. Thomason, A.: The extremal function for complete minors. Journal of Combinatorial Theory. Series B, Volume 81, Number 2. Academic Press, Inc., NY (2001)
24. Yannakakis, M.: Expressing combinatorial optimization problems by linear programs. J. Comput. Syst. Sci. **43**(3), 441–466 (1991)
25. Williams, J.C.: A linear-size zero–one programming model for the minimum spanning tree problem in planar graphs. Networks **39**(1), 53–60 (2002)
26. Wolsey, L.A.: Using extended formulations in practice. Optima **85**, 7–9 (2011)
27. Zhang, S.: Quantum Strategic Game Theory. In Proceedings of the 3rd Innovations in, Theoretical Computer Science, pp. 39–59 (2012)
28. Ziegler, G.M.: Lectures on Polytopes, volume 152 of Graduate Texts in Mathematics. Springer, Berlin (1995)

C

EXTENDED FORMULATIONS FOR POLYGONS

The following article has appeared in *Discrete and Computational Geometry* and is included here as an appendix for completeness.

Discrete Comput Geom (2012) 48:658–668
DOI 10.1007/s00454-012-9421-9

Extended Formulations for Polygons

Samuel Fiorini · Thomas Rothvoß · Hans Raj Tiwary

Received: 13 August 2011 / Revised: 24 February 2012 / Accepted: 28 February 2012 /
Published online: 16 March 2012

Abstract The extension complexity of a polytope P is the smallest integer k such that P is the projection of a polytope Q with k facets. We study the extension complexity of n-gons in the plane. First, we give a new proof that the extension complexity of regular n-gons is $O(\log n)$, a result originating from work by Ben-Tal and Nemirovski (Math. Oper. Res. 26(2), 193–205, 2001). Our proof easily generalizes to other permutahedra and simplifies proofs of recent results by Goemans (2009), and Kaibel and Pashkovich (2011). Second, we prove a lower bound of $\sqrt{2n}$ on the extension complexity of generic n-gons. Finally, we prove that there exist n-gons whose vertices lie on an $O(n) \times O(n^2)$ integer grid with extension complexity $\Omega(\sqrt{n}/\sqrt{\log n})$.

Keywords Extended formulations · Polygon · Polytope · Lower bound

1 Introduction

Consider a (convex) polytope P in $\mathbb{R}^d$. An *extension* (or *extended formulation*) of P is a polytope Q in $\mathbb{R}^e$ such that P is the image of Q under a linear projection from $\mathbb{R}^e$ to $\mathbb{R}^d$. The main motivation for seeking extensions Q of the polytope P is perhaps that the number of facets of Q can sometimes be significantly smaller than

S. Fiorini · H.R. Tiwary (✉)
Department of Mathematics, Université Libre de Bruxelles, Brussels, Belgium
e-mail: htiwary@ulb.ac.be

S. Fiorini
e-mail: sfiorini@ulb.ac.be

T. Rothvoß
Department of Mathematics, MIT, Cambridge, USA
e-mail: rothvoss@math.mit.edu

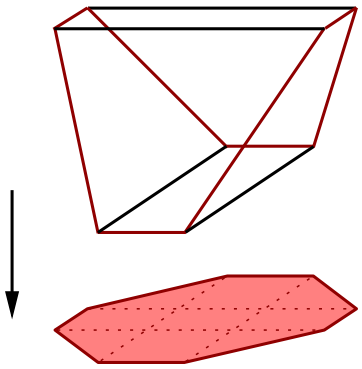


Fig. 1 Proof by picture that the extension complexity of a regular 8-gon is at most 6. Here $P \subseteq \mathbb{R}^2$ is a regular 8-gon, $Q \subseteq \mathbb{R}^3$ is a polytope combinatorially equivalent to a 3-cube, and $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is a linear projection map such that $\pi(Q) = P$

that of P. This phenomenon has already found numerous applications in optimization, and in particular linear and integer programming. To our knowledge, systematic investigations began at the end of the 1980s with the work of Martin [13] and Yannakakis [17], among others. Recently, the subject is receiving an increasing amount of attention. See, e.g., the surveys by Conforti, Cornuéjols and Zambelli [4], Vanderbeck and Wolsey [16], and Kaibel [10].

A striking example, which is relevant to this paper, arises when P is a regular n-gon in $\mathbb{R}^2$. As follows from results of Ben-Tal and Nemirovski [2], for such a polytope P, one can construct an extension Q with as few as $O(\log n)$ facets. It remained an open question to determine to which extent such a dramatic decrease in the number of facets is possible when P is a *non-regular* n-gon.[1] This is the main question we address in this paper.

Before giving an outline of the paper, we state a few more definitions. The *size* of an extension Q is simply the number of facets of Q. The *extension complexity* of P is the minimum size of an extension of P, denoted as $\mathrm{xc}(P)$. See Fig. 1 for an illustration.

Notice that the extension complexity of every n-gon is $\Omega(\log n)$. This follows from the fact that any extension Q with k facets has at most 2^k faces. Since each face of P is the projection of a face of the extension Q, it follows that Q must have at least $\log_2 f$ facets if P has f faces [7]. Thus if P is an n-gon, we have $\mathrm{xc}(P) \geqslant \log_2(2n+2) = \Omega(\log n)$. When P is a regular n-gon, we have $\mathrm{xc}(P) = \Theta(\log n)$.

One of the fundamental results that can be found in Yannakakis' groundbreaking paper [17] is a characterization of the extension complexity of a polytope in terms of the non-negative rank of its slack matrix. Although this is discussed in detail in Sect. 2, we include a brief description here. To each polytope P one can associate a matrix $S(P)$ that records, in the entry that is in the ith row and jth column, the slack of the jth vertex with respect to the ith facet. This matrix is the 'slack matrix' of P. It turns out that computing $\mathrm{xc}(P)$ amounts to determining the minimum number r such that there exists a factorization of the slack matrix of P as $S(P) = TU$, where T is a non-negative matrix with r columns and U is a non-negative matrix with r rows. Such a factorization is called a 'rank r non-negative factorization' of the slack matrix $S(P)$.

In Sect. 3, we give an explicit $O(\log n)$ rank non-negative factorization of the slack matrix of a regular n-gon. This provides a new proof that the extension complexity

[1] This was posed as an open problem during the First Cargese Workshop on Combinatorial Optimization.

of every regular n-gon is $O(\log n)$. Our proof technique directly generalizes to other polytopes, such as the permutahedron. In particular, we obtain a new proof of the fact that the extension complexity of the n-permutahedron is $O(n \log n)$, a result due to Goemans [7]. Our approach builds on a new proof of this result by Kaibel and Pashkovich [11] but is different because it works by directly constructing a non-negative factorization of the slack matrix.

In Sect. 4, we prove that there exist n-gons whose extension complexity is at least $\sqrt{2n}$. However, the proof uses polygons whose coordinates are transcendental numbers, which is perhaps not entirely satisfactory. For instance, one might ask whether a similar result holds when the encoding length of each vertex of the polygon is $O(\log n)$.

In Sect. 5, we settle this last question by proving the existence of n-gons whose vertices belong to an $O(n) \times O(n^2)$ integer grid and with extension complexity $\Omega(\sqrt{n}/\sqrt{\log n})$. This is inspired by recent work of one of the authors on the extension complexity of 0/1-polytopes [14].

2 Slack Matrices and Non-negative Factorizations

Consider a polytope P in $\mathbb{R}^d$ with m facets and n vertices. Let $A_1 x \leqslant b_1, \ldots, A_m x \leqslant b_m$ denote the facet-defining inequalities of P, where $A_1, \ldots, A_m$ are row vectors. Let also $v_1, \ldots, v_n$ denote the vertices of P. The *slack matrix* of P is the non-negative $m \times n$ matrix $S = S(P)$ with $S_{ij} = b_i - A_i v_j$.

A *rank r non-negative factorization* of a non-negative matrix S is an expression of S as product $S = TU$ where T and U are non-negative matrices with r columns and r rows, respectively. The *non-negative rank* of S, denoted by $\operatorname{rank}_+(S)$, is the minimum number r such that S admits a rank r non-negative factorization [3].

The following theorem is (essentially) due to Yannakakis, see also [6].

Theorem 1 (Yannakakis [17]) *For all polytopes P,*

$$\operatorname{xc}(P) = \operatorname{rank}_+\big(S(P)\big).$$

To conclude this section, we briefly indicate how to obtain extensions from non-negative factorizations, and prove half of Theorem 1. Assuming $P = \{x \in \mathbb{R}^d : Ax \leqslant b\}$, consider a rank r non-negative factorization $S(P) = TU$ of the slack matrix of P. Then it can be shown that the image of the polyhedron $Q := \{(x, y) \in \mathbb{R}^{d+r} \mid Ax + Ty = b, y \geqslant 0\}$ under the projection $\mathbb{R}^{d+r} \to \mathbb{R}^d : (x, y) \mapsto x$ is exactly P. Notice that Q has at most r facets. Now if we take $r = \operatorname{rank}_+(S(P))$, then Q is actually a polytope [5]. Thus Q is an extension of P with at most $\operatorname{rank}_+(S(P))$ facets, and hence $\operatorname{xc}(P) \leqslant \operatorname{rank}_+(S(P))$.

3 Regular Polygons

First, we give a new proof of the tight logarithmic upper bound on the extension complexity of a regular n-gon. This result is implicit in work by Ben-Tal and Nemirovski [2] (although for n being a power of two). Another proof can be found in

Kaibel and Pashkovich [11]. Then, we discuss a generalization of the proof to related higher-dimensional polytopes.

Theorem 2 *Let P be a regular n-gon in $\mathbb{R}^2$. Then* $\mathrm{xc}(P) = O(\log n)$.

Proof Without loss of generality, we may assume that the origin is the barycenter of P. After numbering the vertices of P counterclockwise as $v_1, \ldots, v_n$, we define a sequence $\ell_0, \ldots, \ell_{q-1}$ of axes of symmetry of P, as follows.

Initialize i to 0, and k to n. While $k > 1$, repeat the following steps:

- define ℓ_i as the line through the origin and the midpoint of vertices $v_{\lceil \frac{k}{2} \rceil}$ and $v_{\lceil \frac{k+1}{2} \rceil}$;
- replace k by $\lfloor \frac{k+1}{2} \rfloor$;
- increase i by one.

Define q as the final value of i. Thus, q is the number of axes of symmetry ℓ_i defined. Note that when $k = k(i)$ is odd, then ℓ_i passes through one of the vertices of P. Note also that $q = O(\log n)$. For each $i = 0, \ldots, q-1$, one of the two closed half-planes bounded by ℓ_i contains v_1. We denote it ℓ_i^+. We denote the other by ℓ_i^-.

Now, consider a vertex v of P. We define the *folding sequence* $v^{(0)}, v^{(1)}, \ldots, v^{(q)}$ of v as follows. We let $v^{(0)} := v$, and for $i = 0, \ldots, q-1$, we let $v^{(i+1)}$ denote the image of $v^{(i)}$ by the reflection with respect to ℓ_i if $v^{(i)}$ is not in the half-space ℓ_i^+, and we let $v^{(i+1)} := v^{(i)}$ otherwise. In other words, $v^{(i+1)}$ is the image of $v^{(i)}$ under the *conditional reflection* with respect to half-plane ℓ_i^+. By construction, we always have $v^{(q)} = v_1$.

Next, consider a facet F of P. The folding sequence $F^{(0)}, F^{(1)}, \ldots, F^{(q)}$ of facet F is defined similarly as the folding sequence of vertex v. Pick any inequality $a^T x \leqslant \beta$ defining F. We let $a^{(0)} := a$, and for $i = 0, \ldots, q-1$, we let $a^{(i+1)}$ denote the image of $a^{(i)}$ under the conditional reflection with respect to ℓ_i^+. Then $F^{(i)}$ is the facet of P defined by $(a^{(i)})^T x \leqslant \beta$. The last facet $F^{(q)}$ in the folding sequence is always either the segment $[v_1, v_2]$ or the segment $[v_1, v_n]$. See Fig. 2 for an illustration with $n = 15$, and thus $q = 4$.

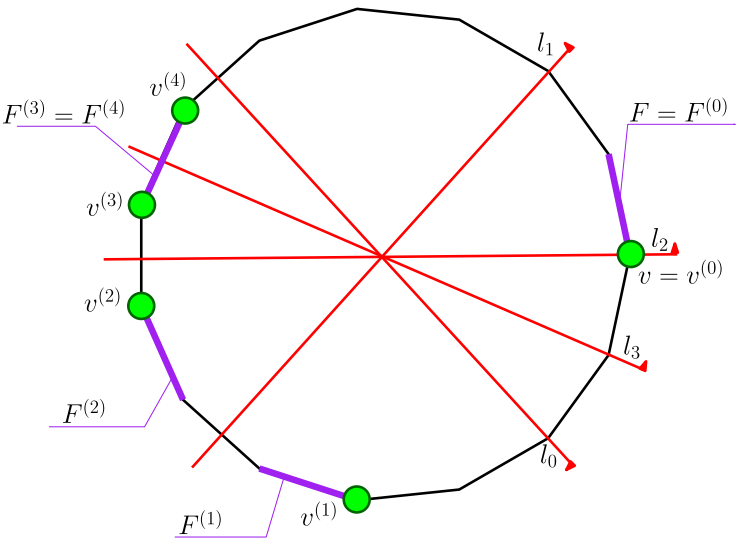


Fig. 2 A 15-gon with four axes of symmetry, a vertex- and a facet-folding sequence

Finally, we define a non-negative factorization $S(P) = TU$ of the slack matrix of P, of rank $2q = O(\log n)$. Below, let $d(x, \ell_i)$ denote the distance of $x \in \mathbb{R}^2$ to line ℓ_i.

In the left factor of the factorization, the row corresponding to facet F is of the form $(t_0, \ldots, t_{q-1})$, where $t_i := (\sqrt{2}\, d(a^{(i)}, \ell_i), 0)$ if $a^{(i)}$ is not in ℓ_i^+ and $t_i := (0, \sqrt{2}\, d(a^{(i)}, \ell_i))$ otherwise. Similarly, in the right factor, the column corresponding to vertex v is of the form $(u_0, \ldots, u_{q-1})^T$, where $u_i := (0, \sqrt{2}\, d(v^{(i)}, \ell_i))^T$ if $v^{(i)}$ is not in ℓ_i^+ and $u_i := (\sqrt{2}\, d(v^{(i)}, \ell_i), 0)^T$ otherwise.

The correctness of the factorization rests on the following simple observation: for $i = 0, \ldots, q-1$ the slack of $v^{(i+1)}$ with respect to $F^{(i+1)}$ equals the slack of $v^{(i)}$ with respect to $F^{(i)}$ plus some correction term. If $a^{(i)}$ and $v^{(i)}$ are on opposite sides of ℓ_i, then the correction term is $2d(a^{(i)}, \ell_i)d(v^{(i)}, \ell_i)$. Otherwise, it is zero (no correction is necessary). Indeed, letting n_i denote a unit vector normal to ℓ_i, and assuming that $v^{(i)}$ and $a^{(i)}$ are on opposite sides of ℓ_i, we have

$$\begin{aligned}
\beta - \big(a^{(i)}\big)^T v^{(i)} &= \beta - \big(a^{(i)}\big)^T \big(v^{(i)} - 2\big(n_i^T v^{(i)}\big)n_i + 2\big(n_i^T v^{(i)}\big)n_i\big) \\
&= \beta - \big(a^{(i+1)}\big)^T v^{(i+1)} - 2\big(\big(a^{(i)}\big)^T n_i\big)\big(n_i^T v^{(i)}\big) \\
&= \beta - \big(a^{(i+1)}\big)^T v^{(i+1)} + 2d\big(a^{(i)}, \ell_i\big)d\big(v^{(i)}, \ell_i\big).
\end{aligned}$$

When $v^{(i)}$ and $a^{(i)}$ are on the same side of ℓ_i, we obviously have

$$\beta - \big(a^{(i)}\big)^T v^{(i)} = \beta - \big(a^{(i+1)}\big)^T v^{(i+1)}.$$

Observe that the slack of $v^{(q)}$ with respect to $F^{(q)}$ is always 0. The theorem follows. □

The *n-permutahedron* is the polytope of dimension $n-1$ in $\mathbb{R}^n$ whose $n!$ vertices are the points obtained by permuting the coordinates of $(1, 2, \ldots, n)^T$. It has $2^n - 2$ facets, defined by the inequalities $\sum_{j \in S} x_j \leqslant g(|S|)$ for all proper non-empty subsets S of $[n] := \{1, 2, \ldots, n\}$, where $g(S) := \binom{n+1}{2} - \binom{n-|S|+1}{2}$.

Let j and k denote two elements of $[n]$ such that $j < k$. We denote $H_{j,k}$ the hyperplane defined by $x_j = x_k$, and $H_{j,k}^+$ the closed half-space defined by $x_j \leqslant x_k$. Applying the conditional reflection with respect to $H_{j,k}^+$ to a vector $x \in \mathbb{R}^n$ amounts to swapping the coordinates x_j and x_k if and only if $x_j > x_k$. Intuitively, the conditional reflection with respect to $H_{j,k}^+$ sorts the coordinates x_j and x_k.

The proof of Theorem 2 can be modified to give a new proof of the existence of $O(n \log n)$ size extension of the n-permutahedron [7], as follows. Since there exists a sorting network of size $O(n \log n)$ for sorting n inputs, a celebrated result of Ajtai, Komlós and Szemerédi [1], there exist $q = O(n \log n)$ half-spaces H_{j_0,k_0}^+, $H_{j_1,k_1}^+, \ldots, H_{j_{q-1},k_{q-1}}^+$ such that sequentially applying the conditional reflection with respect to H_{j_i,k_i}^+ for $i = 0, \ldots, q-1$ to *any* point $x \in \mathbb{R}^n$, sorts this point x.

Therefore, the folding sequence of any vertex v of the n-permutahedron always ends with the vertex $(1, 2, \ldots, n)^T$. Moreover, the folding sequence of the facet defined by $\sum_{j \in S} x_j \leqslant g(|S|)$ always ends with the facet defined by $\sum_{j=n-|S|+1}^{n} x_j \leqslant$

$g(|S|)$. Note that this last facet contains the vertex $(1, 2, \ldots, n)^T$. Hence the proof technique used above for a regular n-gon extends to the n-permutahedron.

In fact, it turns out that the proof technique further extends to the permutahedron of any finite reflection group. One simply has to choose the right sequence of conditional reflections. Such sequences were constructed by Kaibel and Pashkovich [11], with the help of Ajtai–Komlós–Szemerédi sorting networks. Thus we can re-prove their main results about permutahedra of finite reflection groups. Our proof is different in the sense that we explicitly construct a non-negative factorization of the slack matrix.

4 Generic Polygons

We begin by recalling some basic facts about field extensions (see, e.g., Hungerford [9], Lang [12], or Stewart [15]). Let L be a field and K be a subfield of L. Then L is an *extension field* of K, and L/K is a *field extension*. We say that the field extension L/K is *algebraic* if every element of L is algebraic over K, that is, for each element of L there exists a non-zero polynomial with coefficients in K that has the element as one of its roots.

For $\alpha_1, \ldots, \alpha_q \in L$, the inclusion-wise minimal subfield of L that contains both K and $\{\alpha_1, \ldots, \alpha_q\}$ is denoted by $K(\{\alpha_1, \ldots, \alpha_q\})$, or simply $K(\alpha_1, \ldots, \alpha_q)$. It is also the subfield formed by all fractions $\frac{f(\alpha_1,\ldots,\alpha_q)}{g(\alpha_1,\ldots,\alpha_q)}$ where f and g are polynomials with coefficients in K and $g(\alpha_1, \ldots, \alpha_q) \neq 0$.

A subset X of L is said to be *algebraically independent* over K if no non-trivial polynomial relation with coefficients in K holds among the elements of X. The *transcendence degree* of the field extension L/K is defined as the largest cardinality of an algebraically independent subset of L over K. It is also the minimum cardinality of a subset Y of L such that $L/K(Y)$ is algebraic.

We say that a polygon in $\mathbb{R}^2$ is *generic* if the coordinates of its vertices are distinct and form a set that is algebraically independent over the rationals.

Theorem 3 *If P is a generic convex n-gon in $\mathbb{R}^2$ then* $\mathrm{xc}(P) \geqslant \sqrt{2n}$.

Proof Let $\alpha_1, \ldots, \alpha_{2n}$ denote the coordinates of the n vertices of P, listed in any order. Thus $X := \{\alpha_1, \ldots, \alpha_{2n}\}$ is algebraically independent over $\mathbb{Q}$.

Now suppose that P is the projection of a d-dimensional polytope Q with k facets. Without loss of generality, we may assume that Q lives in $\mathbb{R}^d$ and that the projection is onto the two first coordinates.

Consider any linear description of Q. This description is defined by $k(d+1)$ real numbers: the kd entries of the constraint matrix and the k right-hand sides. We denote these reals as $\beta_1, \ldots, \beta_{k(d+1)}$. By Cramer's rule, each α_i can be written as $\alpha_i = \frac{f_i(\beta_1,\ldots,\beta_{k(d+1)})}{g_i(\beta_1,\ldots,\beta_{k(d+1)})}$ where f_i and g_i are polynomials with rational coefficients and $g_i(\beta_1, \ldots, \beta_{k(d+1)}) \neq 0$. In particular, this means that each α_i is in the extension field $L := \mathbb{Q}(\beta_1, \ldots, \beta_{k(d+1)})$.

Since X is algebraically independent over $\mathbb{Q}$ and $X \subseteq L$, the transcendence degree of $L/\mathbb{Q}$ is at least $2n$. But on the other hand, the transcendence degree of $L/\mathbb{Q}$ is at most $k(d+1)$. Indeed, letting $Y := \{\beta_1, \ldots, \beta_{k(d+1)}\}$, we have $\mathbb{Q}(Y) = L$ and thus

$L/\mathbb{Q}(Y)$ is algebraic. It follows that $k(d+1) \geqslant 2n$. Since $k \geqslant d+1$, we see that $k^2 \geqslant 2n$, hence $k \geqslant \sqrt{2n}$. □

5 Polygons with Integer Vertices

Since encoding transcendental numbers would require an infinite number of bits, an objection might be raised that Theorem 3 is not very satisfying. In this section we provide a slightly weaker lower bound with polygons whose vertices can be encoded efficiently. In particular we will now show that for every n there exist polygons with vertices on an $O(n) \times O(n^2)$ grid and whose extension complexity is large. To do this we will need a slightly modified version of a rounding lemma proved by Rothvoß [14], see Lemma 5 below.

For a matrix A let A_ℓ (resp. A^ℓ) denote the ℓth row (resp. ℓth column) of A. Similarly, for a subset I of row indices of A, let A_I denote the submatrix of A obtained by picking the rows indexed by the elements of I.

Let T and U be $m \times r$ and $r \times n$ non-negative matrices. Since below T and U will be respectively the left and right factor of a factorization of some slack matrix, we can assume that no column of T is identically zero and, similarly, no row of U is identically zero. The pair T, U is said to be *normalized* if $\|T^\ell\|_\infty = \|U_\ell\|_\infty$ for every $\ell \in [r]$. Since multiplying a column ℓ of T by $\lambda > 0$ and simultaneously dividing row ℓ of U by λ leaves the product TU unchanged, we can always scale the rows and columns of two matrices so that they are normalized without changing TU.

Lemma 4 (Rothvoß [14]) *If the pair T, U is normalized, then* $\max\{\|T\|_\infty, \|U\|_\infty\} \leqslant \sqrt{\|TU\|_\infty}$.

Proof Let $S := TU$. Suppose, for the sake of contradiction, that the assertion does not hold. Without loss of generality, we may assume that $\|T\|_\infty > \sqrt{\|TU\|_\infty}$. Thus $T_{i\ell} > \sqrt{\|TU\|_\infty}$ for some indices i and ℓ. Since T, U is normalized, $\|U_\ell\|_\infty = \|T^\ell\|_\infty > \sqrt{\|TU\|_\infty}$ and there must be an index j such that $U_{\ell j} > \sqrt{\|TU\|_\infty}$. Then $S_{ij} \geqslant T_{i\ell}U_{\ell j} > \|TU\|_\infty$, which is a contradiction. □

Consider a set of n convex independent points V in the plane lying on an integer grid of size polynomial in n, its convex hull $P := \mathrm{conv}(V)$, and $X := \mathbb{Z}^2 \cap P$. The next crucial lemma (adapted from a similar result in [14]) implies that the description of an extension $Q := \{(x, y) \mid Ax + Ty = b, y \geq 0\}$ for P—potentially containing irrational numbers—can be rounded such that an integer point x is in X if and only if there is a $y \geq 0$ such that $\bar{A}x + \bar{T}y \approx \bar{b}$ holds for the rounded system. Moreover, all coefficients in the rounded system come from a domain which is bounded by a polynomial in n.

Lemma 5 *For $d, N \geq 2$ let $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{Z}^d$ be a convex independent and non-empty set of points with $\|v_i\|_\infty \leqslant N$ for $i \in [n]$. Let $P := \mathrm{conv}(V)$ and let $X := P \cap \mathbb{Z}^d$. Denote $r := \mathrm{xc}(P)$ and $\Delta := ((d+1)N)^d$. Then there are*

matrices $\bar{A} \in \mathbb{Z}^{(d+r)\times d}$, $\bar{T} \in (\frac{1}{4r(d+r)\Delta}\mathbb{Z}_+)^{(d+r)\times r}$ *and a vector* $\bar{b} \in \mathbb{Z}^{d+r}$ *with* $\|\bar{A}\|_\infty, \|\bar{b}\|_\infty, \|\bar{T}\|_\infty \leqslant \Delta$ *such that*

$$X = \left\{ x \in \mathbb{Z}^d \mid \exists y \in [0, \Delta]^r : \|\bar{A}x + \bar{T}y - \bar{b}\|_\infty \leqslant \frac{1}{4(d+r)} \right\}.$$

Proof Let $Ax \leqslant b$ be a non-redundant description of P with integral coefficients. We may assume (see, e.g., [8, Lemma D.4.1]) that $\|A\|_\infty, \|b\|_\infty \leqslant \Delta = ((d+1)N)^d$. Since $\mathrm{xc}(P) = r$, by Yannakakis' Theorem 1 there exist matrices $T \in \mathbb{R}_+^{m\times r}$ and $U \in \mathbb{R}_+^{r\times n}$ such that $S := TU$ is the slack matrix of P, and $P = \{x \in \mathbb{R}^d \mid \exists y \in \mathbb{R}^r : Ax + Ty = b, y \geqslant 0\}$. Without loss of generality assume that the pair T, U is normalized. Note that

$$\|S\|_\infty = \max_{\substack{i\in[m]\\ j\in[n]}} (b_i - A_i v_j) \leqslant \Delta + dN\Delta \leqslant \Delta^2.$$

Since T, U are normalized, using Lemma 4, we have that $\|T\|_\infty \leqslant \Delta$ and $\|U\|_\infty \leqslant \Delta$.

Let $W := \mathrm{span}(\{(A_i, T_i) \mid i \in [m]\})$ be the row span of the constraint matrix of the system $Ax + Ty = b$ and let $k := \dim(W)$ be the dimension of W. Choose $I \subseteq \{1, \ldots, m\}$ of size $|I| = k$ such that the volume of the parallelepiped spanned by the vectors $\{(A_i, T_i) \mid i \in I\}$, denoted by $\mathrm{vol}(\{(A_i, T_i) \mid i \in I\})$, is maximized. Let T_I' be the matrix obtained from rounding the coefficients of T_I to the nearest multiple of $\frac{1}{4r(d+r)\Delta}$. Our choice will be $\bar{A} := A_I$, $\bar{T} := T_I'$ and $\bar{b} := b_I$. Let

$$Y := \left\{ x \in \mathbb{Z}^d \mid \exists y \in [0, \Delta]^r : \left\| A_I x + T_I' y - b_I \right\|_\infty \leqslant \frac{1}{4(d+r)} \right\}.$$

Then it is sufficient to show that $X = Y$.

Claim 6 $X \subseteq Y$.

Proof of claim Consider an arbitrary vertex $v_j \in V$. Since, $S = TU$, we can choose $y := U^j \geqslant 0$ such that $Av_j + Ty = b$. Since T, U are normalized, we have that $\|y\|_\infty \leqslant \|U\|_\infty \leqslant \Delta$. Note that $\|T - T'\|_\infty \leqslant \frac{1}{4r(d+r)\Delta}$. By the triangle inequality,

$$\begin{aligned} \left\| A_I v_j + T_I' y - b_I \right\|_\infty &\leqslant \Big\| \underbrace{A_I v_j + T_I y - b_I}_{=0} + \big(T_I' - T_I\big) y \Big\|_\infty \\ &\leqslant r \cdot \underbrace{\left\| T_I' - T_I \right\|_\infty}_{\leqslant \frac{1}{4r(d+r)\Delta}} \cdot \underbrace{\|y\|_\infty}_{\leqslant \Delta} \leqslant \frac{1}{4(d+r)}. \end{aligned}$$

Thus $v_j \in Y$ and hence $V \subseteq Y$. It follows that $X \subseteq Y$. □

Claim 7 $X \supseteq Y$.

Proof of claim We show that $x \in \mathbb{Z}^d \setminus X$ implies $x \notin Y$. Since $x \notin X$ and $X \subseteq P$, there must be a row ℓ with $A_\ell x > b_\ell$. Since A, b and x are integral, one even has $A_\ell x \geqslant b_\ell + 1$. Note that in general ℓ is not among the selected constraints with row indices in I. But there are unique coefficients $\lambda \in \mathbb{R}^k$ such that we can express constraint $A_\ell x + T_\ell y = b_\ell$ as a linear combination of those with indices in I, i.e.

$$\left(A_\ell, T_\ell\right) = \sum_{i \in I} \lambda_i \left(A_i, T_i\right).$$

It is easy to see that $\sum_{i \in I} \lambda_i b_i = b_\ell$, since otherwise the system $Ax + Ty = b$ could not have any solution (x, y) at all and $P = \varnothing$. The next step is to bound the coefficients λ_i. Here we recall that by Cramer's rule,

$$|\lambda_i| = \frac{\operatorname{vol}(\{(A_{i'}, T_{i'}) \mid i' \in I \setminus \{i\} \cup \{\ell\}\})}{\operatorname{vol}(\{(A_{i'}, T_{i'}) \mid i' \in I\})} \leqslant 1,$$

since we picked I such that $\operatorname{vol}(\{(A_{i'}, T_{i'}) \mid i' \in I\})$ is maximized. Fix an arbitrary $y \in [0, \Delta]^r$, then

$$\begin{aligned} 1 \leqslant |\underbrace{A_\ell x - b_\ell}_{\geqslant 1} + \underbrace{T_\ell y}_{\geqslant 0}| &= \left|\sum_{i \in I} \lambda_i (A_i x - b_i + T_i y)\right| \\ &\leqslant \sum_{i \in I} \underbrace{|\lambda_i|}_{\leqslant 1} \cdot |A_i x - b_i + T_i y| \\ &\leqslant (d + r) \cdot \|A_I x - b_I + T_I y\|_\infty \end{aligned} \tag{1}$$

using the triangle inequality and the fact that $|I| \leqslant d + r$. Again making use of the triangle inequality yields

$$\begin{aligned} \|A_I x - b_I + T_I y\|_\infty &= \left\|A_I x - b_I + T'_I y + \left(T_I - T'_I\right) y\right\|_\infty \\ &\leqslant \left\|A_I x - b_I + T'_I y\right\|_\infty + r \cdot \underbrace{\left\|T_I - T'_I\right\|_\infty}_{\leqslant \frac{1}{4r(d+r)\Delta}} \cdot \underbrace{\|y\|_\infty}_{\leqslant \Delta} \\ &\leqslant \left\|A_I x - b_I + T'_I y\right\|_\infty + \frac{1}{4(d+r)}. \end{aligned} \tag{2}$$

Combining (1) and (2) gives $\|A_I x - b_I + T'_I y\|_\infty \geqslant \frac{1}{d+r} - \frac{1}{4(d+r)} > \frac{1}{4(d+r)}$ for all $y \in [0, \Delta]^r$ and consequently $x \notin Y$. □

The theorem follows. Note that by padding zeros, we can ensure that $\bar{A}$, $\bar{T}$ and $\bar{b}$ have exactly $d + r$ rows. □

Now we are ready to prove our lower bound for the extension complexity of polygons.

Theorem 8 *For every $n \geq 3$, there exists a convex n-gon P with vertices in $[2n] \times [4n^2]$ and $\operatorname{xc}(P) = \Omega(\sqrt{n}/\sqrt{\log n})$.*

Proof The $2n$ points of the set $Z := \{(z, z^2) \mid z \in [2n]\}$ are obviously convex independent. In other words, every subset $X \subseteq Z$ of size $|X| = n$ yields a different convex n-gon. The number of such n-gons is $\binom{2n}{n} \geqslant 2^n$. Let $R := \max\{\mathrm{xc}(\mathrm{conv}(X)) \mid X \subseteq Z, |X| = n\}$. Lemma 5 provides a map Φ which takes X as input and provides the rounded system $(\bar{A}, \bar{T}, \bar{b})$. (If the choice of A, b and I is not unique, make an arbitrary canonical choice.) By padding zeros, we may assume that this system is of size $(2 + R) \times (3 + R)$.

Also, Lemma 5 guarantees that for each system $(\bar{A}, \bar{T}, \bar{b})$, the corresponding set X can be reconstructed. In other words, the map Φ must be injective and the number of such system must be at least 2^n. Thus it suffices to determine the number of such systems: the entries in each system $(\bar{A}, \bar{T}, \bar{b})$ are integer multiples of $\frac{1}{4r(d+r)\Delta} = \frac{1}{4r(2+r)144n^4}$ for some $r \in [R]$ using $d = 2$, $N = 4n^2$, $\Delta = (12n^2)^2 = 144n^4$. Since no entry exceeds Δ, for each entry there are at most $1 + \sum_{r=1}^{R}(165888\, r(2 + r)n^8) \leqslant cn^{11}$ many possible choices for some fixed constant c (note that $R \leqslant n$). Thus the number of such systems is bounded by $(cn^{11})^{(3+R)\cdot(2+R)} \leqslant 2^{c' \log n \cdot R^2}$ for some constant c'.

We conclude that $2^{c' \log_2 n \cdot R^2} \geqslant 2^n$ and thus $R = \Omega(\sqrt{n}/\sqrt{\log n})$. □

6 Concluding Remarks

Although the two lower bounds presented here on the worst case extension complexity of a n-gon are $\tilde{\Omega}(\sqrt{n})$, it is plausible that the true answer is $\tilde{\Omega}(n)$. We leave this as an open problem.

Acknowledgements We thank Stefan Langerman for suggesting the proof of Theorem 3. We also thank Volker Kaibel and Sebastian Pokutta for stimulating discussions. Finally, we thank the anonymous referee for his comments which helped improving the text. S. Fiorini supported by the *Actions de Recherche Concertées* (ARC) fund of the *Communauté française de Belgique*. T. Rothvoß supported by Feodor Lynen Fellowship of the Alexander von Humboldt Foundation, ONR Grant N00014-11-1-0053 and NSF Contract CCF-0829878. H.R. Tiwary supported by *Fonds National de la Recherche Scientifique* (F.R.S.–FNRS).

References

1. Ajtai, M., Komlós, J., Szemerédi, E.: An $O(n \log n)$ sorting network. In: Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, STOC'83, pp. 1–9. ACM, New York (1983)
2. Ben-Tal, A., Nemirovski, A.: On polyhedral approximations of the second-order cone. Math. Oper. Res. **26**(2), 193–205 (2001)
3. Cohen, J.E., Rothblum, U.G.: Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. Linear Algebra Appl. **190**, 149–168 (1993)
4. Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. 4OR **8**(1), 1–48 (2010)
5. Conforti, M., Faenza, Y., Fiorini, S., Grappe, R., Tiwary, H.R.: Extended formulations, non-negative factorizations and randomized communication protocols. http://arxiv.org/abs/1105.4127 (2011)
6. Fiorini, S., Kaibel, V., Pashkovich, K., Theis, D.O.: Combinatorial bounds on nonnegative rank and extended formulations. Working paper (2011)
7. Goemans, M.: Smallest compact formulation for the permutahedron. http://math.mit.edu/~goemans/PAPERS/permutahedron.pdf (2009)

8. Hindry, M., Silverman, J.H.: Diophantine Geometry: An Introduction, 1st edn. Springer, Berlin (2000)
9. Hungerford, T.W.: Algebra. Graduate Texts in Mathematics. Springer, New York (1974)
10. Kaibel, V.: Extended formulations in combinatorial optimization. Optima **85**, 2–7 (2011)
11. Kaibel, V., Pashkovich, K.: Constructing extended formulations from reflection relations. In: Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization (2011, to appear)
12. Lang, S.: Algebra, Graduate Texts in Mathematics. Springer, Berlin (2002)
13. Martin, R.K.: Using separation algorithms to generate mixed integer model reformulations. Oper. Res. Lett. **10**(3), 119–128 (1991)
14. Rothvoß, T.: Some 0/1 polytopes need exponential size extended formulations. http://arxiv.org/abs/1105.0036 (2011)
15. Stewart, I.: Galois Theory, 3rd edn. Chapman & Hall/CRC Mathematics. Chapman & Hall/CRC, Boca Raton (2004)
16. Vanderbeck, F., Wolsey, L.A.: Reformulation and decomposition of integer programs. In: Jünger, M., et al. (eds.) 50 Years of Integer Programming 1958–2008, pp. 431–502. Springer, Berlin (2010)
17. Yannakakis, M.: Expressing combinatorial optimization problems by linear programs. J. Comput. Syst. Sci. **43**(3), 441–466 (1991)

D

ON THE EXTENSION COMPLEXITY OF COMBINATORIAL POLYTOPES

The following article has appeared in *Mathematical Programming* and is included here as an appendix for completeness.

Math. Program., Ser. B (2015) 153:95–115
DOI 10.1007/s10107-014-0764-2

FULL LENGTH PAPER

On the extension complexity of combinatorial polytopes

David Avis · Hans Raj Tiwary

Received: 1 April 2013 / Accepted: 3 February 2014 / Published online: 14 February 2014

Abstract In this paper we extend recent results of Fiorini et al. on the extension complexity of the cut polytope and related polyhedra. We first describe a lifting argument to show exponential extension complexity for a number of NP-complete problems including subset-sum and three dimensional matching. We then obtain a relationship between the extension complexity of the cut polytope of a graph and that of its graph minors. Using this we are able to show exponential extension complexity for the cut polytope of a large number of graphs, including those used in quantum information and suspensions of cubic planar graphs.

Mathematics Subject Classification 52B05

Electronic supplementary material The online version of this article (doi:10.1007/s10107-014-0764-2) contains supplementary material, which is available to authorized users.

D. Avis
GERAD and School of Computer Science, McGill University,
3480 University Street, Montreal, Quebec H3A 2A7, Canada
e-mail: avis@cs.mcgill.ca

D. Avis
Graduate School of Informatics, Kyoto University,
Sakyo-ku, Yoshida Yoshida, Kyoto 606-8501, Japan

H. R. Tiwary (✉)
Department of Mathematics, Université Libre de Bruxelles,
Boulevard du Triomphe, 1050 Brussels, Belgium
e-mail: hans.raj.tiwary@ulb.ac.be

Present address:
H. R. Tiwary
Department of Applied Mathematics (KAM), Institute of Theoretical Computer Science (ITI),
Charles University, Malostranské nám. 25, 118 00 Prague 1, Czech Republic
e-mail: hansraj@kam.mff.cuni.cz

1 Introduction

Cut polytope and related polytopes The cut polytope arises in many application areas and has been extensively studied. Formal definitions of this polytope and its relatives are given in the next section. A comprehensive compilation of facts about the cut polytope is contained in the book by Deza and Laurent [7]. Optimization over the cut polytope is known as the max cut problem, and was included in Karp's original list of problems that he proved to be NP-hard. For the complete graph with n nodes, a complete list of the facets of the cut polytope $\text{CUT}^{\square}{}_n$ is known for $n \leq 7$ (see Section 30.6 of [7]), as well as many classes of facet inducing valid inequalities. The hypermetric inequalities (see Chapter 28 of [7]) are examples of such a class, and it is known that an exponential number of them are facet inducing. Less is known about classes of facets for the cut polytope of an arbitrary graph, $\text{CUT}^{\square}(G)$. Interest in such polytopes arises because of their application to fundamental problems in physics.

In quantum information theory, the cut polytope arises in relation to Bell inequalities. These inequalities, a generalization of Bell's original inequality [4], were introduced to better understand the nonlocality of quantum physics. Bell inequalities for two parties are inequalities valid for the cut polytope of the complete tripartite graph $\text{K}_{1,n,n}$. Avis, Imai, Ito and Sasaki [1] proposed an operation named *triangular elimination*, which is a combination of zero-lifting and Fourier-Motzkin elimination (see e.g. [15]) using the triangle inequality. They proved that triangular elimination maps facet inducing inequalities of the cut polytope of the complete graph to facet inducing inequalities of the cut polytope of $\text{K}_{1,n,n}$. Therefore a standard description of such polyhedra contains an exponential number of facets.

In [2] the method was extended to obtain facets of $\text{CUT}^{\square}(G)$ for an arbitrary graph G from facets of $\text{CUT}^{\square}{}_n$. For most, but not all classes of graphs, $\text{CUT}^{\square}(G)$ has an exponential number of facets. An interesting exception are the graphs with no K_5 minor. Results of Seymour for the cut cone, extended by Barahona and Mahjoub to the cut polytope (see Section 27.3.2 of [7]), show that the facets in this case are just projections of triangle inequalities. It follows that the max cut problem for a graph G on n vertices with no K_5 minor can be solved in polynomial time by optimizing over the *semi-metric polytope*, which has $O(n^3)$ facets. Another way of expressing this is to say that in this case $\text{CUT}^{\square}(G)$ has $O(n^3)$ *extension complexity*, a notion that will be discussed next.

Extended formulations and extensions Even for polynomially solvable problems, the associated polytope may have an exponential number of facets. By working in a higher dimensional space it is often possible to decrease the number of constraints. In some cases, a polynomial increase in dimension can yield an exponential decrease in the number of constraints. The previous paragraph contained an example of this.

For NP-hard problems the notion of extended formulations also comes into play. Even though a natural LP formulation of such a problem in original space has exponential size, this does not rule out a polynomial size formulation in higher dimensions.

In a groundbreaking paper, Yannakakis [14] proved that every symmetric LP for the Traveling Salesman Problem (TSP) has exponential size. Here, an LP is called *symmetric* if every permutation of the cities can be extended to a permutation of all

the variables of the LP that preserves the constraints of the LP. This result refuted various claimed proofs of a polynomial time algorithm for the TSP. In 2012 Fiorini et al. [8] proved that the cut polytope (associated with the max cut problem) also requires exponential size if it is to be solved as an LP. Using this result, they were able to drop the symmetric condition, required by Yannakakis, to get a general super polynomial bound for LP formulations of the TSP.

Our contributions and outline of the paper In this paper, we provide more examples of some polytopes associated with hard combinatorial problems as a way to illustrate a general technique for proving lower bounds for the extension complexity of a polytope. The rest of the paper is organized as follows.

In the next section we give background on cut polytopes, a summary of the approach in [14] and [8], and discuss a general strategy for proving lower bounds. In Sect. 3 we discuss four polytopes arising from the 3SAT, subset sum, 3-dimensional matching, and the maximum stable set problems, and prove superpolynomial extension complexity for them. For the stable set polytope, we improve the result of [8] by proving superpolynomial lower bounds for the stable set polytope of cubic planar graphs.

In Sect. 4 we first reprove the result of [8] for the cut polytope directly without introducing the isomorphic correlation polytope, thus avoiding the introduction of the covariance mapping. We then prove how the bounds propagate when one takes the minors of a graph. We use our results to prove superpolynomial lower bounds for the Bell-inequality polytope $\mathrm{CUT}^{\square}(K_{1,n,n})$ described above. This shows that a complete list of Bell inequalities, for the set up with two persons and n binary measurements each, must have superpolynomial size no matter what the dimension of the underlying set of variables is.

As already noted, the max cut problem can be solved in polynomial time for graphs that are K_5 minor free and their cut polytope has a polynomial size extended formulation. Planar graphs are a subset of this class. A suspension of a graph is formed by adding an additional vertex and joining it to all of the graph's original vertices. Barahona [3] proved that the max cut problem is NP-hard for suspensions of planar graphs and hence for K_6 minor-free graphs. We show that this class of graphs has superpolynomial extension complexity. In fact, the graphs used in our proof are suspensions of cubic planar graphs.

2 Preliminaries

We briefly review basic notions about the cut polytope and extension complexity used in later sections. Definitions, theorems and other results for the cut polytope stated in this section are from [7], which readers are referred to for more information. We assume that readers are familiar with basic notions in convex polytope theory such as convex polytope, facet, projection and Fourier–Motzkin elimination. Readers are referred to a textbook [15] for details.

Throughout this paper, we use the following notation. For a graph $G = (V, E)$ we denote the edge between two vertices u and v by uv, and the neighborhood of a vertex v by $\mathrm{N}_G(v)$. We let $[n]$ denote the integers $\{1, 2, \ldots, n\}$.

2.1 Cut polytope and its relatives

The *cut polytope* of a graph $G = (V, E)$, denoted $\mathrm{CUT}^{\square}(G)$, is the convex hull of the cut vectors $\boldsymbol{\delta}_G(S)$ of G defined by all the subsets $S \subseteq V$ in the $|E|$-dimensional vector space $\mathbb{R}^E$. The cut vector $\boldsymbol{\delta}_G(S)$ of G defined by $S \subseteq V$ is a vector in $\mathbb{R}^E$ whose uv-coordinate is defined as follows:

$$\delta_{uv}(S) = \begin{cases} 1 & \text{if } |S \cap \{u, v\}| = 1, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for } uv \in E.$$

If G is the complete graph K_n, we simply denote $\mathrm{CUT}^{\square}(\mathrm{K}_n)$ by $\mathrm{CUT}^{\square}{}_n$.

For completeness, although we will not use it explicitly, we define the *correlation polytope* $\mathrm{COR}^{\square}_n$. For each subset $S \subseteq \{1, 2, \ldots, n\}$ we define the correlation vector $\pi(S)$ of length $(n+1)n/2$ by setting $\pi(S)_{ij} = 1$ if and only if $i, j \in S$, for all $1 \le i \le j \le n$. $\mathrm{COR}^{\square}_n$ is the convex hull of the 2^n correlation vectors $\pi(S)$. A linear map, known as the covariance map, shows the one-to-one correspondence of $\mathrm{COR}^{\square}_n$ and $\mathrm{CUT}^{\square}{}_{n+1}$ (see [7], Ch. 5).

For a subset F of a set E, the *incidence vector* of F (in E)[1] is the vector $\boldsymbol{x} \in \{0, 1\}^E$ defined by $x_e = 1$ for $e \in F$ and $x_e = 0$ for $e \in E \backslash F$. Using this term, the definition of the cut vector can also be stated as follows: $\boldsymbol{\delta}_G(S)$ is the incidence vector of the cut set $\{uv \in E \mid |S \cap \{u, v\}| = 1\}$ in E. When $G = K_n$ we simply denote the cut-vectors by $\delta(S)$.

We now describe an important well-known general class of valid inequalities for $\mathrm{CUT}^{\square}{}_n$ (see, e.g. [7], Ch. 28).

Lemma 1 *For any $n \ge 2$, let $b_1, b_2, \ldots, b_n$ be any set of n integers. The following inequality is valid for* $\mathrm{CUT}^{\square}{}_n$:

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left\lfloor \frac{\left(\sum_{i=1}^n b_i\right)^2}{4} \right\rfloor \tag{1}$$

Proof Let $\delta(S)$ be any cut vector for the complete graph K_n. Then

$$\sum_{1 \le i < j \le n} b_i b_j \delta(S)_{ij} = \left(\sum_{i \in S} b_i\right)\left(\sum_{i \notin S} b_i\right) \tag{2}$$

Now observe that if the sum of the b_i is even the floor sign is redundant and an elementary calculation shows that the right hand side of (2) is bounded from above by the right hand side of (1). If the sum of the b_i is odd then the same calculation gives an upper bound of $(\sum_{i=1}^n b_i + 1)(\sum_{i=1}^n b_i - 1)/4 = (\sum_{i=1}^n b_i)^2/4 - 1/4$ on the right hand side of (2) and the lemma follows. □

[1] The set E is sometimes not specified explicitly when E is clear from the context or the choice of E does not make any difference.

The inequality (1) is called *hypermetric* (respectively, of *negative type*) if the integers b_i can be partitioned into two subsets whose sum differs by one (respectively, zero). A simple example of hypermetric inequalities are the triangle inequalities, obtained by setting three of the b_i to be $+/-1$ and the others to be zero. The most basic negative type inequality is non-negativity, obtained by setting one b_i to 1, another one to -1, and the others to zero. We note in passing that Deza (see Section 6.1 of [7]) showed that each negative type inequality could be written as a convex combination of hypermetric inequalities, so that none of them are facet inducing for $\mathrm{CUT}^{\square}_n$.

