Extended formulations, non-negative factorizations and randomized communication protocols

Yuri Faenza[1], Samuel Fiorini[2], Roland Grappe[1], and Hans Raj Tiwary[2]

[1]Dipartimento di Matematica Pura e Applicata, Universitá di Padova Via Trieste 63, 35121 Padova, Italy.
Email: {faenza, grappe}@math.unipd.it
[2]Department of Mathematics, Université Libre de Bruxelles CP 216, Boulevard du Triomphe, 1050 Brussels, Belgium.
Email: {sfiorini, htiwary}@ulb.ac.be

September 2, 2011

Abstract

We show that the binary logarithm of the non-negative rank of a non-negative matrix is, up to small constants, equal to the minimum complexity of a randomized communication protocol computing the matrix in expectation. We use this connection to prove new conditional lower bounds on the sizes of extended formulations, in particular, for perfect matching polytopes.

1 Introduction

Extended formulations are a powerful tool for minimizing linear or, more generally, convex functions over polyhedra (see, e.g., Ziegler [20] for background on polyhedra and polytopes). Consider a polyhedron P in $\mathbb{R}^d$ and a convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$, that has to be minimized over P. If a small size linear description of P is known, then minimizing φ over P can be done efficiently using an interior point algorithm, or the simplex algorithm if φ 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 optimization problem at hand difficult. A fundamental result of Grötschel, Lovász and Schrijver [8] 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 useless practically. 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 define an *extended formulation*, or *extension* of P. Minimizing φ over P amounts to minimizing $\varphi \circ \pi$ 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 will not be the main focus here.

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 survey by Conforti, Cornuéjols and Zambelli [3].

In light of the above discussion, it is natural to define the *size* of an extension Q as the number of facets of Q, and the *extension complexity* of a polyhedron P as the minimum size of an extension of P. Following [6], we denote this by $\mathrm{xc}(P)$. The extension complexity of a polyhedron is a far better measure of how "complex" a polyhedron is than, for instance, its number of facets or its number of vertices and extreme rays.

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. In this case, one may assume without loss of generality that Q is also polytope (see below). So 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 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 non-negative $m \times n$ matrix $S = S(P) = (s_{ij})$ with $s_{ij} = h_i(v_j)$.

A *rank r non-negative factorization* of a non-negative matrix M is an expression of M as product $M = AB$ where A and B are non-negative matrices with r columns and r rows, respectively. The *non-negative rank* of M, denoted by $\mathrm{rank}_+(M)$, is the minimum natural r such that M admits a rank r non-negative factorization [2]. Observe that the non-negative rank of M can also be defined as the minimum $r \in \mathbb{N}$ such that M is the sum of r non-negative rank one matrices.

In a seminal paper, Yannakakis [19] proved, among other things, that the extension complexity of a polytope is precisely the non-negative rank of its slack matrix (see also [6]).

Theorem 1. *For all polytopes P,*

$$\mathrm{xc}(P) = \mathrm{rank}_+(S(P)) .$$

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 non-negative factorization $S(P) = FV$ of the slack matrix of P. Then it can be shown that $Q := \{(x, y) \in \mathbb{R}^{d+r} : Ex + Fy = g, y \geqslant 0\}$ is an extension of P. Notice that Q has r facets and r extra variables. Taking $r = \mathrm{rank}_+(S(P))$ implies $xc(P) \leqslant \mathrm{rank}_+(S(P))$. Moreover, in this case Q is bounded, see Lemma 11 in the appendix for a proof.

In the work of Yannakakis [19] also appeared a connection between extended formulations and communication complexity (the book of Kushilevitz and Nisan [11] is a standard reference on communication complexity). Every deterministic communication protocol computing a non-negative matrix M (traditionally M is a binary matrix) yields a non-negative factorization of M, and thus an extended formulation. Indeed, such a protocol defines a partition of the matrix into sub-matrices whose entries are all equal. Notice that the rows and columns of such a "monochromatic" sub-matrix are not necessarily consecutive. Each sub-matrix yields a non-negative rank one matrix, and the sum of the resulting matrices is precisely M. The rank of this non-negative factorization of M is at most 2^c, where c is the complexity of the protocol. When M is the slack matrix of a polytope P, we obtain an extension of P.

Notably, Yannakakis [19] used this connection to obtain a subexponential size extended formulation for the stable set polytope of a n-vertex perfect graph from a deterministic communication protocol computing the corresponding slack matrix with polylogarithmic complexity[1].

---

[1]This protocol was simplified by Hajnal, see the survey by Lovász [13].

The aim of this paper is to prove new results on extended formulations by tightening the connection between extended formulations, non-negative factorizations and communication complexity.

In Section 2, we define the different polytopes considered here, and describe their facets and vertices.

In Section 3, we discuss deterministic and randomized communication protocols and define what it means for a randomized communication protocols with private randomness to compute a given non-negative matrix M in expectation.