For any fixed n there are an infinite number of hypermetric inequalities, but all but a finite number are redundant. This non-trivial fact was proved by Deza, Grishukhin and Laurent (see [7] Section 14.2) and allows us to define the *hypermetric polytope*, which we will refer to again later.

2.2 Extended formulations and extensions

In this paper we make use of the machinery developed and described in Fiorini et al. [8]. A brief summary is given here and the reader is referred to the original paper for more details and proofs.

An *extended formulation* (EF) of a polytope $P \subseteq \mathbb{R}^d$ is a linear system

$$Ex + Fy = g,\ y \geqslant 0 \tag{3}$$

in variables $(x, y) \in \mathbb{R}^{d+r}$, where E, F are real matrices with d, r columns respectively, and g is a column vector, such that $x \in P$ if and only if there exists y such that (3) holds. The *size* of an EF is defined as its number of *inequalities* in the system. Note that for extended formulations written in the form $Ex + Fy = g,\ y \geqslant 0$, the size also equals the number of new variables y as well as the dimension of the nonnegative cone $y \geqslant 0$.

An *extension* of the polytope P is another polytope[2] $Q \subseteq \mathbb{R}^e$ such that P is the image of Q under a linear map. Define the *size* of an extension Q as the number of facets of Q. Furthermore, define the *extension complexity* of P, denoted by $\mathrm{xc}(P)$, as the minimum size of any extension of P.

For a matrix A, let A_i denote the ith row of A and A^j to denote the jth column of A. Let $P = \{x \in \mathbb{R}^d \mid Ax \leqslant b\} = \mathrm{conv}(V)$ be a polytope, with $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$ and $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$. Then $M \in \mathbb{R}_+^{m \times n}$ defined as $M_{ij} := b_i - A_i v_j$ with $i \in [m] := \{1, \ldots, m\}$ and $j \in [n] := \{1, \ldots, n\}$ is the *slack matrix* of P w.r.t. $Ax \leqslant b$ and V. We call the submatrix of M induced by rows corresponding to facets and columns corresponding to vertices the *minimal slack matrix* of P and denote it by $M(P)$. Note that the slack matrix may contain columns that correspond to feasible points that are not vertices of P and rows that correspond to valid inequalities that are not facets of P, and therefore the slack matrix of a polytope is not a uniquely defined object. However every slack matrix of P must contain rows and columns

[2] Even though an extension can also be a polyhedron and not necessarily a polytope, we will consider only those extensions that are polytopes. It is not difficult to see that for a polytope the extension with smallest size would indeed be a polytope.

corresponding to facet-defining inequalities and vertices, respectively. As observed in [8], for proving bounds on the extension complexity of a polytope P it suffices to take any slack matrix of P. Throughout the paper we refer to the minimal slack matrix of P as *the* slack matrix of P and any other slack matrix as *a* slack matrix of P.

A *rank-r nonnegative factorization* of a (nonnegative) matrix M is a factorization $M = QR$ where Q and R are nonnegative matrices with r columns (in case of Q) and r rows (in case of R), respectively. The nonnegative rank of M (denoted by: $\text{rank}_+(M)$) is thus simply the minimum rank of a nonnegative factorization of M. Note that $\text{rank}_+(M)$ is also the minimum r such that M is the sum of r nonnegative rank-1 matrices. In particular, the nonnegative rank of a matrix M is at least the nonnegative rank of any submatrix of M.

The following theorem shows the equivalence of nonnegative rank of the slack matrix, extension and size of an EF.

Theorem 1 (Yannakakis [14]) *Let* $P = \{x \in \mathbb{R}^d \mid Ax \leqslant b\} = \text{conv}(V)$ *be a polytope with* $\dim(P) \geqslant 1$ *with a slack matrix* M. *Then the following are equivalent for all positive integers* r:

(i) M *has nonnegative rank at most* r;
(ii) P *has an extension of size at most* r *(that is, with at most* r *facets);*
(iii) P *has an EF of size at most* r *(that is, with at most* r *inequalities).*

For a given matrix M let $\text{suppmat}(M)$ be the binary support matrix of M, so

$$\text{suppmat}(M)_{ab} = \begin{cases} 1 & \text{if } M_{ab} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

A *rectangle* is the Cartesian product of a set of row indices and a set of column indices. The *rectangle covering bound* is the minimum number of monochromatic rectangles are needed to cover all the 1-entries of the support matrix of M. In general it is difficult to calculate the nonnegative rank of a matrix but sometimes a lower bound can be obtained as shown in the next theorem.

Theorem 2 (Yannakakis [14]) *Let* M *be any matrix with nonnegative real entries and* $\text{suppmat}(M)$ *its support matrix. Then* $\text{rank}_+(M)$ *is lower bounded by the rectangle covering bound for* $\text{suppmat}(M)$.

The following $2^n \times 2^n$ matrix $M^* = M^*(n)$ with rows and columns indexed by n-bit strings a and b, and real nonnegative entries

$$M^*_{ab} := (a^\intercal b - 1)^2.$$

is very useful for obtaining exponential bounds on the EF of various polytopes. This follows from the following result which itself is a consequence of a result of Razborov [13].

Theorem 3 (De Wolf [6]) *Every 1-monochromatic rectangle cover of* $\text{suppmat}(M^*(n))$ *has size* $2^{\Omega(n)}$.

Corollary 1 $\text{rank}_+(M^*(n)) \geqslant 2^{\Omega(n)}$.

Using these ingredients, Fiorini et al. [8] proved the following fundamental result,

Theorem 4 (Lower Bound Theorem) *Let $M(n)$ denote the slack matrix, of* $\text{CUT}^{\square}_n$, *extended with a suitably chosen set of 2^n redundant inequalities. Then $M^*(n-1)$ occurs as a submatrix of $M(n)$ and hence* $\text{CUT}^{\square}_n$ *has extension complexity $2^{\Omega(n)}$.*

They further proved a $2^{\Omega(\sqrt{n})}$ lower bound on the size of extended formulations for the traveling salesman polytope, $\text{TSP}(n)$, by embedding $\text{CUT}^{\square}_n$ as a face of $\text{TSP}(m)$ where $m = O(n^2)$. A similar embedding argument was used to show the same lower bound applies to the stable set polytope, $\text{STAB}(G_n)$ of a graph G_n on n vertices.

2.3 Proving lower bounds for extension complexity

Suppose one wants to prove a lower bound on the extension complexity for a polytope P. Theorem 4 provides a way to do it from scratch: construct a non-negative matrix that has a high non-negative rank and then show that this matrix occurs as a submatrix of a slack matrix of P. Clearly this can be very tricky since there exists neither a general framework for creating such a matrix for each polytope, nor a general way of using a result for one class of polytopes for another.

We now note two observations that are useful in translating results from one polytope to another. Let P and Q be two polytopes. Then,

Proposition 1 *If P is a projection of Q then* $\text{xc}(P) \leqslant \text{xc}(Q)$.

Proposition 2 *If P is a face of Q then* $\text{xc}(P) \leqslant \text{xc}(Q)$.

Naturally there are many other cases where the conditions of neither of these propositions apply and yet a lower bounding argument for one polytope can be derived from another. However we would like to point out that these two propositions already seem to be very powerful. In fact, out of the three lower bounds proved by Fiorini et al. [8] two (for $\text{TSP}(n)$ and $\text{STAB}(G_n)$ for some n vertex graph G_n) use these propositions, while the lower bound on the cut polytope is obtained by showing a direct embedding of $M^*(n)$ in the slack matrix of $\text{CUT}^{\square}_n$.

Fiorini et al. [8] first show $M^*(n)$ is a submatrix of the slack matrix of the correlation polytope $\text{COR}^{\square}_n$ and then use its affine equivalence with $\text{CUT}^{\square}_{n+1}$. This is followed by an embedding of $\text{CUT}^{\square}_{n+1}$ as a face of $\text{STAB}(G(n^2))$ where $G(n^2)$ is a graph with $O(n^2)$ vertices and $O(n^2)$ edges implying a *worst case* lower bound of $2^{\Omega(\sqrt{n})}$ for the extension complexity of the stable set polytope of a graph with n vertices. Similarly, worst case lower bounds are obtained for the traveling salesman polytope by embedding $\text{COR}^{\square}_n$ in a face of $\text{TSP}(n^2)$.

In the next section we will use these propositions to show superpolynomial lower bounds on the extension complexities of polytopes associated with four NP-hard problems.

3 Polytopes for some NP-hard problems

In this section we use the method of Sect. 2.3 to show super polynomial extension complexity for polytopes related to the following problems: subset sum, 3-dimensional

matching and stable set for cubic planar graphs. These proofs are derived by applying this method to standard reductions from 3SAT, which is our starting point.

3.1 3SAT

For any given 3SAT formula Φ with n variables in conjunctive normal form define the polytope $\mathrm{SAT}(\Phi)$ as the convex hull of all satisfying assignments. That is,

$$\mathrm{SAT}(\Phi) := \mathrm{conv}(\{x \in \{0,1\}^n \mid \Phi(x) = 1\})$$

The following theorem and its proof are implicit in [8], making use of the correlation polytope. We provide the proof for completeness, stated this time in terms of the cut polytope.

Theorem 5 *For every n there exists a 3SAT formula Φ with $O(n)$ variables and $O(n)$ clauses such that* $\mathrm{xc}(\mathrm{SAT}(\Phi)) \geqslant 2^{\Omega(\sqrt{n})}$.

Proof For the complete graph K_m we define a boolean formula Φ_m in conjunctive normal form over the variables x_{ij} for $i, j \in \{1, \ldots, m\}$ such that every clause in Φ_m has three literals and $\mathrm{CUT}^{\square}(K_m)$ is a projection of $\mathrm{SAT}(\Phi_m)$.

Consider the relation $x_{ij} = x_{ii} \oplus x_{jj}$, where $\oplus$ is the xor operator. The boolean formula

$$(x_{ii} \vee \overline{x}_{jj} \vee x_{ij}) \wedge (\overline{x}_{ii} \vee x_{jj} \vee x_{ij}) \wedge (x_{ii} \vee x_{jj} \vee \overline{x}_{ij}) \wedge (\overline{x}_{ii} \vee \overline{x}_{jj} \vee \overline{x}_{ij})$$

is true if and only if $x_{ij} = x_{ii} \oplus x_{jj}$ for any assignment of the variables x_{ii}, x_{jj} and x_{ij}.

Now define Φ_m as

$$\Phi_m := \bigwedge_{\substack{i,j \in [m] \\ i \neq j}} \left[(x_{ii} \vee \overline{x}_{jj} \vee x_{ij}) \wedge (\overline{x}_{ii} \vee x_{jj} \vee x_{ij}) \wedge (x_{ii} \vee x_{jj} \vee \overline{x}_{ij}) \wedge (\overline{x}_{ii} \vee \overline{x}_{jj} \vee \overline{x}_{ij}) \right].$$

It is easy to see that any vertex of $\mathrm{SAT}(\Phi_m)$ can be projected to a vertex of $\mathrm{CUT}^{\square}(K_m)$ by projecting out the variables x_{ii} for $i \in \{1, \ldots, m\}$ since $x_{ij} = 1$ if and only if x_{ii} and x_{jj} are assigned different values, and hence the assignment defines a cut in K_m. Furthermore, any vertex of $\mathrm{CUT}^{\square}(K_m)$ can be extended to any of the two assignments that correspond to the cut defined by the vector. That is, if a cut vector of $\mathrm{CUT}^{\square}(K_m)$ partitions the set of vertices into S and $\overline{S}$ then extending the cut vector by assigning $x_{ii} = 1$ if $i \in S$ and $x_{ii} = 0$ if $i \in \overline{S}$ (or the other way round) defines a satisfying assignment for Φ_m and therefore a vertex of $\mathrm{SAT}(\Phi_m)$.

Therefore, $\mathrm{CUT}^{\square}(K_m)$ is a projection of $\mathrm{SAT}(\Phi_m)$, and by Proposition 1 we can conclude that $\mathrm{xc}(\mathrm{SAT}(\Phi_m)) \geqslant \mathrm{xc}(\mathrm{CUT}^{\square}(K_m)) \geqslant 2^{\Omega(m)}$. Note that Φ_m has $O(n^2)$ variables and clauses. Therefore, we have the desired result. □

3.2 Subset sum

The subset sum problem is a special case of the knapsack problem. Given a set of n integers $A = \{a_1, \ldots, a_n\}$ and another integer b, the subset sum problems asks whether any subset of A sums exactly to b. Define the subset sum polytope SUBSETSUM(A, b) as the convex hull of all characteristic vectors of the subsets of A whose sum is exactly b.

$$\text{SUBSETSUM}(A, b) := \text{conv}\left(\left\{x \in \{0, 1\}^n \mid \sum_{i=1}^{n} a_i x_i = b\right\}\right)$$

The subset sum problem then is asking whether SUBSETSUM(A, b) is empty for a given set A and integer b. Note that this polytope is a face of the knapsack polytope

$$\text{KNAPSACK}(A, b) := \text{conv}\left(\left\{x \in \{0, 1\}^n \mid \sum_{i=1}^{n} a_i x_i \leqslant b\right\}\right)$$

In this subsection we prove that the subset sum polytope (and hence the knapsack polytope) can have superpolynomial extension complexity.

Theorem 6 *For every 3SAT formula Φ with n variables and m clauses, there exists a set of integers $A(\Phi)$ and integer b with $|A| = 2n + 2m$ such that* SAT(Φ) *is the projection of* SUBSETSUM(A, b).

Proof Suppose formula Φ is defined in terms of variables $x_1, x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$. We use a standard reduction from 3SAT to subset sum (e.g., [5], Section 34.5.5). We define $A(\Phi)$ and b as follows. Every integer in $A(\Phi)$ as well as b is an $(n + m)$-digit number (in base 10). The first n digit correspond to the variables and the last m digits correspond to each of the clauses.

$$b_j = \begin{cases} 1, & \text{if } 1 \leqslant j \leqslant n \\ 4, & \text{if } n + 1 \leqslant j \leqslant n + m \end{cases}.$$

Next we construct $2n$ integers v_i, v'_i for $i \in \{1, \ldots, n\}$.

$$v_{ij} = \begin{cases} 1, & \text{if } j = i \text{ or } x_i \in C_{j-n} \\ 0, & \text{otherwise} \end{cases},$$

$$v'_{ij} = \begin{cases} 1, & \text{if } j = i \text{ or } \overline{x}_i \in C_{j-n} \\ 0, & \text{otherwise} \end{cases}.$$

Finally, we construct $2m$ integers s_i, s'_i for $i \in \{1, \ldots, m\}$.

$$s_{ij} = \begin{cases} 1, & \text{if } j = n + i \\ 0, & \text{otherwise} \end{cases},$$

Table 1 The base 10 numbers created as an instance of subset-sum for the 3SAT formula $(x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3)$

	x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1	1	0	0	1	0	1	0
v_1'	1	0	0	0	1	0	1
v_2	0	1	0	0	1	1	0
v_2'	0	1	0	1	0	0	1
v_3	0	0	1	1	1	0	0
v_3'	0	0	1	0	0	1	1
s_1	0	0	0	1	0	0	0
s_1'	0	0	0	2	0	0	0
s_2	0	0	0	0	1	0	0
s_2'	0	0	0	0	2	0	0
s_3	0	0	0	0	0	1	0
s_3'	0	0	0	0	0	2	0
s_4	0	0	0	0	0	0	1
s_4'	0	0	0	0	0	0	2
b	1	1	1	4	4	4	4

$$s'_{ij} = \begin{cases} 2, & \text{if } j = n + i \\ 0, & \text{otherwise} \end{cases}.$$

We define the set $A(\Phi) = \{v_1, \ldots, v_n, v'_1, \ldots, v'_n, s_1, \ldots, s_m, s'_1, \ldots, s'_m\}$. Table 1 illustrates the construction for the 3SAT formula $(x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3)$.

Consider the subset-sum instance with $A(\Phi), b$ as constructed above for any 3SAT instance Φ. Let S be any subset of $A(\Phi)$. If the elements of S sum exactly to b then it is clear that for each $i \in \{1, \ldots, n\}$ exactly one of v_i, v'_i belong to S. Furthermore, setting $x_i = 1$ if $v_i \in S$ or $x_i = 0$ if $v'_i \in S$ satisfies every clause. Thus the characteristic vector of S restricted to $\{v_1, \ldots, v_n\}$ is a satisfying assignment for the corresponding SAT formula.

Also, if Φ is satisfiable then the instance of subset sum thus created has a solution corresponding to each satisfying assignment: Pick v_i if $x_i = 1$ or v'_i if $x_i = 0$ in an assignment. Since the assignment is satisfying, every clause is satisfied and so the sum of digits corresponding to each clause is at least 1. Therefore, for a clause C_j either s_j or s'_j or both can be picked to ensure that the sum of the corresponding digits is exactly 4. Note that there is unique way to do this.

This shows that every vertex of the subset sum polytope SUBSETSUM$(A(\Phi), b)$ projects to a vertex of SAT(Φ) and every vertex of SAT(Φ) can be lifted to a vertex of SUBSETSUM$(A(\Phi), b)$. The projection is defined by dropping every coordinate except those corresponding to the numbers v_i in the reduction described above. The lifting is defined by the procedure in the preceding paragraph. Hence, SAT(Φ) is a projection of SUBSETSUM$(A(\Phi), b)$. □

Combining the preceding two theorems we obtain the following.

Corollary 2 *For every natural number $n \geqslant 1$, there exists an instance A, b of the subset-sum problem with $O(n)$ integers in A such that* $\mathrm{xc}(\mathrm{SUBSETSUM}(A, b)) \geqslant 2^{\Omega(\sqrt{n})}$.

As mentioned above, the polytope $\mathrm{SUBSETSUM}(A, b)$ is a face of $\mathrm{KNAPSACK}(A, b)$ and hence Corollary 2 implies a superpolynomial lower bound for the Knapsack polytope. We would like to note that a similar bound for the Knapsack polytope was proved recently and independently by Pokutta and Vyve [12].

3.3 3d-matching

Consider a hypergraph $G = ([n], E)$, where E contains triples for some $i, j, k \in [n]$ where i, j, k are distinct. A subset $E' \subseteq E$ is said to be a 3-dimensional matching if all the triples in E' are disjoint. The $3d$-matching polytope $\mathrm{3DM}(G)$ is defined as the convex hull of the characteristic vectors of every $3d$-matching of G. That is,

$$\mathrm{3DM}(G) := \mathrm{conv}\left(\left\{\chi(E') \mid E' \subseteq E \quad \text{is a } 3d\text{-matching}\right\}\right)$$

It is often customary to consider only hypergraphs defined over three disjoint set of vertices X, Y, Z such that the hyperedges are subsets of $X \times Y \times Z$. Observe that any hypergraph G can be converted into a hypergraph H in such a form by making three copies of the vertex set V, V', V'' and using a hyperedge (i, j', k'') in H if and only if (i, j, k) is a hyperedge in G. The following lemma shows that $\mathrm{xc}(\mathrm{3DM}(G))$ is less than or equal to $\mathrm{xc}(\mathrm{3DM}(H))$. Since we are interested in superpolynomial lower bounds, it suffices for our purposes to consider the former form.

Lemma 2 *Let $G = ([n], E)$ be a hypergraph on n vertices. Consider the hypergraph $H = ([3n], E')$ where $(i, n + j, 2n + k) \in E'$ iff $(i, j, k) \in E$. Then,* $\mathrm{xc}(\mathrm{3DM}(G)) \leqslant \mathrm{xc}(\mathrm{3DM}(H))$.

Proof Identifying the variables corresponding to the hyperedge (i, j, k) in $\mathrm{3DM}(G)$ with the variable corresponding to the hyperedge $(i, n + j, 2n + k)$, $i < j < k$ in $\mathrm{3DM}(H)$, we see that $\mathrm{3DM}(G)$ is a projection of $\mathrm{3DM}(H)$. Thus by Proposition 1 $\mathrm{xc}(\mathrm{3DM}(G)) \leqslant \mathrm{xc}(\mathrm{3DM}(H))$. □

The 3d-matching problem asks: given a hypergraph G, does there exist a 3d-matching that covers all vertices? This problem is known to be NP-complete and was one of Karp's 21 problems proved to be NP-complete [9,11]. Note that this problem can be solved by linear optimization over the polytope $\mathrm{3DM}(G)$ and therefore it is to be expected that $\mathrm{3DM}(G)$ would not have a polynomial size extended formulation.

In this subsection, we show that the 3d-matching polytope has superpolynomial extension complexity in the worst case. We prove this using a standard reduction from 3SAT to 3d-Matching used in the NP-completeness proof for the later problem (See [9]). The form of this reduction, which is very widely used, employs a gadget for each variable along with a gadget for each clause. We omit the exact details for the reduction here because we are only interested in the correctness of the reduction and the variable gadget (See Fig. 1).

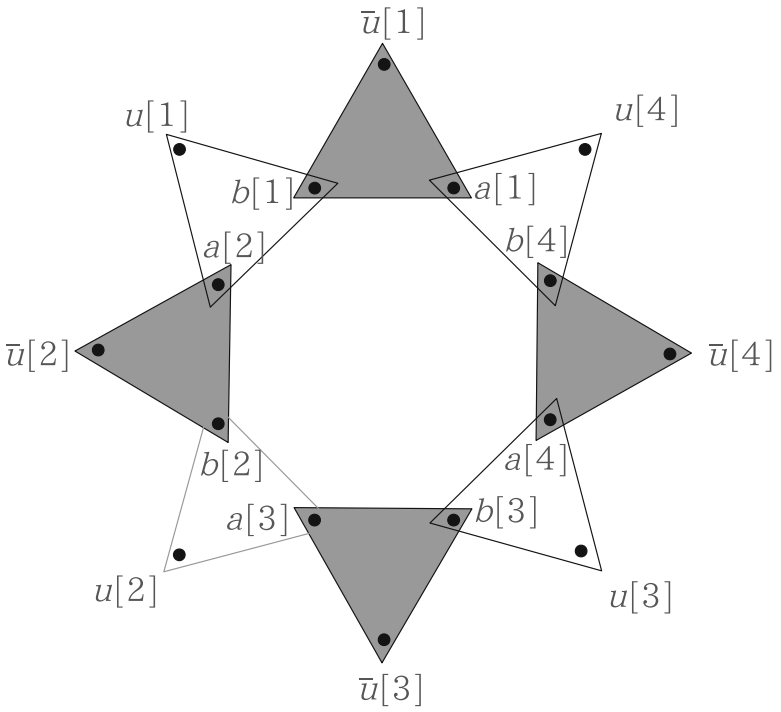


Fig. 1 Gadget for a variable

In the reduction, any 3SAT formula Φ is converted to an instance of a 3d-matching by creating a set of hyperedges for every variable (See Fig. 1) along with some other hyperedges that does not concern us for our result. The crucial property that we require is the following: any satisfiable assignment of Φ defines some (possibly more than one) 3d-matching. Furthermore, in any maximal matching either only the light hyperedges or only the dark hyperedges are picked, corresponding to setting the corresponding variable to, say, true or false respectively. Using these facts we can prove the following:

Theorem 7 *Let Φ be an instance of 3SAT and let H be the hypergraph obtained by the reduction above. Then* $\mathrm{SAT}(\Phi)$ *is the projection of a face of* $\mathrm{3DM}(H)$.

Proof Let the number of hyperedges in the gadget corresponding to a variable x be $2k(x)$. Then, the number of hyperedges picked among these hyperedges in any matching in H is at most $k(x)$. Therefore, if $y_1, \ldots, y_{2k(x)}$ denote the variables corresponding to these hyperedges in the polytope $\mathrm{3DM}(H)$ then $\sum_{i=1}^{2k(x)} y_i \leqslant k(x)$ is a valid inequality for $\mathrm{3DM}(H)$. Consider the face F of $\mathrm{3DM}(H)$ obtained by adding the equation $\sum_{i=1}^{2k(x)} y_i = k(x)$ corresponding to each variable x appearing in Φ.

Any vertex of $\mathrm{3DM}(H)$ lying in F selects either all light hyperedges or all dark hyperedges. Furthermore, if in such a vertex of $\mathrm{3DM}(H)$, light vertices are picked from the gadget corresponding to a variable x then the corresponding satisfying assignment sets x to true. Since, in any vertex of $\mathrm{3DM}(H)$ that lies in F all variables corresponding to light edges have the same value (the value of variable x in the corresponding satisfying assignment), projecting out all variables except one variable y_i corresponding to any fixed (arbitrarily chosen) light hyperedge for each variable in Φ gives a valid satisfying assignment for Φ and thus a vertex of $\mathrm{SAT}(\Phi)$. Conversely, any vertex of $\mathrm{SAT}(\Phi)$ can be extended to a vertex of $\mathrm{3DM}(H)$ lying in F.

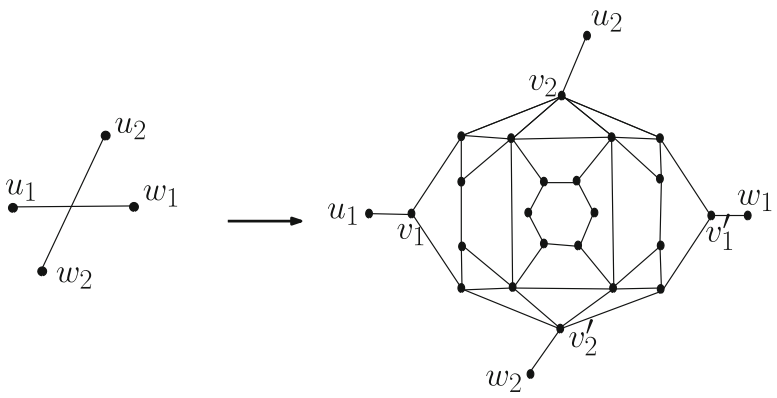


Fig. 2 Gadget to remove a crossing

Therefore, SAT(Φ) is the projection of F. □

The number of vertices in H is $O(nm)$ where n is the number of variables and m the number of clauses in Φ. Considering only the 3SAT formulae with high extension complexity from Sect. 3.1, we have $m = O(n)$. Therefore, considering only the hypergraphs arising from such 3SAT formulae and using Propositions 1 and 2, we have that

Corollary 3 *For every natural number $n \geqslant 1$, there exists a hypergraph H with $O(n)$ vertices such that* $\text{xc}(\text{3DM}(H)) \geqslant 2^{\Omega(n^{1/4})}$.

3.4 Stable set for cubic planar graphs

Now we show that STAB(G) can have superpolynomial extension complexity even when G is a cubic planar graph. Our starting point is the following result proved by Fiorini et al. [8].

Theorem 8 (Fiorini et al. [8]) *For every natural number $n \geqslant 1$ there exists a graph G such that G has $O(n)$ vertices and $O(n)$ edges, and* $\text{xc}(\text{STAB}(G)) \geqslant 2^{\Omega(\sqrt{n})}$.

We start with this graph and convert it into a cubic planar graph G' with $O(n^2)$ vertices and extension complexity at least $2^{\Omega(\sqrt{n})}$.

3.4.1 Making a graph planar

For making any graph G planar without reducing the extension complexity of the associated stable set polytope, we use the same gadget used by Garey, Johnson and Stockmeyer [10] in the proof of NP-completeness of finding maximum stable set in planar graph. Start with any planar drawing of G and replace every crossing with the gadget H with 22 vertices shown in Fig. 2 to obtain a graph G'. The following theorem shows that STAB(G) is the projection of a face of STAB(G').

Theorem 9 *Let G be a graph and let G' be obtained from a planar embedding of G by replacing every edge intersection with a gadget shown in Fig. 2. Then,* STAB(G) *is the projection of a face of* STAB(G').

Table 2 Values of s_{ij}

$i \backslash j$	2	1	0
2	9	8	7
1	9	9	8
0	8	8	7

Proof Let $H_1, \ldots, H_k$ be the gadgets introduced in G to obtain G'. Any stable set S of G' contains some, or possibly no, vertices from the gadgets introduced. For any gadget $H \in \{H_1, \ldots, H_k\}$, let V_H denote the set of vertices of H. Then, $S \cap V_H$ is a stable set for H. Denote by s_{ij} the size of maximum independent set in H containing exactly i vertices out of $\{v_1, v_1'\}$ and exactly j vertices out of $\{v_2, v_2'\}$. Table 2 lists the values of s_{ij} for $i, j \in \{1, 2\}$. The table is essentially Table 1 from [10] but their table lists the size of the minimum vertex cover and so we subtract the entries from the number of nodes in the gadget which is 22.

As we see, every stable set of H has no more than 9 vertices and hence $\sum_{i \in V_H} x_i \leqslant 9$ is a valid inequality for STAB(G'). Consider the face

$$F := \text{STAB}(G') \bigcap_{i=1}^{k} \left\{ x \mid \sum_{j \in V_{H_i}} x_j = 9 \right\}$$

Consider any stable set S of G' lying in the face F. It is clear that at least one vertex must be picked in S out of each $\{v_1, v_1'\}$ and $\{v_2, v_2'\}$. Therefore, for any edge (u, v) in G it is not possible that both u, v are in S and hence projecting out the vertices from the gadgets we get a valid stable set for G. Alternatively, any independent set from G can be extended to a stable set in G' by selecting the appropriate maximum stable set from each of the gadgets. Therefore, STAB(G) is a projection of F. □

Since for any graph G with $O(n)$ edges, the number of gadgets introduced $k \leqslant O(n^2)$, we have that the graph G' in the above theorem has at most $O(n^2)$ vertices and edges. Therefore we have a planar graph G' with at most $O(n^2)$ vertices and $O(n^2)$ edges. This together with Theorem 8, Theorem 9, Proposition 1, and Proposition 2 yields the following corollary.

Corollary 4 *For every n there exists a planar graph G with $O(n^2)$ vertices and $O(n^2)$ edges such that* $\text{xc}(\text{STAB}(G)) \geqslant 2^{\Omega(\sqrt{n})}$.

3.4.2 Making a graph cubic

Suppose we have a graph G and we transform it into another graph G' by performing one of the following operations:

ReduceDegree Replace a vertex v of G of degree $\delta \geqslant 4$ with a cycle $C_v = (v_1, v_1', \ldots, v_\delta, v_\delta')$ of length 2δ and connect the neighbors of v to alternating vertices $(v_1, v_2, \ldots, v_\delta)$ of the cycle (See Fig. 3a).

RemoveBridge Replace any degree two vertex v in G by a four cycle v_1, v_2, v_3, v_4. Let u and w be the neighbors of v in G. Then, add the edges (u, v_1) and (v_3, w). Also add the edge (v_2, v_4) in the graph (See Fig. 3b).

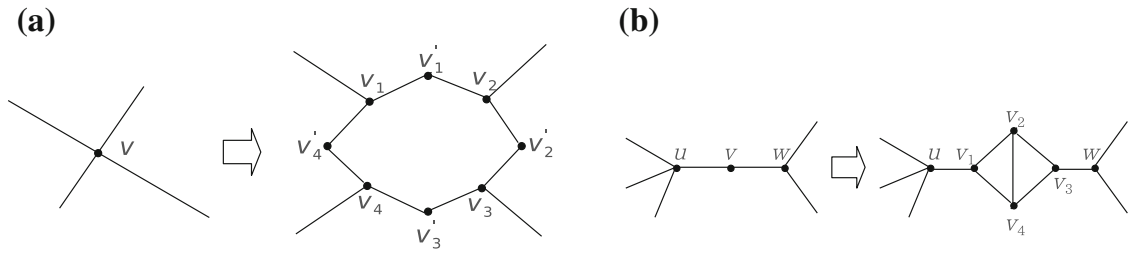


Fig. 3 Gadgets. **a** Replace a degree 4 vertex, **b** remove a degree two vertex

RemoveTerminal Replace any vertex with degree either zero or one with a triangle. In case of degree one, attach any one vertex of the triangle to the neighbor. For example, if uv is an edge with v a vertex of degree one, then we replace v with a triangle and attach u to an arbitrary vertex of the triangle.

Theorem 10 *Let G be any graph and let G' be obtained by performing any number of operation ReduceDegree, RemoveBridge, or RemoveTerminal described above on G. Then* STAB(G) *is the projection of a face of* STAB(G').

Proof It suffices to show that the theorem is true for a single application of either of the three operations.

Consider an application of the operation ReduceDegree. Let C be the gadget that was used to replace a vertex v in G to obtain G'. Let V_C denote the set of vertices of C. Then, for any stable set S of G', the set $S \cap V_C$ is a stable set for C. Every stable set of C has no more than $\delta = |C|/2$ vertices and hence $\sum_{v \in V_C} x_v \leqslant |C|/2$ is a valid inequality for STAB(G'). Consider the face

$$F := \text{STAB}(G') \bigcap_{i=1}^{k} \left\{ x \mid \sum_{v \in V_{C_i}} x_v = |C_i|/2 \right\}$$

Any stable set S lying in the face F must either select all vertices $(v'_1, \ldots, v'_\delta)$ or $(v_1, \ldots, v_\delta)$ for each cycle C of length 2δ. Furthermore, if S contains any neighbor of v then the former set of vertices must be picked in S. Also, any stable set of G can be extended to a stable set of G' that lies in F. For each stable set in F projecting out every vertex of the cycles introduced except any one that has degree 3 gives us a valid stable set of G and therefore, STAB(G) is the projection of a face of STAB(G').

On the other hand, suppose operation RemoveBridge is used to transform any graph G into a graph G'. Let C be the gadget used to replace a vertex v in G. Let $V_C = (v_1, v_2, v_3, v_4)$ denote the set of vertices of C. Then, for any stable set S in G', the set $S \cap V_C$ is a stable set for C. It is easy to see that every stable set of C satisfies the inequality $x_{v_1} + x_{v_3} + 2(x_{v_2} + x_{v_4}) \leqslant 2$ and hence it is a valid inequality for STAB(G'). Define h_C to be the equality obtained from the previous inequality for a gadget C and consider the face

$$F := \text{STAB}(G') \bigcap_{i=1}^{k} h_{C_i}$$

Any stable set S of G' lying in the face F must either select vertices (v_1, v_3) or one of v_2 or v_4 for each gadget C. Furthermore, if S contains any neighbor of v then it contains exactly one of v_2 or v_4 but not both. Also, any stable set of G can be extended to a stable set of G' that lies in F. For each stable set in F projecting out every vertex of the gadget and using the map $x_v = x_{v_2} + x_{v_4}$ gives us a valid stable set of G and therefore, STAB(G) is the projection of F, a face of STAB(G').

Finally it is easy to see that if G' is obtained by applying operation RemoveTerminal on a graph G then STAB(G) is a projection of STAB(G'). □

If G has n vertices and m edges then first applying operation ReduceDegree until every vertex has degree at most 3, and then applying operation RemoveBridge and RemoveTerminal repeatedly until no vertex of degree 0, 1 or 2 is left, produces a graph that has $O(n+m)$ vertices and $O(n+m)$ edges. Furthermore, any application of the three operations do not make a planar graph non-planar. Combining this fact with Theorem 10, Corollary 4, Proposition 1, and Proposition 2, we have

Corollary 5 *For every natural number $n \geqslant 1$ there exists a cubic planar graph G with $O(n)$ vertices and edges such that* $\text{xc}(\text{STAB}(G)) \geqslant 2^{\Omega(n^{1/4})}$.

In summary, starting from a graph whose stable set polytope has high extension complexity, we trade edges for vertices by replacing intersections with vertices thus ensuring planarity at the cost of making the number of vertices quadratic. Then we make the planar graph cubic with just a linear blowup. Taken together, we end up with a cubic planar graph with a quadratic number of vertices and edges compared to the original graph but whose stable set polytope has complexity as large as that of the original graph. The quadratic blowup in the number of vertices results in the lower bound becoming $2^{\Omega(n^{1/4})}$ from $2^{\Omega(\sqrt{n})}$.

4 Extended formulations for CUT$^{\square}(G)$ and its relatives

We use the results described in the previous section to obtain bounds on the extension complexity of the cut polytope of graphs. We begin by reviewing the result in [8] for CUT$^{\square}_n$ using a direct argument that avoids introducing correlation polytopes. For any integer $n \geq 2$ consider the integers $b_1 = \cdots = b_{n-1} = 1$ and $b_n = 3 - n$. Let $b = (b_1, b_2, \ldots, b_n)$ be the corresponding n-vector. Inequality (1) for this b-vector is easily seen to be of negative type and can be written

$$\sum_{1 \leq i < j \leq n-1} x_{ij} \leq 1 + (n-3) \sum_{i=1}^{n-1} x_{in}. \tag{4}$$

Lemma 3 *Let S be any cut in K_n not containing vertex n and let $\delta(S)$ be its corresponding cut vector. Then the slack of $\delta(S)$ with respect to (4) is $(|S|-1)^2$.*

Proof

$$1 + (n-3)\sum_{i=1}^{n-1}\delta(S)_{in} - \sum_{1\le i<j\le n-1}\delta(S)_{ij} = 1 + (n-3)|S| - |S|(n-|S|-1)$$
$$= |S|^2 - 2|S| + 1.$$

□

Let us label a cut S by a binary n-vector a where $a_i = 1$ if and only if $i \in S$. Under the conditions of the lemma we observe that the slack $(|S|-1)^2 = (a^T b - 1)^2$ since we have $a_n = 0$ and $b_1 = \cdots = b_{n-1} = 1$. Now consider consider any subset T of $\{1, 2, \ldots, n-1\}$ and set $b_i = 1$ for $i \in T$, $b_n = 3 - |T|$ and $b_i = 0$ otherwise. We form a 2^{n-1} by 2^{n-1} matrix M as follows. Let the rows and columns be indexed by subsets T and S of $\{1, 2, \ldots, n-1\}$, labeled by the n-vectors a and b as just described. A straight forward application of Lemma 3 shows that $M = M^*(n-1)$. Hence using the fact that the non-negative rank of a matrix is at least as large as that of any of its submatrices, we have that every extended formulation of $\mathrm{CUT}^{\square}_n$ has size $2^{\Omega(n)}$.

Recall the hypermetric polytope, defined in Sect. 2.1, is the intersection of all hypermetric inequalities. As remarked, nonnegative type inequalities are weaker than hypermetric inequalities and so valid for this polytope. In addition all cut vertices satisfy all hypermetric inequalities. Therefore $M = M^*(n-1)$ is also a submatrix of a slack matrix for the hypermetric polytope for K_n. So this polytope also has extension complexity at least $2^{\Omega(n)}$.

Finally let us consider the polytope, which we denote P_n, defined by the inequalities used to define rows of the slack matrix M above. We will show that membership testing for P_n is co-NP-complete.

Theorem 11 *Let P_n be the polytope defined as above, and let $x \in \mathbb{R}^{n(n-1)/2}$. Then it is co-NP-complete to decide if $x \in P_n$.*

Proof Clearly if $x \notin P_n$ then this can be witnessed by a violated inequality of type (4), so the problem is in co-NP.

To see the hardness we do a reduction from the clique problem: given graph $G = (V, E)$ on n vertices and integer k, does G have a clique of size at least k? Since a graph has a clique of size k if and only if its suspension has a clique of size $k+1$ we can assume *wlog* that G is a suspension with vertex v_n connected to every other vertex.

Form a vector x as follows:

$$x_{ij} = \begin{cases} 1/k, & \text{if } j = n \\ 2/k, & \text{if } j \neq n \text{ and } ij \in E, \\ -n^2 & \text{otherwise} \end{cases}$$

Fix an integer t, $2 \le t \le n$ and consider a b-vector with $b_n = 3 - t$, and with $t-1$ other values of $b_i = 1$. Without loss of generality we may assume these are

labeled $1, 2, \ldots, t-1$. Let T be the induced subgraph of G on these vertices. The corresponding non-negative type inequality is:

$$\sum_{1 \le i < j \le t-1} x_{ij} \le 1 + (t-3) \sum_{i=1}^{t-1} x_{in}. \tag{5}$$

Suppose T is a complete subgraph. Then the left hand side minus the right hand side of (5) is

$$\frac{2(t-1)(t-2)}{2k} - \left(1 + \frac{(t-3)(t-1)}{k}\right) = \frac{t-k-1}{k}.$$

This will be positive if and only if $t \ge k+1$, in which case x violates (5). On the other hand if T is not a complete subgraph then the left hand side of (5) is always negative and so the inequality is satisfied. Therefore x satisfies all inequalities defining rows of M if an only if G has no clique of size at least k. □

4.1 Cut polytope for minors of a graph

A graph H is a *minor* of a graph G if H can be obtained from G by contracting some edges, deleting some edges and isolated vertices, and relabeling. In the introduction we noted that if an n vertex graph G has no K_5-minor then $\text{CUT}^{\square}(G)$ has $O(n^3)$ extension complexity. We will now show that the extension complexity of a graph G can be bounded from below in terms of its largest clique minor.

Lemma 4 *Let G be a graph and let H be obtained by deleting an edge e of G, then $\text{CUT}^{\square}(G)$ is an extension of $\text{CUT}^{\square}(H)$. In particular, $\text{xc}(\text{CUT}^{\square}(G)) \geqslant \text{xc}(\text{CUT}^{\square}(H))$.*

Proof Any vertex v of $\text{CUT}^{\square}(H)$ defines a cut on graph H. Let H_1 and H_2 be the two subsets of vertices defined by this cut. Consider the same subsets over the graph G, and the corresponding cut vector for G. This vector is the same as v extended with a coordinate corresponding to the edge e in G which was removed to obtain H. The value on this coordinate is 0 if the end points of this edge belong to different sides of the cut and 1 otherwise. In either case, every vertex of $\text{CUT}^{\square}(G)$ projects to a vertex of $\text{CUT}^{\square}(H)$ and every vertex of $\text{CUT}^{\square}(H)$ can be lifted to a vertex of $\text{CUT}^{\square}(G)$.

Therefore, $\text{CUT}^{\square}(G)$ is an extended formulation of $\text{CUT}^{\square}(H)$ and hence by Proposition 1

$$\text{xc}(\text{CUT}^{\square}(G)) \geqslant \text{xc}(\text{CUT}^{\square}(H)).$$

□

Lemma 5 *Let G be a graph and let H be obtained by deleting a vertex v of G, then $\text{CUT}^{\square}(G)$ is an extension of $\text{CUT}^{\square}(H)$. In particular, $\text{xc}(\text{CUT}^{\square}(G)) \geqslant \text{xc}(\text{CUT}^{\square}(H))$.*

The proof is analogous to that of Lemma 4.

Lemma 6 *Let G be a graph and let H be obtained by contracting an edge $e = (u, v)$ of G, then $\mathrm{CUT}^{\square}(H)$ is the projection of a face of $\mathrm{CUT}^{\square}(G)$. In particular, $\mathrm{xc}(\mathrm{CUT}^{\square}(G)) \geqslant \mathrm{xc}(\mathrm{CUT}^{\square}(H))$.*

Proof Suppose that the vertices u, v are contracted to a new vertex labeled u in H. Consider the inequality $x_e \geqslant 0$. This is a valid inequality for $\mathrm{CUT}^{\square}(G)$. Consider the face

$$F = \mathrm{CUT}^{\square}(G) \cap \{x_e = 0\}.$$

Consider any vertex of F. Project out x_e and also $x_{e'}$ for any $e' = (v, w)$ if (u, w) is an edge in G. Clearly this linear map projects every vertex in F to a vertex of $\mathrm{CUT}^{\square}(H)$. Also, any vertex of $\mathrm{CUT}^{\square}(H)$ can be lifted to a vertex of $\mathrm{CUT}^{\square}(G)$ lying in F as follows. Set $x_e = 0$, and for an edge $e' = (v, w)$ in G we set $x_{vw} = x_{uw}$. It is easy to check that this is a valid cut for G that lies in F.

It is thus clear that $\mathrm{CUT}^{\square}(H)$ is obtained as the projection of a face of $\mathrm{CUT}^{\square}(G)$ by setting $x_e = 0$ for the contracted edge e. Hence by Proposition 2

$$\mathrm{xc}(\mathrm{CUT}^{\square}(G)) \geqslant \mathrm{xc}(\mathrm{CUT}^{\square}(H)).$$

□

Combining Lemma 4, 5, and 6 we get the following theorem.

Theorem 12 *Let G be a graph and H be a minor of G. Then,*

$$\mathrm{xc}(\mathrm{CUT}^{\square}(G)) \geqslant \mathrm{xc}(\mathrm{CUT}^{\square}(H)).$$

Using the above theorem together with the result of [8] that the extension complexity of $\mathrm{CUT}^{\square}(K_n)$ is at least $2^{\Omega(n)}$ we get the following result.

Corollary 6 *The extension complexity of $\mathrm{CUT}^{\square}(G)$ for a graph G with a K_n minor is at least $2^{\Omega(n)}$.*

Using this theorem we can immediately prove that the Bell inequality polytopes mentioned in the introduction have exponential complexity.

Corollary 7 *The extension complexity of $\mathrm{CUT}^{\square}(K_{1,n,n})$ is at least $2^{\Omega(n)}$.*

Proof Pick any matching of size n between the vertices in each of the two parts of cardinality n. Contracting the edges in this matching yields K_{n+1} and the result follows. □

4.2 Cut polytope for K_6 minor-free graphs

Let $G = (V, E)$ be any graph with $V = \{1, \ldots, n\}$. Consider the suspension G' of G obtained by adding an extra vertex labeled 0 with edges to all vertices V.

Theorem 13 *Let $G = (V, E)$ be a graph and let G' be a suspension over G. Then* $\mathrm{STAB}(G)$ *is the projection of a face of* $\mathrm{CUT}^{\square}(G')$.

Proof The polytope $\mathrm{STAB}(G)$ is defined over variables x_i corresponding to each of the vertex $i \in V$ whereas the polytope $\mathrm{CUT}^{\square}(G')$ is defined over the variables x_{ij} for $i, j \in \{0, \ldots, n\}$.

Any cut vertex C of $\mathrm{CUT}^{\square}(G')$ defines sets $S, \overline{S}$ such that $x_{ij} = 1$ if and only if $i \in S$, $j \in \overline{S}$. We may assume that $0 \in \overline{S}$ by interchanging S and $\overline{S}$ if necessary. For every edge $e = (k, l)$ in G consider an inequality $h_e := \{x_{0k} + x_{0l} - x_{kl} \geqslant 0\}$. It is clear that h_e is a valid inequality for $\mathrm{CUT}^{\square}(G')$ for all edges e in G. Furthermore, h_e is tight for a cut vector in G' if and only if either k, l do not lie in the same cut set or k, l both lie in the cut set containing 0. Therefore consider the face

$$F := \mathrm{CUT}^{\square}(G') \bigcap_{(i,j)\in E} \{x_{0i} + x_{0j} - x_{ij} = 0\}.$$

Each vertex in F can be projected to a valid stable set in G by projecting onto the variables $x_{01}, x_{02}, \ldots, x_{0n}$. Furthermore, every stable set S in G can be extended to a cut vector for G' by taking the cut vector corresponding to $S, \overline{S} \cup \{0\}$. Therefore, $\mathrm{STAB}(G)$ is the projection of a face of $\mathrm{CUT}^{\square}(G')$. □

Using this theorem it is easy to show the existence of graphs with a linear number of edges that do not have K_6 as a minor and yet have a high extension complexity. In fact we get a slightly sharper result.

Theorem 14 *For every $n \geqslant 2$ there exists a graph G which is a suspension of a planar graph and for which* $\mathrm{xc}(\mathrm{CUT}^{\square}(G)) \geqslant 2^{\Omega(n^{1/4})}$.

Proof Consider a planar graph $G = (V, E)$ with n vertices for which $\mathrm{xc}(\mathrm{STAB}(G)) \geqslant 2^{\Omega(n^{1/4})}$. Corollary 5 guarantees the existence of such a graph for every n. Then the suspension over G has $n + 1$ vertices and a linear number of edges. The theorem then follows by applying Theorem 13 together with Propositions 1 and 2. □

The above theorem provides a sharp contrast for the complexity of the cut polytope for graphs in terms of their minors. As noted in the introduction, for any K_5 minor-free graph G with n vertices $\mathrm{CUT}^{\square}(G)$ has an extension of size $O(n^3)$ whereas the above result shows that there are K_6 minor free graphs whose cut polytope has superpolynomial extension complexity.

5 Concluding remarks

We have a given a simple polyhedral procedure for proving lower bounds on the extension complexity of a polytope. Using this procedure and some standard NP-completeness reductions we were able to prove lower bounds on the extension complexity of various well-known combinatorial polytopes. For the cut polytope in particular, we are able to draw a sharp line, in terms of minors, for when this complexity becomes super polynomial.

Nevertheless the procedure is not completely 'automatic' in the sense that any NP-completeness reduction of a certain type, say using gadgets, automatically gives a result on the extension complexity of related polytopes. This would seem to be a very promising line of future research.

Acknowledgments Research of the first author is supported by the NSERC and JSPS. The second author was supported by FNRS, Belgium as a postdoctoral researcher during this research.

References

1. Avis, D., Imai, H., Ito, T., Sasaki, Y.: Two-party bell inequalities derived from combinatorics via triangular elimination. J. Phys. A Math. Gen. **38**(50), 10971–10987 (2005)
2. Avis, D., Imai, H., Ito, T.: Generating facets for the cut polytope of a graph by triangular elimination. Math. Program. **112**(2), 303–325 (2008)
3. Barahona, F.: The max-cut problem on graphs not contractible to K_5. Oper. Res. Lett. **2**(3), 107–111 (1983). ISSN 0167–6377
4. Bell, J.S.: On the Einstein–Podolsky–Rosen paradox. Physics **1**(3), 195–290 (1964)
5. Corman, T.H., Leiserson, C. E., Rivest, R. L.: Introduction to Algorithms. MIT Press, Cambridge, MA (2009). ISBN 0-262-03141-8
6. de Wolf, R.: Nondeterministic quantum query and communication complexities. SICOMP. SIAM J. Comput. **32**, 681–699 (2003)
7. Deza, M.M., Laurent, M.: Geometry of Cuts and Metrics, Volume 15 of Algorithms and Combinatorics. Springer, Berlin (1997)
8. Fiorini, S., Massar, S., Pokutta, S., Tiwary, H.R., de Wolf, R.: Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In: STOC, pp. 95–106 (2012)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco, CA (1979)
10. Garey, M.R., Johnson, D.S., Stockmeyer, L.J.: Some simplified NP-complete graph problems. Theor. Comput. Sci. **1**, 237–267 (1976)
11. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press, New York (1972)
12. Pokutta, S., Vyve, M.V.: A note on the extension complexity of the knapsack polytope. Optimization Online (2013)
13. Razborov, A.A.: On the distributional complexity of disjointness. Theor. Comput. Sci. **106**(2), 385–390 (1992)
14. Yannakakis, M.: Expressing combinatorial optimization problems by linear programs. J. Comput. Syst. Sci. **43**(3), 441–466 (1991)
15. Ziegler, G.M.: Lectures on Polytopes, Volume 152 of Graduate Texts in Mathematics. Springer, Berlin (1995)

E

EXTENSION COMPLEXITY OF FORMAL LANGUAGES

The following article is unpublished and is under peer review. It is included here as an appendix for completeness.

Extension Complexity of Formal Languages[1]

Hans Raj Tiwary
hansraj@kam.mff.cuni.cz
KAM/ITI, Charles University,
Malostranské nám. 25,
118 00 Prague 1, Czech Republic

Abstract

In this article we undertake a study of extension complexity from the perspective of formal languages. We define a natural way to associate a family of polytopes with binary languages. This allows us to define the notion of extension complexity of formal languages. We prove several closure properties of languages admitting compact extended formulations. Furthermore, we give a sufficient machine characterization of compact languages. We demonstrate the utility of this machine characterization by obtaining lower bounds in streaming models and upper bounds on extension complexities of several polytopes.

Keywords: Extended formulations, formal languages

[1]This work was partially supported by grant no. P202-13/201414 of GAČR

1. Introduction

A polytope Q is said to be an extended formulation of a polytope P if P can be described as a projection of Q. Measuring the size of a polytope by the number of inequalities required to describe it, one can define the notion of *extension complexity* of a polytope P – denoted by $\mathrm{xc}(P)$ – to be the size of the smallest possible extended formulation.

Let $\boldsymbol{\varphi}$ be a boolean formula. Consider the following polytopes:

$$\begin{aligned} \mathrm{SAT} &= \mathrm{conv}\,\{\boldsymbol{x}\,|\boldsymbol{x} \text{ encodes a satisfiable boolean formula}\} \\ \mathrm{SAT}(\boldsymbol{\varphi}) &= \{\boldsymbol{x}\,|\boldsymbol{\varphi}(\boldsymbol{x}) = 1\} \end{aligned}$$

The former polytope consists of all strings that encode[2] satisfiable boolean formulae, while the latter language consists of all satisfying assignments of a given boolean formula. Which of these represents the boolean satisfiability problem *more naturally*?

Reasonable people will agree that there is no correct choice of a natural polytope for a problem. One complication is that there are various kinds of problems: decision, optimization, enumeration, etc, and very similar problems can have very different behaviour if the notion of problem changes.

Several recent results have established superpolynomial lower bounds on the extension complexity of specific polytopes. For example Fiorini et. al [1] showed that polytopes associated with MAX-CUT, TSP, and Independent Set problems do not admit polynomial sized extended formulations. Shortly afterwards Avis and the present author [2] showed that the same holds for polytopes related to many other NP-hard problems. Subsequently Rothvoß [3] showed that even the perfect matching polytope does not admit polynomial sized extended formulation. These results have been generalized in multiple directions and various lower bounds have been proved related to approximation [4, 5, 6] and semidefinite extensions [7, 8, 9].

A few fundamental questions may be raised about such results:

- How does one choose (a family of) polytopes for a specific problem?
- To what extent does this choice affect the relation between extension complexity of the chosen polytope and the complexity of the underlying problem?
- What good are extension complexity bounds anyway[3]?

The intent of this article is to say something useful (and hopefully interesting) about such problems. In particular, our main contributions are the following:

- We define formally the notion of extension complexity of binary language. Our definition is fairly natural and we do not claim any novelty here. This however is a required step towards any systematic study of problems that admit small extended formulations.
- We define formally what it means to say that a language admits small extended formulation. Again we do not claim novelty here since Rothvoß mentions similar notion in one of the first articles showing existence of polytopes with high extension complexity [10].

[2] Assume some (arbitrary but fixed) encoding of boolean formulae as binary strings.

[3] Perfect Matching remains an easy problem despite exponential lower bound on the extension complexity of the perfect matching polytope. What does an exponential lower bound for the cut polytope tell us about the difficulty of the MAX-CUT problem?

- We prove several closure properties of languages that admit compact extended formulations. Some of these results are trivial and some follows from existing results. For a small number of them we need to provide new arguments.
- We prove a sufficient condition in terms of walks on graphs and in terms of accepting Turing Machines, for a language to have polynomial extension complexity. We show how this characterization can be used to prove space lower bounds for non-deterministic streaming algorithms, and also to construct small extended formulations for various problems by means of a small "verifier algorithm". We provide some small examples to this end.

2. Background Material and Related Work

2.1. Polytopes and Extended Formulations

A polytope $P \subseteq \mathbb{R}^d$ is a closed convex set defined as intersection of a finite number of inequalities. Alternatively, it can be defined as the convex hull of a finite number of points. Any polytope that is full-dimensional has a unique representation in terms of the smallest number of defining inequalities or points. The *size* of a polytope is defined to be the smallest number of inequalities required to define it. For the purposes of this article all polytopes will be assumed to be full-dimensional. While in doing so, no generality is lost for our discussion, we will refrain from discussing such finer points. We refer the reader to [11] for background on Polytopes.

A polytope Q is called an *Extended Formulation* or simply *EF* of a polytope P, if P can be obtained as a projection of Q. The *extension complexity* of a polytope, denoted by $\mathrm{xc}(P)$, is defined to be the smallest size of any possible EF of P.

Extended formulations have a long history of study. Here we refer to only a handful of work that are closely related to this article. For more complete picture related to extended formulations, we refer the reader to the excellents surveys by Conforti et al. [12] and by Kaibel [13] as a point to start.

We will use the following known results related to extended formulations.

Theorem 1 (Balas)**.** *Let P_1 and P_2 be polytopes and let $P = \mathrm{conv}(P_1 \uplus P_2)$, where $\uplus$ denotes the convex hull of the union. Then $\mathrm{xc}(P) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2) + 2$.*

2.2. Online Turing machines

In this article we would be interested in *online* variants of Turing machines. Informally speaking, these machines have access to two tapes: an input tape where the head can only move from left to right (or stay put where it is) and a work tape where the work head can move freely. When the machine halts, the final state determines whether the input has been accepted or not. Such machines - like usual Turing machines - can be either deterministic or non-deterministic. For a non-deterministic machine accepting a binary language $\boldsymbol{L}$ we require that if $\boldsymbol{x} \notin \boldsymbol{L}$ then the machine rejects $\boldsymbol{x}$ for all possible nondeterministic choices, and if $\boldsymbol{x} \in \boldsymbol{L}$ then there be some set of non-deterministic choices that make the machine accept $\boldsymbol{L}$.

The working of an online Turing machine can be thought of as the working of an online algorithm that makes a single pass over the input and decides whether to accept or reject the input. Natural extensions allow the machine to make more than one pass over the input.

Definition 1. The complexity class k-NSPACE($s(n)$) is the class of languages accepted by a k-pass non-deterministic Turing machines using space $s(n)$. Similarly, the complexity class k-DSPACE($s(n)$) is the class of languages accepted by a k-pass deterministic Turing machine using space $s(n)$.

The classes 1L and 1NL were introduced by Hartmanis, Mahaney, and Immerman [14, 15] to study weaker forms of reduction. In our terminology the class 1L would be 1-DSPACE($\log n$) while the class 1NL would be 1-NSPACE($\log n$). The motivation for defining these classes was that if we do not know whether P is different from NP or not, then using a polynomial reduction may not be completely justified in saying that a problem is as hard or harder than another problem, and weaker reductions are probably more meaningful. In any case, these languages have a rich history of study. It is known that non-determinism makes one-pass machines strictly more powerful for $s(n) = \Omega(\log n)$ [16].

2.3. Glued Product of Polytopes

Let $P_1 \subseteq \mathbb{R}^{d_1+k}$ and $P_2 \subseteq \mathbb{R}^{d_2+k}$ be two 0/1 polytopes with vertices $\mathrm{vert}(P_1), \mathrm{vert}(P_2)$ respectively. The *glued product* of P_1 and P_2 where the glueing is done over the last k coordinates is defined to be:

$$P_1 \times_k P_2 := \mathrm{conv}\left\{ \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix} \in \{0,1\}^{d_1+d_2+k} \,\middle|\, \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{pmatrix} \in \mathrm{vert}(P_1), \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix} \in \mathrm{vert}(P_2) \right\}.$$

It is known [17, 18] that if every vertex of P_1 and P_2 contains at most one 1 in the last k coordinates then $\mathrm{xc}(P_1 \times_k P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2)$. We will use this to show that languages in the class 1NL admit polynomial extended formulations. In particular we will use the following Lemma.

Lemma 1. *[17] Let $P_1 \subseteq \mathbb{R}^{d_1+k}$ and $P_2 \subseteq \mathbb{R}^{d_2+k}$ be two 0/1 polytopes such that the every vertex of P_1 and P_2 contains at most one nonzero coordinate entry among the k-coordinates used for the glueing. Then,*

$$\mathrm{xc}(P_1 \times_k P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2).$$

3. Polytopes for Formal Languages

Let $\boldsymbol{L} \subseteq \{0,1\}^*$ be a language over the 0/1 alphabet. For every natural number n define the set $\boldsymbol{L}(n) := \{\boldsymbol{x} \in \{0,1\}^n \mid \boldsymbol{x} \in \boldsymbol{L}\}$. Viewing each string $\boldsymbol{x} \in \boldsymbol{L}(n)$ as a column vector, and ordering the strings lexicographically, we can view the set $\boldsymbol{L}(n)$ as a matrix of size $n \times |\boldsymbol{L}(n)|$. Thus we are in a position to naturally associate a family of polytopes with a given language and the extension complexity of these polytopes can serve as a natural measure of how hard is it to model these languages as Linear Programs.

That is, one can associate with $\boldsymbol{L}$, the family of polytopes $\mathcal{P}(\boldsymbol{L}) = \{P(\boldsymbol{L}(1)), P(\boldsymbol{L}(2)), \ldots\}$ with $P(\boldsymbol{L}(n)) := \mathrm{conv}\{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{L}\}$. The extension complexity $\mathrm{xc}(\mathcal{P}(\boldsymbol{L}))$ is then an intrinsic measure of complexity of the language $\boldsymbol{L}$.

Extension complexity of Languages

Definition 2. The *extension complexity* of a language $\boldsymbol{L}$ – denoted by $\text{xc}(\boldsymbol{L})$ – is defined by $\text{xc}(\boldsymbol{L}) := \text{xc}(\mathcal{P}(\boldsymbol{L}))$.

We say that the *extension complexity* of $\boldsymbol{L}$, denoted by $\text{xc}(\boldsymbol{L})$ is $\boldsymbol{f}(n)$, where $\boldsymbol{f} : \mathbb{N} \to \mathbb{R}_+$ is a non-negative non-decreasing function on natural numbers, if for every polytope $P(\boldsymbol{L}(n)) \in \mathcal{P}(\boldsymbol{L})$ we have that $\text{xc}(P(\boldsymbol{L}(n))) = \boldsymbol{f}(n)$. One can immediately see that this definition is rather useless in its present form since for different values of n, the corresponding polytopes in $\mathcal{P}(\boldsymbol{L})$ may have extension complexities that are not well described by a simple function. For example, the perfect matching polytope would have no strings of length n if n is not of the form $\binom{r}{2}$ for some even positive integer r. To avoid such trivially pathological problems, we will use asymptotic notation to describe the membership extension complexity of languages.

We will say that $\text{xc}(\boldsymbol{L}) = \mathcal{O}(\boldsymbol{f})$ to mean that there exists a constant $c > 0$ and a natural number n_0 such that for every polytope $P(\boldsymbol{L}(n)) \in \mathcal{P}(\boldsymbol{L})$ with $n \geqslant n_0$ we have $\text{xc}(P(\boldsymbol{L}(n))) \leqslant c\boldsymbol{f}(n)$.

We will say that $\text{xc}(\boldsymbol{L}) = \Omega(\boldsymbol{f})$ to mean that there exists a constant $c > 0$ and such that for every natural number n_0 there exists an $n \geqslant n_0$ such that $\text{xc}(P(\boldsymbol{L}(n))) \geqslant c\boldsymbol{f}(n)$. Note the slight difference from the usual Ω notation used in asymptotic analysis of algorithms [4]. The intent here is to be able to say that a polytope family of a certain language contains an infinite family of polytopes that have high extension complexity.

Finally, we will say that $\text{xc}(\boldsymbol{L}) = \Theta(\boldsymbol{f})$ to mean that $\text{xc}(\boldsymbol{L}) = \mathcal{O}(\boldsymbol{f})$ as well as $\text{xc}(\boldsymbol{L}) = \Omega(\boldsymbol{f})$. To give an example of the notation, recent result of Rothvoß[3] proving that perfect matching polytope has high extension complexity would translate in our setting to the following statement:

Theorem. *[3] Let $\boldsymbol{L}$ be the language consisting of the characteristic vectors of perfect matchings of complete graphs. Then, there exists a constant $c > 1$ such that $\text{xc}(\boldsymbol{L}) = \Omega(c^n)$.*

One can extend the above notation to provide more information by being able to use functions described by asysmptotic notation as well. For example, knowing that the perfect matching polytope for K_n has extension complexity at most $2^{\frac{n}{2}}$ [19] together with Rothvoß' result one could say that the language of all perfect matchings of complete graphs has extension complexity $2^{\Theta(n)}$.

Proposition 1. *For every language $\boldsymbol{L} \subseteq \{0,1\}^*$ we have $\text{xc}(\boldsymbol{L}) \leqslant \text{xc}(\boldsymbol{L}) + \text{xc}(\overline{\boldsymbol{L}}) \leqslant 2^n$.*

Proof. The first inequality is trivial. For the last inequality, observe that $\boldsymbol{L}(n)$ and $\overline{\boldsymbol{L}}(n)$ has at most 2^n vertices altogether. □

4. Languages with small extension complexities

Now we are ready to define the class of languages that we are interested in: namely, the languages that have small extension complexities.

Definition 3. $\mathcal{CF}$ is the class of languages admitting $\mathcal{C}$ompact extended $\mathcal{F}$ormulations and is defined as

$$\mathcal{CF} \quad = \quad \{\boldsymbol{L} \subseteq \{0,1\}^* \mid \exists c > 0 \text{ s.t. } \text{xc}(\boldsymbol{L}) \leqslant n^c\}$$

[4] This usage, however, is common among number theorists.

4.1. Some canonical examples

For any given boolean formula $\boldsymbol{\varphi}$ with n variables define the polytope $\mathrm{SAT}(\boldsymbol{\varphi})$ to be the convex hull of all satisfying assignments and $\mathrm{UNSAT}(\boldsymbol{\varphi})$ to be the convex hull of all non-satisfying assigments. That is,

$$\begin{aligned}\mathrm{SAT}(\boldsymbol{\varphi}) &:= \mathrm{conv}(\{\boldsymbol{x} \in \{0,1\}^n \mid \boldsymbol{\varphi}(\boldsymbol{x}) = 1\}),\\ \mathrm{UNSAT}(\boldsymbol{\varphi}) &:= \mathrm{conv}(\{\boldsymbol{x} \in \{0,1\}^n \mid \boldsymbol{\varphi}(\boldsymbol{x}) = 0\})\end{aligned}$$

Let $n \in \mathbb{N}$ and $m = n^2$. For the complete graph K_n define a 3SAT boolean formula $\boldsymbol{\varphi}_m$ such that $\mathrm{CUT}^{\square}(K_n)$ is a projection of $\mathrm{SAT}(\boldsymbol{\varphi}_m)$. Consider the relation $\boldsymbol{x}_{ij} = \boldsymbol{x}_{ii} \oplus \boldsymbol{x}_{jj}$, where $\oplus$ is the xor operator. The boolean formula

$$(\boldsymbol{x}_{ii} \vee \overline{\boldsymbol{x}}_{jj} \vee \boldsymbol{x}_{ij}) \wedge (\overline{\boldsymbol{x}}_{ii} \vee \boldsymbol{x}_{jj} \vee \boldsymbol{x}_{ij}) \wedge (\boldsymbol{x}_{ii} \vee \boldsymbol{x}_{jj} \vee \overline{\boldsymbol{x}}_{ij}) \wedge (\overline{\boldsymbol{x}}_{ii} \vee \overline{\boldsymbol{x}}_{jj} \vee \overline{\boldsymbol{x}}_{ij})$$

is true if and only if $\boldsymbol{x}_{ij} = \boldsymbol{x}_{ii} \oplus \boldsymbol{x}_{jj}$ for any assignment of the variables $\boldsymbol{x}_{ii}, \boldsymbol{x}_{jj}$ and $\boldsymbol{x}_{ij}$.

Therefore we define $\boldsymbol{\varphi}_m$ (with $m = n^2$) as

$$\boldsymbol{\varphi}_m := \bigwedge_{\substack{i,j \in [n] \\ i \neq j}} \left[\begin{array}{l} (\boldsymbol{x}_{ii} \vee \overline{\boldsymbol{x}}_{jj} \vee \boldsymbol{x}_{ij}) \wedge (\overline{\boldsymbol{x}}_{ii} \vee \boldsymbol{x}_{jj} \vee \boldsymbol{x}_{ij}) \wedge \\ (\boldsymbol{x}_{ii} \vee \boldsymbol{x}_{jj} \vee \overline{\boldsymbol{x}}_{ij}) \wedge (\overline{\boldsymbol{x}}_{ii} \vee \overline{\boldsymbol{x}}_{jj} \vee \overline{\boldsymbol{x}}_{ij}) \end{array} \right]. \tag{1}$$

We will call the family of CNF formulae defined by 1 to be the CUTSAT family. It is easy to see the following.

Proposition 2. $\mathrm{xc}(\mathrm{SAT}(\boldsymbol{\varphi}_m)) = 2^{\Omega(n)}$, *where* $m = n^2$.

Proof. The satisfying assignments of $\boldsymbol{\varphi}_m$ when restricted to the variables $\boldsymbol{x}_{ij}$ with $i \neq j$ are exactly the cut vectors of K_n and every cut vector of K_n can be extended to a satisfying assignment of $\boldsymbol{\varphi}$. □

Proposition 3. $\mathrm{xc}(\mathrm{UNSAT}(\boldsymbol{\varphi}_m)) \leqslant \mathcal{O}(n^4)$.

Proof. Let $\boldsymbol{\varphi}$ be a DNF formula with n variables and m clauses. We can show that $\mathrm{xc}(\mathrm{SAT}(\boldsymbol{\varphi})) \leqslant \mathcal{O}(mn)$.

If $\boldsymbol{\varphi}$ consists of a single clause then it is just a conjunction of some literals. In this case $\mathrm{SAT}(\boldsymbol{\varphi})$ is a face of the n-hypercube and has $\mathrm{xc}(\mathrm{SAT}(\boldsymbol{\varphi})) \leqslant 2n$. Furthermore, for DNF formulae $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$ we have that $\mathrm{SAT}(\boldsymbol{\varphi}_1 \vee \boldsymbol{\varphi}_2) = \mathrm{SAT}(\boldsymbol{\varphi}_1) \uplus \mathrm{SAT}(\boldsymbol{\varphi}_2))$. Therefore, using Theorem 1 repeatedly we obtain that for a DNF formula $\boldsymbol{\varphi}$ with n variables and m clauses $\mathrm{SAT}(\boldsymbol{\varphi}) \leqslant \mathcal{O}(mn)$. □

5. Closure properties of compact languages

Now we discuss the closure properties of the class $\mathcal{CF}$ with respect to some common operations on formal languages. The operations that we consider are as follows.

- **Complement** : $\overline{\boldsymbol{L}} = \{\boldsymbol{x} \mid \boldsymbol{x} \notin \boldsymbol{L}\}$
- **Union** : $\boldsymbol{L}_1 \cup \boldsymbol{L}_2 = \{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{L}_1 \vee \boldsymbol{x} \in \boldsymbol{L}_2\}$
- **Intersection** : $\boldsymbol{L}_1 \cap \boldsymbol{L}_2 = \{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{L}_1 \wedge \boldsymbol{x} \in \boldsymbol{L}_2\}$
- **Set difference** : $\boldsymbol{L}_1 \setminus \boldsymbol{L}_2 = \{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{L}_1 \wedge \boldsymbol{x} \notin \boldsymbol{L}_2\}$
- **Concatenation** : $\boldsymbol{L}_1\boldsymbol{L}_2 = \{\boldsymbol{x}\boldsymbol{y} \mid \boldsymbol{x} \in \boldsymbol{L}_1 \wedge \boldsymbol{y} \in \boldsymbol{L}_2\}$
- **Kleene star** : $\boldsymbol{L}^* = \boldsymbol{L} \cup \boldsymbol{L}\boldsymbol{L} \cup \boldsymbol{L}\boldsymbol{L}\boldsymbol{L} \cup \boldsymbol{L}\boldsymbol{L}\boldsymbol{L}\boldsymbol{L} \cup \ldots$

Proposition 4. *$\mathcal{CF}$ is not closed under taking complement.*

Proof. Let $\mathbf{\Phi}$ be the family of 3CNF formula containing CUTSAT formula for $m = n^2$ and containing some tautology with m variables for all other m. Let $\boldsymbol{L}_{\text{sat}}$ be the language containing satisfying assignments of formulae in this family. Similarly, let $\boldsymbol{L}_{\text{unsat}}$ be the language containing non-satisfying assignments of formulae in this family.

It is easy to see that $\boldsymbol{L}_{\text{sat}} = \overline{\boldsymbol{L}}_{\text{unsat}}$. Now, $\boldsymbol{L}_{\text{unsat}} \in \mathcal{CF}$ due to Proposition 3 while $\boldsymbol{L}_{\text{sat}} \notin \mathcal{CF}$ due to Proposition 2. □

Proposition 5. *$\mathcal{CF}$ is closed under taking union.*

Proof. Let $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ be two languages. Then, $\text{xc}(\boldsymbol{L}_1 \cup \boldsymbol{L}_2) \leqslant \text{xc}(\boldsymbol{L}_1) + \text{xc}(\boldsymbol{L}_2) + 2$ (cf. Theorem 1). □

Proposition 6. *$\mathcal{CF}$ is not closed under taking intersection.*

Proof. Let $\boldsymbol{L}_1$ be a language such that a string $\boldsymbol{x} \in \boldsymbol{L}_1$ if and only if it satisfies the following properties.

- $|\boldsymbol{x}| = (n+1)\binom{n}{2}$ for some natural number n, and
- $\boldsymbol{x}_{ij(n+1)} = \boldsymbol{x}_{iji} \oplus \boldsymbol{x}_{ijj}$ if the characters are indexed as $\boldsymbol{x}_{ijk}$ with $1 \leqslant i < j \leqslant n$, $1 \leqslant k \leqslant n+1$.

We claim that $\text{xc}(\boldsymbol{L}_1) = \mathcal{O}(n^3)$. Indeed $P\left(\boldsymbol{L}_1\left((n+1)\cdot\binom{n}{2}\right)\right)$ is the product of polytopes

$$P_{ij} = \left\{\boldsymbol{x} \in \{0,1\}^{n+1} \mid \boldsymbol{x}_{n+1} = \boldsymbol{x}_i \oplus \boldsymbol{x}_j\right\}$$

for $1 \leqslant i < j \leqslant n$ and $\text{xc}(P_{ij}) = \mathcal{O}(n)$.

Now let $\boldsymbol{L}_2$ be a language such that a string $\boldsymbol{x} \in \boldsymbol{L}_2$ if and only if it satisfies the following properties.

- $|\boldsymbol{x}| = (n+1)\binom{n}{2}$ for some natural number n, and
- $\boldsymbol{x}_{i_1 j_1 k} = \boldsymbol{x}_{i_2 j_2 k}$ for all $k \in [n], i \neq j \in [n]$

Each polytope $P\left(\boldsymbol{L}_1\left((n+1)\cdot\binom{n}{2}\right)\right)$ is just an embedding of $\square_{n+\binom{n}{2}}$ in $\mathbb{R}^{(n+1)\binom{n}{2}}$ where $\square_k$ is the k-dimensional hypercube. Therefore, $\text{xc}(\boldsymbol{L}_2) = \mathcal{O}(n^2)$.

Finally, observe that for $m = (n+1)\binom{n}{2}$ the polytope $P((\boldsymbol{L}_1 \cap \boldsymbol{L}_2)(m))$ when projected to the coordinates labelled $\boldsymbol{x}_{ij(n+1)}$ is just the polytope $\text{CUT}_n^{\square}$ (cf. Proposition 2). Therefore, $\text{xc}(\boldsymbol{L}_1 \cap \boldsymbol{L}_2) = 2^{\Omega(n)}$ and even though $\boldsymbol{L}_1, \boldsymbol{L}_2 \in \mathcal{CF}$, the intersection $\boldsymbol{L}_1 \cap \boldsymbol{L}_2 \notin \mathcal{CF}$. □

Proposition 7. *$\mathcal{CF}$ is not closed under taking set difference.*

Proof. The complete language $\{0,1\}^*$ clearly belongs to $\mathcal{CF}$. For any language $\boldsymbol{L}$ we have $\overline{\boldsymbol{L}} = \{0,1\}^* \setminus \boldsymbol{L}$. If $\mathcal{CF}$ were closed under taking set-difference, it would also be closed under taking complements. But as pointed out in Proposition 4, it is not. □

Proposition 8. *$\mathcal{CF}$ is closed under concatenation.*

Proof. $P(\boldsymbol{L}_1\boldsymbol{L}_2(n))$ is the union of the polytopes $P(\boldsymbol{L}_1(i)) \times P(\boldsymbol{L}_2(n-i))$ for $i \in [n]$. Therefore, we have that $\text{xc}(\boldsymbol{L}_1\boldsymbol{L}_2) \leqslant \mathcal{O}(n(\text{xc}(\boldsymbol{L}_1) + \text{xc}(\boldsymbol{L}_2)))$. □

Proposition 9. *$\mathcal{CF}$ is closed under taking Kleene star.*

Proof. Let $\boldsymbol{L} \in \mathcal{CF}$. For $0 \leqslant k \leqslant n$, consider the polytope P_k defined as

$$P_k := \operatorname{conv}\left(\left\{ \begin{pmatrix} \boldsymbol{e}_{i+1}^{n+1} \\ \boldsymbol{0}^{i} \\ \boldsymbol{x} \\ \boldsymbol{0}^{n-i-k} \\ \boldsymbol{e}_{i+|\boldsymbol{x}|+1}^{n+1} \end{pmatrix} \in \{0,1\}^{3n+2} \;\middle|\; \begin{array}{ll} & \boldsymbol{x} \in \boldsymbol{L} \\ \wedge & |\boldsymbol{x}| = k \\ \wedge & 0 \leqslant i \leqslant n-k \end{array} \right\}\right)$$

Define $P := \cup_{j=0}^{n} P_j$. Then, $\operatorname{xc}(P) \leqslant \sum_{k=0}^{n} \operatorname{xc}(P_k) \leqslant \sum_{k=0}^{n} (n \operatorname{xc}(P(\boldsymbol{L}(k)))) \leqslant \mathcal{O}(n^2 \operatorname{xc}(\boldsymbol{L}))$.

Let S_0 be the face of P defined by the first n coordinates being 0 and the $(n{+}1)$-th coordinate being 1. Construct S_{i+1} by taking the glued product of S_i with P over the last $n{+}1$ coordinates of S_i and the first $n{+}1$ coordinates of Q.

Take the face R of S_n defined by the last n coordinates being 0 and the $(n{+}1)$-th penultimate coordinate being 1. Then, R is an EF for $P(\boldsymbol{L}^*(n))$. Moreover, $\operatorname{xc}(R) \leq \operatorname{xc}(S_n) \leqslant (n{+}1) \operatorname{xc}(P) \leqslant \mathcal{O}(n^3 \operatorname{xc}(\boldsymbol{L}))$.

Therefore, $\operatorname{xc}(\boldsymbol{L}^*) = \mathcal{O}(n^3 \operatorname{xc}(\boldsymbol{L}))$ and $\boldsymbol{L}^* \in \mathcal{CF}$. □

6. Computational power of compact languages

We would like to start the discussion in this section by pointing out that in the class of compact languages is in some sense too powerful. This power comes just from non-uniformity in the definition.

Proposition 10. *$\mathcal{CF}$ contains undecidable languages.*

It is easy to construct undecidable languages that are in $\mathcal{CF}$. Take any uncomputable function $\boldsymbol{f} : \mathbb{N} \to \{0,1\}$ and define the language $\boldsymbol{L}$ containing all strings of length n if $\boldsymbol{f}(n) = 1$ and no strings of length n if $\boldsymbol{f}(n) = 0$. The extension complexity of $\boldsymbol{L}$ is $\Theta(2n)$.

At the moment we do not want to start a discussion about controlling the beast that non-uniformity unleashes. Rather we will focus on something more positive. We will show that if a language is accepted by a non-deterministic LOGSPACE online Turing machine, then its extension complexity is polynomial. This brings into fold many non-regular languages already. And as we will see, this characterization allows us to give simple proofs for polynomial extension complexity for some polytopes.

Before we proceed, we would also like to point out that, in the following discussion, the assumption on the input tape being accessed in a one-way fashion is not something one can remove easily. There are languages in LOGSPACE and AC^0 that have exponential extension complexity: for example, the string of all perfect matchings of K_n.

6.1. Polytopes of walks in graphs

Definition 4. Let $D = (V, A)$ be a directed graph with every edge labeled either zero or one. Consider two nodes $u, v \in V$ and a walk ω of length n from u to v. The *signature* of ω – denoted by σ_ω – is the sequence of edge labels along the walk ω. The node u is called the *source* of the walk and the node v the *destination*.

Definition 5. Consider the convex hull of all zero-one vectors of the form (u, σ, v) where u and v are indices of two nodes in D and σ is the signature of some walk of length n from u to v. This polytope – denoted by $\mathrm{P}_{\mathrm{markov}}(D, n)$ – is called the *Markovian polytope of* D.

Proposition 11. *Let $D = (V, A)$ be directed graph (possibly with self-loops and multiple edges) with every edge labeled either zero or one. Then, $\mathrm{P}_{\mathrm{markov}}(D, n)$ has extension complexity at most $2|V| + |A| \cdot n$.*

Proof. Let us encode every vertex of D with a zero-one vector of length V such that the unit vector e_i represents vertex i.

Define polytope $P_{\mathrm{trans}} \subset \{0,1\}^{|V|+1+|V|}$ with $(a, z, b) \in \{0,1\}^{|V|+1+|V|}$ a vertex of P_{trans} if and only if it encodes a possible transition in D. That is, a and b encode vertices of V, and the coordinate z represents the label of the edge following which one can move from a to b. Since P_{trans} has at most $|E|$ vertices $\mathrm{xc}(P_{\mathrm{trans}}) \leqslant |E|$.

Let P_0 be the convex hull of $(i, \boldsymbol{e}_i)$ for $i \in V$ and P_f be the convex hull of $(\boldsymbol{e}_i, i)$ for $i \in V$. Observe that the two polytopes are the same except for relabeling of coordinates. Also, $\mathrm{xc}(P_0) = \mathrm{xc}(P_f) \leqslant |V|$.

Let $P_1 = P_{\mathrm{trans}}$. For $2 \leqslant i \leqslant n$, construct the polytope P_i by glueing the last $|V|$ coordinates of P_{i-1} with the first $|V|$ coordinates of P_{trans}. By Lemma 1 we have that $\mathrm{xc}(P_n) \leqslant |E| \cdot n$.

Finally, let P be the polytope obtained by glueing last $|V|$ coordinates of P_0 with the first $|V|$ coordinates of P_n, and then glueing the last $|V|$ vertices of the result with the first $|V|$ coordinates of P_f. Note that $\mathrm{xc}(P) \leqslant 2|V| + |E| \cdot n$.

To complete the proof, notice that P is an extended formulation for $\mathrm{P}_{\mathrm{markov}}(D, n)$. In particular, projecting out every coordinate except the ones corresponding to the source node in P_0, the ones corresponding to the destination node in P_f, and ones that correspond to the z coordinates in all the copies of P_{trans} produces exactly the vertices of $\mathrm{P}_{\mathrm{markov}}(D, n)$. The z-coordinate corresponding to the i-th copy of P_{trans} corresponds to the i-th index of signatures in the vectors in $\mathrm{P}_{\mathrm{markov}}(D, n)$. □

6.2. Polytopes for Online Turing Machines

Proposition 12. *Let $\boldsymbol{L} \in k$-NSPACE$(s(n))$. Then, $\boldsymbol{L} \in 1$-NSPACE$(ks(n))$.*

Proof. Let M_n be the Turing machine that accepts strings of length n. We will simulate M_n using a multi-tape single pass nondeterministic Turing machine called the simulator S. S is supplied with $p(n)$ work tapes. S starts by guessing the initial work state of M_n at the start of i-th pass and writing them on the i-th work tape. S then simulates (using extra space on each work tape) each of the passes independently starting from their respective initial configuration. Once the entire input has been scanned, the simulator verifies that the work space of M_n on the i-th tape at the end of the pass matches the guess for the initial content for the $(i+1)$-th tape. S will accept only if the last tape is in an accepting state.

To store the content of work tape and the current state, S needs $s(n) + o(s(n))$ space for each pass. Thus S uses a single pass and total space of $p(n)s(n)(1 + o(1))$. By Proposition 14 the extension complexity of the strings accepted by M_n is then $2^{\mathcal{O}(p(n)s(n))}n$. □

Thus for our purposes it suffices to restrict our attention to single pass TMs.

Definition 6. The *configuration graph* for input of length n for a given one-pass Turing machine (deterministic or non-determinisitic) is constructed as follows. For each fixed n, consider

the directed graph whose nodes are marked with a label consisting of $s(n) + \lceil \log(s(n)) \rceil$ characters. The labels encode the complete configuration of the Turing machine: the content of the worktape and head position on the worktape. We make directed edges between two nodes u and v if the machine can reach from configuration u to configuration v by a sequence of transitions with exactly one input bit read in between. The directed edge is labeled by the input bit read during this sequence of transition.

Finally, we add two special nodes: a start node with a directed edge to each possible starting configuration of the machine, and a finish node with a directed edge from each possible accepting configuration. Each of these directed edges are labeled by zero.

Proposition 13. *The configuration graph for input of length n for a one-pass Turing machine has $\mathcal{O}(2^{s(n)}s(n))$ nodes. If the Turing machine is non-deterministic, this graph has $\mathcal{O}(4^{s(n)}(s(n))^2)$ edges. If the Turing machine is deterministic then this graph has $O(2^{s(n)}s(n))$ edges.*

Proof. The bound for number of nodes is clear from the construction of the configuration graph. We can have at most two transition edges between any two (possibly non-distinct) nodes: one corresponding to reading a zero on the input tape, and one corresponding to reading a one. Therefore, asymptotically the configuration graph can have number of edges that is at most square of the number of nodes.

For deterministic Turing machine, each node in the configuration graph has exactly two outgoing edges (possibly to the same node). Therefore the number of edges is asymptotically the same as the number of vertices. □

Now Proposition 11 can be used to bound the extension complexity of language accepted by one-pass machines.

Proposition 14. *Let $\boldsymbol{L} \in$ 1-NSPACE$(s(n))$. Then,* $\mathrm{xc}(\boldsymbol{L}) = \mathcal{O}(4^{s(n)}(s(n))^2 \cdot n)$.

Proof. Let $\boldsymbol{L} \in$ 1-NSPACE$(s(n))$ be a language. That is, there exists a Turing machine that when supplied with a string on the one-way input tape uses at most $s(n)$ cells on the worktape, makes a single pass over the input and then accepts or rejects the input. If the input string is in $\boldsymbol{L}$, some sequence of non-deterministic choices lead the machine to an accepting state, otherwise the machine always rejects.

The length-n strings that are accepted by such a Turing machine correspond exactly to the signatures of length $n+2$ walks on the corresponding configuration graph D. The first and the last character of these strings is always zero. Therefore, an extended formulation for $P(\boldsymbol{L}(n))$ is obtained by taking the face of $\mathrm{P}_{\mathrm{markov}}(D, n+2)$ corresponding to walks that start and the start node and finish at the finish node. By Proposition 11 $\mathrm{P}_{\mathrm{markov}}(D, n+2)$ has extension complexity $\mathcal{O}(4^{s(n)}(s(n))^2 \cdot n$, and so does the desired face. □

If $\boldsymbol{L}$ is accepted by a one-pass deterministic TM then one can do better because the configuration graph has fewer edges.

Proposition 15. *Let $\boldsymbol{L} \in$ 1-DSPACE$(s(n))$. Then,* $\mathrm{xc}(\boldsymbol{L}) = \mathcal{O}(2^{s(n)}s(n) \cdot n)$.

6.3. Extensions for multiple-pass machines

Proposition 16. *Let $\boldsymbol{L} \in p$-NSPACE$(s(n))$. Then,* $\mathrm{xc}(\boldsymbol{L}) = 2^{\mathcal{O}(p(n)s(n))}n$.

Proof. This follows immediately from Propositions 12 and 14. □

Proposition 17. *Let $\mathcal{M}$ be a (not necessarily uniform) family of deterministic online Turing machines. Let the number of passes and the space used by the family be bounded by functions, $p(n), s(n)$ respectively. Let $\boldsymbol{L}(M)$ be the language accepted by M. Then,* $\mathrm{xc}(L(\mathcal{M})) \leqslant 2^{\mathcal{O}(p(n)s(n))}n$.

Proposition 18. *If $\boldsymbol{L}$ is accepted by a fixed-pass non-deterministic* LOGSPACE *Turing machine then $\boldsymbol{L} \in \mathcal{CF}$.*

We end this section with the following remark. For a language to be compact (that is, to have polynomial extension complexity), it is sufficient to be accepted by an online Turing machine (deterministic or not) that requires only logarithmic space. However, this requirement is clearly not necessary. This can be proved by contradiction: Suppose that the condition is necessary. Then the class of compact languages must be closed under taking intersection. (Simply chain the two accepting machines and accept only if both do). Since we have already established (cf. Proposition 6) that the class of compact languages is not closed under taking intersection, we have a contradiction.

7. Applications

7.1. Streaming lower bounds

Reading Proposition 16 in converse immediately yields lower bounds in the streaming model of computation. We illustrate this by an example.

Example 1. We know that the perfect matching polytope of the complete graph K_n has extension complexity $2^{\Omega(n)}$. Any $p(n)$-pass algorithm requiring space $s(n)$, that correctly determines whether a given stream of $\binom{n}{2}$ is the characteristic vector of a perfect matching in K_n, must have $p(n)s(n) = \Omega(n)$. This bound applies even to non-deterministic algorithms.

In fact Proposition 11 provides an even stronger lower bound.

Definition 7. Let $\boldsymbol{L} \subseteq \{0,1\}^n$ be a language. $\boldsymbol{L}$ is said to be online μ-magic if there exists a Turing machine T that accepts $\boldsymbol{L}$ with the following oracle access. On an input of length n on the one-way input tape, the machine T scans the input only once. At any time (possibly multiple times) during the scanning of the input, T may prepare its working tape to describe any well-formed function $\boldsymbol{f} : \{0,1\}^{\mu(n)} \to \{0,1\}^{\mu(n)}$ and a particular input $\boldsymbol{x}$ and invoke the oracle that changes the contents of the work-tape to $\boldsymbol{f}(\boldsymbol{x})$. The machine must always reject strings not in $\boldsymbol{L}$. For strings in $\boldsymbol{L}$ there must be some possible execution resulting in accept.

Notice that the working of even such a machine can be encoded in terms of the configuration graph where the transitions may depend arbitrarily but in a well-formed way on the contents of the work-tape.

Proposition 19. *If the set of characteristic vectors of perfect matchings in K_n are accepted by an online μ-magic Turing machine, then $\mu(n) = \Omega(n)$.*

Thus we see that extension complexity lower bounds highlight deep limitations of the streaming model: even powerful oracles do not help solve in sublinear space problems that are LOGSPACE solvable if the one-way restriction on the input is removed.

7.2. Upper bounds from online algorithms

Parity Polytope

As an example, consider the language containing strings where the last bit indicates the parity of the previous bits. This language can be accepted by a deterministic LOGSPACE turing machine requiring a single pass over the input and a single bit of space. Therefore, the parity polytope has extension complexity $\mathcal{O}(n)$.

The parity polytope is known to have extension complexity at most $4n - 4$ [20].

Integer Partition Polytope

For non-negative integer n the Integer Partition Polytope, IPP_n, is defined as

$$\text{IPP}_n := \text{conv}\{x \in \mathbb{Z}_+^n | \sum_{k=1}^{n} kx_k = n\}.$$

It is known that $\text{xc}(\text{IPP}_n) = \mathcal{O}(n^3)$ [21].

Consider the polytope in $\mathbb{R}^{\lceil \log n \rceil \times n}$ that encodes each x_i as a binary string. For example, for $n = 4$ the vector $(2, 1, 0, 0)$ is encoded as $(1, 0, 0, 1, 0, 0, 0, 0)$. This polytope is clearly an extended formulation of the Integer Partition Polytope. Call this polytope BIPP_n. The following single pass determinisitic algorithm accepts a string $(x_1, x_2, \ldots, x_n) \in \{0, 1\}^{\lceil \log n \rceil \times n}$ if and only if the string represents a vertex of BIPP_n.

```
Data: Binary string of length n⌈log n⌉
Result: Accept if the input encodes a vertex of the BIPP_n
s = 0; i = 0; l = 0;
while i < n do
    b =read_next_bit;
    if (s + (i + 1)2^l b) > n then
        reject;
    else
        s = (s + (i + 1)2^l b);
        l = (l + 1)%⌈log n⌉;
        if l == 0 then
            i++;
        end
    end
end
if s == n then
    accept;
else
    reject;
end
```

Algorithm 1: One pass algorithm for accepting vertices of BIPP_n.

The above algorithm together with Proposition 15 shows that $\text{xc}(\text{IPP}_n) \leqslant \text{xc}(\text{BIPP}_n) \leqslant \mathcal{O}(n^3 \log^2 n)$.

Knapsack Polytopes

For a given sequence of (non-negative) integers $(a, b) = (a_1, a_2, \ldots, a_n, b)$, the Knapsack polytope $KS(a, b)$ is defined as

$$KS(a, b) := \left\{ x \in \{0, 1\}^n \,\middle|\, \sum_{i=1}^{n} a_i x_i \leqslant b \right\}.$$

The Knapsack polytope is known to have extension complexity super-polynomial in n. However, optimizing over $KS(a, b)$ can be done via dynamic programming in time $O(nW)$ where W is the largest number among $a_1, \ldots, a_n, b$.

Suppose the integers a_i, b are arriving in a stream with a bit in between indicating whether $x_i = 0$ or $x_i = 1$. With a space of W bits, an online Turing machine can store and update

$\sum_{i=1}^{n} a_i x_i$. At the end, it can subtract b and accept or reject depending on whether the result is 0 or not. Any overflow during intermediate steps can be used to safely reject the input. Therefore, the extension complexity of the Knapsack polytope is $O(nW \log W)$. Note however the extension obtained this way is actually an extended formulation of a polytope encoding all the instances together with their solutions.

*Languages in co-***DLIN**

Let $\boldsymbol{L}$ be a language such that $\overline{\boldsymbol{L}}$ is generated by a determinisitic linear grammar [22]. The following was proved by Babu, Limaye, and Varma [23].

Theorem 2 (BLV)**.** *Let* $\boldsymbol{L} \in$ DLIN*. Then there exists a probabilistic one-pass streaming algorithm using* $\mathcal{O}(\log n)$ *space that accepts every string in* $\boldsymbol{L}$ *and rejects every other string with probability at least* $1/n^c$*.*

Using the above algorithm together with Proposition 16 we get the following.

Proposition 20. *If* $\boldsymbol{L} \in$ DLIN*, then* $\overline{\boldsymbol{L}} \in \mathcal{CF}$*.*

8. Conclusion and Outlook

We have initiated a study of extension complexity of formal languages in this article. We have shown various closure properties of compact languages. This is only a first step in what we hope will be a productive path. We have proved a sufficient machine characterization of compact languages in terms of acceptance by online Turing machines. This property is clearly not necessary. What – in terms of computational complexity – characterizes whether or not a language can be represented by small polytopes? We do not know (yet).

References

References

[1] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, R. de Wolf, Exponential lower bounds for polytopes in combinatorial optimization, J. ACM 62 (2) (2015) 17. doi:10.1145/2716307.

[2] D. Avis, H. R. Tiwary, On the extension complexity of combinatorial polytopes, Math. Program. 153 (1) (2015) 95–115. doi:10.1007/s10107-014-0764-2.

[3] T. Rothvoß, The matching polytope has exponential extension complexity, in: Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014, 2014, pp. 263–272. doi:10.1145/2591796.2591834.

[4] G. Braun, R. Jain, T. Lee, S. Pokutta, Information-theoretic approximations of the non-negative rank, ECCC:2013-158.

[5] S. Chan, J. Lee, P. Raghavendra, D. Steurer, Approximate constraint satisfaction requires large LP relaxations, FOCS'13 (2013).

[6] G. Braun, S. Fiorini, S. Pokutta, D. Steurer, Approximation Limits of Linear Programs (Beyond Hierarchies), in: Proc. FOCS 2012, 2012, pp. 480–489.

[7] H. Fawzi, P. Parrilo, Exponential lower bounds on fixed-size psd rank and semidefinite extension complexity, arXiv:1311.2571 (2013).

[8] J. Briët, D. Dadush, S. Pokutta, On the existence of 0/1 polytopes with high semidefinite extension complexity (2013).

[9] T. Lee, D. O. Theis, Support-based lower bounds for the positive semidefinite rank of a nonnegative matrix, arXiv:1203.3961 (2012).

[10] T. Rothvoß, Some 0/1 polytopes need exponential size extended formulations, Math. Program. 142 (1-2) (2013) 255–268. doi:10.1007/s10107-012-0574-3.

[11] G. M. Ziegler, Lectures on polytopes, Vol. 152 of Graduate Texts in Mathematics, Springer-Verlag, Berlin-Heidelberg-New York-London-Paris-Tokyo-Hong Kong-Barcelona-Budapest, 1995.

[12] M. Conforti, G. Cornuéjols, G. Zambelli, Extended formulations in combinatorial optimization, Annals OR 204 (1) (2013) 97–143.

[13] V. Kaibel, Extended formulations in combinatorial optimization, Optima 85 (2011) 2–7.

[14] J. Hartmanis, N. Immerman, S. R. Mahaney, One-way log-tape reductions, in: 19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978, 1978, pp. 65–72. doi:10.1109/SFCS.1978.31.

[15] Hartmanis, Mahaney, Languages simultaneously complete for one-way and two-way log-tape automata, SICOMP: SIAM Journal on Computing 10.

[16] Szepietowski, Weak and strong one-way space complexity classes, IPL: Information Processing Letters 68.

[17] F. Margot, Composition de polytopes combinatoires: une approche par projection, Ph.D. thesis, École polytechnique fédérale de Lausanne (1994).

[18] M. Conforti, K. Pashkovich, The projected faces property and polyhedral relations, Mathematical Programming (2015) 1–12doi:10.1007/s10107-015-0882-5.

[19] Y. Faenza, S. Fiorini, R. Grappe, H. R. Tiwary, Extended formulations, nonnegative factorizations, and randomized communication protocols, Math. Program. 153 (1) (2015) 75–94. doi:10.1007/s10107-014-0755-3.

[20] R. Carr, G. Konjevod, Polyhedral combinatorics, in: H. G (Ed.), Tutorials on Emerging Methodologies and Applications in Operations Research, Vol. 76 of International Series in Operations Research & Management Science, Springer New York, 2005, pp. 2–1–2–46.

[21] S. Onn, V. A. Shlyk, Some efficiently solvable problems over integer partition polytopes, Discrete Applied Mathematics 180 (2015) 135–140.

[22] C. de la Higuera, J. Oncina, Inferring deterministic linear languages, in: Computational Learning Theory, 15th Annual Conference on Computational Learning Theory, COLT 2002, Sydney, Australia, July 8-10, 2002, Proceedings, 2002, pp. 185–200. doi:10.1007/3-540-45435-7_13.

[23] A. Babu, N. Limaye, J. Radhakrishnan, G. Varma, Streaming algorithms for language recognition problems, Theor. Comput. Sci. 494 (2013) 13–23. doi:10.1016/j.tcs.2012.12.028.

F

PARAMETERIZED EXTENSION COMPLEXITY OF INDEPENDENT SET AND RELATED PROBLEMS

The following article is unpublished and is under peer review. It is included here as an appendix for completeness.

Parameterized Extension Complexity of Independent Set and Related Problems

Jakub Gajarský[a], Petr Hliněný[a], Hans Raj Tiwary[b]

[a] *Faculty of Informatics, Masaryk University Brno, Czech Republic.*
[b] *KAM/ITI, MFF, Charles University in Prague, Czech Republic.*

Abstract

Let G be a graph on n vertices and $\mathrm{STAB}_k(G)$ be the convex hull of characteristic vectors of its independent sets of size at most k. It is known that optimizing over $\mathrm{STAB}_k(G)$ is $W[1]$-hard and is FPT tractable for graphs of bounded expansion. We show analogous results for the extension complexity of $\mathrm{STAB}_k(G)$. In particular, we show that when G is a graph from a class of bounded expansion then $\mathrm{xc}(\mathrm{STAB}_k(G)) \leqslant \mathcal{O}(f(k) \cdot n)$ for some function f (depending only on the class). This result can be extended in a simple way to a wide range of similarly defined graph polytopes. In case of general graphs we show that there is *no function* f such that, for all natural numbers k and for all graphs on n vertices, the extension complexity of $\mathrm{STAB}_k(G)$ is at most $f(k) \cdot n^{\mathcal{O}(1)}$.

Keywords: FPT extension complexity, independent set, FO logic

1. Introduction

Polyhedral (aka LP) formulations of combinatorial problems belong to the basic toolbox of combinatorial optimization. In a nutshell, a set of feasible solutions of some problem is suitably encoded by a set of vectors, whose convex hull forms a polytope over which one can then optimize using established tools. A polytope Q is said to be an *extended formulation* or *extension* of a polytope P if P is a projection of Q. Measuring the size of a polytope by the minimum number of inequalities required to describe it, one can define the extension complexity of a polytope to be the smallest size extension of the polytope. This notion has a rich history in combinatorial optimization where by adding extra variables one can sometimes obtain significantly smaller polytopes. For some recent survey on extended formulations in the context of combinatorial optimization and integer programming see [1, 2, 3, 4].

Since linear (or indeed convex) optimization of a polytope P can instead be indirectly done by optimizing over an extended formulation of P, this concept provides a powerful model for solving many combinatorial problems. Various Linear Program (LP) solvers exist today that perform quite well in practice and it is desirable if a problem can be modeled as a small-sized polytope over which one can use an existing LP solver for linear optimization. However, in recent year super-polynomial lower bound on the extension complexity of polytopes associated with many combinatorial problems have been established. These bounds have been generalized to various settings, such as convex extended formulations, approximation algorithms, etc. These results are too numerous for a comprehensive listing but we refer the interested readers to some of the landmark papers in this regard [5, 6, 7, 8].

Many of the recent lower bounds on the extension complexity of various combinatorial polytopes mimic the computational complexity of the underlying problem. For example, it is known that the *extension complexities of polytopes* related to various NP-hard problems are super-polynomial [5, 9, 10, 6]. One satisfying feature of these lower bounds is that they are independent of traditional complexity-theoretic assumptions such as $P \neq NP$. Though, there also exist polytopes corresponding to polynomial

Email addresses: xgajar@fi.muni.cz (Jakub Gajarský), hlineny@fi.muni.cz (Petr Hliněný), hansraj@kam.mff.cuni.cz (Hans Raj Tiwary)

time solvable optimization problems whose extension complexity is super-polynomial. In particular, the perfect matching polytope was shown to have super-polynomial extension complexity by Rothvoß [7].

Notwithstanding the latter weakness, one can ask the related questions in the realm of *parameterized complexity theory*. In this rapidly grown field each problem instance comes additionally equipped with an integer parameter, and the "efficient" class denoted by FPT (*fixed-parameter tractable*) is the one of problems solvable, for every fixed value of the parameter, in polynomial time of degree independent of the parameter. See Section 2 for details.

Similarly as parameterized complexity provides a finer resolution of algorithmic tractability of problems, parameterized polytopes extensions can provide a finer resolution of extension complexities of polytopes of the problems. We follow this direction of research with a case study of the *independent-set polytope* of a graph, naturally parameterized by the solution size. We confirm that the extension complexity of the independent-set polytope indeed mimics the parameterized complexity of the underlying problem, in the following sense:

- We prove that this polytope does not have FPT extension for all graphs (Section 3), but
- an FPT extensions of it exists on every graph class of bounded expansion (Section 4).

We conclude the paper with a straightforward extension of this result to many other similarly defined problems, namely those expressible in existential FO logic of graphs (Section 5), followed by some further thoughts on the topic (Section 6).

2. Preliminaries

We follow standard terminology of graph theory and consider finite simple undirected graphs. We refer to the vertex and edge sets of a graph G as to $V(G)$ and $E(G)$, respectively. In particular, an *independent set* X of vertices of a graph is such that no two elements of X are joined by an edge. By a *cut* in a graph G we mean an edge cut, that is a minimal (by inclusion) set of edges $C \subseteq E(G)$ such that $G \setminus C$ has more connected components than G.

For fundamental concepts of parameterized complexity we refer the readers, e.g., to the monograph [11]. Here we just very briefly recall the needed notions. Considering a problem $\mathcal{P}$ with input of the form $(x, k) \in \Sigma^* \times \mathbb{N}$ (where k is a *parameter*), we say that $\mathcal{A}$ is *fixed-parameter tractable* (shortly FPT) if there is an algorithm solving $\mathcal{A}$ in time $f(k) \cdot n^{\mathcal{O}(1)}$ where f is an arbitrary computable function. In the (parameterized) *k-independent set problem* the input is (G, k) where G is a graph and $k \in \mathbb{N}$, and the question is whether G has an independent set of size at least k.

There is no known FPT algorithm for the k-independent set problem in general and, in fact, the theory of parameterized complexity [11] defines complexity classes $W[t]$, $t \geq 1$, such that the k-independent set problem is complete for $W[1]$. Problems that are $W[1]$-hard do not admit an FPT algorithm unless the Exponential Time Hypothesis fails.

2.1. Fixed-parameter extension complexity

The *size* of a polytope P, denoted by $\mathrm{size}(P)$, is defined to be the number of facets of P, which is the minimum number of inequalities needed to describe P if it is full-dimensional. A polytope Q is called an *extension* of a polytope P if P can be obtained as a linear projection of Q. As a shorthand we will say that in this case Q is an EF of P. As noted in the Introduction, the following is a useful notion:

Definition 2.1 (Extension complexity)**.** The *extension complexity of a polytope P*, denoted by $\mathrm{xc}(P)$, is defined to be the size of the smallest extension. More precisely,

$$\mathrm{xc}(P) := \min_{Q \text{ an EF of } P} \mathrm{size}(Q).$$

In the context of *fixed-parameter extension complexity*, we deal with families of polytopes $\mathcal{P}_n$ where $n \in \mathbb{N}$, and a parameter k. For example, for the independent set problem parameterized by a nonnegative integer k, the family $\mathcal{P}_n$ could be the family of k-independent set polytopes (cf. Subsection 2.2) for a family of n-vertex graphs. The prime question is whether there exists a computable function f such that $\mathrm{xc}(P) \leq f(k) \cdot n^{\mathcal{O}(1)}$ for all k, n and all $P \in \mathcal{P}_n$. As a shorthand we will say in such case that the collection of families $\{\mathcal{P}_n : n \in \mathbb{N}\}$ has *FPT extension complexity* (analogously to the FPT complexity class).

Buchanan in a recent article [12] studied the fixed-parameter extension complexity of the k-vertex cover problem, and proved that for any graph G with n vertices, the k-vertex cover polytope has extension complexity at most $\mathcal{O}(c^k n)$ for some constant $c < 2$. Hence this is a nontrivial example of a polytope class with FPT extension complexity. Buchanan also raised the question whether the k-independent set polytope (Definition 2.4) admits an FPT extension. We answer this in the negative in Theorem 3.6. Note that our negative answer does *not rely* on any complexity theoretical assumptions (such as $FPT \neq W[1]$).

We also look at the positive side of the k-independent set problem. It is known that this problem admits an FPT algorithm (w.r.t. k) on quite rich restricted graph classes, e.g., on classes of bounded expansion [13] (see Subsection 2.3 for the definition). While this finding, in general, does not imply anything about the extension complexity of the k-independent set polytope, we manage to apply the tools of [13] in our setting, and confirm – in Theorem 4.3 – FPT extension complexity of the k-independent set polytope on any graph class of bounded expansion. We also study a meta-generalization of this result in Section 5.

In the course of proving aforementioned Theorems 3.6 and 4.3, we are going to use the following established results on the topic of polytope extension complexity.

Theorem 2.2 (Balas [14])**.** *Let $P_1, P_2, \ldots, P_s$ be polytopes and let $P := \mathrm{conv}(\bigcup_{i=1}^{s} P_i)$. Then,* $\mathrm{xc}(P) \leqslant s + \sum_{i=1}^{s} \mathrm{xc}(P_i)$.

For a graph G, a cut vector is a 0/1 vector of length $|E(G)|$ whose coordinates correspond to whether an edge of G is in a cut $C \subseteq E(G)$ or not. The *cut polytope* is then a convex hull of all the cut vectors of G. Our negative result relies on the following lower bound.

Theorem 2.3 (Fiorini et al. [5])**.** *The extension complexity of the cut polytope of the complete graph K_n on n vertices is $2^{\Omega(n)}$.*

2.2. The k-independent set polytope

Let G be a graph on n vertices. Every subset of vertices of G can be encoded as a characteristic vector of length n. That is, for a subset $S \subseteq V$, define the characteristic vector χ^S as follows:

$$\chi^S_v = \begin{cases} 1 & \text{if } v \in S \\ 0 & \text{otherwise} \end{cases}$$

Definition 2.4 (independent set polytope)**.** The *k-independent set polytope of G*, denoted by $\mathrm{STAB}_k(G)$, is defined to be the convex hull of the characteristic vectors of every independent set of size at most k. That is,

$$\mathrm{STAB}_k(G) = \mathrm{conv}\left(\left\{\chi^S \in \{0,1\}^n |\ S \subseteq V \text{ is an independent set of } G; |S| \leqslant k\right\}\right).$$

In case that $k = n$ we simply speak about the *independent set polytope of G*; $\mathrm{STAB}(G)$.

Alternatively, one could define the polytope $\mathrm{STAB}^{=}_k(G)$ to be the convex hull of all independent sets of size exactly equal to k. That is,

$$\mathrm{STAB}^{=}_k(G) = \mathrm{conv}\left(\left\{\chi^S \in \{0,1\}^n |\ S \subseteq V \text{ is an independent set of } G; |S| = k\right\}\right).$$

To simplify our situation, we note the following:

Lemma 2.5. $\mathrm{xc}(\mathrm{STAB}_k^=(G)) \leqslant \mathrm{xc}(\mathrm{STAB}_k(G)) \leqslant k + \sum_{i=0}^{k} \mathrm{xc}(\mathrm{STAB}_i^=(G))$.

Proof. Clearly, $\mathrm{STAB}_k^=(G)$ is a face of $\mathrm{STAB}_k(G)$. Therefore, $\mathrm{xc}(\mathrm{STAB}_k^=(G)) \leqslant \mathrm{xc}(\mathrm{STAB}_k(G))$.

On the other hand, $\mathrm{STAB}_k(G) = \mathrm{conv}(\bigcup_{i=1}^{k} \mathrm{STAB}_i^=(G))$, and therefore $\mathrm{xc}(\mathrm{STAB}_k(G)) \leqslant k + \sum_{i=0}^{k} \mathrm{xc}(\mathrm{STAB}_i^=(G))$ by Theorem 2.2. □

The above Lemma 2.5 shows that any bounds (lower or upper) that are valid for $\mathrm{xc}(\mathrm{STAB}_k^=(G))$ are also asymptotically valid for $\mathrm{xc}(\mathrm{STAB}_k(G))$. Therefore, in the rest we will use the notation $\mathrm{STAB}_k(G)$ to actually mean $\mathrm{STAB}_k^=(G)$, to keep the notation uncluttered. We stress that this is just for simplicity of notation and does not cause any loss of generality.

We shall also use the following result.

Theorem 2.6 (Buchanan and Butenko [15])**.** *The extension complexity of a graph's independent set polytope is $\mathcal{O}(2^{tw}n)$, where n is the number of vertices and tw denotes its treewidth.*

Note that Buchanan and Butenko give an explicit description of an extension of the independent set polytope.

2.3. Sparsity and bounded expansion

A useful toolbox in our research is the theory of sparse graph classes, largely developed by Nešetřil and Ossona de Mendez. We follow their monograph [16].

We start by defining the notion of edge contraction. Given an edge $e = uv$ of a graph G, we let G/e denote the graph obtained from G by *contracting* the edge e, which amounts to deleting the endpoints of e, introducing a new vertex w_e and making it adjacent to all vertices in the union of the neighborhoods of u and v (excluding u, v themselves). A *minor* of G is a graph obtained from a subgraph of G by contracting zero or more edges. In a more general view, if H is isomorphic to a minor of G, then we call H a minor of G as well, and we write $H \preceq G$.

Alternatively, H is a minor of G if there exists a bijection $\psi\colon V(H) \to \{V_1, \ldots, V_p\}$ where $V_1, \ldots, V_p$ are pairwise disjoint subsets of $V(G)$ inducing connected subgraphs of G, and $uv \in E(H)$ only if there is an edge in G with an endpoint in each of $\psi(u)$ and $\psi(v)$. If, moreover, it is required that each subgraph $G[V_i]$ has radius at most d, meaning that there exist $c_i \in V_i$ (a center) such that every vertex in V_i is within distance at most d from c_i in $G[V_i]$; then H is called a *shallow minor at depth d* of G (shortly, a *d-shallow minor*).

Note that if $u, v \in V(H)$ in a d-shallow minor, and $u_1 \in \psi(u)$ and $v_1 \in \psi(v)$, then $d_G(u_1, v_1) \leq (2d+1) \cdot d_H(u, v)$. The class of d-shallow minors of G is denoted by $G\nabla d$, and this is extended to graph classes $\mathcal{G}$ as well; $\mathcal{G}\nabla d = \bigcup_{G \in \mathcal{G}} G\nabla d$.

One of the most prominent [16] notions of "sparsity" for graph classes is the following one:

Definition 2.7 (Grad and bounded expansion [17])**.** Let $\mathcal{G}$ be a graph class. Then the *greatest reduced average density* of $\mathcal{G}$ with *rank* d is defined as

$$\nabla_d(\mathcal{G}) = \sup_{H \in \mathcal{G}\nabla d} \frac{|E(H)|}{|V(H)|}.$$

A graph class $\mathcal{G}$ has *bounded expansion* if there exists a function $f : \mathbb{N} \to \mathbb{R}$ (called the *expansion function*) such that for all $d \in \mathbb{N}$, $\nabla_d(\mathcal{G}) \leq f(d)$.

We provide a brief informal explanation of Definition 2.7. A graph to be considered "sparse" should not, in particular, contain subgraphs with relatively many edges. Since $G\nabla 0$ is the set of all subgraphs of G, this is captured by $2\nabla_0(G)$ being the maximum average degree over all subgraphs of G. However, the definition requires more; even after contracting edges up to limited depth d, the resulting shallow minors stay free of relatively dense subgraphs, with the maximum average degree bounded by $2\nabla_d(\mathcal{G}) \leq 2f(d)$.

For example, the class $\mathcal{P}$ of all planar graphs has bounded expansion (even with a constant expansion function). On the other hand, a class $\mathcal{Q}$ obtained from all cliques by subdividing each edge twice, although also having relatively few edges, does not have bounded expansion since $\mathcal{Q}\nabla 1$ contains all graphs.

3. Lower Bound: Paired Local-Cut Graphs

We use a shorthand notation $[n] = \{1, 2, \ldots, n\}$. Given positive integers k and n, we define a graph called a *Paired Local-Cut Graph* and denoted $\mathrm{PLC}(k, n)$.

First we create $k2^{\lfloor \log n \rfloor}$ vertices labeled with tuples (i, S) for $i \in [k]$ and $S \subseteq [\lfloor \log n \rfloor]$. These vertices will be called *cut vertices*. Then we create $2\binom{k}{2}2^{2\lfloor \log n \rfloor}$ vertices labeled with tuples (i, j, S_1, S_2) where $1 \leqslant i \neq j \leqslant k$ and $S_1, S_1 \subseteq [\lfloor \log n \rfloor]$. These vertices will be called *pairing vertices*.

We add edges to these vertices of $\mathrm{PLC}(k, n)$ as follows. For each fixed $i \in [k]$ we add the edges between all cut nodes that have labels (i, S). Furthermore, for each fixed pair $i, j \in [k]$ we add the edges between all pairing nodes that have labels (i, j, S_1, S_2). Finally, let u be a cut vertex labeled (i, S) and let v be a pairing vertex labeled (j_1, j_2, S_1, S_2). If $i = j_1$ but $S \neq S_1$ we add edge uv. Symmetrically, if $i = j_2$ but $S \neq S_2$ we add edge uv.

For ease of exposition we will identify vertices of $\mathrm{PLC}(k, n)$ with their labels whenever convenient.

Observation 3.1. The number of vertices of the graph $\mathrm{PLC}(k, n)$ equals $2\binom{k}{2}2^{2\lfloor \log n \rfloor} + k2^{\lfloor \log n \rfloor} \leqslant (kn)^2$.

Observation 3.2. Let (i, S) and (j_1, j_2, S_1, S_2) be two vertices of $\mathrm{PLC}(k, n)$ that are not joined by an edge. If $i = j_1$ then $S = S_1$, and if $i = j_2$ then $S = S_2$.

This together with the next lemma will ensure that in any independent set I of $\mathrm{PLC}(k, n)$ that has size k^2, every index $i \in [k]$ can be uniquely associated with a subset $S_i \subseteq \lfloor \log n \rfloor$.

Lemma 3.3. *Let I be an independent set in $\mathrm{PLC}(k, n)$. Then, $|I| \leqslant k^2$. Moreover, an equality holds if and only if I contains exactly one cut vertex for each $1 \leqslant i \leqslant k$ and exactly one pairing vertex for each $1 \leqslant i \neq j \leqslant k$.*

Proof. Clearly, the set I can contain at most k cut vertices – at most one vertex (i, S_i) for each $1 \leqslant i \leqslant k$. Also, I can contain at most $2\binom{k}{2} = k^2 - k$ pairing vertices – at most one vertex (i, j, S_i, S_j) for each ordered pair $1 \leqslant i, j \leqslant k$. □

We will now relate the vertices of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$ with the vertices of the *polytope* $\mathrm{CUT}(K_r)$ where $r = k \lfloor \log n \rfloor$, to be defined as follows. We denote the vertices and edges of the complete graph K_r (on r vertices) by V_r and E_r, respectively. We group the vertices of K_r into k groups, each of size $\lfloor \log n \rfloor$, and label the vertices v^i_j where $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant \lfloor \log n \rfloor$. Finally, we order the vertices lexicographically according to their labels.

A cut vector of K_r – corresponding to a cut C – is a 0/1 vector of length $\binom{r}{2}$ whose coordinates correspond to whether an edge of K_r is in the cut C or not. The edges of K_r are labeled with pairs (i_1, j_1, i_2, j_2) where $1 \leqslant i_1, i_2 \leqslant k$; $1 \leqslant j_1, j_2 \leqslant \lfloor \log n \rfloor$, and $(i_1, j_1) \leqslant (i_2, j_2)$ lexicographically. So, if z is a cut vector corresponding to a given cut $C \subset E_r$, then $z_{i_1,j_1,i_2,j_2} = 1$ if and only if the edge (i_1, j_1, i_2, j_2) is in C. $\mathrm{CUT}(K_r)$ is the convex hull of all such cut vectors.

Similarly, an independent-set vector of $\mathrm{PLC}(k, n)$ – corresponding to an independent set I – is a 0/1 vector of length $2\binom{k}{2}2^{2\lfloor \log n \rfloor} + k2^{\lfloor \log n \rfloor}$ (see Observation 3.1) whose coordinates correspond to whether the corresponding vertex is in I or not. Recall that the cut vertices of $\mathrm{PLC}(k, n)$ are labeled with a pair consisting of an index from $[k]$, and a subset of $[\lfloor \log n \rfloor]$. Also, the pairing vertices of $\mathrm{PLC}(k, n)$ are labeled with a tuple consisting of two indices from $[k]$ and two subsets of $[\lfloor \log n \rfloor]$.

Let $\mathcal{C}$ be the set of all cuts in K_r, and let $\mathcal{I}$ be the set of all independent sets of size k^2 in $\mathrm{PLC}(k, n)$. Any cut $C \in \mathcal{C}$ creates a bipartition $(S, \overline{S})$ of the vertices of K_r. Recall that the vertices of K_r have been split in k groups. The partition $(S, \overline{S})$ thus induces a partition $(S_i, \overline{S}_i)$ within each of these groups.

Lemma 3.4. *For every pair of natural numbers (k, n) and $r = k \lfloor \log n \rfloor$ it holds that $\mathrm{CUT}(K_r)$ is a projection of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$.*

Proof. Let s denote the length of the independent set vectors of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$. That is $s = 2\binom{k}{2}2^{2\lfloor \log n \rfloor} + k2^{\lfloor \log n \rfloor}$. We describe an affine map $\pi : \mathbb{R}^s \to \mathbb{R}^{\binom{r}{2}}$ such that for every vertex C of $\mathrm{CUT}(K_r)$ there exists a vertex of I of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))$ such that $\pi(I) = C$. Moreover, for every

vertex I of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k,n))$ we show that $\pi(I)$ is a vertex of $\mathrm{CUT}(K_r)$. Since π is an affine map, this will complete the proof.

First we identify the coordinates of $\mathbb{R}^s$ with vertices of $\mathrm{PLC}(k,n)$. To make it easy to refer to this identification we label the coordinates with tuples (i,j,S_1,S_2) defined as follows. For a coordinate corresponding to a cut vertex (i,S) we label the coordinate with (i,i,S,S). For a coordinate corresponding to a pairing vertex (i,j,S_1,S_2) we label the coordinate with the same label. Similarly, we identify the coordinates of $\mathbb{R}^{\binom{r}{2}}$ with the pairs of vertices of K_r, i.e, the coordinate corresponding to an edge between two distinct vertices $v^i_{\ell_1}$ and $v^j_{\ell_2}$ is to be labeled with the integer tuple (i,ℓ_1,j,ℓ_2), assuming that $v^i_{\ell_1} \leqslant v^j_{\ell_2}$ (that is, $i \leqslant j$ and if $i = j$ then $\ell_1 < \ell_2$). Also note that $1 \leqslant i,j \leqslant k$ and $1 \leqslant \ell_1,\ell_2 \leqslant \lfloor \log n \rfloor$.

Given a vector $y \in \mathbb{R}^s$ we define $\pi(y) := z \in \mathbb{R}^{\binom{r}{2}}$ where

$$z_{i_1,\ell_1,i_2,\ell_2} = \begin{cases} \displaystyle\sum_{\substack{S_1 \subseteq [\lfloor \log n \rfloor] \\ S_2 \subseteq [\lfloor \log n \rfloor] \\ \ell_1 \notin S_1 \\ \ell_2 \in S_2}} y_{i_1,i_2,S_1,S_2} + \sum_{\substack{S_1 \subseteq [\lfloor \log n \rfloor] \\ S_2 \subseteq [\lfloor \log n \rfloor] \\ \ell_1 \in S_1 \\ \ell_2 \notin S_2}} y_{i_1,i_2,S_1,S_2} & \text{if } i_1 \neq i_2, \\ \displaystyle\sum_{\substack{S \subseteq [\lfloor \log n \rfloor] \\ \ell_1 \notin S \\ \ell_2 \in S}} y_{i_1,i_1,S,S} + \sum_{\substack{S \subseteq [\lfloor \log n \rfloor] \\ \ell_1 \in S \\ \ell_2 \notin S}} y_{i_1,i_1,S,S} & \text{if } i_1 = i_2\,. \end{cases}$$

Let $y \in \mathbb{R}^s$ be a vertex of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k,n))$. That is, y is the characteristic vector of an independent set $I \in \mathcal{I}$. Since I is of size k^2, for every $1 \leqslant i \leqslant k$ exactly one cut vertex (i,S_i) is picked in I, by Lemma 3.3. Furthermore, by Observation 3.2, for any pairing vertex (i_1,i_2,S,S') picked in I the sets S,S' are unique for given i_1,i_2; if $i = i_1$ then $S = S_i$, and if $i = i_2$ then $S' = S_i$. Consider the subsets $S(I),\overline{S}(I)$ of vertices of K_r defined as follows:

$$\begin{aligned} S(I) &= \left\{ v^i_j \middle|\, 1 \leqslant i \leqslant k; j \in S_i \right\} \\ \overline{S}(I) &= \left\{ v^i_j \middle|\, 1 \leqslant i \leqslant k; j \notin S_i \right\} \end{aligned}$$

It is not difficult to see that $\pi(y)$ is exactly the characteristic vector of the cut defined by $S(I),\overline{S}(I)$ because $z_{i_1,\ell_1,i_2,\ell_2} = 1$ if and only if $v^{i_1}_{\ell_1}$ and $v^{i_2}_{\ell_2}$ do not both lie in $S(I)$ or both in $\overline{S}(I)$.

On the other hand, if z is the characteristic vector of a cut defined by subsets $S,\overline{S}$ of vertices of K_r, then we can define $S_i = S \cap \left\{ v^i_j \middle|\, 1 \leqslant j \leqslant \lfloor \log n \rfloor \right\}$ so that $\left\{ (i,S_i) | 1 \leqslant i \leqslant k \right\} \cup \left\{ (i,j,S_i,S_j) | 1 \leqslant i < j \leqslant k \right\}$ is an independent set I of $\mathrm{PLC}(k,n)$ whose size is k^2 and whose characteristic vector projects to z under π.

Hence π defines a projection from $\mathrm{PLC}(k,n)$ to $\mathrm{CUT}(K_r)$. □

Corollary 3.5. *There exists a constant $c' > 0$ such that for $k,n \in \mathbb{N}$,*

$$\mathrm{xc}\left(\mathrm{STAB}_{k^2}(\mathrm{PLC}(k,n))\right) \geqslant n^{c'k}.$$

Proof. By Lemma 3.4, $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k,n))$ is an extended formulation of $\mathrm{CUT}(K_r)$ with $r = k\lfloor \log n \rfloor$. So any extended formulation of $\mathrm{STAB}_{k^2}(\mathrm{PLC}(k,n))$ is also an extended formulation of $\mathrm{CUT}(K_r)$. By Theorem 2.3, $\mathrm{xc}\left(\mathrm{CUT}(K_r)\right) \geqslant 2^{\Omega(r)}$. Therefore, $\mathrm{xc}\left(\mathrm{STAB}_{k^2}(\mathrm{PLC}(k,n))\right) \geqslant \mathrm{xc}\left(\mathrm{CUT}(K_r)\right) \geqslant 2^{\Omega(r)} \geqslant n^{c'k}$ for some constant $c' > 0$. □

We can now easily finish with the main result of this section.

Theorem 3.6. *There is no function $f : \mathbb{N} \to \mathbb{R}$ such that $\mathrm{xc}(\mathrm{STAB}_k(G)) \leqslant f(k) \cdot n^{\mathcal{O}(1)}$ for all natural numbers k and all graphs G on n vertices.*

Proof. Suppose, on the contrary, that such a function f does exist. That is, there is a constant c such that for every pair of natural numbers (ℓ, m) and for all m-vertex graphs G it holds that $\mathrm{xc}(\mathrm{STAB}_\ell(G)) \leqslant f(\ell) \cdot m^c$.

Given a pair (k, n) of natural numbers consider the graph $\mathrm{PLC}(k, n)$. By Corollary 3.5, we have that $\mathrm{xc}\left(\mathrm{STAB}_{k^2}(\mathrm{PLC}(k, n))\right) \geqslant n^{c'k}$ for some constant $c' > 0$. On the other hand, from our assumption for $\ell = k^2$ and $m \leqslant (kn)^2$ we have that $\mathrm{xc}\left(\mathrm{STAB}_{k^2}(G)\right) \leqslant f(k^2) \cdot (kn)^{2c}$. Therefore, $n^{c'k} \leqslant f(k^2) \cdot (kn)^{2c}$ and so

$$c'k \log n \leqslant \log f(k^2) + 2c(\log k + \log n)$$

$$(c'k - 2c) \log n \leqslant \log f(k^2) + 2c \log k.$$

Clearly, the latter cannot hold true for a sufficiently large but fixed parameter k and arbitrary n, a contradiction. Hence no such function f exists. □

4. Upper Bound: Bounded Expansion Classes

While Theorem 3.6 asserts that FPT extensions are not possible for the k-independent set polytopes of all graphs, there is still a good chance to prove a positive result for restricted classes of graphs. An example of such restriction is, by a simple modification of Theorem 2.6, presented by graph classes of bounded tree-width; although, this is somehow too restrictive. We show that in the case of k being a fixed parameter, one can go much further.

The underlying idea of our approach can be informally explained as follows. Imagine we can "guess", in advance, a (short) list of well-structured subgraphs of our graph such that every possible independent set is fully contained in at least one of them. Then we can separately construct an independent set polytope for each one of the subgraphs, and make their union at the end (Theorem 2.2). This ambitious plan indeed turns out to be viable for graph classes of bounded expansion (Definition 2.7), and the key to the success is a combination of a powerful structural characterization of bounded expansion (Theorem 4.2) with the size bound k on the independent sets.

In order to state the desired structural characterization, we need the notion of treedepth. In this context, a *rooted forest* is a disjoint union of rooted trees. The height of a rooted forest is the maximum distance from one of the forest's roots to a vertex in the same tree. The *closure* $\mathrm{clos}(\mathcal{F})$ of a rooted forest $\mathcal{F}$ is the graph with the vertex set $\bigcup_{T \in \mathcal{F}} V(T)$ and the edge set $\{xy : x \text{ is an ancestor of } y \text{ in a tree of } \mathcal{F}\}$. The *treedepth* $\mathrm{td}(G)$ of a graph G is the minimum height plus one of a rooted forest $\mathcal{F}$ such that $G \subseteq \mathrm{clos}(\mathcal{F})$.

For simplicity, we skip here the rather complicated definition of treewidth (which is never directly used in our arguments, anyway). Instead, we just use the following fact which is easy to establish directly from the definitions:

Observation 4.1. For any G, the treewidth of G is at most the treedepth of G minus one.

The amazing connection between graph classes of bounded expansion and treedepth is captured by the notion of low treedepth coloring: For an integer $d \geq 1$, an assignment of colors to the vertices of a graph G is a *low tree-depth coloring of order d* if, for every $s = 1, 2, \ldots, d$, the union of any s color classes induces a subgraph of G of treedepth at most s.

In particular, every low tree-depth coloring of G is a proper coloring of G (but not the other way round), and the union of any two color classes induces a forest of stars. The following result is crucial:

Theorem 4.2 (Nešetřil and Ossona de Mendez [13, 17])**.** *If $\mathcal{G}$ is a class of graphs of bounded expansion, then there is a function $N_{\mathcal{G}} : \mathbb{N} \to \mathbb{N}$ (depending on the expansion function of $\mathcal{G}$) such that for any graph $G \in \mathcal{G}$ and k, there exists a low tree-depth coloring of order k of G using at most $N_{\mathcal{G}}(k)$ colors. This coloring can be found in linear time for fixed k.*

We are now ready to state and prove the main theorem of this section.

Theorem 4.3. *Let $\mathcal{G}$ be any graph class of bounded expansion. Then there exists a computable function $f : \mathbb{N} \to \mathbb{N}$, depending on the expansion function of $\mathcal{G}$, such that*

$$\mathrm{xc}\big(\mathrm{STAB}_k(G)\big) \leqslant f(k) \cdot n$$

holds for every integer n and every n-vertex graph $G \in \mathcal{G}$. Moreover, an explicit extension of $\mathrm{STAB}_k(G)$ of size at most $f(k) \cdot n$ can be found in linear time for fixed k and $\mathcal{G}$.

Proof. Since $\mathcal{G}$ is a graph class of bounded expansion, by Theorem 4.2 we can for any $G \in \mathcal{G}$ and given k find an assignment $c : V(G) \to [N_{\mathcal{G}}(k)]$ such that c is a low tree-depth coloring of order k. Let $\mathcal{J}_k := \binom{[N_{\mathcal{G}}(k)]}{k}$ denote the set of k-element subsets of $[N_{\mathcal{G}}(k)]$, and let a subgraph $G_J \subseteq G$ where $J \in \mathcal{J}_k$, be defined as the subgraph of G induced on $\bigcup_{j \in \mathcal{J}_k} c^{-1}(j)$ – the color classes indexed by J.

Note the following two immediate facts:

a) by the definition, each G_J, $J \in \mathcal{J}_k$, is of treedepth at most $|J| = k$;

b) for every set $X \subseteq V(G)$ (independent or not) of size $|X| \leq k$, there is $J \in \mathcal{J}_k$ such that $X \subseteq V(G_J)$.

Consequently,

$$\mathrm{STAB}_k(G) = \mathrm{conv}\left(\bigcup\nolimits_{J \in \mathcal{J}_k} \mathrm{STAB}_k(G_J)\right)$$

and it is sufficient to bound the extension complexity of each $\mathrm{STAB}_k(G_J)$.

By (a) and Observation 4.1, $tw(G_J) \leq k-1$ and Theorem 2.6 applies here: $\mathrm{xc}\big(\mathrm{STAB}_k(G_J)\big) \leqslant \mathcal{O}(2^k \cdot |G_J|) \leqslant \mathcal{O}(2^k \cdot n) \leqslant c' 2^k \cdot n$ for a suitable constant c'. Then, by Theorem 2.2, we have

$$\begin{aligned}\mathrm{xc}\big(\mathrm{STAB}_k(G)\big) &\leqslant |\mathcal{J}_k| + \sum_{J \in \mathcal{J}_k} \mathrm{xc}\big(\mathrm{STAB}_k(G_J)\big) \\ &\leqslant |\mathcal{J}_k| \cdot (1 + c' 2^k \cdot n) \leqslant \binom{N_{\mathcal{G}}(k)}{k} (1 + c' 2^k) \cdot n \leqslant f(k) \cdot n\,.\end{aligned}$$

Note that this extended formulation can be constructed in linear time for fixed k since the low tree-depth coloring in Theorem 4.2 can be found in linear time, the extended formulation in Theorem 2.6 is explicit, and the extended formulation of union of polytopes can be constructed in linear time from the extensions of the component polytopes [14]. □

5. Generalizing the Upper Bound

The tools used in Section 4 for graph classes of bounded expansion apply not only to the particular case of independent sets of size at most k, but to a much wider range of problems. This can be formulated as a meta-result about graph polytopes definable in the first-order logic.

The *first-order logic of graphs* (abbreviated as FO) applies the standard language of first-order logic to a graph G viewed as a relational structure with the domain $V(G)$ and the single binary (symmetric) relation $E(G)$. For example, the formula $\iota(x_1, \ldots, x_k) \equiv \bigwedge_{i \neq j} \big(\neg edge(x_i, x_j) \wedge x_i \neq x_j\big)$ asserts that $\{x_1, \ldots, x_k\}$ is an independent set of size exactly k. A slightly more involved example describes a vertex cover tuple as $\gamma(x_1, \ldots, x_k) \equiv \forall y,z\big(edge(y,z) \to \bigvee_{i=1}^{k} (y = x_i \vee z = x_i)\big)$.

To any FO formula $\phi(x_1, \ldots, x_k)$ one can assign a graph polytope, although this task is not as straightforward as with the independent set polytope since the order of arguments of ϕ matters, and the same vertex may be repeated among the arguments. For an ordered k-tuple of vertices $W = (w_1, \ldots, w_k) \in V(G)^k$ we thus define its characteristic vector χ^W of length $k|V(G)|$ by

$$\chi^W_{v,i} = \begin{cases} 1 & \text{if } v = w_i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that χ^W always satisfies $\sum_{v \in V(G)} \chi^W_{v,i} = 1$ for each $i = 1, \ldots, k$, by the definition.

If $W = (w_1, \ldots, w_k) \in V(G)^k$ is such that $\phi(w_1, \ldots, w_k)$ holds true in G, we write $G \models \phi(w_1, \ldots, w_k)$. We can now give the following definition:

Definition 5.1 (FO polytope)**.** Let $\phi(x_1, \ldots, x_k)$ be an FO formula with k free variables. The *(first-order) ϕ-polytope of G*, denoted by $\mathrm{FOP}_\phi(G)$, is defined to be the convex hull of the characteristic vectors of every k-tuple of vertices of G such that $\phi(w_1, \ldots, w_k)$ holds true in G. That is,

$$\mathrm{FOP}_\phi(G) = \mathrm{conv}\left(\left\{\chi^W \in \{0,1\}^n \middle| \, W = (w_1, \ldots, w_k) \in V(G)^k,\ G \models \phi(w_1, \ldots, w_k)\right\}\right).$$

This definition is closely related to Definition 2.4 via the following observation:

Lemma 5.2. *Let $\iota(x_1, \ldots, x_k) \equiv \bigwedge_{i \neq j} \left(\neg edge(x_i, x_j) \wedge x_i \neq x_j\right)$ (the k-independent set formula). For every graph G, the ι-polytope $\mathrm{FOP}_\iota(G)$ is an extension of $\mathrm{STAB}_k(G)$.*

Proof. If G has n vertices then $\mathrm{STAB}_k(G) = \left\{ y \in \mathbb{R}^n \,\middle|\, y_v = \sum_{i=1}^k \chi^W_{v,i},\ \chi^W \in \mathrm{FOP}_\iota(G) \right\}$. Therefore, $\mathrm{STAB}_k(G)$ is a projection of $\mathrm{FOP}_\iota(G)$ given by the projection map described by $y_v = \sum_{i=1}^k \chi^W_{v,i}$ for all vertices v of G. □

Using the decomposition provided by Theorem 4.2, we can in fact obtain a more general result which we will discuss now. We will use the following weaker form[1] of a recent result of Kolman et al. [18].

Theorem 5.3 ((Kolman, Koutecký and Tiwary [18]))**.** *Let $\phi(x_1, \ldots, x_k)$ be an existential FO formula with k free variables and ℓ quantifiers. Then there exists a computable function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, such that*

$$\mathrm{xc}\big(\mathrm{FOP}_\phi(G)\big) \leqslant g(k+\ell, \tau) \cdot n$$

holds for every integer n and every n-vertex graph G of treewidth τ. Furthermore, this extension can be computed in linear time for fixed k, ℓ and τ.

We are now able to directly extend Theorem 4.3 to the following restrictive fragment of FO logic. We say that an FO formula $\phi(x_1, \ldots, x_k)$ is *existential FO* if it can be written as $\phi(x_1, \ldots, x_k) \equiv \exists y_1 \ldots y_\ell \, \psi(x_1, \ldots, x_k, y_1, \ldots, y_\ell)$, where ψ is quantifier-free. The number ℓ of quantified variables in ϕ is called the *quantifier rank of ϕ*.

Theorem 5.4. *Let $\phi(x_1, \ldots, x_k)$ be an existential FO formula with k free variables and quantifier rank ℓ. Also, let $\mathcal{G}$ be any graph class of bounded expansion. Then there exists a computable function $f : \mathbb{N} \to \mathbb{N}$, depending on the expansion function of $\mathcal{G}$, such that*

$$\mathrm{xc}\big(\mathrm{FOP}_\phi(G)\big) \leqslant f(k+\ell) \cdot n$$

holds for every integer n and every n-vertex graph $G \in \mathcal{G}$. Furthermore, an explicit extension of $\mathrm{FOP}_\phi(G)$ of size at most $f(k+\ell) \cdot n$ can be found in linear time for fixed k, ℓ and $\mathcal{G}$.

Since the proof of Theorem 5.4 is a straightforward extension of the proof of Theorem 4.3, we skip it in this restricted paper.

Proof. We start with two simple facts of model theory:

a) If H is an induced subgraph of G, and $H \models \phi(w_1, \ldots, w_k)$ for $w_1, \ldots, w_k \in V(H)$, then $G \models \phi(w_1, \ldots, w_k)$ (since ϕ is existential).

b) If $G \models \phi(w_1, \ldots, w_k)$ for any $W = \{w_1, \ldots, w_k\} \subseteq V(G)$, then there is $U \subseteq V(G)$, $|U| \leq \ell$, such that $G[W \cup U] \models \phi(w_1, \ldots, w_k)$ where $G[W \cup U]$ is the subgraph of G induced on $W \cup U$ (since the quantifier rank of existential ϕ is $\leq \ell$).

[1] The result of Kolman et al. applies to Monadic Second Order logic: a logic that subsumes FO logic.

We can hence apply the same technique as in the proof of Theorem 4.3 – using a low treedepth coloring c of G now by $N_{\mathcal{G}}(k+\ell)$ colors from Theorem 4.2. Again, let $\mathcal{J}_{k+\ell} := \binom{[N_{\mathcal{G}}(k+\ell)]}{k+\ell}$ denote the set of $(k+\ell)$-element subsets of $[N_{\mathcal{G}}(k+\ell)]$, and let a subgraph $G_J \subseteq G$ where $J \in \mathcal{J}_{k+\ell}$, be defined as the subgraph of G induced on $\bigcup_{j\in\mathcal{J}_{k+\ell}} c^{-1}(j)$ – the color classes of c indexed by J. By a),b) we immediately get

$$\mathrm{FOP}_\phi(G) = \mathrm{conv}\left(\bigcup\nolimits_{J\in\mathcal{J}_{k+\ell}} \mathrm{FOP}_\phi(G_J)\right).$$

From Theorems 2.2 and 5.3 (via Observation 4.1) we then analogously conclude

$$\begin{aligned}\mathrm{xc}\big(\mathrm{FOP}_\phi(G)\big) &\leqslant |\mathcal{J}_{k+\ell}| + \sum_{J\in\mathcal{J}_{k+\ell}} \mathrm{xc}\big(\mathrm{FOP}_\phi(G_J)\big)\\ &\leqslant |\mathcal{J}_{k+\ell}| \cdot \big(1 + g(k+\ell, k+\ell-1)\cdot n\big) \leqslant f(k+\ell)\cdot n\,.\end{aligned}$$

Again, this extended formulation can be constructed in linear time for fixed k, ℓ. □

6. Conclusions

We have begun to study the question: to which extent FPT tractability of the k-independent set problem on graph classes is related to the FPT extension complexity of the (corresponding) k-independent-set polytope? Not surprisingly, we confirm that there cannot be FPT extensions of this polytope in the class of all graphs (note, though, that our proof is absolute and does not rely on the assumption $FPT \neq W[1]$). On the other hand, the k-independent-set problem is linear-time FPT tractable on graph classes of bounded expansion [13], and we construct a linear FPT extension for its polytope on such classes. This positive result easily carries over to all existential FO problems of graphs.

There are two possible directions of future research in this regard:

1. A graph class $\mathcal{G}$ is *nowhere dense* [16] if there is no integer d such that $\mathcal{G}\nabla d$ contains all graphs. Every graph class of bounded expansion is nowhere dense, but the converse is not true. The k-independent-set problem is also known to be in FPT on every nowhere dense class, which follows from a more general result of [19]. It is natural to ask whether the same can hold for the fixed-parameter extension complexity of its polytope.

2. The deep tractability results of [20, 19] (on bounded expansion and nowhere dense classes, respectively) address problems in full FO logic of graphs, i.e., allowing also for universal quantifiers in the problem expression. This suggests that perhaps, for every FO formula ϕ, the related ϕ-polytope can also have FPT extension complexity on, say, graph classes of bounded expansion. Though, the involved proof techniques of [20, 19] do not seem to easily translate to the extension complexity setting.

In a broader view, one may regard the property of (the problem polytope) having an FPT extension complexity as a finer (case-by-case) resolution of the class FPT. For an explanation; the well-established assumption $FPT \neq W[1]$ implies that problems not in FPT do not have FPT extensions, while on the other hand the example of the matching polytope [7] suggests that there may also be FPT problems whose polytopes do not have FPT extensions. We believe this task is worth further detailed investigation.

One may try to proceed even further and ask a general question:

3. Is it true that all $W[t]$-hard problems for some $t \geq 1$ do not admit FPT extensions?

However, this question is not even easy to formulate since the polytope we associate with a problem remains a specific choice which by no means is the only choice.

Moreover, it can be argued that either possible answer to the very broad question (3) would be a significant breakthrough in the complexity world. Say, since some FPT problems such as k-vertex cover do admit FPT extensions [12], an affirmative answer to (3) would imply that this problem is

not $W[t]$-complete and so $FPT \neq W[t]$. On the other hand, if the answer to (3) was no, then this would imply the existence of non-uniform FPT circuits for $W[t]$-complete problems which is considered unlikely.

References

[1] M. Conforti, G. Cornuéjols, G. Zambelli, Extended formulations in combinatorial optimization, 4OR 8 (2010) 1–48.

[2] F. Vanderbeck, L. A. Wolsey, Reformulation and decomposition of integer programs, in: M. J. et al. (Ed.), 50 Years of Integer Programming 1958-2008, Springer, 2010, pp. 431–502.

[3] V. Kaibel, Extended formulations in combinatorial optimization, Optima 85 (2011) 2–7.

[4] L. A. Wolsey, Using extended formulations in practice, Optima 85 (2011) 7–9.

[5] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, R. de Wolf, Exponential lower bounds for polytopes in combinatorial optimization, J. ACM 62 (2) (2015) 17.

[6] G. Braun, S. Fiorini, S. Pokutta, D. Steurer, Approximation limits of linear programs (beyond hierarchies), Math. Oper. Res. 40 (3) (2015) 756–772. doi:10.1287/moor.2014.0694.

[7] T. Rothvoß, The matching polytope has exponential extension complexity, in: Proc. of the 46th ACM Symposium on Theory of Computing, (STOC), 2014, pp. 263–272.

[8] S. Chan, J. Lee, P. Raghavendra, D. Steurer, Approximate constraint satisfaction requires large LP relaxations, FOCS'13 (2013).

[9] D. Avis, H. R. Tiwary, On the extension complexity of combinatorial polytopes, in: Proc. ICALP(1) 2013, 2013, pp. 57–68.

[10] S. Pokutta, M. Van Vyve, A note on the extension complexity of the knapsack polytope, Operations Research Letters 41 (4) (2013) 347–350.

[11] R. G. Downey, M. R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013. doi:10.1007/978-1-4471-5559-1.

[12] A. Buchanan, Extended formulations for vertex cover, available on Optimization Online (2015).

[13] J. Nešetřil, P. Ossona de Mendez, Linear time low tree-width partitions and algorithmic consequences, in: STOC'06, ACM, 2006, pp. 391–400. doi:10.1145/1132516.1132575.

[14] E. Balas, Disjunctive programming: Properties of the convex hull of feasible points, Discrete Applied Mathematics 89 (1–3) (1998) 3–44.

[15] A. Buchanan, S. Butenko, Tight extended formulations for independent set, available on Optimization Online (2014).

[16] J. Nešetřil, P. Ossona de Mendez, Sparsity: Graphs, Structures, and Algorithms, Vol. 28 of Algorithms and Combinatorics, Springer, 2012.

[17] J. Nešetřil, P. Ossona de Mendez, Grad and classes with bounded expansion I. Decompositions, European J. Combin. 29 (3) (2008) 760–776.

[18] P. Kolman, M. Koutecký, H. R. Tiwary, Extension complexity, mso logic, and treewidth, CoRR abs/1507.04907.
URL http://arxiv.org/abs/1507.04907

[19] M. Grohe, S. Kreutzer, S. Siebertz, Deciding first-order properties of nowhere dense graphs, in: STOC'14, ACM, 2014, pp. 89–98. doi:10.1145/2591796.2591851.

[20] Z. Dvořák, D. Král', R. Thomas, Deciding first-order properties for sparse graphs, in: FOCS'10, IEEE Computer Society, 2010, pp. 133–142.

G

EXTENSION COMPLEXITY, MSO LOGIC, AND TREEWIDTH

The following articles has been accepted for publication in the proceedings of the *15th Scandinavian Symposium and Workshop on Algorithm Theory*. It is included here as an appendix for completeness.

Extension Complexity, MSO Logic, and Treewidth *

Petr Kolman[1], Martin Koutecký[1], and Hans Raj Tiwary[1]

1 **Department of Applied Mathematics (KAM) & Institute of Theoretical Computer Science (ITI), Faculty of Mathematics and Physics (MFF), Charles University in Prague, Czech Republic.**
{kolman,koutecky,hansraj}@kam.mff.cuni.cz

Abstract

We consider the convex hull $P_\varphi(G)$ of all satisfying assignments of a given MSO_2 formula φ on a given graph G. We show that there exists an extended formulation of the polytope $P_\varphi(G)$ that can be described by $f(|\varphi|, \tau) \cdot n$ inequalities, where n is the number of vertices in G, τ is the treewidth of G and f is a computable function depending only on φ and τ.

In other words, we prove that the extension complexity of $P_\varphi(G)$ is linear in the size of the graph G, with a constant depending on the treewidth of G and the formula φ. This provides a very general yet very simple meta-theorem about the extension complexity of polytopes related to a wide class of problems and graphs.

1998 ACM Subject Classification F.2.2, F.4.1, G.1.6, G.2.1, G.2.2

Keywords and phrases Extension Complexity – FPT – Courcelle's Theorem – MSO Logic

Digital Object Identifier 10.4230/LIPIcs.SWAT.2016.

1 Introduction

In the '70s and '80s, it was repeatedly observed that various NP-hard problems are solvable in polynomial time on graphs resembling trees. The graph property of *resembling a tree* was eventually formalized as having bounded treewidth, and in the beginning of the '90s, the class of problems efficiently solvable on graphs of bounded treewidth was shown to contain the class of problems definable by the Monadic Second Order Logic (MSO_2) (Courcelle [11], Arnborg et al. [1], Courcelle and Mosbah [13]). Using similar techniques, analogous results for weaker logics were then proven for wider graph classes such as graphs of bounded cliquewidth and rankwidth [12]. Results of this kind are usually referred to as *Courcelle's theorem* for a specific class of structures.

In this paper we study the class of problems definable by the MSO logic from the perspective of extension complexity. While small extended formulations are known for various special classes of polytopes, we are not aware of any other result in the theory of extended formulations that works on a wide class of polytopes the way Courcelle's theorem works for a wide class of problems and graphs.

Our Contribution. We prove that satisfying assignments of an MSO_2 formula φ on a graph of bounded treewidth can be *expressed* by a "small" linear program. More precisely, there exists a computable function f such that the convex hull – $P_\varphi(G)$ – of satisfying assignments

* This research was partially supported by the project P202-13/201414 of GA ČR.

of φ on a graph G on n vertices with treewidth τ can be obtained as the projection of a polytope described by $f(|\varphi|, \tau) \cdot n$ linear inequalities; we call $P_\varphi(G)$ the *MSO polytope*. All our results can be extended to general finite structures where the restriction on treewidth applies to the treewidth of their Gaifman graph [30].

Our proof essentially works by "merging the common wisdom" from the areas of extended formulations and fixed parameter tractability. It is known that dynamic programming can usually be turned into a compact extended formulation [32, 18], and that Courcelle's theorem can be seen as an instance of dynamic programming [26], and therefore it should be expected that the polytope of satisfying assignments of an MSO formula of a bounded treewidth graph be small.

However, there are a few roadblocks in trying to merge these two folklore wisdoms. For one, while Courcelle's theorem being an instance of dynamic programming in some sense may be obvious to an FPT theorist, it is far from clear to anyone else what that sentence may even mean. On the other hand, being able to turn a dynamic program into a compact polytope may be a theoretical possibility for an expert on extended formulations, but it is by no means an easy statement for an outsider to comprehend. What complicates the matters even further is that the result of Martin et al. [32] is not a result that can be used in a black box fashion. That is, a certain condition must be satisfied to get a compact extended formulation out of a dynamic program. This is far from a trivial task, especially for a theorem like Courcelle's theorem.

The rest of the article is organized as follows. In Section 2 we review some previous work related to Courcelle's theorem and extended formulations. In Section 3 we describe the relevant notions related to polytopes, extended formulations, graphs, treewidth and MSO logic. In Section 4 we prove the existence of compact extended formulations for MSO polytopes parameterized by the length of the given MSO formula and the treewidth of the given graph. In Section 5 we describe how to efficiently construct such a polytope given a tree decomposition of a graph.

2 Related Work

2.1 MSO Logic vs. Treewidth

Because of the wide relevance of the treewidth parameter in many areas (cf. the survey of Bodlaender [5]) and the large expressivity of the MSO and its extensions (cf. the survey of Langer et al. [27]), considerable attention was given to Courcelle's theorem by theorists from various fields, reinterpreting it into their own setting. These reinterpretations helped uncover several interesting connections.

The classical way of proving Courcelle's theorem is constructing a tree automaton A in time only dependent on φ and the treewidth τ, such that A accepts a tree decomposition of a graph of treewidth τ if and only if the corresponding graph satisfies φ; this is the automata theory perspective [11]. Another perspective comes from finite model theory where one can prove that a certain equivalence on the set of graphs of treewidth at most τ has only finitely many (depending on φ and τ) equivalence classes and that it *behaves well* [16]. Another approach proves that a quite different equivalence on so-called extended model checking games has finitely many equivalence classes [23] as well; this is the game-theoretic perspective. It can be observed that the finiteness in either perspective stems from the same roots.

Another related result is an *expressivity* result: Gottlob et al. [16] prove that on bounded treewidth graphs, a certain subset of the database query language Datalog has the same

expressive power as the MSO. This provides an interesting connection between the automata theory and the database theory.

2.2 Extended Formulations

Sellmann, Mercier, and Leventhal [34] claimed to show compact extended formulation for binary Constraint Satisfaction Problems (CSP) for graphs of bounded treewidth, but their proof is not correct [33]. The first two authors of this paper gave extended formulations for CSP that has polynomial size for instances whose constraint graph has bounded treewidth [25] using a different technique. Bienstock and Munoz [3] prove similar results for the approximate and exact version of the problem. In the exact case, Bienstock and Munoz's bounds are slightly worse than those of Kolman and Koutecký [25]. It is worth noting that CSPs are a restricted subclass of problems that can be modeled using MSO logic. Laurent [28] provides extended formulations for the independent set and max cut polytopes of size $O(2^\tau n)$ for n-vertex graphs of treewidth τ and, independently, Buchanan and Butenko [8] provide an extended formulation for the independent set polytope of the same size.

A lot of recent work on extended formulations has focussed on establishing lower bounds in various settings: exact, approximate, linear vs. semidefinite, etc. (See for example [15, 2, 6, 29]). A wide variety of tools have been developed and used for these results including connections to nonnegative matrix factorizations [37], communication complexity [14], information theory [7], and quantum communication [15] among others.

For proving upper bounds on extended formulations, several authors have proposed various tools as well. Kaibel and Loos [19] describe a setting of branched polyhedral systems which was later used by Kaibel and Pashkovich [20] to provide a way to construct polytopes using reflection relations.

A particularly specific composition rule, which we term glued product (cf. Subsection 3.1), was studied by Margot in his PhD thesis [31]. Margot showed that a property called the projected face property suffices to glue two polytopes efficiently. Conforti and Pashkovich [10] describe and strengthen Margot's result to make the projected face property to be a necessary and sufficient condition to describe the glued product in a particularly efficient way.

Martin et al. [32] have shown that under certain conditions, an efficient dynamic programming based algorithm can be turned into a compact extended formulation. Kaibel [18] summarizes this and various other methods.

3 Preliminaries

3.1 Polytopes, Extended Formulations and Extension Complexity

For background on polytopes we refer the reader to Grünbaum [17] and Ziegler [38]. To simplify reading of the paper for the audience that is not working often in the area of polyhedral combinatorics, we provide here a brief glossary of common polyhedral notions that are used in this article.

A *hyperplane* in $\mathbb{R}^n$ is a closed convex set of the form $\{x|a^\intercal x = b\}$ where $a \in \mathbb{R}^n, b \in \mathbb{R}$. A *halfspace* in $\mathbb{R}^n$ is a closed convex set of the form $\{x|a^\intercal x \leqslant b\}$ where $a \in \mathbb{R}^n, b \in \mathbb{R}$. The inequality $a^\intercal x \leqslant b$ is said to define the corresponding halfspace. A *polytope* $P \subseteq \mathbb{R}^n$ is a bounded subset defined by intersection of finite number of halfspaces. A result of Minkowsky-Weyl states that equivalently, every polytope is the convex hull of a finite number of points. Let h be a halfspace defined by an inequality $a^\intercal x \leqslant b$; the inequality is said to

be *valid* for a polytope P if $P = P \cap h$. Let h be a halfspace defined by a valid inequality $a^\intercal x \leqslant b$; then, $P \cap \{x | a^\intercal x = b\}$ is said to be a *face* of P.

Note that, taking a to be the zero vector and $b = 0$ results in the face being P itself. Also, taking a to be the zero vector and $b = 1$ results in the empty set. These two faces are often called the trivial faces and they are polytopes "living in" dimensions n and -1, respectively. Every face - that is not trivial - is itself a polytope of dimension d where $0 \leqslant d \leqslant n-1$.

It is not uncommon to refer to three separate (but related) objects as a face: the actual face as defined above, the valid inequality defining it, and the equation corresponding to the valid inequality. While this is clearly a misuse of notation, the context usually makes it clear as to exactly which object is being referred to.

The zero dimensional faces of a polytope are called its vertices, and the $(n-1)$-dimensional faces are called its facets.

Let P be a polytope in $\mathbb{R}^d$. A polytope Q in $\mathbb{R}^{d+r}$ is called an *extended formulation* or an *extension* of P if P is a projection of Q onto the first d coordinates. Note that for any linear map $\pi : \mathbb{R}^{d+r} \to \mathbb{R}^d$ such that $P = \pi(Q)$, a polytope Q' exists such that P is obtained by dropping all but the first d coordinates on Q' and, moreover, Q and Q' have the same number of facets.

The *size* of a polytope is defined to be the number of its facet-defining inequalities. Finally, the *extension complexity* of a polytope P, denoted by $\mathrm{xc}(P)$, is the size of its smallest extended formulation. We refer the readers to the surveys [9, 35, 18, 36] for details and background of the subject and we only state three basic propositions about extended formulations here.

▶ **Proposition 1.** *Let P be a polytope with a vertex set $V = \{v_1, \ldots, v_n\}$. Then* $\mathrm{xc}(P) \leqslant n$.

Proof. Let $P = \mathrm{conv}(\{v_1, \ldots, v_n\})$ be a polytope. Then, P is the projection of

$$Q = \left\{ (x, \lambda) \,\middle|\, x = \sum_{i=1}^{n} \lambda_i v_i; \sum_{i=1}^{n} \lambda_i = 1; \lambda_i \geqslant 0 \text{ for } i \in \{1, \ldots, n\} \right\}.$$

It is clear that Q has at most n facets and therefore $\mathrm{xc}(P) \leqslant n$. ◀

▶ **Proposition 2.** *Let P be a polytope obtained by intersecting a set H of hyperplanes with a polytope Q. Then* $\mathrm{xc}(P) \leqslant \mathrm{xc}(Q)$.

Proof. Note that any extended formulation of Q, when intersected with H, gives an extended formulation of P. Intersecting a polytope with hyperplanes does not increase the number of facet-defining inequalities (and only possibly reduces it). ◀

The (cartesian) *product* of two polytopes P_1 and P_2 is defined as

$$P_1 \times P_2 = \mathrm{conv}(\{(x, y) \mid x \in P_1, y \in P_2\}).$$

▶ **Proposition 3.** *Let P_1, P_2 be two polytopes. Then*

$$\mathrm{xc}(P_1 \times P_2) \leqslant \mathrm{xc}(P_1) + \mathrm{xc}(P_2) \ .$$

Proof. Let Q_1 and Q_2 be extended formulations of P_1 and P_2, respectively. Then, $Q_1 \times Q_2$ is an extended formulation of $P_1 \times P_2$. Now assume that $Q_1 = \{x \mid Ax \leqslant b\}$ and $Q_2 = \{y \mid Cy \leqslant d\}$ and that these are the smallest extended formulations of P_1 and P_2, resp. Then

$$Q_1 \times Q_2 = \{(x, y) \mid Ax \leqslant b, Cy \leqslant d\} \ .$$

That is, we have an extended formulation of $P_1 \times P_2$ of size at most $\mathrm{xc}(P_1) + \mathrm{xc}(P_2)$. ◀

We are going to define the glued product of polytopes, a slight generalization of the usual product of polytopes. We use a case where the extension complexity of the glued product of two polytopes is upper bounded by the sum of the extension complexities of the two polytopes, and use it in Section 4 to describe a small extended formulation for the MSO polytope $P_\varphi(G)$ on graphs with bounded treewidth.

Let $P \subseteq \mathbb{R}^{d_1+k}$ and $Q \subseteq \mathbb{R}^{d_2+k}$ be 0/1-polytopes defined by m_1 and m_2 inequalities and with vertex sets $\mathrm{vert}(P)$ and $\mathrm{vert}(Q)$, respectively. Let $I_P \subseteq \{1, \dots d_1 + k\}$ be a subset of coordinates of size k, $I_Q \subseteq \{1, \dots d_2 + k\}$ be a subset of coordinates of size k, and let $I'_P = \{1, \dots d_1 + k\} \setminus I_P$. For a vector x, and a subset I of coordinates, we denote by $x|_I$ the subvector of x specified by the coordinates I. The *glued product* of P and Q, (glued) with respect to the k coordinates I_P and I_Q, denoted by $P \times_k Q$, is defined as

$$P \times_k Q = \mathrm{conv}\left(\left\{(x|_{I'_P}, y) \in \mathbb{R}^{d_1+d_2+k} \mid x \in \mathrm{vert}(P), y \in \mathrm{vert}(Q), x|_{I_P} = y|_{I_Q}\right\}\right).$$

We adopt the following convention while discussing glued products in the rest of this article. In the above scenario, we say that $P \times_k Q$ is obtained by gluing P and Q along the k coordinates I_P of P with the k coordinates I_Q of Q. If, for example, these coordinates are named z in P and w in Q, then we also say that P and Q have been glued along the z and w coordinates and we refer to the coordinates z and w as the *glued coordinates*. In the special case that we glue along the last k coordinates, the definition of the glued product simplifies to

$$P \times_k Q = \mathrm{conv}\left(\left\{(x, y, z) \in \mathbb{R}^{d_1+d_2+k} \mid (x, z) \in \mathrm{vert}(P), (y, z) \in \mathrm{vert}(Q)\right\}\right).$$

This notion was studied by Margot [31] who provided a sufficient condition for being able to write the glued product in a specific (and efficient) way from the descriptions of P and Q. We will use this particular way in Lemma 1. The existing work [31, 10], however, is more focused on characterizing exactly when this particular method works. We do not need the result in its full generality and would be interested in a very specific case for our purposes, so we will describe the terms that we will use in our context and then state a specific version of Margot's result.

▶ **Lemma 1** (Gluing lemma). *Let P and Q be 0/1-polytopes and let the k (glued) coordinates in P be labeled $z_1, \dots, z_k$, and the k (glued) coordinates in Q be labeled $w_1, \dots, w_k$. Suppose that $\mathbf{1}^\intercal z \leqslant 1$ is valid for P and $\mathbf{1}^\intercal w \leqslant 1$ is valid for Q. Then $\mathrm{xc}(P \times_k Q) \leqslant \mathrm{xc}(P) + \mathrm{xc}(Q)$.*

As mentioned before, this is a special case of Margot's result, but for completeness we include a proof.

Proof. Let (x', z', y', w') be a point from $P \times Q \cap \{(x, z, y, w) | z = w\}$. Observe that the point (x', z') is a convex combination of points $(x', 0), (x', e_1), \dots, (x', e_k)$ from P with coefficients $(1 - \sum_{i=1}^k z'_i), z'_1, z'_2, \dots, z'_k$ where e_i is the i-th unit vector. Similarly, the point (y', w') is a convex combination of points $(y', 0), (y', e_1), \dots, (y', e_k)$ from Q with coefficients $(1 - \sum_{i=1}^k w'_i), w'_1, w'_2, \dots, w'_k$. Notice that for every $j \in [k]$, (x'_j, e_j, y'_j) is a point from the glued product. As $w_i = z_i$ for every $i \in [k]$, we conclude that $(x', w', z') \in P \times_k Q$. Thus, by Proposition 2 the extension complexity of $P \times_k Q$ is at most that of $P \times Q$ which is at most $\mathrm{xc}(P) + \mathrm{xc}(Q)$ by Proposition **??**. ◀

3.2 Graphs and Treewidth

For notions related to the treewidth of a graph and nice tree decomposition, in most cases we stick to the standard terminology as given in the book by Kloks [22]; the only deviation

is in the leaf nodes of the nice tree decomposition where we assume that the bags are empty. For a vertex $v \in V$ of a graph $G = (V, E)$, we denote by $\delta(v)$ the set of neighbors of v in G, that is, $\delta(v) = \{u \in V \mid \{u, v\} \in E\}$.

A *tree decomposition* of a graph $G = (V, E)$ is a tree T in which each *node* $a \in T$ has an assigned set of vertices $B(a) \subseteq V$ (called a *bag*) such that $\bigcup_{a \in T} B(a) = V$ with the following properties:

- for any $uv \in E$, there exists a node $a \in T$ such that $u, v \in B(a)$.
- if $v \in B(a)$ and $v \in B(b)$, then $v \in B(c)$ for all c on the path from a to b in T.

The *treewidth* $tw(T)$ *of a tree decomposition* T is the size of the largest bag of T minus one. The *treewidth* $tw(G)$ *of a graph* G is the minimum treewidth over all possible tree decompositions of G.

A *nice tree decomposition* is a tree decomposition with one special node r called the *root* in which each node is one of the following types:

- *Leaf node*: a leaf a of T with $B(a) = \emptyset$.
- *Introduce node*: an internal node a of T with one child b for which $B(a) = B(b) \cup \{v\}$ for some $v \in B(a)$.
- *Forget node*: an internal node a of T with one child b for which $B(a) = B(b) \setminus \{v\}$ for some $v \in B(b)$.
- *Join node*: an internal node a with two children b and c with $B(a) = B(b) = B(c)$.

For a vertex $v \in V$, we denote by $top(v)$ the topmost node of the nice tree decomposition T that contains v in its bag. For any graph G on n vertices, a nice tree decomposition of G with at most $8n$ nodes can be computed in time $\mathcal{O}(n)$ [4, 22].

Given a graph $G = (V, E)$ and a subset of vertices $\{v_1, \ldots, v_d\} \subseteq V$, we denote by $G[v_1, \ldots, v_d]$ the subgraph of G induced by the vertices $v_1, \ldots, v_d$. Given a tree decomposition T and a node $a \in V(T)$, we denote by T_a the subtree of T rooted in a, and by G_a the subgraph of G induced by all vertices in bags of T_a, that is, $G_a = G[\bigcup_{b \in V(T_a)} B(b)]$. Throughout this paper we assume that for every graph, its vertex set is a subset of $\mathbb{N}$. We define the following operator σ: for any set $U = \{v_1, v_2, \ldots, v_l\} \subseteq \mathbb{N}$, $\sigma(U) = (v_{i_1}, v_{i_2}, \ldots, v_{i_l})$ such that $v_{i_1} < v_{i_2} \cdots < v_{i_l}$.

For an integer $m \geq 0$, an $[m]$-*colored graph* is a pair $(G, \vec{V})$ where $G = (V, E)$ is a graph and $\vec{V} = (V_1, \ldots, V_m)$ is an m-tuple of subsets of vertices of G called an m-*coloring of* G. For integers $m \geq 0$ and $\tau \geq 0$, an $[m]$-*colored* τ-*boundaried graph* is a triple $(G, \vec{V}, \vec{p})$ where $(G, \vec{V})$ is an $[m]$-colored graph and $\vec{p} = (p_1, \ldots, p_\tau)$ is a τ-tuple of vertices of G called a *boundary* of G. If the tuples $\vec{V}$ and $\vec{p}$ are clear from the context or if their content is not important, we simply denote an $[m]$-colored τ-boundaried graph by $G^{[m],\tau}$. For a tuple $\vec{p} = (p_1, \ldots, p_\tau)$, we denote by p the corresponding set, that is, $p = \{p_1, \ldots, p_\tau\}$.

Two $[m]$-colored τ-boundaried *graphs* $(G_1, \vec{V}, \vec{p})$ and $(G_2, \vec{U}, \vec{q})$ are *compatible* if the function $h : \vec{p} \to \vec{q}$, defined by $h(p_i) = q_i$ for each i, is an isomorphism of the induced subgraphs $G_1[p_1, \ldots, p_\tau]$ and $G_2[q_1, \ldots, q_\tau]$, and if for each i and j, $p_i \in V_j \Leftrightarrow q_i \in U_j$.

Given two compatible $[m]$-colored τ-boundaried graphs $G_1^{[m],\tau} = (G_1, \vec{U}, \vec{p})$ and $G_2^{[m],\tau} = (G_2, \vec{W}, \vec{q})$, the *join* of $G_1^{[m],\tau}$ and $G_2^{[m],\tau}$, denoted by $G_1^{[m],\tau} \oplus G_2^{[m],\tau}$, is the $[m]$-colored τ-boundaried graph $G^{[m],\tau} = (G, \vec{V}, \vec{p})$ where

- G is the graph obtained by taking the disjoint union of G_1 and G_2, and for each i, identifying the vertex p_i with the vertex q_i and keeping the label p_i for it;
- $\vec{V} = (V_1, \ldots, V_m)$ with $V_j = U_j \cup W_j$ and every q_i replaced by p_i, for each j;
- $\vec{p} = (p_1, \ldots, p_\tau)$ with p_i being the node in $V(G)$ obtained by the identification of $p_i \in V(G_1)$ and $q_i \in V(G_2)$, for each i.

Because of the choice of referring to the boundary vertices by their names in $G_1^{[m],\tau}$, it does not always hold that $G_1^{[m],\tau} \oplus G_2^{[m],\tau} = G_2^{[m],\tau} \oplus G_1^{[m],\tau}$; however, the two structures are isomorphic and equivalent for our purposes (see below).

3.3 Monadic Second Order Logic and Types of Graphs

In most cases, we stick to standard notation as given by Libkin [30].

A *vocabulary* σ is a finite collection of *constant* symbols $c_1, c_2, \ldots$ and *relation* symbols $P_1, P_2, \ldots$. Each relation symbol P_i has an associated arity r_i. A *σ-structure* is a tuple $\mathcal{A} = (A, \{c_i^{\mathcal{A}}\}, \{P_i^{\mathcal{A}}\})$ that consists of a universe A together with an interpretation of the constant and relation symbols: each constant symbol c_i from σ is associated with an element $c_i^{\mathcal{A}} \in A$ and each relation symbol P_i from σ is associated with an r_i-ary relation $P_i^{\mathcal{A}} \subseteq A^{r_i}$.

To give an example, a graph $G = (V, E)$ can be viewed as a σ_1-structure $(V, \emptyset, \{E\})$ where E is a symmetric binary relation on $V \times V$ and the vocabulary σ_1 contains a single relation symbol. Alternatively, for another vocabulary σ_2 containing three relation symbols, one of arity two and two of arity one, one can view a graph $G = (V, E)$ also as a σ_2-structure $I(G) = (V_I, \emptyset, \{E_I, L_V, L_E\})$, with $V_I = V \cup E$, $E_I = \{\{v, e\} \mid v \in e, e \in E\}$, $L_V = V$ and $L_E = E$; we will call $I(G)$ the *incidence graph* of G. In our approach we will make use of the well known fact that the treewidths of G and $I(G)$, viewed as a σ_1- and σ_2- structures as explained above, differ by one at most [24].

The main subject of this paper are formulas for graphs in monadic second order logic (MSO) which is an extension of first order logic that allows quantification over monadic predicates (i.e., over sets of vertices). By MSO_2 we denote the extension of MSO that allows in addition quantification over sets of edges. As every MSO_2 formula φ over σ_1 can be turned into an MSO_1 formula φ' over σ_2 such that for every graph G, $G \models \varphi$ if and only if $I(G) \models \varphi'$ [folklore], for the sake of presentation we restrict our attention, without loss of generality, to MSO_1 formulae over the σ_2 vocabulary. To further simplify the presentation, without loss of generality (cf. [21]) we assume that the input formulae are given in a variant of MSO_1 that uses only set variables (and no element variables).

An important kind of structures that are necessary in the proofs in this paper are the $[m]$-colored τ-boundaried graphs. An $[m]$-colored τ-boundaried graph $G = (V, E)$ with boundary $p_1, \ldots, p_\tau$ colored with $V_1, \ldots, V_m$ is viewed as a structure $(V_I, \{p_1, \ldots, p_\tau\}, \{E_I, L_V, L_E, V_1, \ldots, V_m\})$; for notational simplicity, we stick to the notation $G^{[m],\tau}$ or $(G, \vec{V}, \vec{p})$. The corresponding vocabulary is denoted by $\sigma_{m,\tau}$.

A variable X is *free* in φ if it does not appear in any quantification in φ. If $\vec{X}$ is the tuple of all free variables in φ, we write $\varphi(\vec{X})$. A variable X is *bound* in φ if it is not free. By $qr(\varphi)$ we denote the *quantifier rank* of φ which is the number of quantifiers of φ when transformed into the prenex form (i.e., all quantifiers are at the beginning of the formula). We denote by $\mathrm{MSO}[k, \tau, m]$ the set of all MSO_1 formulae φ over the vocabulary $\sigma_{\tau,m}$ with $qr(\varphi) \leq k$.

Two $[m]$-colored τ-boundaried graphs $G_1^{[m],\tau}$ and $G_2^{[m],\tau}$ are $\mathrm{MSO}[k]$-*elementarily equivalent* if they satisfy the same $\mathrm{MSO}[k, \tau, m]$ formulae; this is denoted by $G_1^{[m],\tau} \equiv_k^{MSO} G_2^{[m],\tau}$. The main tool in the model theoretic approach to Courcelle's theorem, that will also play a crucial role in our approach, can be stated as the following theorem.

▶ **Theorem 2** (follows from Proposition 7.5 and Theorem 7.7 [30]). *For any fixed $\tau, k, m \in \mathbb{N}$, the equivalence relation $\equiv_k^{MSO}$ has a finite number of equivalence classes.*

Let us denote the equivalence classes of the relation $\equiv_k^{MSO}$ by $\mathcal{C} = \{\alpha_1 \ldots, \alpha_w\}$, fixing an ordering such that α_1 is the class containing the empty graph. Note that the size of

$\mathcal{C}$ depends only on k, m and τ, that is, $|\mathcal{C}| = f(k,m,\tau)$ for some computable function f. For a given MSO formula φ with m free variables, we define an *indicator function* $\rho_\varphi : \{1,\ldots,|\mathcal{C}|\} \to \{0,1\}$ as follows: for every i, if there exists a graph $G^{[m],\tau} \in \alpha_i$ such that $G^{[m],\tau} \models \varphi$, we set $\rho_\varphi(i) = 1$, and we set $\rho_\varphi(i) = 0$ otherwise; note that if there exists a graph $G^{[m],\tau} \in \alpha_i$ such that $G^{[m],\tau} \models \varphi$, then $G'^{[m],\tau} \models \varphi$ for every $G'^{[m],\tau} \in \alpha_i$.

For every $[m]$-colored τ-boundaried graph $G^{[m],\tau}$, its *type*, with respect to the relation $\equiv_k^{MSO}$, is the class to which $G^{[m],\tau}$ belongs. We say that *types* α_i and α_j are *compatible* if there exist two $[m]$-colored τ-boundaried graphs of types α_i and α_j that are compatible; note that this is well defined as all $[m]$-colored τ-boundaried graphs of a given type are compatible. For every $i \geq 1$, we will encode the type α_i naturally as a binary vector $\{0,1\}^{|\mathcal{C}|}$ with exactly one 1, namely with 1 on the position i.

An important property of the types and the join operation is that the type of a join of two $[m]$-colored τ-boundaried graphs depends on their types only.

▶ **Lemma 3** (Lemma 7.11 [30] and Lemma 3.5 [16]). *Let $G_a^{[m],\tau}$, $G_{a'}^{[m],\tau}$, $G_b^{[m],\tau}$ and $G_{b'}^{[m],\tau}$ be $[m]$-colored τ-boundaried graphs such that $G_a^{[m],\tau} \equiv_k^{MSO} G_{a'}^{[m],\tau}$ and $G_b^{[m],\tau} \equiv_k^{MSO} G_{b'}^{[m],\tau}$. Then $(G_a^{[m],\tau} \oplus G_b^{[m],\tau}) \equiv_k^{MSO} (G_{a'}^{[m],\tau} \oplus G_{b'}^{[m],\tau})$.*

The importance of the lemma rests in the fact that for determination of the type of a join of two $[m]$-colored τ-boundaried graphs, it suffices to know only a *small* amount of information about the two graphs, namely their types. The following two lemmas deal in a similar way with the type of a graph in other situations.

▶ **Lemma 4** (implicitly in [16]). *Let $(G_a, \vec{X}, \vec{p})$, $(G_b, \vec{Y}, \vec{q})$ be $[m]$-colored τ-boundaried graphs and let $(G_{a'}, \vec{X'}, \vec{p'})$, $(G_{b'}, \vec{Y'}, \vec{q'})$ be $[m]$-colored $(\tau+1)$-boundaried graphs with $G_a = (V,E)$, $G_{a'} = (V',E')$, $G_b = (W,F)$, $G_{b'} = (W',F')$ such that*

1. *$(G_a, \vec{X}, \vec{p}) \equiv_k^{MSO} (G_b, \vec{Y}, \vec{q})$;*
2. *$V' = V \cup \{v\}$ for some $v \notin V$, $\delta(v) \subseteq p$, $\vec{p}$ is a subtuple of $\vec{p'}$ and $(G_{a'}[V], \vec{X'}[V], \vec{p'}[V]) = (G_a, \vec{X}, \vec{p})$;*
3. *$W' = W \cup \{w\}$ for some $w \notin W$, $\delta(w) \subseteq q$, $\vec{q}$ is a subtuple of $\vec{q'}$ and $(G_{b'}[W], \vec{Y'}[W], \vec{q'}[W]) = (G_b, \vec{Y}, \vec{q})$;*
4. *$(G_{a'}, \vec{X'}, \vec{p'})$ and $(G_{b'}, \vec{Y'}, \vec{q'})$ are compatible.*

Then $(G_{a'}, \vec{X'}, \vec{p'}) \equiv_k^{MSO} (G_{b'}, \vec{Y'}, \vec{q'})$.

▶ **Lemma 5** (implicitly in [16]). *Let $(G_a, \vec{X}, \vec{p})$, $(G_b, \vec{Y}, \vec{q})$ be $[m]$-colored τ-boundaried graphs and let $(G_{a'}, \vec{X'}, \vec{p'})$, $(G_{b'}, \vec{Y'}, \vec{q'})$ be $[m]$-colored $(\tau+1)$-boundaried graphs with $G_a = (V,E)$, $G_{a'} = (V',E')$, $G_b = (W,F)$, $G_{b'} = (W',F')$ such that*

1. *$(G_{a'}, \vec{X'}, \vec{p'}) \equiv_k^{MSO} (G_{b'}, \vec{Y'}, \vec{q'})$;*
2. *$V \subseteq V'$, $|V'| = |V| + 1$, $\vec{p}$ is a subtuple of $\vec{p'}$ and $(G_{a'}[V], \vec{X'}[V], \vec{p'}[V]) = (G_a, \vec{X}, \vec{p})$;*
3. *$W \subseteq W'$, $|W'| = |W| + 1$, $\vec{q}$ is a subtuple of $\vec{q'}$ and $(G_{b'}[W], \vec{Y'}[W], \vec{q'}[W]) = (G_b, \vec{Y}, \vec{q})$.*

Then $(G_a, \vec{X}, \vec{p}) \equiv_k^{MSO} (G_b, \vec{Y}, \vec{q})$.

3.4 Feasible Types

Suppose that we are given an MSO_1 formula φ over σ_2 with m free variables and a quantifier rank at most k, a graph G of treewidth at most τ, and a nice tree decomposition T of the graph G.

For every node of T we are going to define certain types and tuples of types as *feasible*. For a node $b \in V(T)$ of any kind (leaf, introduce, forget, join) and for $\alpha \in \mathcal{C}$, we say that α is a *feasible type of the node b* if there exist $X_1, \ldots, X_m \subseteq V(G_b)$ such that $(G_b, \vec{X}, \sigma(B(b)))$ is

of type α where $\vec{X} = (X_1, \ldots, X_m)$; we say that $\vec{X}$ *realizes type* α *on the node* b. We denote the set of feasible types of the node b by $\mathcal{F}(b)$.

For an *introduce* node $b \in V(T)$ with a child $a \in V(T)$ (assuming that v is the new vertex), for $\alpha \in \mathcal{F}(a)$ and $\beta \in \mathcal{F}(b)$, we say that (α, β) is a *feasible pair of types for* b if there exist $\vec{X} = (X_1, \ldots, X_m)$ and $\vec{X'} = (X'_1, \ldots, X'_m)$ realizing types α and β on the nodes a and b, respectively, such that for each i, either $X'_i = X_i$ or $X'_i = X_i \cup \{v\}$. We denote the set of feasible pairs of types of the introduce node b by $\mathcal{F}_p(b)$.

For a *forget* node $b \in V(T)$ with a child $a \in V(T)$ and for $\beta \in \mathcal{F}(b)$ and $\alpha \in \mathcal{F}(a)$, we say (α, β) is a *feasible pair of types for* b if there exists $\vec{X}$ realizing β on b and α on a. We denote the set of feasible pairs of types of the forget node b by $\mathcal{F}_p(b)$.

For a *join* node $c \in V(T)$ with children $a, b \in V(T)$ and for $\alpha \in \mathcal{F}(c)$, $\gamma_1 \in \mathcal{F}(a)$ and $\gamma_2 \in \mathcal{F}(b)$, we say that $(\gamma_1, \gamma_2, \alpha)$ is a *feasible triple of types for* c if γ_1, γ_2 and α are mutually compatible and there exist $\vec{X^1}, \vec{X^2}$ realizing γ_1 and γ_2 on a and b, respectively, such that $\vec{X} = (X^1_1 \cup X^2_1, \ldots, X^1_m \cup X^2_m)$ realizes α on c. We denote the set of feasible triples of types of the join node c by $\mathcal{F}_t(c)$.

We define an indicator function $\mu : \mathcal{C} \times V(G) \times \{1, \ldots, m\} \to \{0, 1\}$ such that $\mu(\beta, v, i) = 1$ if and only if there exists $\vec{X} = (X_1, \ldots, X_m)$ realizing the type β on the node $top(v) \in V(T)$ and $v \in X_i$.

4 Extension Complexity of the MSO Polytope

For a given MSO_1 formula $\varphi(\vec{X})$ over σ_2 with m free set variables $X_1, \ldots, X_m$, we define a polytope of satisfying assignments on a given graph G, represented as a σ_2 structure $I(G) = (V_I, \emptyset, \{E_I, L_V, L_E\})$ with domain of size n, in a natural way. We encode any assignment of elements of $I(G)$ to the sets $X_1, \ldots, X_m$ as follows. For each X_i in φ and each v in V_I, we introduce a binary variable y^v_i. We set y^v_i to be one if $v \in X_i$ and zero otherwise. For a given 0/1 vector y, we say that y *satisfies* φ if interpreting the coordinates of y as described above yields a satisfying assignment for φ. The polytope of satisfying assignments, also called the *MSO polytope*, is defined as

$$P_\varphi(G) = \mathrm{conv}\left(\{y \in \{0,1\}^{nm} \mid y \text{ satisfies } \varphi\}\right) .$$

▶ **Theorem 6** (Extension Complexity of the MSO Polytope)**.** *For every graph G and for every MSO_1 formula φ, $\mathrm{xc}(P_\varphi(G)) = f(|\varphi|, \tau) \cdot n$ where f is some computable function, $\tau = tw(G)$ and $n = |V_I|$.*

Proof. Let T be a fixed nice tree decomposition of treewidth τ of the given graph G represented as $I(G)$ and let k denote the quantifier rank of φ and m the number of free variables of φ. Let $\mathcal{C}$ be the set of equivalence classes of the relation $\equiv_k^{MSO}$. For each node b of T we introduce $|\mathcal{C}|$ binary variables that will represent a feasible type of the node b; we denote the vector of them by t_b (i.e., $t_b \in \{0,1\}^{|\mathcal{C}|}$). For each introduce and each forget node b of T, we introduce additional $|\mathcal{C}|$ binary variables that will represent a feasible type of the child (descendant) of b; we denote the vector of them by d_b (i.e., $d_b \in \{0,1\}^{|\mathcal{C}|}$). Similarly, for each join node b we introduce additional $|\mathcal{C}|$ binary variables, denoted by l_b, that will represent a feasible type of the left child of b, and other $|\mathcal{C}|$ binary variables, denoted by r_b, that will represent a feasible type of the right child of b (i.e., $l_b, r_b \in \{0,1\}^{|\mathcal{C}|}$).

We are going to describe inductively a polytope in the dimension given (roughly) by all the binary variables of all nodes of the given nice tree decomposition. Then we show that its extension complexity is small and that a properly chosen face of it is an extension of $P_\varphi(G)$.

First, for each node b of T, depending on its type, we define a polytope P_b as follows:

- b is a *leaf*. P_b consists of a single point $P_b = \{\overbrace{100\ldots0}^{|\mathcal{C}|}\}$.
- b is an *introduce* or *forget* node. For each feasible pair of types $(\alpha_i, \alpha_j) \in \mathcal{F}_p(b)$ of the node b, we create a vector $(d_b, t_b) \in \{0,1\}^{2|\mathcal{C}|}$ with $d_b[i] = t_b[j] = 1$ and all other coordinates zero. P_b is defined as the convex hull of all such vectors.
- b is a *join* node. For each feasible triple of types $(\alpha_h, \alpha_i, \alpha_j) \in \mathcal{F}_t(b)$ of the node b, we create a vector $(l_b, r_b, t_b) \in \{0,1\}^{3|\mathcal{C}|}$ with $l_b[h] = r_b[i] = t_b[j] = 1$ and all other coordinates zero. P_b is defined as the convex hull of all such vectors.

It is clear that for every node b in T, the polytope P_b contains at most $|\mathcal{C}|^3$ vertices, and, thus, by Proposition 1 it has extension complexity at most $\mathrm{xc}(P_b) \leqslant |\mathcal{C}|^3$. Recalling our discussion in Section 3 about the size of $\mathcal{C}$, we conclude that there exists a function f such that for every $b \in V(T)$, it holds that $\mathrm{xc}(P_b) \leqslant f(|\varphi|, \tau)$.

We create an extended formulation for $P_\varphi(G)$ by gluing these polytopes together, starting in the leaves of T and processing T in a bottom up fashion. We create polytopes Q_b for each node b in T recursively as follows:

- If b is a leaf then $Q_b = P_b$.
- If b is an introduce or forget node, then $Q_b = Q_a \times_{|\mathcal{C}|} P_b$ where a is the child of b and the gluing is done along the coordinates t_a in Q_a and d_b in P_b.
- If b is a join node, then we first define $R_b = Q_a \times_{|\mathcal{C}|} P_b$ where a is the left child of b and the gluing is done along the coordinates t_a in Q_a and l_b in P_b. Then Q_b is obtained by gluing R_b with Q_c along the coordinates t_c in Q_c and r_b in R_b where c is the right child of b.

The following lemma states the key property of the polytopes Q_b's.

▶ **Lemma 7.** *For every vertex y of the polytope Q_b there exist $X_1, \ldots, X_m \subseteq V(G_b)$ such that $(G_b, (X_1, \ldots, X_m), \sigma(B(b)))$ is of type α where α is the unique type such that the coordinate of y corresponding to the binary variable $t_b(\alpha)$ is equal to one.*

Proof. The proof is by induction, starting in the leaves of T and going up towards the root. For leaves, the lemma easily follows from the definition of the polytopes P_b's.

For the inductive step, we consider an inner node b of T and we distinguish three cases:

- If b is a join node, then the claim for b follows from the inductive assumptions for the children of b, definition of a feasible triple, definition of the polytope P_b, Lemma 3 and the construction of the polytope Q_b.
- If b is an introduce node or a forget node, respectively, then, analogously, the claim for b follows from the inductive assumption for the child of b, definition of a feasible pair, definition of the polytope P_b, Lemma 4 or Lemma 5, respectively, and the construction of the polytope Q_b.

◀

Let c be the root node of the tree decomposition T. Consider the polytope Q_c. From the construction of Q_c, our previous discussion and the Gluing lemma, it follows that $\mathrm{xc}(Q_c) \leqslant \sum_{b \in V(T)} \mathrm{xc}(P_b) \leqslant f(|\varphi|, \tau) \cdot \mathcal{O}(n)$. It remains to show that a properly chosen face of Q_c is an extension of $P_\varphi(G)$. We start by observing that $\sum_{i=1}^{|\mathcal{C}|} t_c[i] \leq 1$ and $\sum_{i=1}^{|\mathcal{C}|} \rho_\varphi(i) \cdot t_c[i] \leq 1$, where ρ_φ is the indicator function, are valid inequalities for Q_c.

Let Q_φ be the face of Q_c corresponding to the valid inequality $\sum_{i=1}^{|\mathcal{C}|} \rho_\varphi(i) \cdot t_c[i] \leq 1$. Then Q_φ represents those $[m]$-colorings of G for which φ holds. The corresponding feasible assignments of φ on G are obtained as follows: for every vertex $v \in V(G)$ and every

$i \in \{1, \ldots, m\}$ we set $y_i^v = \sum_{j=1}^{|\mathcal{C}|} \mu(\alpha_j, v, i) \cdot t_{top(v)}[j]$. The sum is 1 if and only if there exists a type j such that $t_{top(v)}[j] = 1$ and at the same time $\mu(\alpha_j, v, i) = 1$; by the definition of the indicator function μ in Subsection 3.4, this implies that $v \in X_i$. Thus, by applying the above projection to Q_φ we obtain $P_\varphi(G)$, as desired.

It is worth mentioning at this point that the polytope Q_c depends only on the quantifier rank k of φ and the number of free variables of φ. The dependence on the formula φ itself only manifests in the choice of the face Q_φ of Q_c and its projection to $P_\varphi(G)$. ◀

▶ **Corollary 8.** *The extension complexity of the convex hull of all satisfying assignments of a given MSO_2 formula φ on a given graph G is linear in the size of the graph G.*

5 Efficient Construction of the MSO Polytope

In the previous section we have proven that $P_\varphi(G)$ *has* a compact extended formulation but our definition of feasible tuples and the indicator functions μ and ρ_φ did not explicitly provide a way how to actually *obtain* it efficiently. That is what we do in this section. We also briefly mention some implications of our results for optimization versions of Courcelle's theorem.

As in the previous section we assume that we are given a graph G of treewidth τ and an MSO formula φ with m free variables and quantifier rank k. We start by constructing a nice tree decomposition T of G of treewidth τ in linear time.

Let $\mathcal{C}$ denote the set of equivalence classes of $\equiv_k^{MSO}$. Because $\mathcal{C}$ is finite and its size is independent of the size of G (Theorem 2), for each class $\alpha \in \mathcal{C}$, there exists an $[m]$-colored τ-boundaried graph $(G^\alpha, \vec{X}^\alpha, \vec{p^\alpha})$ of type α whose size is upper-bounded by a function of k, m and τ. For each $\alpha \in \mathcal{C}$, we fix one such graph, denote it by $W(\alpha)$ and call it the *witness of* α. Let $\mathcal{W} = \{W(\alpha) \mid \alpha \in \mathcal{C}\}$. The witnesses make it possible to easily compute the indicator function ρ_φ: for every $\alpha \in \mathcal{C}$, we set $\rho_\varphi(\alpha) = 1$ if and only if $W(\alpha) \models \varphi$, and we set $\rho_\varphi(\alpha) = 0$ otherwise.

▶ **Lemma 9** (implicitly in [16] in the proof of Theorem 4.6 and Corollary 4.7)**.** *The set $\mathcal{W}$ and the indicator function ρ_φ can be computed in time $f(k, m, \tau)$, for some computable function f.*

It will be important to have an efficient algorithmic test for MSO$[k, \tau]$-elementary equivalence. This can be done using the Ehrenfeucht-Fraïssé games:

▶ **Lemma 10** (Theorem 7.7 [30])**.** *Given two $[m]$-colored τ-boundaried graph $G_1^{[m],\tau}$ and $G_2^{[m],\tau}$, it can be decided in time $f(m, k, \tau, |G_1|, |G_2|)$ whether $G_1^{[m],\tau} \equiv_k^{MSO} G_2^{[m],\tau}$, for some computable function f.*

▶ **Corollary 11.** *Recognizing the type of an $[m]$-colored τ-boundaried graph $G^{[m],\tau}$ can be done in time $f(m, k, \tau, |G|)$, for some computable function f.*

Now we describe a linear time construction of the sets of feasible types, pairs and triples of types $\mathcal{F}(b)$, $\mathcal{F}_p(b)$ and $\mathcal{F}_t(b)$ for all relevant nodes b in T. In the initialization phase we construct the set $\mathcal{W}$, using the algorithm from Lemma 7. The rest of the construction is inductive, starting in the leaves of T and advancing in a bottom up fashion towards the root of T. The idea is to always replace a possibly *large* graph $G_a^{[m],\tau}$ of type α by the *small* witness $W(\alpha)$ when computing the set of feasible types for the father of a node a.

Leaf node. For every leaf node $a \in V(T)$ we set $\mathcal{F}(a) = \{\alpha_1\}$. Obviously, this corresponds to the definition in Section 3.

Introduce node. Assume that $b \in V(T)$ is an introduce node with a child $a \in V(T)$ for which $\mathcal{F}(a)$ has already been computed, and $v \in V(G)$ is the new vertex. For every $\alpha \in \mathcal{F}(a)$, we first produce a τ'-boundaried graph $H^{\tau'} = (H^\alpha, \vec{q})$ from $W(\alpha) = (G^\alpha, \vec{X^\alpha}, \vec{p^\alpha})$ as follows: let $\tau' = |\vec{p^\alpha}| + 1$ and H^α be obtained from G^α by attaching to it a new vertex in the same way as v is attached to G_a. The boundary $\vec{q}$ is obtained from the boundary $\vec{p^\alpha}$ by inserting in it the new vertex at the same position that v has in the boundary of $(G_a, \sigma(B(a)))$. For every subset $I \subseteq \{1, \ldots, m\}$ we construct an $[m]$-coloring $\vec{Y}^{\alpha,I}$ from $\vec{X^\alpha}$ by setting $Y_i^{\alpha,I} = X_i^\alpha \cup \{v\}$, for every $i \in I$, and $Y_i^{\alpha,I} = X_i^\alpha$, for every $i \notin I$. Each of these $[m]$-colorings $\vec{Y}^{\alpha,I}$ is used to produce an $[m]$-colored τ'-boundaried graph $(H^\alpha, \vec{Y}^{\alpha,I}, \vec{q})$ and the types of all these $[m]$-colored τ'-boundaried graphs are added to the set $\mathcal{F}(b)$ of feasible types of b, and, similarly, the pairs (α, β) where β is a feasible type of some of the $[m]$-colored τ'-boundaried graph $(H^\alpha, \vec{Y}^{\alpha,I}, \vec{q})$, are added to the set $\mathcal{F}_p(b)$ of all feasible pairs of types of b. The correctness of the construction of the sets $\mathcal{F}(b)$ and $\mathcal{F}_p(b)$ for the node b of T follows from Lemma 4.

Forget node. Assume that $b \in V(T)$ is a forget node with a child $a \in V(T)$ for which $\mathcal{F}(a)$ has already been computed and that the d-th vertex of the boundary $\sigma(B(a))$ is the vertex being forgotten. We proceed in a similar way as in the case of the introduce node. For every $\alpha \in \mathcal{F}(a)$ we produce an $[m]$-colored τ'-boundaried graph $(H^\alpha, \vec{Y}^\alpha, \vec{q})$ from $W(\alpha) = (G^\alpha, \vec{X}^\alpha, \vec{p}^\alpha)$ as follows: let $\tau' = |\vec{p^\alpha}| - 1$, $H^\alpha = G^\alpha$, $\vec{Y}^\alpha = \vec{X}^\alpha$ and $\vec{q} = (p_1, \ldots, p_{d-1}, p_{d+1}, \ldots, p_{\tau'+1})$. For every $\alpha \in \mathcal{F}(a)$, the type β of the constructed graph is added to $\mathcal{F}(b)$, and, similarly, the pairs (α, β) are added to $\mathcal{F}_p(b)$. The correctness of the construction of the sets $\mathcal{F}(b)$ and $\mathcal{F}_p(b)$ for the node b of T follows from Lemma 5.

Join node. Assume that $c \in V(T)$ is a join node with children $a, b \in V(T)$ for which $\mathcal{F}(a)$ and $\mathcal{F}(b)$ have already been computed. For every pair of compatible types $\alpha \in \mathcal{F}(a)$ and $\beta \in \mathcal{F}(b)$, we add the type γ of $W(\alpha) \oplus W(\beta)$ to $\mathcal{F}(c)$, and the triple (α, β, γ) to $\mathcal{F}_t(c)$. The correctness of the construction of the sets $\mathcal{F}(c)$ and $\mathcal{F}_t(c)$ for the node b of T follows from Lemma 3.

It remains to construct the indicator function μ. We do it during the construction of the sets of feasible types as follows. We initialize μ to zero. Then, every time we process a node b in T and we find a new feasible type β of b, for every $v \in B(b)$ and for every i for which d-th vertex in the boundary of $W(\beta) = (G^\beta, \vec{X}, \vec{p})$ belongs to X_i, we set $\mu(\beta, v, i) = 1$ where d is the order of v in the boundary of $(G_b, \sigma(B(b))$. The correctness follows from the definition of μ and the definition of feasible types.

Concerning the time complexity of the inductive construction, we observe, exploiting Corollary 9, that for every node b in T, the number of steps, the sizes of graphs that we worked with when dealing with the node b, and the time needed for each of the steps, depends on k, m and τ only. We summarize the main result of this section in the following theorem.

▶ **Theorem 12.** *Under the assumptions of Theorem 6, the polytope $P_\varphi(G)$ can be constructed in time $f'(|\varphi|, \tau) \cdot n$, for some computable function f'.*

5.1 Courcelle's Theorem and Optimization.

It is worth noting that even though linear time *optimization* versions of Courcelle's theorem are known, our result provides a linear size LP for these problems out of the box. Together with a polynomial algorithm for solving linear programming we immediately get the following:

▶ **Theorem 13.** *Given a graph G on n vertices with treewidth τ, a formula $\varphi \in$ MSO with*

m free variables and real weights w_v^i*, for every* $v \in V(G)$ *and* $i \in \{1, \ldots, m\}$*, the problem*

$$opt \left\{ \sum_{v \in V(G)} \sum_{i=1}^{m} w_v^i \cdot y_v^i \;\middle|\; y \text{ satisfies } \varphi \right\}$$

where opt is min *or* max*, is solvable in time polynomial in the input size.*

6 Acknowledgements

We thank the anonymous reviewers for pointing out existing work and shorter proof of the Glueing lemma, among various other improvements.

References

1 S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, June 1991.
2 D. Avis and H. R. Tiwary. On the extension complexity of combinatorial polytopes. In *Proc. ICALP(1)*, pages 57–68, 2013.
3 D. Bienstock and G. Munoz. LP approximations to mixed-integer polynomial optimization problems. *ArXiv e-prints*, Jan. 2015.
4 H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proc. STOC*, pages 226–234, 1993.
5 H. L. Bodlaender. Treewidth: characterizations, applications, and computations. In *Proc. of WG*, volume 4271 of *LNCS*, pages 1–14. Springer, 2006.
6 G. Braun, S. Fiorini, S. Pokutta, and D. Steurer. Approximation limits of linear programs (beyond hierarchies). *Math. Oper. Res.*, 40(3):756–772, 2015.
7 G. Braun, R. Jain, T. Lee, and S. Pokutta. Information-theoretic approximations of the nonnegative rank. *Electronic Colloquium on Computational Complexity*.
8 A. Buchanan and S. Butenko. Tight extended formulations for independent set, 2014. Available on Optimization Online.
9 M. Conforti, G. Cornuéjols, and G. Zambelli. Extended formulations in combinatorial optimization. *Annals of Operations Research*, 204(1):97–143, 2013.
10 M. Conforti and K. Pashkovich. The projected faces property and polyhedral relations. *Mathematical Programming*, pages 1–12, 2015.
11 B. Courcelle. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.
12 B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique width. In *Proc. of (WG)*, volume 1517 of *LNCS*, pages 125–150, 1998.
13 B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109(1–2):49–82, 1 Mar. 1993.
14 Y. Faenza, S. Fiorini, R. Grappe, and H. R. Tiwary. Extended formulations, nonnegative factorizations, and randomized com. protocols. *Math. Program.*, 153(1):75–94, 2015.
15 S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2):17, 2015.
16 G. Gottlob, R. Pichler, and F. Wei. Monadic datalog over finite structures with bounded treewidth. In *Proc. PODS*, pages 165–174, 2007.
17 B. Grünbaum. *Convex Polytopes*. Wiley Interscience Publ., London, 1967.
18 V. Kaibel. Extended formulations in combinatorial optimization. *Optima*, 85:2–7, 2011.

19 V. Kaibel and A. Loos. Branched polyhedral systems. In *Proc. IPCO*, volume 6080 of *LNCS*, pages 177–190. Springer, 2010.
20 V. Kaibel and K. Pashkovich. Constructing extended formulations from reflection relations. In *Proc. IPCO*, volume 6655 of *LNCS*, pages 287–300. Springer, 2011.
21 L. Kaiser, M. Lang, S. Leßenich, and C. Löding. A Unified Approach to Boundedness Properties in MSO. In *Proc. of CSL*, volume 41 of *LIPIcs*, pages 441–456, 2015.
22 T. Kloks. *Treewidth: Computations and Approximations*, volume 842 of *LNCS*. Springer, 1994.
23 J. Kneis, A. Langer, and P. Rossmanith. Courcelle's theorem - A game-theoretic approach. *Discrete Optimization*, 8(4):568–594, 2011.
24 P. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. In *Proc. PODS*, 1998.
25 P. Kolman and M. Koutecký. Extended formulation for CSP that is compact for instances of bounded treewidth. *Arxiv*, abs/1502.05361, 2015.
26 S. Kreutzer. Algorithmic meta-theorems. In *Proc. of IWPEC*, volume 5018 of *LNCS*, pages 10–12. Springer, 2008.
27 A. Langer, F. Reidl, P. Rossmanith, and S. Sikdar. Practical algorithms for MSO model-checking on tree-decomposable graphs. *Computer Science Review*, 13-14:39–74, 2014.
28 M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.
29 J. R. Lee, P. Raghavendra, and D. Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proc. STOC*, pages 567–576, 2015.
30 L. Libkin. *Elements of Finite Model Theory*. Springer-Verlag, Berlin, 2004.
31 F. Margot. *Composition de polytopes combinatoires: une approche par projection*. PhD thesis, École polytechnique fédérale de Lausanne, 1994.
32 R. K. Martin, R. L. Rardin, and B. A. Campbell. Polyhedral characterization of discrete dynamic programming. *Oper. Res.*, 38(1):127–138, Feb. 1990.
33 M. Sellmann. The polytope of tree-structured binary constraint satisfaction problems. In *Proc. CPAIOR*, volume 5015 of *LNCS*, pages 367–371. Springer, 2008.
34 M. Sellmann, L. Mercier, and D. H. Leventhal. The linear programming polytope of binary constraint problems with bounded tree-width. In *Proc. CPAIOR*, volume 4510 of *LNCS*, pages 275–287. Springer, 2007.
35 F. Vanderbeck and L. A. Wolsey. Reformulation and decomposition of integer programs. In *50 Years of Integer Programming 1958-2008*, pages 431–502. Springer, 2010.
36 L. A. Wolsey. Using extended formulations in practice. *Optima*, 85:7–9, 2011.
37 M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *J. Comput. Syst. Sci.*, 43(3):441–466, 1991.
38 G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, 1995.

H

A GENERALISATION OF EXTENSION COMPLEXITY THAT CAPTURES P

The following article has appeared in *Information Processing Letters* and is included here as an appendix for completeness.

Information Processing Letters 115 (2015) 588–593

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A generalization of extension complexity that captures P

David Avis [a,b], Hans Raj Tiwary [c,*]

[a] GERAD and School of Computer Science, McGill University, 3480 University Street, Montreal, Quebec, H3A 2A7, Canada
[b] Graduate School of Informatics, Kyoto University, Sakyo-ku, Yoshida, Kyoto 606-8501, Japan
[c] Department of Applied Mathematics (KAM) and Institute of Theoretical Computer Science (ITI), Charles University, Malostranské nám. 25, 118 00 Prague 1, Czech Republic

ARTICLE INFO

Article history:
Received 11 April 2014
Accepted 8 February 2015
Available online 17 February 2015
Communicated by R. Uehara

Keywords:
Combinatorial problems
Computational complexity
Theory of computation
Polytopes
Extended formulations
Extension complexity
Lower bounds
Linear programming

ABSTRACT

In this paper we propose a generalization of the extension complexity of a polyhedron Q. On the one hand it is general enough so that all problems in P can be formulated as linear programs with polynomial size extension complexity. On the other hand it still allows non-polynomial lower bounds to be proved for NP-hard problems independently of whether or not $P = NP$. The generalization, called H-free extension complexity, allows for a set of valid inequalities H to be excluded in computing the extension complexity of Q. We give results on the H-free extension complexity of hard matching problems (when H are the odd-set inequalities) and the traveling salesman problem (when H are the subtour elimination constraints).

1. Introduction

Since linear programming is in P, we will not be able to solve an NP-hard problem X in polynomial time (poly-time) by linear programming unless $P = NP$. On the other hand, since linear programming is P-complete, we will not be able to prove a super-polynomial lower bound on solving X by any linear program (LP) without showing that $P \neq NP$. One way to make progress on this problem is to consider restricted versions of linear programming which have two properties.

Property (1): Problems in P will still be solvable in the poly-time even in the restricted version of linear programming.

Property (2): Known NP-hard problems with natural LP formulations will have provable super-polynomial lower bounds under the restricted version of linear programming.

Note that results of type (1) and (2) will still be true, independently of whether or not $P = NP$.

A candidate for such a restricted LP model is extension complexity. In formulating optimization problems as LPs, adding extra variables can greatly reduce the size of the LP [4]. An extension of a polytope is such a formulation that projects onto the original LP formulation of the problem. In this model, LP formulations of some problems in P that have exponential size can be reduced to polynomial size in higher dimensions. For example Martin [9] showed that the minimum spanning tree problem has an extended formulation of size $O(n^3)$ even though its natural formulation requires exponentially many inequalities.

Various authors have shown that extended formulations of various NP-hard problems have exponential lower bounds on their size [14,7,1,11]. However this promising restricted model for LP unfortunately does not satisfy prop-

* Corresponding author.
E-mail addresses: avis@cs.mcgill.ca (D. Avis), hansraj@kam.mff.cuni.cz (H.R. Tiwary).

http://dx.doi.org/10.1016/j.ipl.2015.02.005
0020-0190/

erty (1): Rothvoß [12] recently proved that the matching problem has exponential extension complexity.

Here we propose a stronger version of extension complexity which satisfies property (1). We also exhibit some NP-hard problems that satisfy property (2). In the proposed model we concentrate on the separation problem rather than the polynomial time equivalent optimization problem.

Let Q be a polytope with half-space representation $F(Q)$ and let H be a valid set of inequalities for Q. We delete from $F(Q)$ all half-spaces that are redundant with respect to H and call the resulting (possibly empty) polyhedron Q_H. The *H-free extension complexity of* Q is defined to be the extension complexity of Q_H. If this extension complexity is polynomial and H can be separated in poly-time then we can solve LPs over Q in poly-time. Note that this is true *even if H itself has super-polynomial extension complexity*. On the other hand if Q_H has super-polynomial extension complexity, then even if H can be poly-time separated, any LP that requires an explicit formulation of Q_H will have super-polynomial size. This allows us to strengthen existing results on the extension complexity of NP-hard problems. To illustrate this, we give results on the H-free extension complexity of hard matching problems (when H are the odd-set inequalities) and the traveling salesman problem (when H are the subtour elimination inequalities).

2. Background

We begin by recalling some basic definitions related to extended formulations of polytopes. The reader is referred to [4,6] for more details. An *extended formulation* (EF) of a polytope $Q \subseteq \mathbb{R}^d$ is a linear system

$$Ex + Fy = g,\ y \geqslant \mathbf{0} \tag{1}$$

in variables $(x, y) \in \mathbb{R}^{d+r}$, where E, F are real matrices with d, r columns respectively, and g is a column vector, such that $x \in Q$ if and only if there exists y such that (1) holds. The *size* of an EF is defined as its number of *inequalities* in the system.

An *extension* of the polytope Q is another polytope $Q' \subseteq \mathbb{R}^e$ such that Q is the image of Q' under a linear map. We define the *size* of an extension Q' as the number of facets of Q'. Furthermore, we define the *extension complexity* of Q, denoted by $\mathrm{xc}(Q)$, as the minimum size of any extension of Q.

For a matrix A, let A_i denote the ith row of A and A^j to denote the jth column of A. Let $Q = \{x \in \mathbb{R}^d \mid Ax \leqslant b\} = \mathrm{conv}(V)$ be a polytope, with $A \in \mathbb{R}^{m\times d}$, $b \in \mathbb{R}^m$ and $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$. Then $M \in \mathbb{R}_+^{m\times n}$ defined as $M_{ij} := b_i - A_i v_j$ with $i \in [m] := \{1, \ldots, m\}$ and $j \in [n] := \{1, \ldots, n\}$ is the *slack matrix* of Q w.r.t. $Ax \leqslant b$ and V.

We call the submatrix of M induced by rows corresponding to facets and columns corresponding to vertices the *minimal slack matrix* of Q and denote it by $M(Q)$. Note that the slack matrix may contain columns that correspond to feasible points that are not vertices of Q and rows that correspond to valid inequalities that are not facets of Q, and therefore the slack matrix of a polytope is not a uniquely defined object. However every slack matrix of Q must contain rows and columns corresponding to facet-defining inequalities and vertices, respectively.

As observed in [7], for proving bounds on the extension complexity of a polytope Q it suffices to take any slack matrix of Q. Throughout the paper we refer to the minimal slack matrix of Q as *the* slack matrix of Q and any other slack matrix as *a* slack matrix of Q.

3. *H*-free extensions of polytopes

Let X be some computational problem that can be solved by an LP over a polytope Q. For the applications considered in this paper, it is convenient to consider the case where Q is given by an implicit description of its vertices. So for the matching problem, Q is the convex hull of all 0/1 matching vectors, and for the TSP problem it is the convex hull of all 0/1 incidence vectors of Hamiltonian circuits.

For the given polytope Q let $F(Q)$ be a non-redundant half-space representation. If Q has full dimension $F(Q)$ is unique and each half-space supports a facet of Q. Otherwise we may assume that $F(Q)$ is defined relative to some canonical representation of the linearity space of Q. Our restricted LP model will allow a restricted separation oracle for Q.

Let $H = H(Q)$ be a possibly super-polynomial size set of valid inequalities for Q equipped with an H-separation oracle. We delete from $F(Q)$ all half-spaces that are redundant with respect to H (that is, implied by the inequalities in H) and call the resulting (possibly empty) polyhedron Q_H.

We can solve the separation problem for Q for a point x by first solving it for H and then, if necessary, for Q_H. Suppose x is not in Q. If x is not in H we get a violated inequality by the oracle. Otherwise x must violate a facet of Q_H. We will allow separation for Q_H to be performed using any extension Q'_H of Q_H by explicitly checking the facets of Q'_H for the lifting of x. We call Q'_H an *H-free EF for Q.* Using this separation algorithm and the ellipsoid method we have a way to solve LPs over Q. We call such a restricted method of solving LPs an *H-free LP for Q.* The *H-free extension complexity of* Q is defined to be $\mathrm{xc}(Q_H)$.

We say that an *H-free EF for Q has polynomial size* if:

(a) The H-separation oracle runs in poly-time. That is, there is a poly-time separation algorithm for the polytope defined by the inequalities in H, and
(b) $\mathrm{xc}(Q_H)$ is polynomial in the input size of X.

In this case we also have an H-free LP for Q that can be solved in polynomial time. On the other hand, if for given H, $\mathrm{xc}(Q_H)$ is super-polynomial in the size of X then we say that all H-free LPs for X require super-polynomial size. Note that this statement is independent of whether or not $P = NP$. When H is empty all of the above definitions reduce to standard definitions for EFs and extension complexity.

We illustrate these concepts with a few examples. For the matching problem if H is the set of odd-set inequali-

ties then Q_H is empty. In this case we have an H-free EF for matching of poly-size even though matching has exponential extension complexity.

This example generalizes to show that every problem X in P has a poly-size H-free EF for some H. Indeed, since LP is P-complete, X can be solved by optimizing over a polytope Q. Let H be the entire facet list $F(Q)$ so that Q_H is again empty. Optimization over Q can be performed in poly-time so, by the equivalence of optimization and separation, separation over H can be performed in poly-time also. Therefore (a) and (b) are satisfied as required.

For the TSP, let H be the sub-tour constraints. In this case Q_H is non-empty and in fact we will show in the next section that it has exponential extension complexity. Therefore H-free LPs for the TSP require exponential time, extending the existing extension complexity result for this problem.

We remark that H is an essential parameter here. Matching, for example, has poly-size H-free extension complexity when H are the odd-set inequalities, but not when H is empty. Nevertheless, any problem with poly-size H-free extension complexity for some H can of course be solved in poly-time. For a given hard problem, one gets stronger hardness results by letting H be larger and larger sets of poly-size separable inequalities, as long as one can still prove that Q_H has super-polynomial extension complexity. We give some examples to illustrate this in subsequent sections of the paper.

Before we proceed further, we note an intersection lemma that will be useful for proving lower bounds for H-free extensions of polytopes. This lemma is a polar formulation of the following result of Balas [2]:

Lemma 1. *Let P_1 and P_2 be two polytopes with extension complexity r_1 and r_2 respectively. Then, the extension complexity of $conv(P_1 \cup P_2)$ is at most $r_1 + r_2 + 1$.*

Lemma 2. *Let P_1 and P_2 be two polytopes with extension complexity r_1 and r_2 respectively. Then, the extension complexity of $P_1 \cap P_2$ is at most $r_1 + r_2 + 1$.*

Proof. If the intersection of P_1 and P_2 is not full dimensional, then we can intersect the two polytopes with a suitable affine subspace without increasing their extension complexity. So it suffices to prove this result for the case where $P_1 \cap P_2$ is full dimensional.

Taking the polar dual of the two polytopes with respect to some point in the interior of $P_1 \cap P_2$, we see that $(P_1 \cap P_2)^* = \text{conv}(P_1^* \cup P_2^*)$. Since the slack matrix of the dual of a polytope P is just the transpose of the slack matrix of P (where the dual is obtained with respect to a point in the interior of P), the dual has the same extension complexity. Thus applying Lemma 1 we obtain the desired result. □

4. The travelling salesman problem (TSP)

An undirected TSP instance is defined by a set of integer weights w_{ij}, $1 \le i < j \le n$, for each edge of the complete graph K_n. A tour is a Hamiltonian cycle in K_n defined by a permutation of its vertices. It is required to compute a tour of minimum weight. We define the polytope Q to be the convex hull of the 0/1 incidence vectors $x = (x_{ij} : 1 \le i < j \le n)$ of the tours. It is known that $\text{xc}(Q) = 2^{\Omega(n)}$ [12].

We define H to be the set of *subtour elimination* constraints:

$$\sum_{i,j \in S, i \neq j} x_{ij} \le |S| - 1, \quad S \subseteq \{1, 2, \ldots, n-1\}, \ |S| \ge 2. \tag{2}$$

$$x_{ij} \ge 0, \quad 1 \le i < j \le n \tag{3}$$

It is well known that the subtour elimination constraints can be poly-time separated by using network flows. These constraints by themselves define the convex hull of all forests in K_{n-1} and Martin [9] has given an EF for them that has size $O(n^3)$.

Therefore, $\text{xc}(Q_H) = 2^{\Omega(n)}$, otherwise together with Martin's result and the intersection lemma (Lemma 2), it would imply an upper bound of $2^{o(n)}$ for the travelling salesman polytope. It follows that every H-free LP for the TSP runs in exponential time, where H are the subtour inequalities.

5. Matching problems

A matching in a graph $G = (V, E)$ is a set of edges that do not share any common vertices. Let $n = |V|$. A matching is called perfect if it contains exactly $n/2$ edges. For a given instance G we define the matching polytope Q to be the convex hull of the 0/1 incidence vectors $x = (x_e : e \in E)$ of matchings.

For any $S \subseteq V$ and $e \in E$, we write that $e \in S$ whenever both endpoints of e are in S. Edmonds [5] proved that Q has the following half-space representation:

$$\sum_{e \in S} x_e \le (|S| - 1)/2, \quad S \subseteq V, \ |S| \text{ is odd} \tag{4}$$

$$0 \le x_e \le 1, \quad e \in E. \tag{5}$$

Rothvoß [12] recently proved that $\text{xc}(Q) = 2^{\Omega(n)}$ and that a similar result holds for the convex hull of all perfect matchings. Let H be this half-space representation of Q. Since optimization over Q can be performed in poly-time by Edmonds algorithm there is a poly-time separation algorithm for H. It follows that the matching problem has a poly-size H-free EF.

In the next three subsections we give NP-hard generalizations of the matching problem which have super-polynomial lower bounds on their H-free extension complexity, where H are the odd-set inequalities (4). The method used is similar to that described in detail in [1].

5.1. Induced matchings

A matching in a graph $G = (V, E)$ is called induced if there is no edge in G between any pair of matching edges. Stockmeyer and Vazirani [13] and Cameron [3] proved that the problem of finding a maximum cardinality induced matching is NP-hard. Let Q be the convex hull of the incidence vectors of all induced matchings in G. Let H be the odd set inequalities (4). Clearly H are valid for Q, and as

remarked above, they admit a poly-time separation oracle. We will prove that $\text{xc}(Q_H)$ is super-polynomial. Our proof makes use of the reduction in [3].

Theorem 1. *For every n there exists a bipartite graph G with $O(n)$ edges and vertices such that the induced matching polytope of G has extension complexity $2^{\Omega(\sqrt[4]{n})}$.*

Proof. For every graph $G = (V, E)$ one can construct in polynomial time another graph $G' = (V', E')$ with $|V'| = 2|V|$ and $|E'| = |V| + 28|E|$ such that the stable set polytope of G is the projection of a face of the induced matching polytope of G' [3]. Furthermore, G' is bipartite. Since, for every n there exist graphs with $O(n)$ edges and vertices such that the stable set polytope of the graph has extension complexity $2^{\Omega(\sqrt[4]{n})}$ [1], the result follows. □

Since the above theorem applies to bipartite graphs G, each of the odd-set inequalities (4) is redundant for the induced matching polytope of G. Therefore the H-free extension complexity of the induced matching polytope is super-polynomial in the worst case.

Although this example offers an example H-free extension complexity, it suffers from one obvious weakness. For every graph, all of the inequalities in H are redundant with respect to Q even for non-bipartite graphs! A graph is called *hypomatchable* if the deletion of any vertex yields a graph with a perfect matching. Pulleyblank proved in 1973 (see [8]) that facet-inducing inequalities in (4) correspond to subsets S that span 2-connected hypomatchable subgraphs of G. Let x be the incidence vector for any matching M in G that satisfies such an inequality as an equation. Since S spans a 2-connected subgraph, M cannot be an induced matching.

In order to avoid such trivial cases it is desirable that most, if not all, inequalities of H define facets for at least one polytope Q that corresponds to some instance of the given problem.

5.2. Maximal matchings

A matching in a graph $G = (V, E)$ is called maximal if its edge set is not included in a larger matching. Rather naturally, we will call the convex hull of the characteristic vectors of all maximal matchings of G the maximal matching polytope of G and denote it by $MM(G)$. It is known that finding the minimum maximal matching is NP-hard [15]. Now we show that for every n there exists a graph with n vertices such that $MM(G)$ has super-polynomial H-free extension complexity where H denotes the set of odd cut inequalities.

For every 3-CNF formula ϕ we call the convex hull of all satisfying assignments the satisfiability polytope of ϕ.

Theorem 2. *For every n there exists a 3-CNF formula in $O(n)$ variables such that the satisfiability polytope has super-polynomial extension complexity. Furthermore, in the formula every variable appears at most twice non-negated and at most once negated.*

Proof. The statement is known to be true without the restriction on the number of occurrences of the literals [7,1]. To impose the restriction that every variable appear at most twice non-negated and at most once negated, once can perform the following simple operations.

For any given 3-CNF formula ϕ construct another formula ψ as follows. For each variable x_i, replace the occurrence of x_i in a clause C_j by the variable x_i^j and the occurrence of $\bar{x}_i$ in a clause C_j by the variable y_i^j. Suppose $x_i^1, \ldots, x_i^k$, $y_i^1, \ldots, y_i^l$ are the variables replacing x_i. We add extra clauses corresponding to the conditions $x_i^j \Longrightarrow x_i^{j+1}$, that is, $\bar{x}_i^j \vee x_i^{j+1}$ for $i = 1, \ldots, k-1$. We also add more clauses corresponding to the conditions $\bar{y}_i^j \Longrightarrow \bar{y}_i^{j+1}$, that is, $y_i^j \vee \bar{y}_i^{j+1}$ for $i = 1, \ldots, l-1$. We add two additional clauses: $\bar{x}_i^k \vee \bar{y}_i^1$ ensures that $x_i^k \Longrightarrow \bar{y}_i^1$ and $y_i^l \vee x_i^1$. ensures that $\bar{y}_i^l \Longrightarrow x_i^1$. Finally since the new clauses contain two literals, they are converted to clauses with three literals each in the usual way: duplicate each clause, add a new variable to one clause and its complement to the other. This gives a 3-CNF formula ψ with the required properties and where the number of variables and clauses is polynomial in the size of ϕ.

It is easy to see that the satisfiability polytope for ϕ is obtained by projecting the satisfiability polytope of ψ along one of the variables x_i^k for each i. Therefore the extension complexity of the satisfiability polytope of ψ is at least as high as that of ϕ and we have a family of 3-CNF formula with the desired restricted occurrences that have super-polynomial extension complexity in the worst case. □

Theorem 3. *For every n there exists a bipartite graph $G = (V_1 \cup V_2, E)$ such that $MM(G)$ has extension complexity super-polynomial in n.*

Proof. For every restricted 3-SAT formula ϕ one can construct, in polynomial time, a bipartite graph G such that the maximal matching polytope $MM(G)$ is an extended formulation of the satisfiability polytope of ϕ. This can be done using the exact same construction used in the NP-hardness proof in [15]. Therefore, the extension complexity of $MM(G)$ is super-polynomial for the formulae used in Theorem 2. □

Again, since the graphs G in the above theorem are bipartite, each of the odd-set inequalities (4) is redundant for the maximal matching polytope of G. Therefore the H-free extension complexity of the induced matching polytope is super-polynomial in the worst case.

This example differs from the example in the previous subsection in that (4) are facet defining for maximum matching polytopes of non-bipartite graphs. Too see this, fix a graph G and odd-set S of its vertices. Pulleyblanks's characterisation [8] states that (4) is facet defining for the matching polytope of G whenever S spans a 2-connected hypomatchable subgraph. The only matchings in G that lie on this facet have precisely $(|S|-1)/2$ edges from the set S and are therefore maximal on S. Each of these matchings can be extended to a maximal matching in G which

appears as a vertex of $MM(G)$. Therefore, provided these extensions do not lie in a lower dimensional subspace and $MM(G)$ is full dimensional, (4) is also facet inducing for $MM(G)$ for the given set S. For example, the odd cycles C_{2k+1}, $k \geq 3$ with the addition of a chord cutting off a triangle are a family of such graphs.

5.3. Edge disjoint matching and perfect matching

Given a bipartite graph $G(V_1 \cup V_2, E)$ and a natural number k, it is NP-hard to decide whether G contains a perfect matching M and a matching M' of size k such that M and M' do not share an edge [10].

For a given graph G with n vertices and m edges consider a polytope in the variables $x_1, \ldots, x_m$, $y_1, \ldots, y_m$. For a subset of edges encoding a perfect matching M and a matching M' of size k the we construct a vector with

$$x_i = \begin{cases} 1, & \text{if } e_i \in M \\ 0, & \text{if } e_i \notin M \end{cases}, \quad y_i = \begin{cases} 1, & \text{if } e_i \in M' \\ 0, & \text{if } e_i \notin M' \end{cases}$$

Let us denote the convex hull of all the vectors encoding an edge disjoint perfect matching and a matching of size at least k as $MPM(G,k)$. We would like to remark that one can also define a "natural" polytope here without using separate variables for a matching and a perfect matching and instead using the characteristic vectors of all subsets of edges that are an edge-disjoint union of a matching and a perfect matching. However, the formulation that we consider allows different cost functions to be applied to the matching and the perfect matching.

Now we show that for every n there exists a bipartite graph G with n vertices and a constant $0 < c < \frac{1}{2}$ such that $MPM(G, cn)$ has extension complexity super-polynomial in n. We will use the same reduction as in [10], which is a reduction from MAX-2-SAT. So we first prove a super-polynomial lower bound for the satisfiability polytope of 2-SAT formulas.

Theorem 4. *For every n there exists a 2-SAT formula ϕ in n variables such that the satisfiability polytope of ϕ has extension complexity at least $2^{\Omega(\sqrt[4]{n})}$.*

Proof. It was shown in [1] that for every n there exists a graph G with $O(n)$ edges and vertices such that the stable set polytope of G has extension complexity $2^{\Omega(\sqrt[4]{n})}$. Since the stable sets of a graph can be encodes as a 2-SAT formula as: $\bigwedge_{(i,j)\in E} (\bar{x}_i \vee \bar{x}_j)$, we obtain a family of 2-SAT formulas whose satisfiability polytope for extension complexity at least $2^{\Omega(\sqrt[4]{n})}$. □

Note that the 2-SAT instances required in the above theorem are always satisfiable.

Theorem 5. *For every n there exists a bipartite graph G on n vertices and a constant $0 < c < \frac{1}{2}$ such that $MPM(G, cn)$ has super-polynomial extension complexity.*

Proof. The construction in [10] implicitly provides an algorithm that given any 2-SAT formula ϕ with n variables and m clauses constructs a bipartite graph G with $4mn + 4m$ vertices and maximum degree 3 such that for $k = 2mn + s$, there is an assignment of variables that satisfy at least s clauses of ϕ if and only if G has and edge disjoint perfect matching and a matching of size k. Further the satisfiability polytope of ϕ is the projection of $MPM(G,k)$ and so we obtain a family of bipartite graphs with super-polynomial extension complexity. □

Note that for every pair of odd subsets S_1, S_2 of G two odd-set inequalities can be written: one corresponding to the odd-set inequalities for perfect matching polytope on variables x_i, and the other corresponding to the odd-set inequalities for matching polytope on variables y_i. For a subset of vertices S, let $\delta(S)$ denote the subset of edges with exactly one endpoint in S. The two sets of inequalities are:

$$\sum_{e\in\delta(S_1)} x_e \geqslant 1, \qquad S_1 \subseteq V,\ |S_1| \text{ is odd} \tag{6}$$

$$\sum_{e\in S_2} y_e \leqslant \frac{|S_2|-1}{2}, \quad S_2 \subseteq V,\ |S_2| \text{ is odd} \tag{7}$$

Again the graphs G in the above theorem are bipartite so each of the odd-set inequalities (6), (7) is redundant for $MPM(G)$. Therefore taking H to be the set of these inequalities we have that the H-free extension complexity of the these polytopes is super-polynomial in the worst case.

6. Concluding remarks

We have proposed a generalization of extended formulations and extension complexity which allows partial use of an oracle to separate valid inequalities from a specified set H. Our restricted LP model allows use of the oracle and an explicit half-space representation of the remaining non-redundant inequalities Q_H. In this restricted LP model, all problems in P are solvable in poly-time even if H itself has super-polynomial extension complexity. On the other hand, if $\text{xc}(Q_H)$ is super-polynomial then so is the running time of any LP in our restricted model.

This model allows for progressively stronger lower bounds as more valid inequalities are included in the set H. For example, for the TSP, it would be of interest to include poly-time separable comb inequalities along with the subtour elimination inequalities in H and see if one can still prove a super-polynomial bound on $\text{xc}(Q_H)$.

Acknowledgements

Research of the first author is supported by a Grant-in-Aid for Scientific Research on Innovative Areas – Exploring the Limits of Computation, MEXT, Japan. Research of the second author is partially supported by the Center of Excellence – Institute for Theoretical Computer Science, Prague (project P202/12/G061 of GA ČR).

References

[1] D. Avis, H.R. Tiwary, On the extension complexity of combinatorial polytopes, in: ICALP (1), 2013, pp. 57–68.

[2] E. Balas, Disjunctive programming and a hierarchy of relaxations for discrete optimization problems, SIAM J. Algebr. Discrete Methods 6 (3) (1985) 466–486.
[3] K. Cameron, Induced matchings, Discrete Appl. Math. 24 (1–3) (1989) 97–102.
[4] M. Conforti, G. Cornuéjols, G. Zambelli, Extended formulations in combinatorial optimization, 4OR 8 (2010) 1–48.
[5] J. Edmonds, Maximum matching and a polyhedron with 0, 1 vertices, J. Res. Natl. Bur. Stand. 69 B: 125–130 (1965).
[6] Y. Faenza, S. Fiorini, R. Grappe, H.R. Tiwary, Extended formulations, nonnegative factorizations, and randomized communication protocols, in: ISCO, 2012, pp. 129–140.
[7] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, R. de Wolf, Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds, in: STOC, 2012, pp. 95–106.
[8] L. Lovász, M. Plummer, Matching Theory, Akadémiai Kiadó, Budapest, 1986, also published as Vol. 121 of the North-Holland Mathematics Studies, North-Holland Publishing, Amsterdam.
[9] R. Martin, Using separation algorithms to generate mixed integer model reformulations, Oper. Res. Lett. 10 (3) (Apr. 1991) 119–128.
[10] D. Pálvölgyi, Partitioning to three matchings of given size is NP-complete for bipartite graphs, Technical Report QP-2013-01, Egerváry Research Group, Budapest, 2013.
[11] S. Pokutta, M.V. Vyve, A note on the extension complexity of the knapsack polytope, Inf. Process. Lett. 41 (4) (2013) 347–350.
[12] T. Rothvoß, The matching polytope has exponential extension complexity, arXiv:1311.2369 [CoRR], 2013.
[13] L.J. Stockmeyer, V.V. Vazirani, NP-completeness of some generalizations of the maximum matching problem, Inf. Process. Lett. 15 (1) (Aug. 1982) 14–19.
[14] M. Yannakakis, Expressing combinatorial optimization problems by linear programs, J. Comput. Syst. Sci. 43 (3) (1991) 441–466.
[15] M. Yannakakis, F. Gavril, Edge dominating sets in graphs, SIAM J. Appl. Math. 38 (1980) 364–372.

I

ON THE $\mathcal{H}$-FREE EXTENSION COMPLEXITY OF THE TSP

The following article has been accepted for publication in *Optimization Letters* and is in press. It is included here as an appendix for completeness.

Optim Lett
DOI 10.1007/s11590-016-1029-1

ORIGINAL PAPER

On the $\mathcal{H}$-free extension complexity of the TSP

David Avis[1] · **Hans Raj Tiwary**[2]

Received: 29 June 2015 / Accepted: 16 March 2016

Abstract It is known that the extension complexity of the TSP polytope for the complete graph K_n is exponential in n even if the subtour inequalities are excluded. In this article we study the polytopes formed by removing other subsets $\mathcal{H}$ of facet-defining inequalities of the TSP polytope. In particular, we consider the case when $\mathcal{H}$ is either the set of blossom inequalities or the simple comb inequalities. These inequalities are routinely used in cutting plane algorithms for the TSP. We show that the extension complexity remains exponential even if we exclude these inequalities. In addition we show that the extension complexity of polytope formed by all comb inequalities is exponential. For our proofs, we introduce a subclass of comb inequalities, called (h, t)-uniform inequalities, which may be of independent interest.

Keywords Traveling salesman polytope · Extended formulations · Comb inequalities · Lower bounds

1 Introduction

A polytope Q is called an extended formulation or an extension of polytope P if P can be obtained as a projection of Q. Extended formulations are of natural interest in combinatorial optimization because even if P has a large number of facets and vertices, there may exist a small extended formulation for it, allowing one to optimize a linear

✉ Hans Raj Tiwary
hansraj@kam.mff.cuni.cz

David Avis
avis@i.kyoto-u.ac.jp

[1] Graduate School of Informatics, Kyoto University, Sakyo-ku, Yoshida, Kyoto 606-8501, Japan

[2] KAM/ITI, Charles University, Malostranské nám. 25, 118 00 Prague 1, Czech Republic

Published online: 30 March 2016

function over P indirectly by optimizing instead over Q. Indeed, many polytopes of interest admit small extended formulations (see [3], for example, for a survey).

Recent years have seen many strong lower bounds on the size of extended formulations. In particular, Fiorini et al. [5] showed superpolynomial lower bounds for polytopes related to the MAX-CUT, TSP, and Independent Set problems. This was extended to more examples of polytopes related to other NP-hard problems having superpolynomial lower bounds [1,9]. Even though these results are remarkable, they are hardly surprising since existence of a small extension for any of these polytopes would have extremly unexpected consequences in complexity theory.

Subsequently, Rothvoß showed that the perfect matching polytope of Edmonds does not admit a polynomial sized extended formulation [10], even though one can separate over it in polynomial time despite the polytope having exponentially many vertices and facets. To reconcile this apparent lack of power of compact extended formulations to capture even "easy" problems like perfect matching, the authors of this article introduced the notion of $\mathcal{H}$-free extended formulations [2].

Intuitively, in this setting, given a polytope P (presumably with a high extension complexity) and a set of valid inequalities $\mathcal{H}$, one would like to understand the extent to which the inequalities in $\mathcal{H}$ cause a bottleneck in finding a good extended formulation for P. More formally, the $\mathcal{H}$-free extension complexity of a polytope P measures the extension complexity of the polytope formed by removing the inequalities in $\mathcal{H}$ from the facet-defining inequalities of P. Particularly interesting classes of inequalities, for any polytope, are those for which one can construct an efficient separation oracle.

Clearly, in this setting, nothing interesting happens if the inequalities to be "removed" are redundant. In this article, we consider the traveling salesman polytope and study its $\mathcal{H}$-free extension complexity when $\mathcal{H}$ is the set of simple comb inequalities or the set of 2-matching inequalities. Both sets of inequalities form important classes of inequalities for the TSP polytope. Whereas efficient separation algorithms are known for the 2-matching inequalities, no such algorithm is known for comb inequalities, which generalize the set of 2-matching inequalities [6,8].

In this article we identify a parameterized subset of comb inequalities which we call (h, t)-*uniform comb inequalities* where the parameters require a uniform intersection between the handle and all the teeth of the comb. We use these inequalities to show that the intersection of comb inequalities defines a polytope with exponential extension complexity. Furthermore we show that if $\mathcal{H}$ is a set of valid inequalities for the TSP polytope such that $\mathcal{H}$ does not contain the (h, t)-uniform comb inequalities for some values of parameters h and t, then the $\mathcal{H}$-free extension complexity of the TSP polytope on K_n is at least $2^{\Omega(n/t)}$. As corollaries we obtain exponential lower bounds for the $\mathcal{H}$-free extension complexity of the TSP polytope with respect to 2-matching inequalities and simple comb inequalities.

The rest of this article is organized as follows. In the next section we describe the comb and 2-matching inequalities and introduce the (h, t)-uniform comb inequalities. We also introduce the central tool that we use: subdivided prisms of graphs. After a brief motivation for the study of subdivided prisms in Sect. 3, we prove our main lemma in Sect. 4. We show that over suitably subdivided prisms of the complete graph, there exists a canonical way to translate perfect matchings into TSP tours that can be done without regard to any specific comb inequality. This translation, together

with known tools developed in [4] connecting extension complexity with randomized communication protocols gives the desired results for the problems of interest. Finally, we discuss applications of the main result in Sect. 5.

2 Definitions

Let P be a polytope in $\mathbb{R}^d$. The extension complexity of P—denoted by $\mathrm{xc}(P)$—is defined to be the smallest number r such that there exists an extended formulation Q of P with r facets.

Let $G = (V, E)$ be a graph. For any subset S of vertices, we denote the edges crossing the boundary of S by $\delta(S)$. That is, $\delta(S)$ denotes the set of edges $(u, v) \in E$ such that $|S \cap \{u, v\}| = 1$.

The TSP polytope for the complete graph K_n is defined as the convex hull of the characteristic vectors of all TSP tours in K_n, and is denoted by TSP_n. Similary, PM_n denotes the convex hull of all perfect matchings in K_n. We say that any inequality $a^\intercal x \leqslant b$ is valid for a polytope P if every point in P satisfies this inequality. For a point v in P, the slack of v with respect to a valid inequality $a^\intercal x \leqslant b$ is defined to be the nonnegative number $b - a^\intercal v$.

2.1 Comb inequalities for TSP

For a graph $G = (V, E)$, a comb is defined by a subset of vertices H called the handle and a set of subsets of vertices $T_i, 1 \leqslant i \leqslant k$ where k is an odd number at least three. The sets T_i are called the teeth. The handle and the teeth satisfy the following properties:

$$H \cap T_i \neq \emptyset, \tag{1}$$

$$T_i \cap T_j = \emptyset, \quad \forall i \neq j \tag{2}$$

$$H \setminus \bigcup_{i=1}^{k} T_i \neq \emptyset. \tag{3}$$

The following inequality is valid for the TSP polytope of G and is called the comb inequality for the comb defined by handle H and teeth T_i as above.

$$x(\delta(H)) + \sum_{i=1}^{k} x(\delta(T_i)) \geqslant 3k + 1.$$

Grötschel and Padberg [7] showed that every comb inequality defines a facet of TSP_n for each $n \geqslant 6$. It is not known whether separating over comb inequalities is NP-hard, neither is a polynomial time algorithm known.

For a given comb C and a TSP tour T of G, the slack between the corresponding comb inequality and T is denoted by $\mathrm{sl}_{\mathrm{comb}}(C, T)$.

2-Matching inequalities

A comb inequality corresponding to a handle H and k teeth T_i is called a 2-matching inequality if each tooth T_i has size exactly two. In particular this means that $|H \cap T_i| = 1$ and $|T_i \setminus H| = 1$ for each $1 \leqslant i \leqslant k$. These inequalities are sometimes also referred to as blossom inequalities. Padberg and Rao [8] gave a polynomial time algorithm to separate over the 2-matching inequalities.

Simple comb inequalities

A comb inequality corresponding to a handle H and k teeth T_i is called a simple comb inequality if $|H \cap T_i| = 1$ or $|T_i \setminus H| = 1$ for each $1 \leqslant i \leqslant k$. Simple comb inequalities contain all the 2-matching inequalities. It is not known whether one can separate over them in polynomial time.

(h, t)-uniform comb inequalities

Let us define a subclass of comb inequalities called (h, t)-*uniform comb inequalities* associated with what we will call (h, t)-*uniform combs* for arbitrary $1 \leqslant h < t$. A comb, with handle H and k teeth T_i, is said be (h, t)-uniform if $|T_i| = t$ and $H \cap T_i = h$, for all $1 \leqslant i \leqslant k$.

2.2 Odd set inequalities for perfect matching

Let V denote the vertex set of K_n. A subset $U \subset V$ is called an odd set if the cardinality of U is odd. For every odd set U the following inequality is valid for the perfect matching polytope PM_n and is called an odd set inequality.

$$x(\delta(U)) \geqslant 1.$$

For a given odd set S and a perfect matching M of K_n, the slack between the corresponding odd set inequality and M is denoted by $\mathrm{sl}_{\mathrm{odd}}(S, M)$.

2.3 t-Subdivided prisms of a graph

A prism over a graph G is obtained by taking two copies of G and connecting corresponding vertices. It is helpful to visualise this as stacking the two copies one over the other and then connecting corresponding vertices in the two copies by a vertical edge. A t-subdivided prism is then obtained by subdividing the vertical edges by putting $t - 2$ extra vertices on them. See Fig. 1 for an example.

Let G be the t-subdivided prism of K_n. Let the vertices of the two copies be labeled $u_1^1, \ldots, u_n^1$ and $u_1^t, \ldots, u_n^t$. As a shorthand we will denote the path $u_i^1, u_i^2, \ldots, u_i^t$ as $u_i^1 \rightsquigarrow u_i^t$. Similarily, $u_i^t \rightsquigarrow u_i^1$ will denote $u_i^t, \ldots, u_i^2, u_i^1$.

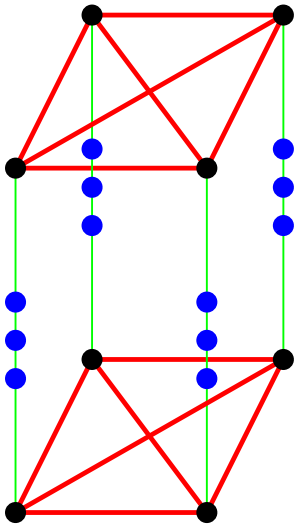


Fig. 1 A 5-subdivided prism over K_4

The graph G has path $u_i^1 \rightsquigarrow u_i^t$ for all $i \in [n]$ and $(u_i^1, u_j^1), (u_i^t, u_j^t)$ for all $i \neq j, i, j \in [n]$. Thus G has tn vertices and $2\binom{n}{2} + (t-1)n$ edges.

3 Motivation

The motivation for looking at t-subdivided prisms stems from a simple observation which we state in the form of a proof of the following proposition:

Proposition 1 *Let $2MP(n)$ be the convex hull of the incidence vectors of all 2-matchings of the complete graph K_n. Then,* $\text{xc}(2MP(n)) \geqslant 2^{\Omega(n)}$.

Proof Let G be a graph with n vertices and m edges and let G' be the 3-subdivided prism of G. G' has $3n$ vertices and $2m + 2n$ edges. Any 2-matching in G' contains all the vertical edges and thus when restricted to a single copy – say the bottom one – of G gives a matching in G. Conversely, any matching in G can be extended to a (not necessarily unique) 2-matching in G'.

Taking G as K_n we obtain a G' that is a subgraph of K_{3n}. The 2-matching polytope of G' lies on a face of the 2-matching polytope of the complete graph on $3n$ vertices (corresponding to all missing edges having value 0). Therefore, the extension complexity of the 2-matching polytope $2MP(n)$ is at least as large as that of the perfect matching polytope. That is, $\text{xc}(2MP(n)) \geqslant 2^{\Omega(n)}$. □

The above generalizes to p-matching polytopes for arbitrary p in the obvious way, and is probably part of folklore.[1]

The generalization of the 3-subdivided prism to larger subdivisions allows us to be able to argue not only about the 2-matching inequalities—which are the facet-defining inequalities for the 2-matching polytope—but also about comb inequalities by using the vertical paths as teeth for constructing combs.

[1] W. Cook (private communication) attributes the same argument to T. Rothvoß.

4 Main tools

4.1 EF-protocols

Given a matrix M, a randomized communication protocol computing M in expectation is a protocol between two players Alice and Bob. The players, having full knowledge of the matrix M, agree upon some strategy. Next, Alice receives a row index i and Bob receives a column index j. Based on their agreed-upon strategy and their respective indices, they exchange a few bits and either one of them outputs a non-negative number, say X_{ij}. For brevity, we will call such protocols EF-protocols. An EF-protocol is said to correctly compute M if for every pair i, j of indices, $\mathbb{E}[X_{ij}] = M_{ij}$, where $\mathbb{E}[X_{ij}]$ is the expected value of the random variable X_{ij}.

The complexity of the protocol is measured by the number of bits exchanged by Alice and Bob in the worst case. It is known that the base-2 logarithm of the extension complexity of any polytope P is equal to the complexity of the best EF-protocol that correctly computes the slack matrix of P [4]. We will use this fact to show our lower bounds by showing that a sublinear EF-protocol for problems of our interest would yield a sublinear EF-protocol for the slack matrix of the perfect matching polytope. First we restate some known results about EF-protocols and extension complexity of perfect matching polytope in a language that will be readily usable to us.

Proposition 2 [4] *Let P be a polytope and $S(P)$ its slack matrix. There exists an EF-protocol of complexity $\Theta(k)$ that correctly computes $S(P)$ if and only if there exists an extended formulation of P of size $2^{\Theta(k)}$.*

Combining lower bounds by Rothvoß [10] with the above mentioned equivalence by Faenza et al. [4], it is easy to see that no sublinear protocol computes the slack matrix of the perfect matching polytope.

Proposition 3 [10] *Any EF-protocol that correctly computes the slack matrix of the perfect matching polytope of K_n requires an exchange of $\Omega(n)$ bits.*

4.2 Uniform combs of odd sets

Let n and t be positive integers. In the rest of the article we will assume that n is a multiple of t. Since we are interested in asymptotic statements only, this does not result in any loss of generality. Let G be the t-subdivided prism of $K_{n/t}$ for some $t \geqslant 2$. Given an odd set S and a perfect matching M in $K_{n/t}$, and arbitrary $1 \leqslant h < t$, we are interested in constructing a comb C and a TSP tour T in K_n such that the following conditions hold:

(C1) C is a (h, t)-uniform comb.
(C2) C depends only on S and 2 edges of M.
(C3) T depends only on M.
(C4) $\mathrm{sl}_{\mathrm{comb}}(C, T) = \mathrm{sl}_{\mathrm{odd}}(S, M)$.

If such a pair (C, T) of a comb and a TSP tour is shown to exist for every pair (S, M) of an odd set and a perfect matching, then we can show that any EF-protocol

for computing the slack $\mathrm{sl}_{\mathrm{comb}}(C, T)$ can be used to construct an EF-protocol for computing $\mathrm{sl}_{\mathrm{odd}}(S, M)$ due to condition (C4). Furthermore, due to conditions (C2) and (C3) the number of bits required for the later protocol will not be much larger than the number of bits required for the former, as C can be locally constructed from S after an exchange of two edges, and T can be locally constructed from M.

Now we show that such a pair does exist if at least two edges of M are contained in S and $|S| \geqslant 5$.

Lemma 1 *Let (S, M) be a pair of an odd set and a perfect matching in $K_{n/t}$, and let $1 \leqslant h < t$. Suppose that $|S| \geqslant 5$, and let $w_1, w_2, w_3, w_4 \in S$ be distinct with (w_1, w_2) and (w_3, w_4) in M. Then, there exists a pair (C, T) of a comb C and a TSP tour T in K_n satisfying the four conditions* (C1)–(C4).

Proof Let $|S| = s$. For simplicity of exposition, we assume that the vertices of S are labeled $w_1, \ldots, w_s$. By w_i^j, we denote the copy of w_i in the j-th layer of the t-subdivided prism over $K_{n/t}$.

The comb C is constructed as follows. The handle H is obtained by taking all vertices in S and the copies $w_1^2, \ldots, w_1^t$ and $w_3^2, \ldots, w_3^t$. For every other vertex $w \in S$ the vertices $w^2, \ldots, w^h$ are also added to H. The teeth T_i are formed by pairing each vertex v in $S \backslash \{w_1, w_3\}$ with its copies $v^2, \ldots, v^t$ producing $s - 2$ teeth. See Fig. 2 for an illustration. Since $s \geqslant 5$ is odd, the number of teeth is odd and at least 3. Thus, the constructed comb is (h, t)-uniform satisfying conditions (C1) and (C2), and the corresponding comb inequality is

$$x(\delta(H)) + \sum_{i=1}^{s-2} x(\delta(T_i)) \geqslant 3(s-2) + 1. \tag{4}$$

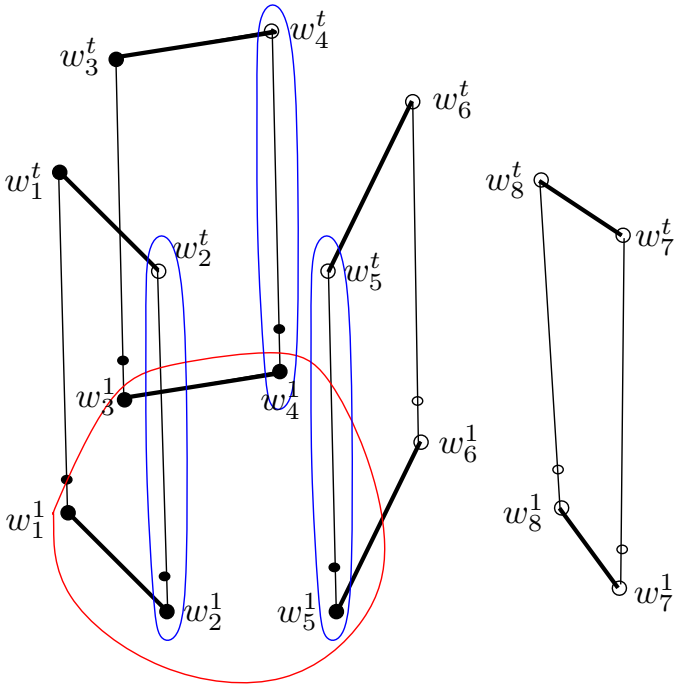


Fig. 2 Construction of a comb from given odd set: the odd set consists of five vertices displayed as *big filled circles in the bottom* copy. The corresponding handle consists of all vertices represented by *filled circles*. The teeth are represented by the *vertical* ellipsoidal enclosures.
The *big circles* represent vertices of the original graph and their *top* copies. The *small circles* represent the hth copy, while the other copies have been omitted here. *Bold edges at the bottom* are matching edges. All other *edges* displayed are just for illustration of the relationship of various copies of vertices

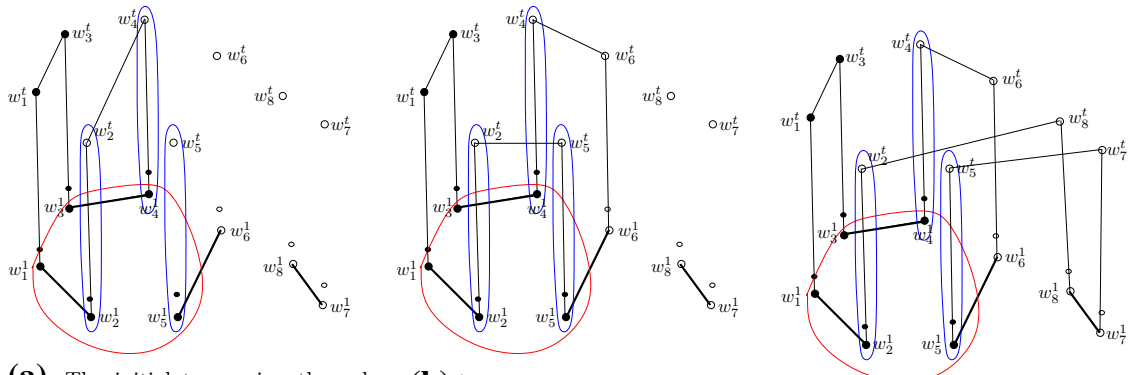
To construct a tour T from the given perfect matching M such that conditions (C3) and (C4) are satisfied, we start with a subtour $(w_1^1 \rightsquigarrow w_1^t, w_3^t \rightsquigarrow w_3^1, w_4^1 \rightsquigarrow w_4^t, w_2^t \rightsquigarrow w_2^1, w_1^1)$. At each stage we maintain a subtour that contains all matching edges on the induced vertices in the lower copy, the edge (w_1^t, w_3^t), and at least one top edge different from (w_1^t, w_3^t). Clearly the starting subtour satisfies these requirements. As long as we have some matching edges in M that are not in our subtour, we pick an arbitrary edge (w_a, w_b) in M and extend our subtour as follows. Select a top edge (w_q^t, w_r^t) different from (w_1^t, w_3^t), remove the edge and add the path $(w_q^t, w_a^t \rightsquigarrow w_a^1, w_b^1 \rightsquigarrow w_b^t, w_r^t)$. The new subtour contains the selected perfect matching edge (w_a^1, w_b^1), the paths $w_a^1 \rightsquigarrow w_a^t$ and $w_b^1 \rightsquigarrow w_b^t$ and has one more top edge distinct from (w_1^t, w_3^t) than in the previous subtour. See Fig. 3 for an example.

At the completion of the procedure, we have a TSP tour that satisfies the following properties:

1. Each edge of M is used in the tour.
2. Each vertical path $w_i^1 \rightsquigarrow w_i^t$ for all $i \in [n]$ is used in the tour.
3. Edge (w_1^t, w_3^t) is used in the tour.

From the construction, edges in $|\delta(H) \cap T|$ are precisely the edges in $|\delta(S) \cap M|$ together with $s-2$ other edges exiting the comb: one through each of the $s-2$ teeth. Therefore, $|\delta(H) \cap T| = |\delta(S) \cap M| + s - 2$. Also, the tour T enters and exits each tooth precisely once so $|\delta(T_i) \cap T| = 2$ for each of the $s-2$ teeth. Substituting these values in the inequality (4), we obtain the slack $\mathrm{sl}_{\mathrm{comb}}(C, T) = |\delta(S) \cap M| + (s - 2) + 2(s - 2) - 3(s - 2) - 1 = \mathrm{sl}_{\mathrm{odd}}(S, M)$. This completes the proof because the pair (C, T) satisfies conditions (C1)–(C4). □

We are finally ready to state the main Lemma of this article. Using the existence of the pair (C, T) as described earlier and the fact that any EF-protocol for the perfect matching polytope requires an exchange of a linear number of bits, we will lower bound the number of bits exchanged by any EF-protocol computing the slack of (h, t)-uniform comb inequalities with respect to TSP tours. In the next Section we will use this Lemma multiple times by fixing different values for the parameters h and t.



(a) The initial tour going through $w_1^1, w_1^t, w_3^t, w_3^1, w_4^1, w_4^t, w_2^t, w_2^1, w_1^1$

(b) Adding a new matching edge (w_5, w_6) by removing (w_2^t, w_4^t)

(c) The final tour

Fig. 3 Constructing a TSP tour from a perfect matching

Lemma 2 *Any EF-protocol computing the slack of (h,t)-uniform comb inequalities with respect to the TSP tours of K_n, requires an exchange of $\Omega(n/t)$ bits. Equivalently, the extension complexity of the polytope of (h,t)-uniform comb inequalities is $2^{\Omega(n/t)}$.*

Proof Due to Proposition 3, it suffices to show if such a protocol uses r bits, then an EF-protocol for the perfect matching polytope for $K_{n/t}$ can be constructed, that uses $r + \mathcal{O}(\log(n/t))$ bits. The protocol for computing the slack of an odd set inequality with respect to a perfect matching in $K_{n/t}$ works as follows.

Suppose Alice has an odd set S in $K_{n/t}$, with $|S| = s$, and Bob has a matching M in $K_{n/t}$. The slack of the odd-set inequality corresponding to S with respect to matching M in the perfect matching polytope for $K_{n/t}$ is $|\delta(S) \cap M| - 1$.

We assume that $s \geqslant 5$. Otherwise, Alice can send the identity of the entire set S with at most $4\log(n/t)$ bits and Bob can output the slack exactly.

Alice first sends an arbitrary vertex $w_1 \in S$, to Bob. Bob replies with the matching vertex of w_1, say w_2. Alice then sends another arbitrary vertex $w_3 \in S$, $w_3 \neq w_2$ to Bob who again replies with the matching vertex for w_3, say w_4. So far the number of bits exchanged is $4\lceil\log(n/t)\rceil$.

Now there are two possibilities: either at least one of the vertices w_2, w_4 is not in S, or both w_2, w_4 are in S. Alice sends one bit to communicate which of the possibilities has occurred and accordingly they switch to one of the two protocols as described next.

In the former case, Alice has identified an edge, say e, in $\delta(S) \cap M$. Now Bob selects an edge e' of his matching uniformly at random (i.e. with probability $2/n$) and sends it to Alice. If e' is in $\delta(S)\backslash\{e\}$, Alice outputs $n/2$. Otherwise, Alice outputs zero. The expected contribution by edges in $(\delta(S) \cap M)\backslash\{e\}$ is then exactly one while the expected contribution of all other edges is zero. Therefore the expected output is $|\delta(S) \cap M| - 1$, and the number of bits exchanged for this step is $\lceil\log m\rceil$ where m is the number of edges in $K_{n/t}$. Thus the total cost in this case is $\mathcal{O}(\log(n/t))$ bits.

In the latter case, the matching edges (w_1, w_2) and (w_3, w_4) lie inside S. Alice constructs a comb C in the t-subdivided prism of $K_{n/t}$, and Bob a TSP tour T in the t-subdivided prism of $K_{n/t}$ such that (C, T) satisfies conditions (C1)–(C4). By Lemma 1 they can do this without exchanging any more bits. Since $\mathrm{sl}_{\mathrm{comb}}(C, T) = \mathrm{sl}_{\mathrm{odd}}(S, M)$, they proceed to compute the corresponding slack with the new inequality and tour, exchanging r bits. The total number of bits exchanged in this case is $r + 4\lceil\log(n/t)\rceil + 1 = r + \mathcal{O}(\log(n/t))$. □

5 Applications

In this section we consider the extension complexity of the polytope of comb inequalities and $\mathcal{H}$-free extension complexity of the TSP polytope when $\mathcal{H}$ is the set of simple comb inequalities. As we will see, the results in this section are obtained by instantiating Lemma 2 with different values of the parameters h and t.

5.1 Extension complexity of Comb inequalities

We show that the polytope defined by the Comb inequalities has high extension complexity.

Theorem 1 *Let COMB(n) be the polytope defined by the intersection of all comb inequalities for* TSP$_n$. *Then* $\mathrm{xc}(COMB(n)) \geqslant 2^{\Omega(n)}$.

Proof Suppose there exists an EF-protocol that computes the slack of COMB(n) that uses r bits. Since $(1, 2)$-uniform comb inequalities are valid for TSP$_n$ we can use the given protocol to compute the slack of these inequalities with respect to the TSP tours of K_n using r bits. Then, using Lemma 2, the slack matrix of the perfect matching polytope for $K_{n/2}$ can be computed using $r + \mathcal{O}(\log n)$ bits. By Proposition 3, this must be $\Omega(n)$. Finally, by Proposition 2 this implies that $\mathrm{xc}(\mathrm{COMB}(n)) \geqslant 2^{\Omega(n)}$. □

5.2 $\mathcal{H}$-free extension complexity

Let $\mathcal{C}_{h,t}$ be the set of (h, t)-uniform comb inequalities for fixed values of h and t. Observe that, since at least three teeth are required to define a comb and the handle must contain some vertex not in any teeth, for (h, t)-uniform combs on n vertices we must have $t \leqslant \left\lfloor \frac{n-1}{3} \right\rfloor$. So for any values of $1 \leqslant h < t \leqslant \left\lfloor \frac{n-1}{3} \right\rfloor$, the set $\mathcal{C}_{h,t}$ is a nonempty set of facet-defining inequalities for TSP$_n$, and for any other values of h and t the set $\mathcal{C}_{h,t}$ is empty.

Theorem 2 *If $\mathcal{H}$ is a set of inequalities valid for the polytope* TSP$_n$, *such that $\mathcal{H} \cap \mathcal{C}_{h,t} = \emptyset$ for some nonempty $\mathcal{C}_{h,t}$, then the $\mathcal{H}$-free extension complexity of* TSP$_n$ *is at least* $2^{\Omega(n/t)}$.

Proof Let $1 \leqslant h < t$ be integers such that $\mathcal{H} \cap \mathcal{C}_{h,t} = \emptyset$. That is, the set $\mathcal{H}$ does not contain any (h, t)-uniform comb inequalities. Let P be the polytope formed from TSP$_n$ by throwing away any facet-defining inequalities that are in $\mathcal{H}$. Then, any EF-protocol computing the slack matrix of P correctly must use $\Omega(n/t)$ bits due to Lemma 2. The claim then follows from Proposition 2. □

The above theorem shows that for every set $\mathcal{H}$ of valid inequalities of TSP$_n$, if the extension complexity of the TSP polytope becomes polynomial after removing the inequalities in $\mathcal{H}$, then $\mathcal{H}$ must contain some inequalities from every (h, t)-uniform comb inequality class, for all $t = o(n/\log n)$. The theorem can easily be made stronger by replacing the requirement $\mathcal{H} \cap \mathcal{C}_{h,t} = \emptyset$ with $|\mathcal{H} \cap \mathcal{C}_{h,t}| \leqslant \mathrm{poly}(n)$. (See the discussion about $\mathcal{H}$-free extension complexity of TSP$_n$ with respect to subtour inequalities in Avis and Tiwary [2] for clarification.)

We can use the above theorem to give lower bounds for $\mathcal{H}$-free extension complexity of the TSP polytope with respect to important classes of valid inequalities by simply demonstrating some class of (h, t)-uniform comb inequalities that has been missed.

2-Matching inequalities

Corollary 1 *Let P be the polytope obtained by removing the 2-matching inequalities from the TSP polytope. Then,* $\mathrm{xc}(P) = 2^{\Omega(n)}$.

Proof The 2-matching inequalities are defined by combs for which each tooth has size exactly two. Therefore the set of $(1, 3)$-uniform combs are not 2-matching inequalities, and Theorem 2 applies. □

Simple comb inequalities

Corollary 2 *Let P is the polytope obtained by removing the set of simple comb inequalities from the TSP polytope. Then,* $\mathrm{xc}(P) = 2^{\Omega(n)}$.

Proof Recall that a comb is called simple if $|H \cap T_i| = 1$ or $|T_i \setminus H| = 1$ for all $1 \leqslant i \leqslant k$ where k is the (odd) number of teeth in the comb and H is the handle. Clearly, $(2, 4)$-uniform combs are not simple and Theorem 2 applies.

As mentioned before, simple comb inequalities define a superclass of 2-matching inequalities and a polynomial time separation algorithm is known for 2-matching inequalities. Althought a similar result was claimed for simple comb inequalities, the proof was apparently incorrect, as pointed out by Fleischer et al. [6]. This latter paper includes a polynomial time separation algorithm for the wider class of simple domino-parity inequalities that we do not consider here.

We leave as an open problem whether there exists a polynomial time separation algorithm for the (h, t)-uniform comb inequalities.

Acknowledgments Research of the first author is supported by a Grant-in-Aid for Scientific Research on Innovative Areas—Exploring the Limits of Computation, MEXT, Japan. Research of the second author is partially supported by GA ČR Grant P202-13/201414.

References

1. Avis, D., Tiwary, H.R.: On the extension complexity of combinatorial polytopes. In: ICALP, pp. 57–68 (2013)
2. Avis, D., Tiwary, H.R.: A generalization of extension complexity that captures P. Inf. Process. Lett. **115**(6–8), 588–593 (2015)
3. Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. 4OR **8**, 1–48 (2010)
4. Faenza, Y., Fiorini, S., Grappe, R., Tiwary, H.R.: Extended formulations, nonnegative factorizations, and randomized communication protocols. In: ISCO, pp. 129–140 (2012)
5. Fiorini, S., Massar, S., Pokutta, S., Tiwary, H.R., de Wolf, R.: Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In: STOC, pp. 95–106 (2012)
6. Fleischer, L., Letchford, A.N., Lodi, A.: Polynomial-time separation of a superclass of simple comb inequalities. Math. Oper. Res. **31**(4), 696–713 (2006)
7. Grötschel, M., Padberg, M.: On the symmetric travelling salesman problem II: lifting theorems and facets. Math. Program. **16**(1), 281–302 (1979)
8. Padberg, M., Rao, M.R.: Odd minimum cut-sets and b-matchings. Math. Oper. Res. **7**, 67–80 (1982)
9. Pokutta, S., Vyve, M.V.: A note on the extension complexity of the knapsack polytope. Oper. Res. Lett. **41**(4), 347–350 (2013)
10. Rothvoß, T.: The matching polytope has exponential extension complexity. In: STOC, pp. 263–272 (2014)

J

POLYNOMIAL SIZE LINEAR PROGRAMS FOR PROBLEMS IN P

The following article is unpublished and is under peer review. It is included here as an appendix for completeness.

Polynomial size linear programs for problems in P

David Avis[1,2] * David Bremner[3] † Hans Raj Tiwary[4] ‡ Osamu Watanabe[5] §

April 2, 2015

Abstract

A perfect matching in an undirected graph $G = (V, E)$ is a set of vertex disjoint edges from E that include all vertices in V. The perfect matching problem is to decide if G has such a matching. Recently Rothvoß proved the striking result that the Edmonds' matching polytope has exponential extension complexity. Here for each $n = |V|$ we describe a perfect matching polytope that is different from Edmonds' polytope and define a weaker notion of extended formulation. We show that the new polytope has a weak extended formulation (WEF) Q of polynomial size. For each graph G with n vertices we can readily construct an objective function so that solving the resulting linear program over Q decides whether or not G has a perfect matching. With this construction, a straightforward $O(n^4)$ implementation of Edmonds' matching algorithm using $O(n^2)$ bits of space would yield a WEF Q with $O(n^6 \log n)$ inequalities and variables. The construction is uniform in the sense that, for each n, a single polytope is defined for the class of all graphs with n nodes. The method extends to solve polynomial time optimization problems, such as the weighted matching problem. In this case a logarithmic (in the weight of the optimum solution) number of optimizations are made over the constructed WEF.

The method described in the paper involves the construction of a compiler that converts an algorithm given in a prescribed pseudocode into a polytope. It can therefore be used to construct a polytope for any decision problem in **P** which can be solved by a well defined algorithm. Compared with earlier results of Dobkin-Lipton-Reiss and Valiant our method allows the construction of explicit linear programs directly from algorithms written for a standard register model, without intermediate transformations. We apply our results to obtain polynomial upper bounds on the non-negative rank of certain slack matrices related to membership testing of languages in **P/Poly**.

Keywords: Polytopes, extended formulation, extension complexity, perfect matching, linear programming, non-negative rank

1 Introduction

A perfect matching in an undirected graph $G = (V, E)$ is a set of vertex disjoint edges from E that include all vertices in V. We let n denote the number of vertices and assume n is even throughout the paper. The perfect matching problem is to determine if G contains a perfect matching and this can be decided in polynomial time by running Edmonds' algorithm [7]. As well as this combinatorial algorithm, Edmonds also introduced a related polytope [8] which we will call the *Edmonds' polytope* EP_n:

$$\mathrm{EP}_n = \mathrm{CH}\{x \in \{0,1\}^{\binom{n}{2}} : x \text{ is the edge-vector of a perfect matching in } K_n\} \tag{1}$$

For any $S \subseteq V$ and edge $ij \in E$, we write that $ij \in \delta(S)$ whenever exactly one of the vertices i and j is

*Email: avis@cs.mcgill.ca
†Email: bremner@unb.ca
‡Email: hansraj@kam.mff.cuni.cz
§Email: watanabe@is.titech.ac.jp
[1]GERAD and School of Computer Science, McGill University, 3480 University Street, Montreal, Quebec, Canada H3A 2A7.
[2]Graduate School of Informatics, Kyoto University, Sakyo-ku, Yoshida Yoshida, Kyoto 606-8501, Japan
[3]Faculty of Computer Science, University of New Brunswick
[4]Department of Applied Mathematics (KAM) and Institute of Theoretical Computer Science (ITI), Charles University, Malostranské nám. 25, 118 00 Prague 1, Czech Republic
[5]Department of Mathematical and Computing Sciences, Tokyo Institute of Technology

in S. Edmonds [8] proved that EP_n has the following halfspace representation:

$$\begin{aligned}
\sum_{ij\in\delta(S)} x_{ij} &\geqslant 1, && S\subseteq V,\ |S|\geqslant 3,\ |S| \textit{ is odd}\\
\sum_{ij\in\delta(i)} x_{ij} &= 1 && i=1,2,...,n\\
0\leqslant\ x_{ij} &\leqslant 1, && 1\leqslant i<j\leqslant n
\end{aligned}$$

This description is exponential in size. Nevertheless, the perfect matching problem can be solved in polynomial time by solving a linear program (LP) over this polytope. Indeed, define an objective function $c^T x = \sum_{1\leqslant i<j\leqslant n} c_{ij}x_{ij}$, where $c_{ij}=1$ if $ij\in E$ and $c_{ij}=0$ otherwise. The LP is:

$$\begin{aligned}
z^* = max\ z &= c^T x\\
x &\in \mathrm{EP}_n
\end{aligned} \tag{2}$$

It is easy to verify that if G has a perfect matching then $z^*=n/2$ otherwise $z^*<n/2$. Since the inequalities defining EP_n can be separated in polynomial time, the LP can be solved in polynomial time [12].

Since the perfect matching problem is in **P**, it seemed possible that EP_n could be written as the projection of a polytope with a polynomial size description. This is the topic of *extension complexity* (see, e.g., Fiorini et al. [10]). We recall the basic definitions here, referring the reader to [10] for further details.

An *extended formulation* (EF) of a polytope $Q\subseteq\mathbb{R}^d$ is a linear system

$$Ex+Fy=g,\ y\geqslant\mathbf{0} \tag{3}$$

in variables $(x,y)\in\mathbb{R}^{q+r}$, where E,F are real matrices with q,r columns respectively, and g is a column vector, such that $x\in Q$ if and only if there exists y such that (3) holds. The *size* of an EF is defined as the number of *inequalities* in the system.

An *extension* of the polytope Q is another polytope $Q'\subseteq\mathbb{R}^e$ such that Q is the image of Q' under a linear map. We define the *size* of an extension Q' as the number of facets of Q'. Furthermore, we define the *extension complexity* of Q, denoted by $\mathrm{xc}(Q)$, as the minimum size of any extension of Q.

Rothvoß [16] recently proved the surprising result that $\mathrm{xc}(\mathrm{EP}_n)$ is exponential. Since extension complexity seemed a promising candidate to obtain computational models that separate problems in **P** from those that are **NP**-hard, this was a setback. A way of strengthening extension complexity to handle this problem was recently proposed by Avis and Tiwary [3].

Dobkin et al. [6] and Valiant [18] showed that linear programming is **P**-complete from which it follows that every problem in **P** has an LP-formulation. We will review this result in Section 3 giving Valiant's construction. Valiant's construction is for the non-uniform circuit model and so, of course, also applies to uniform families of circuits. The main result is of this paper is to give a direct method to produce polynomial size LPs from polynomial time algorithms. Specifically we will construct LPs directly from a polynomial time algorithm expressed in pseudocode that solves a decision problem. Of course a trivial LP formulation can be obtained by first solving the decision problem for a given input and setting $c=1$ if the answer is yes and $c=0$ otherwise. Then solving the one dimensional LP: $max\ cx, 0\leqslant x\leqslant 1$ solves the original problem. To avoid such trivial LPs we limit how much work can be done in constructing the objective function. One such limitation might be, for example, to insist that the objective function can be computed in linear time in terms of the input size. The objective functions we consider in this paper satisfy this condition.

For concreteness, we focus on an explicit construction of a poly-size LP that can be used to solve the perfect matching problem. Firstly we describe another 'natural' polytope, PM_n, for the perfect matching problem. Then we will introduce the notion of a weak extended formulation (WEF). Instead of requiring projection onto PM_n we will simply require that LPs solved over the WEF solve the original problem. The objective function used is basically just a ±1 encoding of the input graph. The approach used is quite general and can be applied to any problem in P for which an explicit algorithm is known. It extends to polynomial time solvable optimization problems also. However in this case a logarithmic (in the weight of the optimum solution) number of optimizations are made over the constructed WEF. Note that when an EF exists *both* the optimization and decision problems can be solved in a single LP optimization. Hence a WEF is indeed weaker than an EF in this sense. We discuss this further in Section 6.

The paper is organized as follows. In the next section we introduce a new polytope for the perfect matching problem and give some basic results about its facet structure. We define the notion of weak extended

formulation and state the main theorem. In Section 3 we first give a simple example to illustrate the technique we use to build extended formulations from boolean circuits. Then we prove the main theorem of the paper. In Section 4 we generalize our method to algorithms given in pseudocode rather than as a circuit. We show how programs written in a simple pseudocode can be converted to WEFs. Our method is modelled on Sahni's proof of Cook's theorem given in [13]. Since our pseudocode is clearly strong enough to implement Edmonds' algorithm in polynomial time, our method gives a poly-size WEF for the perfect matching problem. In Section 5 we use our main theorem to show that the non-negative rank of certain matrices is polynomially bounded above. Finally in Section 6 we give some concluding remarks including a discussion of applying this technique to polynomial time *optimization* problems such as the maximum weighted matching problem.

2 Polytopes for decision problems

2.1 Another perfect matching polytope

We use the notation $\mathbb{1}_t$ to denote the t-vector of all ones, dropping the subscript when it is clear from the context. Let n be an even integer and let x be a binary vector of length $\binom{n}{2}$. We let $G(x) = (V, E)$ denote the graph with edge incidence vector given by x, let n be the number of its vertices and $m = \mathbb{1}^T x$ the number of its edges. Furthermore, let $w_x = 1$ if $G(x)$ has a perfect matching and zero otherwise. We define the polytope PM_n as:

$$\mathrm{PM}_n = \mathrm{CH}\{(x, w_x) : x \in \{0,1\}^{\binom{n}{2}}\} \tag{4}$$

PM_n may be visualized by starting with a hypercube in dimension $\binom{n}{2}$ and embedding it in one higher dimension with extra coordinate w. For vertices of the cube corresponding to graphs with perfect matchings $w = 1$ else $w = 0$. It is easy to see that PM_n has precisely $2^{\binom{n}{2}}$ vertices. EP_n is closely related to PM_n, in fact it forms a face.

Proposition 1. *EP_n is a face of PM_n and can be defined by*

$$\mathrm{EP}_n = \{x : (x, w) \in \mathrm{PM}_n \cap \{\mathbb{1}^T x + (1 - w)n^2 = \frac{n}{2}\ \}\} \tag{5}$$

Proof. We first show that the inequality

$$\mathbb{1}^T x + (1 - w)n^2 \geqslant \frac{n}{2} \tag{6}$$

is valid for PM_n. We need only verify it for the extreme points (x, w_x) given in (4). If $w_x = 0$, (6) holds since $\mathbb{1}^T x + n^2 \geqslant \frac{n}{2}$. Otherwise $w_x = 1$, x is the incidence vector of graph containing a perfect matching, so $\mathbb{1}^T x \geqslant n/2$. The vectors x with $w_x = 1$ and $\mathbb{1}^T x = n/2$ are the incidence vectors of perfect matchings of K_n and are precisely those used to define EP_n in (1). □

For a given input graph $G(\bar{x}) = (V, E)$ we define the vector $c = (c_{ij})$ by:

$$c_{ij} = 1 \quad ij \in E \qquad c_{ij} = -1 \quad ij \notin E \qquad 1 \leqslant i < j \leqslant n \tag{7}$$

and let d be a constant such that $0 < d \leqslant 1/2$. We construct the LP:

$$\begin{aligned} z^* = max\ z &= c^T x + dw \\ (x, w) &\in \mathrm{PM}_n \end{aligned} \tag{8}$$

Proposition 2. *For any edge incidence vector $\bar{x} \in [0,1]^{\binom{n}{2}}$ let $m = \mathbb{1}^T \bar{x}$. The optimum solution to (8) is unique, $z^* = m + d$ if $G(\bar{x})$ has a perfect matching, and $z^* = m$ otherwise.*

Proof. Let c be the objective function defined by (7) and set $m = \mathbb{1}^T \bar{x}$. Note that $c^T \bar{x} = m$ and that $c^T x \leqslant m - 1$ for any other vertex x of the $\binom{n}{2}$-cube. If $G(\bar{x})$ has a perfect matching then $(x, w) = (\bar{x}, 1)$ is a feasible solution to (8) with $z = m + d$. Since $x \neq \bar{x}$, $c^T x + dw \leqslant m - 1 + d$ this is the unique optimum solution.

If $G(\bar{x})$ does not have a perfect matching then $(x, w) = (\bar{x}, 0)$ is a feasible solution to (8) with $z = m$. Consider any other vertex x of the cube. Then $z = c^T x + dw \leqslant m - 1 + 1/2 = m - 1/2$. It follows that $z^* = m$ is the unique optimum solution. □

2.2 Polytopes for decision problems and weak extended formulations

The basic ideas above can be extended to arbitrary polynomial time decision problems. Let X denote a poly-time decision problem defined on binary input vectors $x = (x_1, ..., x_q)$, and an additional bit w_x, where $w_x = 1$ if x results in a "yes" answer and $w_x = 0$ otherwise. We define the polytope P as:

$$P = \text{CH}\{(x, w_x) : x \in \{0,1\}^q\} \tag{9}$$

For a given binary input vector $\bar{x}$ we define the vector $c = (c_j)$ by:

$$c_j = 1 \quad \bar{x}_j = 1 \quad \text{and} \quad c_j = -1 \quad \bar{x}_j = 0 \qquad 1 \leqslant j \leqslant q \tag{10}$$

and let d be a constant such that $0 < d \leqslant 1/2$. As before we construct an LP:

$$\begin{aligned} z^* = max\ z &= c^T x + dw \\ (x, w) &\in P \end{aligned} \tag{11}$$

The following proposition can be proved in an identical way to Proposition 2.

Proposition 3. *For any $\bar{x} \in [0,1]^q$ let $m = \mathbb{1}^T \bar{x}$. The optimum solution to (11) is unique, $z^* = m + d$ if $\bar{x}$ has a "yes" answer and $z^* = m$ otherwise.*

By an n-cube we mean the n-dimensional hypercube whose vertices are the 2^n binary vectors of length n.

Definition 1. Let Q be a polytope which is a subset of the $(q+t)$-cube with variables labeled $x_1, ..., x_q, y_1, ..., y_t$. We say that Q has the *x-0/1 property* if each of the 2^q ways of assigning 0/1 to the x variables uniquely extends to a vertex (x, y) of Q and, furthermore, y is 0/1 valued. Q may have additional fractional vertices.

In polyhedral terms, this says that the intersection of Q with the hyperplanes $x_j = e_j, j = 1, ..., q$ is a 0/1 vertex, for each assignment of zero or one to the e_j's. We will show that we can solve a poly-time decision problem X by replacing P in (8) by a polytope Q of polynomial size, while maintaining the same objective functions. We call Q a *weak extended formulation* as it does not necessarily project onto P.

Definition 2. A polytope

$$Q = \{(x, w, s) : x \in [0,1]^q, w \in [0,1], s \in [0,1]^r, Ax + bw + Cs \leqslant h\}$$

is a *weak extended formulation (WEF)* of P if

- Q has the x-0/1 property.
- For any binary vector $\bar{x} \in [0,1]^q$ let $m = \mathbb{1}^T \bar{x}$. Let c be defined by (10) and let $0 < d \leqslant 1/2$. The optimum solution $z^* = max\ \{c^T x + dw : (x, w, s) \in Q\}$ is unique and takes the value $z^* = m + d$ if $\bar{x}$ has a "yes" answer. Otherwise $z^* < m + d$ and for all sufficiently small d, $z^* = m$ and is unique.

For example, let X be the perfect matching problem so that $P = \text{PM}_n$. Let $Q = Q_n$ be a WEF as given by this definition. It follows from Proposition 2 that we can determine whether an input graph G has a perfect matching by solving an LP over either PM_n or Q_n using the same objective function which is derived directly from the edge adjacency vector of G. As a very simple example, consider $n = 2$ giving $\text{PM}_2 = \text{CH}\{(0,0), (1,1)\}$. A WEF, for example, is given by:

$$Q_2 = \text{CH}\{(0,0,0), (1,1,1), (1/4, 1, 1/2)\}$$

Initially let $d = 1/2$. When $G(\bar{x})$ is an edge, $m = 1$, $c_{12} = 1$ and $z = c^T x + dw$ obtains the same optimum solution of $z^* = 3/2 = m + d$ over both PM_2 and Q_2. When $G(\bar{x})$ is a non-edge, $m = 0$, $c_{12} = -1$ and $z = c^T x + dw$ obtains the optimum solution of $z^* = 0 = m$ over PM_2 and $z^* = 1/4 < 1/2 = m + d$ over Q_2, at the fractional vertex (1/4,1,1/2). However, if $0 < d < 1/4$ then $z = c^T x + dw$ obtains the unique optimum solution of $z^* = 0 = m$ over both PM_2 and Q_2. We see that Q_2 projects onto a triangle in the (x, w)-space, whereas PM_2 is a line segment.

In the next section we prove the following result:

Theorem 1. *Every decision problem X in* **P/poly** *admits a weak extended formulation Q of polynomial size.*

3 From Circuits to Polytopes

In order to show that Linear Programming is **P**-complete, Valiant [18] gave a construction to transform boolean circuits into a linear sized set of linear inequalities with the x-0/1 property (where x_i are the variables corresponding to the inputs of the circuit); a similar construction was used by Yannakakis [19] in the context of the Hamiltonian Circuit problem. In this section we show that Valiant's construction implies Theorem 1. Valiant's point of view is slightly different from ours in that he explicitly fixes the values of the input variables before solving an LP-feasibility problem (as opposed to using different objective functions with a fixed set of inequalities). Showing that the result of this fixing is a 0/1-vertex is precisely our x-0/1 property.

We begin with a standard definition[1]:

Definition 3. A *(boolean) circuit* with q input bits $x = (x_1, x_2, ..., x_q)$ is a directed acyclic graph in which each of its t nodes, called *gates*, is either an AND($\wedge$) gate, an OR($\vee$) or a NOT($\neg$) gate. We label each gate by its output bit. One of these gates is designated as the *output gate* and gives output bit w. The *size* of a circuit is the number of gates it contains and its *depth* is the maximal length of a path from an input gate to the output gate.

For example, the circuit shown in Figure 1 can be used to compute whether or not a graph on 4 nodes has a perfect matching. The input is the binary edge-vector of the graph and the output is $w = 1$ if the graph has a matching (e.g. G_1) or $w = 0$ if it does not (e.g. G_2). If the graph has a perfect matching, exactly one of y_{12}, y_{13} or y_{14} is one, defining the matching. For each gate we have labeled the output bit by a new variable. We will construct a polytope from the circuit by constructing a system of inequalities on the same variables.

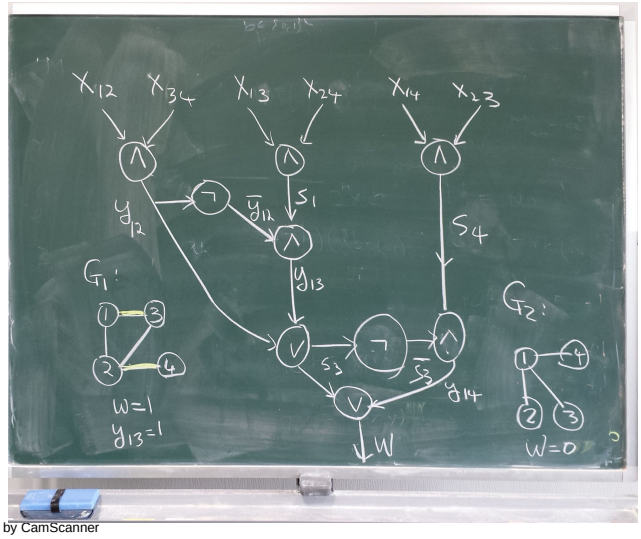


Figure 1: A circuit to compute whether a 4 node graph has a perfect matching

From an AND gate, say $y_{12} = x_{12} \wedge x_{34}$, we generate the inequalities:

$$\begin{array}{rcl} x_{12} + x_{34} - y_{12} & \leqslant & 1 \\ -x_{12} \qquad + y_{12} & \leqslant & 0 \\ -x_{34} + y_{12} & \leqslant & 0 \\ y_{12} & \geqslant & 0 \end{array} \tag{12}$$

The system (12) defines a polytope in three variables whose 4 vertices represent the truth table for the

[1] See, e.g., the text by Savage [17]

AND gate:

x_{12}	x_{34}	y_{12}
0	0	0
0	1	0
1	0	0
1	1	1

Note that the variables x_{12}, x_{34} define a 2-cube and so the polytope is an extension of the 2-cube. In the terminology of the last section, it has the $\{x_{12}, x_{34}\}$-0/1 property.

From an OR gate, say $s_3 = y_{12} \vee y_{13}$, we generate the inequalities:

$$\begin{aligned} -y_{12} - y_{13} + s_3 &\leqslant 0 \\ y_{12} - s_3 &\leqslant 0 \\ y_{13} - s_3 &\leqslant 0 \\ s_3 &\leqslant 1 \end{aligned} \tag{13}$$

The system (13) defines a polytope in three variables whose 4 vertices represent the truth table for the OR gate, as can easily be checked. Indeed, this polytope has the $\{y_{12}, y_{13}\}$-0/1 property.

From a NOT gate, say $\bar{y}_{12} = \neg y_{12}$, we could generate the equation

$$\bar{y}_{12} = 1 - y_{12} \tag{14}$$

However it is equivalent to just replace all instances of $\bar{y}_{12}$ by $1 - y_{12}$ in the inequality system, and this is what we will do in the sequel.

The circuit in Figure 1 contains 5 AND gates and 2 OR gates. By suitably replacing variables in (12) and (13) we obtain a system of 28 inequalities in 13 variables. As just mentioned, the NOT gates are handled by variable substitution rather than explicit equations. Let Q_4 denote the corresponding polytope. It will follow by the general argument below that Q_4 is a weak extended formulation (WEF) of PM_4.

We now show that the above construction can be applied to any boolean circuit C to obtain a polytope Q which has the 0/1 property with respect to the inputs of C.

Lemma 1 ([18])**.** *Let C be a boolean circuit with q input bits $x = (x_1, x_2, ..., x_q)$, t gates labeled by their output bits $y = (y_1, y_2, ..., y_t)$ and with circuit output bit $w = y_t$. Construct the polytope Q with $4t$ inequalities and $q + t$ variables using the systems (12) and (13) respectively. Q has the the x-0/1 property and for every input x the value of w computed by C corresponds to the value of y_t in the unique extension $(x, y) \in Q$ of x.*

Proof. Since C is an acyclic directed graph it contains a topological ordering of its nodes (gates) and we can assume that the labelling $y_1, y_2, ..., y_t$ is such an ordering. Note we can assume $w = y_t$ comes last since it cannot be an input to any other gate. For any given input x the output of the circuit can be obtained by evaluating each gate in the order $y_1, ..., y_t$. Since it is a topological ordering, each input for a gate has been determined before the gate is evaluated.

We proceed by induction. Let Q_k be the polytope defined by the $4k$ inequalities corresponding to gates $y_1, ..., y_k$. The inductive hypothesis is that for $k = 1, 2, ..., t$

- Q_k has the x-0/1 property, and
- for each x the value of y_k calculated by C corresponds to the value of y_k in the unique extension of x in Q_k.

This is clearly true for $k = 1$ as the analysis following (12) and (13) shows. We assume the hypothesis is true for $k = 1, 2, ..., j$, where $1 \leqslant j < t$, and prove it for $j + 1$. Indeed, since Q_j has the x-0/1 property for each x the values of $y_1, ..., y_j$ are uniquely defined and have 0/1 values. By induction they correspond to the values computed by C. Therefore the analysis following (12) and (13) shows that y_{j+1} will also be uniquely defined, 0/1 valued, and will correspond to the value computed by C. This verifies the inductive hypothesis for $j + 1$ and since $Q = Q_t$ the proof is complete. □

Lemma 2. *Let C be a circuit that solves a decision problem X with q input bits $x = (x_1, x_2, ..., x_q)$ and has associated polytope P as defined in (9). The polytope Q constructed in Lemma 1 is a WEF for P.*

Proof. In order to make the correspondence with Definition 2 we relabel the variables in Q, constructed in Lemma 1, so that $s = (y_1, y_2, ..., y_{t-1})$ and $w = y_t$. By Lemma 1 we know Q has the x-0/1 property so it remains to prove the second condition in Definition 2.

Let $\bar{x}$ be any binary q-vector and set $m = \mathbb{1}^T\bar{x}$. Since Q has the x-0/1 property $\bar{x}$ extends to a unique binary vertex $(\bar{x}, \bar{w}, \bar{s})$ of Q. Define c as in (10). Fix some d, $0 < d \leqslant 1/2$ and consider the optimum solution

$$z^* = max\ \{c^T x + dw : (x, w, s) \in Q\}.$$

Since Q has the x-0/1 property the maximum of $c^T x$ over Q is obtained at $c^T\bar{x} = m$ at the unique vertex $(\bar{x}, \bar{w}, \bar{s})$ of Q. For any other $(x, w, s) \in Q$, since x is in the q-cube and not equal to $\bar{x}$, we have $c^T x < m$ and, since $w \leqslant 1$, $z = c^T x + dw < m + d$. Therefore, if $\bar{x}$ has a "yes" answer then $w = 1$, $z^* = m + d$ and $(\bar{x}, \bar{w}, \bar{s})$ is the unique optimum solution.

If $\bar{x}$ has a "no" answer then $z = c^T\bar{x} + d\bar{w} = m$, since $\bar{w} = 0$. In this case the optimum z^* may be obtained at a fractional vertex (x, w, s). But then, as observed, $c^T x < m$. Since $w \leqslant 1$ we must have $z^* = c^T x + dw = m + d - \epsilon_{\bar{x}}$, for some $\epsilon_{\bar{x}} > 0$. So if $d < \epsilon_{\bar{x}}$ then $z^* = m$ is unique and obtained at $(\bar{x}, \bar{w}, \bar{s})$. By choosing $d < min\{\epsilon_x\ :\ x \in \{0,1\}^q\}$ we obtain the final part of the second condition of Definition 2. The lemma follows. We remark that a suitable value of d (with a polynomial sized encoding) can be computed using Hadamard bounds on determinants. □

Theorem 1 follows from Lemmas 1 and 2. Since there is no limitation of uniformity on the circuits used, the theorem holds for all decision problems in **P/poly**. Since each gate in the circuit gives rise to 4 inequalities and one new variable, we have the following corollary.

Corollary 1. *Let X be a decision problem with corresponding polytope P defined by (9). A set of circuits for X with size $p(n)$ generate a WEF Q for P with $4p(n)$ inequalities and variables.*

In this section we showed how to construct a poly-size LP from a poly-size circuit so that the optimum solution of the LP gives the output of the circuit. However it is not immediately clear how to use this to obtain a polytope for the perfect matching problem. It would be required to convert Edmonds' algorithm to a family of circuits. In the next section we bypass this step by showing how to convert a simple pseudocode directly into a polytope without first computing a circuit (See Theorem 2). This can be used to convert poly-time algorithms into poly-size LPs directly.

We would like to remark that our construction in Theorem 2 of a WEF from a pseudocode may not be optimal. For example, it would be possible to get roughly $O(T(n)\log T(n))$- size circuits simulating a given $T(n)$-time bounded Turing machine (see, e.g., Chapter 1 of [1]) from which we can construct a WEF with $O(T(n)\log T(n))$ inequalities. But since Turing Machines are not commonly used for designing algorithms, we leave the interested reader to check whether a similar idea can be used to define a WEF with smaller size.

4 Constructing an LP from pseudocode

In this section we introduce a rudimentary pseudocode that can be used for decision problems. This pseudocode follows the usual practices of specifying algorithms and the tradition of so-called *register machines* (see e.g. [5]). We show how the code can be translated into a linear program, in a way similar to that shown for circuits in the previous section. We assume there is a polynomial function $p(n)$ so that the pseudocode terminates within $p(n)$ steps for any input of size n. We will show that the corresponding LP will also have polynomial size in n.

The pseudocode we use and its translation into an LP is adapted from a proof of Cook's theorem given in [13] which is attributed to Sartaj Sahni. In Sahni's construction the underlying algorithm may be non-deterministic, but we will consider only deterministic algorithms. Furthermore, Sahni describes how to convert his pseudocode into a satisfiability expression. Although it would be possible to convert this expression into an LP, considerable simplifications are obtained by doing a direct conversion from pseudocode to an LP. In this section, for simplicity, we describe only those features of the pseudocode that are necessary for implementing Edmonds' algorithm for the perfect matching problem. Additional features would be needed to handle more sophisticated problems, such as the weighted matching problem. For full details, the reader is referred to Section 11.2 of [13].

Our pseudocode `A` has the following form. We assume W is a fixed integer which will represent the word size for integer variables.

- *Variables* are binary valued except for indices, which are W-bit *integers*. Arrays of binary values are allowed and may be one or two dimensional. Dimension information is specified at the beginning of `A`. We let $q(n)$ denote the maximum number of bits required to represent all variables for an input size of n. Sahni argues that $q(n) = O(p(n))$ however in our case $q(n)$ is significantly smaller. Statements in `A` are numbered sequentially from 1 to l.

- An *expression* contains at most one boolean operator or is the incrementation of an index. Array variables are not used in expressions but may be assigned to simple variables and vice versa.

- `A` contains no read statements and obtains input via its parameters. All other variables are initially zero.

- `A` may contain control statements **go to** k and **if** $c = 1$ **then go to** k **endif**. Here k is an instruction number and c is a simple binary variable.

- `A` terminates by setting a binary variable w to one if the input results in a *yes* outcome and to zero otherwise. The program then halts.

In our implementation we also allow higher level commands such **while** and **for** loops which are first pre-compiled into the basic statements listed above. As a simple example, here is a pseudocode that produces essentially the same result as the circuit in Figure 1.

$y_{12} = x_{12} \wedge x_{34}$
$y_{13} = 0$
$y_{14} = 0$
if y_{12} then go to 50 endif
$y_{13} = x_{13} \wedge x_{24}$
if y_{13} then go to 50 endif
$y_{14} = x_{14} \wedge x_{23}$
if y_{14} then go to 50 endif
$w = 0$
return
50: $w = 1$
return

Note that the lines of the pseudocode which are executed depend on the input values x. This is different from the circuit where all gates are executed for every input. We return to this point below.

The variables in the LP are denoted as follows. They correspond to variables in `A` as it is being executed on a specific input I.

- *Binary variables* $B(i,t), 1 \leqslant i \leqslant q(n), 0 \leqslant t < p(n)$.
  $B(i,t)$ represents the value of binary variable i in `A` after t steps of computation. For convenience we may group W consecutive bits together as an *integer variable* i. $I(i,j,t)$ represents the value of the j-th bit of integer variable i in `A` after t steps of computation. The bits are numbered from right to left, the rightmost bit being numbered 1.

- *Binary arrays* A binary array $R[m], m = 0, 1, ..., u$ is stored in consecutive binary variables $B(\alpha+m, t), 0 \leqslant m \leqslant u, 0 \leqslant t \leqslant p(n)$ from some base location α. The array index m is stored as a W-bit integer $I(*, *, t)$ and so we must have $u \leqslant 2^W - 1$.

- *2-dimensional binary arrays* A two dimensional binary array $R[m][c]$, $m = 0, 1, ..., u$, $c = 0, 1, ..., v$ is stored in row major order in consecutive binary variables $B[\alpha + j, t-1]$, $0 \leqslant j \leqslant uv + u + v, 0 \leqslant t \leqslant p(n)$ from some base location α. The array indices m and c are stored as a W-bit integers $I(*, *, t)$ and so we must have $u, v \leqslant 2^W - 1$.

- *Step counter* $S(i,t), 1 \leqslant i \leqslant l, 1 \leqslant t \leqslant p(n)$.
  Variable $S(i,t)$ represents the instruction to be executed at time t. It takes value 1 if line i of `A` is being executed at time t and 0 otherwise.

All of the above variables are specifically bounded to be between zero and one in our LP. The last set of variables, the step counter, indicates an essential difference between the circuit model and the pseudocode model. In the former model, all gates are executed for each possible input. The gates can be executed in any topological order consistent with the circuit. For the pseudocode model, however, the step to be executed at any time t will usually depend on the actual input. For each time step t and line i of pseudocode we will develop a system of inequalities which have the x-0/1 property, for some subset of variables x, *if line i is executed at time t*. I.e., the inequalities should uniquely determine a 0/1 value of all variables given any 0/1 setting of the x variables. However, if step i is *not executed* at time t then the variables should be free to hold any 0/1 values and these values will be determined by the step that *is* executed at time t. So in each set of inequalities a control variable (in our case the variable $S(i,t)$) will appear for this purpose. More formally, we make the following definition which generalizes Definition 1:

Definition 4. Let $Cx+Dy \leqslant e$ be a system of inequalities that satisfy the x-0/1 property, i.e. each 0/1 setting of the x variables uniquely defines a 0/1 setting of the y variables. Suppose that $Cx + Dy \leqslant e + 1$ is feasible for all 0/1 settings of x and y variables, and let z be a binary variable. The system $\mathbb{1}z + Cx + Dy \leqslant e + \mathbb{1}$ has the (z) *controlled x-0/1 property.*

Note that if $z = 0$ the new system is always feasible for any 0/1 setting of x and y. If $z = 1$ then the new system reduces to the old system that has the x-0/1 property.

We now define the 5 different types of linear inequalities needed to simulate the pseudocode which, following Sahni, we label C,D,E,F and G.[2] Recall that the $S(i,t)$-variables ensure that at each time t a unique line i is executed, taking the value $S(i,t) = 1$ if it is and 0 otherwise. The inequalities listed below all have the $S(i,t)$ controlled x-0/1 property, and so have the form $S(i,t) + Cx + Dy \leqslant e + 1$ for suitably chosen C, D, e, x, y.

C: *(Variable initialization)* The variables $B(i,0), I(i,j,0), 1 \leqslant i \leqslant q(n), 1 \leqslant j \leqslant W$ are set equal to their initial value, if any, else set to zero.

D: *(Step counter initialization)* Instruction 1 is executed at time $t = 1$.

$$S(1,1) = 1$$

E: *(Unique step execution)* A unique instruction is executed at each time t.

$$\sum_{j=1}^{l} S(i,t) = 1, \qquad 1 \leqslant t \leqslant p(n)$$

F: *(Flow control)* The instruction to execute at time $t+1$ is determined, assuming we are at line i of `A` at time t, i.e. $S(i,t) = 1$. If not, i.e. $S(i,t) = 0$, then all inequalities below are trivially satisfied. This follows since the other variables are constrained to be between zero and one. There are 4 subcases depending on the instruction at line i. Inequalities are generated for each t, $1 \leqslant t \leqslant p(n)$.

(i) **(assignment statement)** Go to the next instruction.

$$S(i,t) - S(i+1,t+1) \leqslant 0$$

(ii) (**go to** k)

$$S(i,t) - S(k,t+1) \leqslant 0$$

(iii) (**return**) Loop on this line until time runs out.

$$S(i,t) - S(i,t+1) \leqslant 0$$

(iv) (**if** $c = 1$ **then go to** k **endif**) We assume that bit c is represented by variable $B(j,t-1)$.

$$\begin{aligned} S(i,t) + B(j,t-1) - S(k,t+1) &\leqslant 1 \\ S(i,t) - B(j,t-1) - S(i+1,t+1) &\leqslant 0 \end{aligned}$$

[2]Sahni also has a constraint set H which relates to the certificate checking function of his algorithm, and is not needed here.

When $S(i,t) = 1$ cases (i)-(iii) fix the next line to be executed and trivially have the controlled x-0/1 property, where x is empty. For (iv), note we have also the equations E above. When $S(i,t) = 1$, if $B(j,t-1) = 1$ then the first inequality fixes $S(k,t+1) = 1$ otherwise the second inequality fixes $S(i+1,t+1) = 1$. The inequalities (iv) have the controlled $B(j,t-1)$-0/1 property.

G: *(Control of variables)* If we are at line i of `A` at time t, i.e. $S(i,t) = 1$, all variables are updated to their correct values at time $t+1$ following the execution of line i. If not, i.e. $S(i,t) = 0$, then all inequalities below are trivially satisfied. Again there are several cases depending on the instruction at line i. Inequalities are generated for each t, $1 \leqslant t \leqslant p(n)$.

(i) **(Reassignment of unchanged variables)** All variables left unchanged at a given step t need to be reassigned their previous values. For Let k index some bit unchanged at step t.

$$\begin{aligned} S(i,t) + B(k,t-1) - B(k,t) &\leqslant 1 \\ S(i,t) - B(k,t-1) + B(k,t) &\leqslant 1 \end{aligned}$$

Note that when $S(i,t) = 1$ these inequalities imply that $B(k,t-1) = B(k,t)$. They have the controlled $B(k,t-1)$-0/1 property. Similar inequalities are generated for each integer variable $I(k,j,t), 1 \leqslant j \leqslant W$.

In what follows, the above inequalities need to be generated for all variables $B(k,t)$ and $I(k,j,t)$ not being assigned values at time t in the particular instruction i being considered.

(ii) (**assignment:** $s = x$ **and** $s = \neg x$) Assume that x, s are stored in $B(q,t-1), B(s,t)$ respectively. For $s = x$ we generate the two inequalities:

$$\begin{aligned} S(i,t) + B(q,t-1) - B(s,t) &\leqslant 1 \\ S(i,t) - B(q,t-1) + B(s,t) &\leqslant 1 \end{aligned}$$

When $S(i,t) = 1$ the inequalities imply $B(s,t) = B(q,t-1)$ as desired. They have the controlled $B(q,t-1)$-property. For $s = \neg x$ we generate the two inequalities:

$$\begin{aligned} S(i,t) + B(q,t-1) + B(s,t) &\leqslant 2 \\ S(i,t) - B(q,t-1) - B(s,t) &\leqslant 0 \end{aligned}$$

The analysis is similar to that for $s = x$.

(iii) (**assignment:** $s = x \oplus y$) Assume that x, y, s are stored in $B(q,t-1), B(r,t-1), B(s,t)$ respectively.

$$\begin{aligned} S(i,t) + B(q,t-1) - B(r,t-1) - B(s,t) &\leqslant 1 \\ S(i,t) - B(q,t-1) - B(r,t-1) + B(s,t) &\leqslant 1 \\ S(i,t) - B(q,t-1) + B(r,t-1) - B(s,t) &\leqslant 1 \\ S(i,t) + B(q,t-1) + B(r,t-1) + B(s,t) &\leqslant 3 \end{aligned}$$

If $S(i,t) = 1$ then all constants on the right hand side are reduced by one and $S(i,t)$ can be deleted. It is easy to check the inequalities have the controlled $\{B(q,t-1), B(r,t-1)\}$-0/1 property, and that for each such 0/1 assignment $B(s,t)$ is correctly set.

(iv) (**assignment:** $s = x \wedge y$) Assume that x, y, s are stored in $B(q,t-1), B(r,t-1), B(s,t)$ respectively.

$$\begin{aligned} S(i,t) - B(q,t-1) \qquad\qquad + B(s,t) &\leqslant 1 \\ S(i,t) \qquad\qquad - B(r,t-1) + B(s,t) &\leqslant 1 \\ S(i,t) + B(q,t-1) + B(r,t-1) - B(s,t) &\leqslant 2 \end{aligned}$$

If $S(i,t) = 1$ then all constants on the right hand side are reduced by one and $S(i,t)$ can be deleted. It is easy to check the inequalities have the controlled $\{B(q,t-1), B(r,t-1)\}$-0/1 property, and that for each such 0/1 assignment $B(s,t)$ is correctly set.

(v) (**assignment:** $s = x \vee y$**,** $s = x_1 \vee x_2 \vee ... \vee x_k$)
Assume that x, y, s are stored in $B(q, t-1), B(r, t-1), B(s, t)$ respectively.

$$\begin{aligned} S(i,t) + B(q,t-1) \qquad\qquad - B(s,t) &\leqslant 1 \\ S(i,t) \qquad\qquad + B(r,t-1) - B(s,t) &\leqslant 1 \\ S(i,t) - B(q,t-1) - B(r,t-1) + B(s,t) &\leqslant 1 \end{aligned}$$

The analysis is similar to G(iv) and is omitted. The inequalities have the controlled {B(q,t-1),B(r,t-1)}-0/1 property.
The k-way *or* is an easy generalization which will be needed in the sequel, where we assume that x_j is stored in $B(q_j, t-1), j = 1, 2, ..., k$. It is defined by the following inequalities:

$$\begin{aligned} S(i,t) + B(q_j,t-1) - B(s,t) &\leqslant 1 \qquad 1 \leqslant j \leqslant k \\ S(i,t) - \sum_{j=1}^{k} B(q_j,t-1) + B(s,t) &\leqslant 1. \end{aligned}$$

(vi) (**increment integer variable**) Assume that the integer variable is stored in $I(q, j, t-1), 1 \leqslant j \leqslant W$ and is to be incremented by 1. We require another integer $I(r, j, t), 1 \leqslant j \leqslant W$ to hold the binary carries. On overflow, $I(r, W, t) = 1$ and $I(q, j, t) = 0, 1 \leqslant j \leqslant W$. The incrementer makes use of two previous operations, G(iii) and G(iv):

$$\begin{aligned} I(q,1,t) &= I(q,1,t-1) \oplus 1 \\ I(r,1,t) &= I(q,1,t-1) \wedge 1 \\ I(q,j,t) &= I(q,j,t-1) \oplus I(r,j-1,t) \qquad 2 \leqslant j \leqslant W \\ I(r,j,t) &= I(q,j,t-1) \wedge I(r,j-1,t) \qquad 2 \leqslant j \leqslant W \end{aligned}$$

By appropriate formal substitution of variables, each of the above assignments is transformed into inequalities of the form G(iii) and G(iv), which are controlled by the step counter $S(i, t)$. It can be verified that the full system satisfies the controlled $\{I(q, j, t), 1 \leqslant j \leqslant W\}$-0/1 property because for each 0/1 setting of these variables all other variables are fixed by the above system of equations.

(vii) (**equality test for integer variables**) Assume that the integer variables are stored in $I(q, j, t-1)$ and $I(r, j, t-1)$, $1 \leqslant j \leqslant W$. We require $W+1$ temporary variables w.l.o.g. $B(j, t), 1 \leqslant j \leqslant W+1$. If the two integer variables are equal then $B(W+1, t)$ is set to one else it is set to zero.

$$\begin{aligned} B(j,t) &= I(q,j,t-1) \oplus I(r,j,t-1) \qquad 1 \leqslant j \leqslant W \\ B(W+1,t) &= \neg \bigvee_{j=1}^{W} B(j,t) \end{aligned}$$

The first equations makes repeated use of G(iii) after appropriate substitution. By combining G(ii) and the k-way *or* from $G(v)$ we may implement the second equation by the inequalities.

$$\begin{aligned} S(i,t) + B(j,t) + B(W+1,t) &\leqslant 2 \qquad 1 \leqslant j \leqslant W \\ S(i,t) - \sum_{j=1}^{k} B(j,t) - B(W+1,t) &\leqslant 0. \end{aligned}$$

The inequalities have the controlled $\{I(q, j, t-1), I(r, j, t-1),\ 1 \leqslant j \leqslant W\}$-0/1 property.

(viii) (**array assignment**) $R[m] = x$ (and $x = R[m]$) We assume that R has dimension u, is stored in $B[\alpha+j, t-1]$, $0 \leqslant j \leqslant u$ and that x is stored in $B(x, t-1)$. We further assume that m is stored in an integer variable $I(m, k, t-1), 1 \leqslant k \leqslant W$. We need additional binary variables $M(j, t), 0 \leqslant j \leqslant u$ to hold intermediate results. Initially we write down some equations and then we use previous results to convert these to inequalities. Firstly we need to discover the memory location for $R[m]$. For any $0 \leqslant j \leqslant u$ let $j_W j_{W-1} ... j_1$ be the binary representation of j. Then we formally define for $k = 1, 2, ..., W$

$$T_j(m,k,t-1) = \begin{cases} I(m,k,t-1) & j_k = 0 \\ 1 - I(m,k,t-1) & j_k = 1 \end{cases}$$

Note this definition is purely formal and has nothing to do with the execution of A. We will assign $M(j,t)$ a value via the W-way *or* given in G(v). For $0 \leqslant j \leqslant u$:

$$\begin{aligned} S(i,t) + T_j(m,k,t-1) - M(j,t) &\leqslant 1 \qquad 1 \leqslant k \leqslant W \\ S(i,t) - \sum_{k=1}^{W} T_j(m,k,t-1) + M(j,t) &\leqslant 1 \end{aligned} \tag{15}$$

When $S(i,t) = 1$, it can be verified that $M(j,t) = 0$ whenever $j = m$ and is one otherwise. Now we may update all array elements of R at time t and make the assignment $R[m] = x$ by the system of inequalities, for all $0 \leqslant j \leqslant u$:

$$\begin{aligned} S(i,t) + B(x,t-1) - B(\alpha+j,t) - M(j,t) &\leqslant 1 \\ S(i,t) - B(x,t-1) + B(\alpha+j,t) - M(j,t) &\leqslant 1 \\ S(i,t) + B(\alpha+j,t-1) - B(\alpha+j,t) + M(j,t) &\leqslant 2 \\ S(i,t) - B(\alpha+j,t-1) + B(\alpha+j,t) + M(j,t) &\leqslant 2 \end{aligned}$$

To understand these inequalities, first note that they are trivially satisfied unless $S(i,t) = 1$. When $j = k$ we have $M(j,t) = 0$ and the first two inequalities are tight. We have $B(\alpha+j,t) = B(x,t-1)$ updating the array element to x. The second two inequalities are trivially satisfied. Otherwise $j \neq k$, $M(j,t) = 1$, the first two inequalities are trivially satisfied and the second two are tight. We have $B(\alpha+j,t) = B(\alpha+j,t-1)$ copying the array element over to time t from time $t-1$. We remark that there are $O(uW)$ inequalities generated above.

Finally note that we can implement $x = R[m]$ by using the inequalities

$$\begin{aligned} S(i,t) + B(x,t) - B(\alpha+j,t-1) - M(j,t) &\leqslant 1 \\ S(i,t) - B(x,t) + B(\alpha+j,t-1) - M(j,t) &\leqslant 1 \end{aligned}$$

and letting the array $R[m]$ be copied at time t using G(i). Both of these two inequality systems have the controlled $\{B(x,t-1), B(\alpha+j,t-1), j = 0,...,u\}$-0/1 property.

(ix) (**2-dimensional array assignment**) $R[m][c] = x$ (or $x = R[m][c]$). This is a natural generalization of G(viii). We assume that R has dimensions u and v, is stored in row major order in $B[\alpha+j,t-1]$, $0 \leqslant j \leqslant uv+u+v$ and that x is stored in $B(x,t-1)$. We further assume that m and c are stored in an integer variables $I(m,k,t-1), I(c,k,t-1), 1 \leqslant k \leqslant W$ respectively. We need additional binary variables $M(j,t), 0 \leqslant j \leqslant u$ and $N(j,t), 0 \leqslant j \leqslant v$ to hold intermediate results. Firstly we need to discover the memory location for $R[m][c]$. We again use the equations (15) for the row index.

For the column index, as in G(viii), for any $0 \leqslant j \leqslant v$ let $j_W j_{W-1} ... j_1$ be the binary representation of j. We formally define for $k = 1, 2, ..., W$

$$T_j(c,k,t-1) = \begin{cases} I(c,k,t-1) & j_k = 0 \\ 1 - I(c,k,t-1) & j_k = 1 \end{cases}$$

We will assign $N(j,t)$ a value via the W-way *or* given in G(v). For $0 \leqslant j \leqslant v$:

$$\begin{aligned} S(i,t) + T_j(c,k,t-1) - N(j,t) &\leqslant 1 \qquad 1 \leqslant k \leqslant W \\ S(i,t) - \sum_{k=1}^{W} T_j(c,k,t-1) + N(j,t) &\leqslant 1 \end{aligned}$$

When $S(i,t) = 1$, it can be verified that $N(j,t) = 0$ whenever $j = c$ and is one otherwise. Now we may update all array elements of R at time t and make the assignment $R[m][c] = x$ by the following

system of inequalities. For all $0 \leqslant j_1 \leqslant u$, $0 \leqslant j_2 \leqslant v$, $r = j_1(u+1) + j_2$:

$$\begin{aligned}
S(i,t) + B(x,t-1) - B(\alpha+r,t) - M(j_1,t) - N(j_2,t) &\leqslant 1 \\
S(i,t) - B(x,t-1) + B(\alpha+r,t) - M(j_1,t) - N(j_2,t) &\leqslant 1 \\
S(i,t) + B(\alpha+r,t-1) - B(\alpha+r,t) + M(j_1,t) &\leqslant 2 \\
S(i,t) - B(\alpha+r,t-1) + B(\alpha+r,t) + M(j_1,t) &\leqslant 2 \\
S(i,t) + B(\alpha+r,t-1) - B(\alpha+r,t) + N(j_2,t) &\leqslant 2 \\
S(i,t) - B(\alpha+r,t-1) + B(\alpha+r,t) + N(j_2,t) &\leqslant 2
\end{aligned}$$

The analysis is similar to G(viii). The above inequalities are all trivial unless $S(i,t) = 1$. Note that for each j_1 and j_2, index r gives the relative location in the array. If $j_1 = m$, $j_2 = c$ then $M(j_1,t) = N(j_2,t) = 0$, the first two inequalities are tight and the last four loose. The first two inequalities give $B(\alpha+r,t) = B(x,t-1)$. Otherwise either $M(j_1,t) = 1$ or $N(j_2,t) = 1$ or both, and the first two inequalities are trivially satisfied. In the former case the two middle inequalities are tight and we have the equation $B(\alpha+r,t) = B(\alpha+r,t-1)$. In the latter case this equation is formed from the last two inequalities. We remark that there are $O(uv + uW + vW)$ inequalities generated above.

For the assignment $x = R[m][c]$ we need the inequalities

$$\begin{aligned}
S(i,t) + B(x,t) - B(\alpha+r,t-1) - M(j_1,t) - N(j_2,t) &\leqslant 1 \\
S(i,t) - B(x,t) + B(\alpha+r,t-1) - M(j_1,t) - N(j_2,t) &\leqslant 1
\end{aligned}$$

for $r = j_1(u+1) + j_2$. All array elements of R must also be copied from time $t-1$ to time t as in G(i).

Both of these two inequality systems have the controlled $\{B(x,t-1), B(\alpha+j,t), j = 0, ..., uv+u+v\}$-0/1 property.

Remark: In applications using graphs, a symmetric 2-dimensional array is often used to hold the adjacency matrix. Such symetric matrices may be implemented in pseudocode by replacing a statement such as $R[m][c] = x$ by the two statements $R[m][c] = x$ and $R[c][m] = x$. Assignment statements such as $x = R[m][c]$ may be left as is.

To show the correctness of the above procedure we give two lemmas that are analogous to Lemmas 1 and 2 of the last section. First we show that the above construction can be applied to any pseudocode `A`, written in the language described, to produce a polytope Q which has the 0/1 property with respect to the inputs of `A`.

Lemma 3. *Let* `A` *be a pseudocode, written in the above language, which takes n input bits $x = (x_1, x_2, ..., x_n)$, and terminates by setting a bit w. Construct the polytope Q as described above relabelling $B(0, p(n))$ as w and the additional variables as $s = (s_1, s_2, ..., s_N)$ for some integer N. Q has the the x-0/1 property and for every input x the value of w computed by* `A` *corresponds to the value of w in the unique extension $(x, w, s) \in Q$ of x.*

Proof. (Sketch) As with Lemma 1 the proof is by induction, but this time we use the step counter. By assumption `A` terminates after $p(n)$ steps. Let $k = 1, 2, ..., p(n)$ represent the step counter. Define Q_k to be the polytope consisting of precisely those inequalities in Q that use variables: $B(i,t), 1 \leqslant i \leqslant q(n), 1 \leqslant t \leqslant k$, $I(i,j,t), 1 \leqslant i \leqslant q(n), 1 \leqslant j \leqslant W, 1 \leqslant t \leqslant k$ and $S(i,t), 1 \leqslant i \leqslant l, 1 \leqslant t \leqslant k$.

- Q_k has the x-0/1 property, and
- for each x, at step k, `A` with input x is executing line i corresponding to the unique index i where $S(i,k) = 1$ and all variables at that step have the values corresponding to the values of $B(i,k), 1 \leqslant i \leqslant q(n)$ and $I(i,j,k), 1 \leqslant i \leqslant q(n), 1 \leqslant j \leqslant W$.

The inequalities of Q_1 consist of those in groups C, D, and part of E above and the induction hypothesis is readily verified. We assume the hypothesis is true for $k = 1, 2, ..., T$, where $1 \leqslant T < p(n)$, and prove it for $T+1$. Indeed, since Q_k has the x-0/1 property for each x the values of all variables with index $t \leqslant T$ have been correctly set. It follows that for precisely one index i we have $S(i,T) = 1$, meaning that line i of the pseudocode is executed at time T for this particular input. The inequalities defined in group G all have the

controlled x-0/1 property for the control variable $S(i,T)$. The variables B and I with index $t = T+1$ are correctly set by the analysis in group G above. The analysis in group F implies that the values of $S(i,T+1)$ will also be uniquely determined and 0/1, correctly indicating the next line of `A` to be executed at $t = T+1$. This verifies the inductive hypothesis for $T+1$ and since $Q = Q_{p(n)}$ this concludes the proof. □

The next lemma is simply a restatement of Lemma 2 in the context of our pseudocode rather than circuits. The proof of Lemma 2 makes no reference to how Q was computed, so the same proof holds.

Lemma 4. *Let* `A` *be a algorithm, written in the above pseudocode, which solves a decision problem X with n input bits $x = (x_1, x_2, ..., x_n)$ and has associated polytope P as defined in (9). The polytope Q described in Lemma 3 is a WEF for P.*

Lemmas 3 and 4 justify the correctness of the method outlined in this section.

We now analyze the size of the WEF Q created. Recall that $q(n)$ is the number of bits of storage required by the algorithm A, which of course consists of a constant number of lines of pseudocode. The variables of Q are the variables $B(j,t)$, $S(i,t)$ and additional temporary variables created in some of the groups C–G. It can be verified that their number is $O(p(n)q(n))$. For fixed t, each of the sets of inequalities described in groups C–G have size at most $O(q(n))$ except possibly the array assignment inequalities described in G(viii) and G(ix). As remarked there, an array of dimension u generates $O(uW)$ inequalities. A 2-dimensional array of dimension r by c generates $O(rc + rW + cW)$ inequalities. We may assume that $W \in O(\log n)$. Then $O(q(n)\log n)$ is an upper bound the number of inequalities generated in either G(viii) or G(ix). Since t is bounded by $p(n)$ we see that the WEF has at most $O(p(n)(q(n)\log n))$ inequalities also. We have:

Theorem 2. *Let X be a decision problem with corresponding polytope P defined by (9). A algorithm for X written in the pseudocode described above requiring $q(n)$ space and terminating after $p(n)$ steps generates a WEF Q for P with $O(p(n)q(n)\log n)$ inequalities and variables.*

Since Edmonds' algorithm can be implemented in poly-time in the pseudocode presented our method gives a polynomial size WEF for PM_n. So for example, a straightforward $O(n^4)$ implementation of Edmonds' algorithm using $O(n^2)$ space would yield a WEF with $O(n^6 \log n)$ inequalities and variables. If the $O(n^{2.5})$ time algorithm of Even and Kariv [9] can be implemented in our pseudocode it would yield a considerably smaller polytope.

We are currently building a compiler along the lines described here to automatically generate a WEF corresponding to any given pseudocode. The elements described in C–G above have been implemented and tested as well as a few small complete examples of pseudocode. The polytopes generated are rather large even for short pseudocodes. For example, the pseudocode at the beginning of this section generated a polytope with about 3200 inequalities![3] This should be compared with 28 inequalities for the circuit in Figure 1 and 4 odd set inequalities for Edmonds' polytope EP_4. Nevertheless the WEF generated by this method should be significantly smaller than EP_n even for relatively small n. The details of the implementation of the compiler and its application to Edmonds' algorithm will be described in a subsequent paper.

5 Connections to non-negative rank

In this section we reformulate the results in previous sections in terms of non-negative ranks of certain matrices. Non-negative rank is a very useful tool that captures the extension complexity of polytopes [19]. We use a slightly modified version of non-negative rank to given an alternate characterization of the complexity class P/poly, which is the class of decision problems that can be solved in polynomial time given a polynomial size advice string [1]. This characterization potentially opens the door to proving that a given problem does not belong to P/poly by demonstrating high non-negative rank of the associated characteristic matrix. We will come back to this discussion again after describing the characterization.

A matrix S is called non-negative if all its entries are non-negative. The non-negative rank of a non-negative matrix S, denoted by $\mathrm{rank}_+(S)$, is the smallest number r such that there exist non-negative matrices T and U such that T has r columns, U has r rows and $S = TU$. If we require the left factor T in the above definition to only contain numbers that can be encoded using a number of bits only polynomial in the number of bits required to encode any entry of S then the smallest such r is called the *succinct* non-negative rank of S and is

[3] This polytope and other examples produced by the compiler can be found at `http://www.cs.unb.ca/~bremner/research/sparks_lp`

denoted by $\text{rank}_{\oplus}(S)$. To see the usefulness of this apparently asymmetric restriction on the factors, note that when S is the slack matrix of a polytope $Ax \geqslant b$ then such a factorization allows one to describe an extended formulation for the polytope using only the entries of A, T and b. So if T is required to polynomial in the bit complexity of the entries of S and then one can represent the polytope as the shadow of another polytope that can be encoded using a polynomial number of bits.

Let P_{in} and P_{out} be two polytopes in $\mathbb{R}^k$ such that $P_{\text{in}} \subseteq P_{\text{out}}$. We say that such a pair defines a *polytopal sandwich*. With every polytopal sandwich we can associate a non-negative matrix which encodes the slack of the inequalities defining P_{out} with respect to the vertices of P_{in}. That is, if $P_{\text{in}} = \text{conv}(\{v_1, \ldots, v_n\})$ and $P_{\text{out}} = \{x \in \mathbb{R}^k \mid a_i^\intercal x \leqslant b_i, 1 \leqslant i \leqslant m\}$ then the slack matrix associated with the polytopal sandwich thus defined is $S(P_{\text{out}}, P_{\text{in}}) = S$ with $S_{ij} = b_i - a_i^\intercal v_j$. When P_{in} and P_{out} define the same polytope P we denote the corresponding slack matrix simply as $S(P)$. The next lemma shows the relation between the non-negative rank of the slack of a polytopal sandwich and the smallest size polytope whose shadow fits in the sandwich. We will assume that the polytopes defining our sandwiches are full-dimensional. The same lemma appears in [4] and has roots in [11, 14]. However the proof is simple enough to attribute it to folklore.

Lemma 5. *Let $P_{in} = \text{conv}(\{v_1, \ldots, v_n\})$ and $P_{out} = \{x \in \mathbb{R}^k \mid a_i^\intercal x \leqslant b_i, 1 \leqslant i \leqslant m\}$. Let P be a polytope with smallest extension complexity such that $P_{in} \subseteq P \subseteq P_{out}$. Then,* $\text{xc}(P) = \text{rank}_+(S(P_{out}, P_{in}))$.

Proof. Suppose $P_{\text{in}} \subseteq P \subseteq P_{\text{out}}$. We can describe P as the convex hull of the vertices of P together with the vertices of P_{in}. Similarly we can describe P as the intersection of all its facet-defining inequalities and the facet-defining inequalities of P_{out}. Now the matrix $S(P_{\text{out}}, P_{\text{in}})$ is a submatrix of the slack matrix $S(P)$ of this particular representation of P. Therefore, $\text{rank}_+(S(P)) \geqslant \text{rank}_+(S(P_{\text{out}}, P_{\text{in}}))$. It is easy to see (see, for example, [10]) that the non-negative rank of the slack matrix of a polytope is not changed by adding redundant inequalities and points in its representation. Also, since the non-negative rank of the slack matrix of a polytope is equal to its extension complexity ([10]), we have that $\text{xc}(P) \geqslant \text{rank}_+(S(P_{\text{out}}, P_{\text{in}}))$.

Now, suppose that $\text{rank}_+(S(P_{\text{out}}, P_{\text{in}})) = r$. That is there exist non-negative matrices T and U with r columns and rows respectively, such that $S(P_{\text{out}}, P_{\text{in}}) = TU$. Denote by T_i the i-th row of T and U^j the j-th column of U. Consider the polytope

$$Q = \{(x, y) \in \mathbb{R}^{k+r} \mid a_i^\intercal x + T_i y = b_i, 1 \leqslant i \leqslant m, y \geqslant 0\}$$

and let

$$P = \{x \in \mathbb{R}^k \mid \exists y \in \mathbb{R}^r, (x, y) \in Q\}.$$

Since, by definition, P is a projection of Q and Q has at most r inequalities, we have that $\text{xc}(P) \leqslant \text{rank}_+(S(P_{\text{out}}, P_{\text{in}}))$. If we show that $P_{\text{in}} \subseteq P \subseteq P_{\text{out}}$ then $\text{xc}(P) \geqslant \text{rank}_+(S(P_{\text{out}}, P_{\text{in}}))$, implying the lemma.

Suppose $x \in P$. Then $\exists y, (x, y) \in Q$. That is $y \geqslant 0$ and $a_i^\intercal x + T_i y = b_i$ for all i. Since T is non-negative, $T_i y \geqslant 0$ and therefore $a_i^\intercal x \leqslant b_i$ for all i. That is, $x \in P_{\text{out}}$. Therefore, $P \subseteq P_{\text{out}}$.

Suppose $x \in P_{\text{in}}$. Then $x = \sum_{j=1}^n \lambda_j v_j, \sum_{j=1}^n \lambda_j = 1, \lambda_j \geqslant 0$, for some λ. Consider $y = \sum_{j=1}^n \lambda_j U^j$. Then, for each $i = 1, 2, .., m$ we have that $a_i^\intercal x + T_i y = \sum_{j=1}^n \lambda_j (a_i^\intercal v_j + T_i U^j) = \sum_{j=1}^n \lambda_j (b_i) = b_i$. Clearly $y \geqslant 0$ since U is non-negative. So $(x, y) \in Q$ and thus $x \in P$. Therefore $P_{\text{in}} \subseteq P$. This completes the proof. □

Note that given a polytopal sandwich $P_{\text{in}}, P_{\text{out}}$, any lower bound on the non-negative rank of its slack matrix $S(P_{\text{in}}, P_{\text{out}})$ is also the lower bound on the succinct non-negative rank of $S(P_{\text{in}}, P_{\text{out}})$. Conversely, any upper bound on the succinct non-negative rank of $S(P_{\text{in}}, P_{\text{out}})$ is also an upper bound on the non-negative rank of $S(P_{\text{in}}, P_{\text{out}})$. In the next subsection we will define canonical polytopal sandwiches for binary languages and discuss the relation of the non-negative rank and succinct non-negative rank of the associated slack matrix with whether or not membership testing in the language belongs to P/poly or not.

5.1 Languages and their sandwiches

Let $L \subseteq \{0, 1\}^*$ be a language over the 0/1 alphabet. For every natural number n define the set $L(n)$ as

$$L(n) := \{x \in \{0, 1\}^n \mid x \in L\}$$

For every n consider the $2^n \times 2^n$ non-negative matrix $M(L(n))$ defined as follows. Rows and columns of $M(L(n))$ are indexed by 0/1 vectors a, b of length n and

$$M_{a,b}(L(n)) = a^\intercal \mathbb{1}_n - 2a^\intercal b + \mathbb{1}^\intercal b + \alpha(a, b)$$

where

$$\alpha(a,b) = \begin{cases} d & \text{if } a \in L, b \notin L \\ -d & \text{if } a \notin L, b \in L \\ 0 & \text{otherwise} \end{cases},$$

for some constant $0 < d < \frac{1}{2}$. This constant is a universal constant that depends only on n and not on L. The appropriate value of d can be obtained from Lemma 2 (See last part of the proof).

Corresponding to any language L let us define a polytopal sandwich given by a pair of polytopes. The inner polytope is described by its vertices and is contained in the outer polytope, which in turn is described by a set of inequalities. Both the vertices of the inner polytope and the inequalities for the outer polytope depend only on the language L. We call such a sandwich the *characteristic sandwich* of $L(n)$ and $M(L(n))$ is the slack of this sandwich (Lemma 6).

Corresponding to every language $L \subseteq \{0,1\}^*$ we define characteristic functions $\psi : \{0,1\}^* \to \{0,1\}$ and $\phi : \{0,1\}^* \to \{-1,1\}^*$ with

$$\psi(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases},$$

$$\phi(x)_i = \begin{cases} 1 & \text{if } x_i = 1 \\ -1 & \text{if } x_i = 0 \end{cases}$$

To make the connection with Section 2.2 note that $\psi(x)$ will play the role of w_x and $\phi(x)$ will play the role of the objective function vector c. The inner polytope $V(L(n))$ is then defined to be $\text{conv}(\{(x, \psi(x)) \mid x \in \{0,1\}^n\})$. In terms of Section 2.2, $V(L(n))$ plays the role of P. In terms of matchings it is PM_n. The outer polytope $H(L(n))$ is defined by the inequalities

$$H(L(n)) := \left\{ (x,w) \in \mathbb{R}^{n+1} \;\middle|\; \begin{array}{ll} \phi(a)^\intercal x + dw \leqslant a^\intercal \mathbb{1} + d & \forall a \in L(n) \\ \phi(a)^\intercal x + dw \leqslant a^\intercal \mathbb{1} & \forall a \notin L(n) \end{array} \right\}.$$

Note that the normal vectors of the inequalities defining $H(L(n))$ are just the optimization directions (c, d) that were used in Section 2.2.

Lemma 6. *The slack of $H(L(n))$ with respect to $V(L(n))$ is the matrix $M(L(n))$.*

Proof. Consider two vectors $a, b \in \{0,1\}^n$. The slack of the inequality corresponding to $\phi(a)$ with respect to $(b, \psi(b))$ is

$$\begin{cases} a^\intercal \mathbb{1} + d - d\psi(b) - \phi^\intercal(a)b & \text{if } a \in L \\ a^\intercal \mathbb{1} - d\psi(b) - \phi^\intercal(a)b & \text{if } a \notin L \end{cases}$$

Observing that $\phi(a) = a - (\mathbb{1} - a) = 2a - \mathbb{1}$ we can see that $a^\intercal \mathbb{1} - \phi^\intercal(a)b = a^\intercal \mathbb{1} + b^\intercal \mathbb{1} - 2a^\intercal b$. Therefore the slack of the inequality corresponding to $\phi(a)$ with respect to $(b, \psi(b))$ is

$$\begin{cases} a^\intercal \mathbb{1} + b^\intercal \mathbb{1} - 2a^\intercal b + d & \text{if } a \in L, b \notin L \\ a^\intercal \mathbb{1} + b^\intercal \mathbb{1} - 2a^\intercal b - d & \text{if } a \notin L, b \in L \\ a^\intercal \mathbb{1} + b^\intercal \mathbb{1} - 2a^\intercal b & \text{otherwise} \end{cases}$$

□

The following lemma is analogous to Proposition 3.

Lemma 7. *Let P be a polytope such that $V(L(n)) \subseteq P \subseteq H(L(n))$. Then, deciding whether a vector $a \in \{0,1\}^n$ is in L or not can be achieved by optimizing over P along the direction $(\phi(a), d)$ for some constant $0 < d \leqslant 1/2$.*

Proof. Let a be a given vector in $\{0,1\}^n$. Consider the maxima z_v, z_p, z_h of $\phi^\intercal(a)x + dw$ when $(x,w) \in V(L(n))$, $(x,w) \in P$, and $(x,w) \in H(L(n))$ respectively. Since $V(L(n)) \subseteq P \subseteq H(L(n))$ we have that $z_v \leqslant z_p \leqslant z_h$. If $a \in L$ then $z_v = z_h = a^\intercal \mathbb{1} + d$, otherwise $z_v = a^\intercal \mathbb{1}, z_h \leqslant a^\intercal \mathbb{1}$. Therefore whether $z_p = a^\intercal \mathbb{1} + d$ or not tells us whether $a \in L$ or not. □

Theorem 3. *For any language L if membership testing in L belongs to the class* **P/poly**, *then $M(L(n))$ has non-negative rank polynomial in n. Moreover if the succinct non-negative rank of $M(L(n))$ is polynomial in n, then membership testing in L belongs to the class* **P/poly**.

Proof. Suppose L belongs to the class **P/poly**, then we can construct a polynomial size polytope Q as in Lemma 1. By the argument used in the proof of Lemma 2 we can see that the projection of Q onto the first $n+1$ coordinates is contained in $H(L(n))$ and contains $V(L(n))$. Since $M(L(n))$ is the slack of $H(L(n))$ with respect to $V(L(n))$, the non-negative rank of $M(L(n))$ is upper bounded by the extension complexity of any polytope P sandwiched between the two polytopes (see proof of Lemma 5). Since the size of Q is a polynomial in n and Q is the extension of some polytope that can be sandwiched between $H(L(n))$ and $V(L(n))$ we have that $\text{rank}_+(M(L(n))$ is a polynomial in n.

For the other direction, suppose that $M(n)$ has succinct non-negative rank r which is polynomial in n. Since $M(n)$ is the slack of $H(L(n))$ with respect to $V(L(n))$, by Lemma 5 there exists a polytope P such that the extension complexity of P is r and $V(L(n)) \subseteq P \subseteq H(L(n))$. In other words, there exists polytopes P and Q such that Q has r facets, projects down to P and $V(L(n)) \subseteq P \subseteq H(L(n))$. Furthermore, the description of Q contains only numbers polynomially bounded in n because the rank-r non-negative factorization is succinct by our assumption. By Lemma 7 optimizing over P can be used to decide whether $x \in L$ or not for a given x. Furthermore, optimizing over P can be done by optimizing over Q instead. Since Q has polynomial size, we can use interior point methods to do the optimization and so determine membership in L in polynomial time. □

The above theorem in principle paves a way to show that membership testing in a certain language is not in **P/poly**. This can be done by showing that the non-negative rank of the associated sandwich is super-polynomial. Various techniques exist to show lower bounds for the non-negative rank of matrices and have been used to prove super-polynomial lower bounds for the extension complexity of important polytopes like the CUT polytope, the TSP polytope, and the Perfect Matching polytope of Edmonds, among others [10, 16, 2, 15]. Whether one can apply such techniques to show super-polynomial lower bounds on the non-negative rank of the slack of the characteristic sandwich of some language is left as as open problem.

6 Concluding remarks

The discussion in this paper was centred around decision problems and one may wonder if it can be applied to optimization problems also. Before addressing this let us recall some discussion on the topic in Yannakakis' paper [19]:

> Linear programming is complete for decision problems in P; the P = NP? question is equivalent to a weaker requirement of the LP (than that expressing the TSP polytope), *in some sense reflecting the difference between decision and optimization problems.* (P. 445, emphasis ours)

The term "expressing" in the citation refers to an extended formulation of polynomial size. The method we have described can indeed be used to construct polynomial sized LPs for optimization problems which have polynomial time algorithms. Consider, for example, the problem of finding a maximum weight matching for the complete graph K_n, where the edge weights are integers of length W bits. For simplicity we assume the weights are non-negative, but a small extension would handle the general case. We construct a polytope P as defined by (9) as follows. The binary vectors x have length $Wn(n-1)/2 + W + \lceil \log_2 n \rceil$, where the first $Wn(n-1)/2$ bits encode the edge weights and the remaining bits encode an integer $k, 0 \leq k < 2^W n$. The bit w_x is defined by setting $w_x = 1$ whenever the edge weights specified in x admit a matching of weight k or greater and $w_x = 0$ otherwise. Note that the unweighted maximum matching problem for graphs on n vertices fits into this framework by setting $W = 1$.

Applying the method of Section 4 to the weighted version of Edmonds' algorithm we obtain a polynomial sized WEF Q for P. We can decide by solving a single LP over Q if a given weighted K_n has a matching of weight at least k, for any fixed k: the last $W + \lceil \log_2 n \rceil$ coefficients of the objective function (10) vary depending on k. Therefore, by binary search on k we can solve the maximum weight matching problem for a given input by optimizing $O(W + \log n)$ times over Q with objective functions depending on k. We do not, however, know how to solve the weighted matching problem by means of a single polynomially sized linear program.

Acknowledgments

We would like to thank three anonymous referees of an extended abstract of an earlier version of this paper for their helpful comments. This research is supported by a Grant-in-Aid for Scientific Research on Innovative Areas – Exploring the Limits of Computation(ELC), MEXT, Japan. Research of the second author is partially supported by NSERC Canada. Research of the third author is also partially supported by the Center of Excellence – Institute for Theoretical Computer Science, Prague (project P202/12/G061 of GA ČR).

References

[1] S. Arora and B. Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. ISBN 978-0-521-42426-4. URL `http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264`.

[2] D. Avis and H. R. Tiwary. On the extension complexity of combinatorial polytopes. In *ICALP (1)*, pages 57–68, 2013.

[3] D. Avis and H. R. Tiwary. A generalization of extension complexity that captures P. *CoRR*, abs/1402.5950, 2014.

[4] G. Braun, S. Fiorini, S. Pokutta, and D. Steurer. Approximation limits of linear programs (beyond hierarchies). In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 480–489, 2012. doi: 10.1109/FOCS.2012.10. URL `http://dx.doi.org/10.1109/FOCS.2012.10`.

[5] S. A. Cook and R. A. Reckhow. Time bounded random access machines. *J. Comput. System Sci.*, 7: 354–375, 1973. ISSN 0022-0000. Fourth Annual ACM Symposium on the Theory of Computing (Denver, Colo., 1972).

[6] D. P. Dobkin, R. J. Lipton, and S. P. Reiss. Linear programming is log-space hard for P. *Inf. Process. Lett.*, 8:96–97, 1979.

[7] J. Edmonds. Paths, trees, and flowers. *Canad. J. Math.*, 17:449–467, 1965.

[8] J. Edmonds. Maximum matching and a polyhedron with $0,1$ vertices. *J. of Res. the Nat. Bureau of Standards*, 69 B:125–130, 1965.

[9] S. Even and O. Kariv. $O(n^{2.5})$ algorithm for maximum matching in general graphs. In *FOCS*, pages 100–112, 1975.

[10] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In *STOC*, pages 95–106, 2012.

[11] N. Gillis and F. Glineur. On the geometric interpretation of the nonnegative rank, Sept. 04 2010. URL `http://arxiv.org/abs/1009.0880`.

[12] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993. ISBN 3-540-56740-2. doi: 10.1007/978-3-642-78240-4. URL `http://dx.doi.org/10.1007/978-3-642-78240-4`.

[13] E. Horowitz and S. Sahni. *Fundamentals of Computer Algorithms.* Computer Science Press, 1978.

[14] K. Pashkovich. *Extended formulations for combinatorial polytopes.* PhD thesis, Otto-von-Guericke-Universitẗ, Magdeburg, 2012.

[15] S. Pokutta and M. V. Vyve. A note on the extension complexity of the knapsack polytope. *To appear in Operations Research Letters*, 2013.

[16] T. Rothvoß. The matching polytope has exponential extension complexity. *CoRR*, abs/1311.2369, 2013.

[17] J. E. Savage. *Models of Computation: Exploring the Power of Computation.* `http://cs.brown.edu/~jes/book`, 2015.

[18] L. G. Valiant. Reducibility by algebraic projections. *Enseign. Math. (2)*, 28(3-4):253–268, 1982. ISSN 0013-8584.

[19] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.

Appendices

A Valid inequalities and facets of PM_n

We give here two classes of valid inequalities for PM_n. Firstly, let $M \subseteq E$ define a perfect matching in G. We have:

$$w \geqslant \sum_{ij \in M} x_{ij} - \frac{n}{2} + 1 \tag{16}$$

To see the validity of this inequality, note that if M is a perfect matching the sum becomes $n/2$ and the inequality states that $w \geqslant 1$, i.e. $G(x)$ has a perfect matching. We show below that the each inequality of type (16) define a facet of PM_n. Since the number of perfect matchings in K_n is the double factorial $(n-1)!! = (n-1)\cdot(n-3)...3\cdot 1$ the number of facet defining inequalities of PM_n is therefore super-polynomial.

For a second set of valid inequalities, first let E_n be the set of edges of K_n. A proper subset $S \subset E_n$ is *hypo-matchable* if it has no matching of size $n/2$ but the addition of any other edge from $E_n \setminus S$ to S yields such a matching. Then we have:

$$w \leqslant \sum_{ij \in E_n \setminus S} x_{ij} \tag{17}$$

To see the validity of this inequality note that if the sum is zero then no edges from $E_n \setminus S$ are in $G(x)$. So $G(x)$ has no perfect matching and so w must be zero.

We now prove that the inequalities (16) are facet defining for PM_{2n}, where we have replaced n by $2n$ to avoid fractions. For any integer s we use the notations $I_{s\times s}$, $\mathbb{1}_{s\times s}$ and $\mathbb{0}_{s\times s}$ to represent, respectively, the $s \times s$ identity matrix, matrix of all ones, and matrix of all zeroes. With only one subscript, the latter two notations represent the corresponding vectors. For an integer n we let $t = 2n(n-1)$. Without loss of generality, consider a perfect matching M in K_{2n} consisting of the n edges $12, 34, 56, ..., (2n-1)2n$ and let E_t be the t edges of K_{2n} that are not in M. We construct a set of $t+n = n(2n-1)$ graphs $G(x)$ for which inequality (16) is tight and for which the x vectors are affinely independent. The corresponding $(t+n+1)\times(t+n+1)$-matrix A of edge vectors x is:

$$A = \left[\begin{array}{c|c|c} I_{t\times t} & \mathbb{1}_{t\times n} & \mathbb{1}_t \\ \hline \mathbb{0}_{n\times t} & \mathbb{1}_{n\times n} - I_{n\times n} & \mathbb{0}_n \\ \hline \mathbb{0}_t & \mathbb{1}_n & 1 \end{array}\right] \tag{18}$$

We label the columns of A as follows. The first t columns correspond to the edges in E_t listed in lexicographical order by ij. The next n columns are indexed by the edges $12, 34, ..., (2n-1)2n$ of M and the final column by w. The first t rows of A consist of the edge vectors of graphs which contain M and precisely one other edge ij not in M, arranged in lexicographic order by ij. This means that the top left hand block in A is the identity matrix. Since all these graphs contain M, which is a perfect matching, all these remaining entries in the first t rows of A are ones.

The next n rows of A correspond to graphs with edge vectors $M \setminus \{ij\}$, where ij ranges over the perfect matching $12, 24, ..., (2n-1)2n$. Clearly the first block of these rows are all zeroes and the second block is $\mathbb{1}_{n\times n} - I_{n\times n}$. The last column is all zero since none of these graphs has a perfect matching. The final row of A corresponds to the graph M.

It is straight forward to perform row operations on A to transform it into an upper triangular matrix with ± 1 on the main diagonal. This can be performed by subtracting the last row from the preceding n rows. The middle block of A is now $-I_{n\times n}$. Finally these rows can then be added to the last row, which is then divided by $n-1$. It is then all zero except for the last column, which is -1. This completes the proof.

DECLARATION

The content of this work are based primarily on articles that I have coauthored. In cases of works by others I have attempted to cite the appropriate source correctly.

Prague, 2016

Hans Raj Tiwary