Then we prove in Sections 4 and 5 that the minimum complexity of a randomized protocol computing M in expectation is, up to small additive constants, the binary logarithm of the non-negative rank of M. This is done in two parts. Let c denote the minimum complexity of a randomized protocol computing M in expectation, and let $r := \text{rank}_+(M)$. First, in Section 4, we prove the inequality $\lg r \leqslant c$. Second, in Section 5, we prove the converse inequality $c \leqslant \lg r + O(1)$. The two inequalities together imply $\lg r = c + \Theta(1)$. By Theorem 1, we obtain a new characterization of the extension complexity of polytopes.

Finally, in Section 6, we use this characterization to prove new results on extended formulations of perfect matching polytopes, a prominent family of polytopes for which the extension complexity is unknown. Yannakakis [19] proved every *symmetric* extension that the perfect matching polytope of the complete graph K_n has exponential size. Here, we show roughly that there is a tradeoff between the amount of randomness used by an extension of the perfect matching polytope K_n, regarded as a randomized protocol, and the size of this extension. In particular, we prove that if the protocol detects non-zero entries of the slack matrix with constant probability, then the extension has exponential size.

Kaibel, Pashkovich and Theis [9] showed that the restriction to only symmetric extensions is a rather strong one. They showed that there exists polytopes with no polynomial size symmetric extension but that admit polynomial size extensions when the symmetry restriction is dropped. At the end of the section, we observe a similar phenomenon with our conditional lower bound. We show that every extension of the spanning tree polytope of K_n that, seen as a randomized protocol, detects non-zero entries of the slack matrix with constant probability has exponential size, even though an $O(n^3)$ size extension exists.

Finally we conclude this paper with a discussion of some related problems, conjectures, and future work.

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 [17].

Let I be a finite ground set. The *incidence 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 (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,

$$\begin{aligned} E[U] &= \{uv \in E : u \in U, v \in U\}, \text{ and} \\ \delta(U) &= \{uv \in E : u \in U, v \notin U\}. \end{aligned}$$

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 $E(K^n) = \{ij : i, j \in [n], i \neq j\}$.

2.1 Spanning Tree Polytope

A *spanning tree* of G is tree T 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 incidence vectors of the spanning trees of G, i.e.,

$$P_{\text{spanning tree}}(G) = \text{conv}\{\chi^{E(T)} \in \mathbb{R}^E : T \text{ spanning tree of } G\} .$$

Edmonds [5] showed that this polytope admits the following linear description (See also [17], page 861) :

$$
\begin{align}
x(E[U]) &\leqslant |U| - 1 \quad \text{for nonempty } U \subseteq V , \tag{1} \\
x(E) &= |V| - 1 , \tag{2} \\
x_e &\geqslant 0 \quad \text{for } e \in E . \tag{3}
\end{align}
$$

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 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 incidence vectors of the perfect matchings of G, i.e.,

$$P_{\text{perfect matching}}(G) = \text{conv}\{\chi^M \in \mathbb{R}^E : M \text{ perfect matching of } G\} .$$

Edmonds [4] showed that the perfect matching polytope of G is described by the following linear constraints (See also [16], page 438):

$$
\begin{align}
x(\delta(U)) &\geqslant 1 \quad \text{for } U \subseteq V \text{ with } |U| \text{ odd} , \tag{4} \\
x(\delta(v)) &= 1 \quad \text{for each } v \in V , \tag{5} \\
x_e &\geqslant 0 \quad \text{for } e \in E . \tag{6}
\end{align}
$$

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* $\text{STAB}(G)$ of a graph $G(V, E)$ is the convex hull of the incidence vectors of the stable sets in G, i.e.,

$$\text{STAB}(G) = \text{conv}\{\chi^S \in \mathbb{R}^V : S \text{ stable set of } G\} .$$

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 $\text{STAB}(G)$ for any graph G:

$$
\begin{align}
x(K) &\leqslant 1 \quad \text{for each clique } K \text{ of } G , \tag{7} \\
x_v &\geqslant 0 \quad \text{for } v \in V . \tag{8}
\end{align}
$$

Inequalities (7) are called the *clique inequalities*. See Schrijver [17] 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 the above inequalities completely describe $\text{STAB}(G)$.

3 Communication complexity

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 the value of $f(x,y)$, even though each player possesses unlimited computational power.

The communication is carried out as a protocol that is agreed on beforehand by Alice and Bob, on the sole basis of the function f. At each step of the protocol, one of the player has the token. He/she sends a bit to the other, that depends only on his/her input and on previously sent bits. The transmitted bit determines which player has the token in the next step. This is repeated until the value of f on (x,y) is known by 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.

In this section we describe deterministic protocols briefly and then randomized protocols (with private random bits). In the literature, a randomized protocol is said to compute a function f if for all inputs $(x,y) \in X \times Y$ the protocol outputs the correct value, namely $f(x,y)$, with high probability. Here we consider a new notion of computation where the value output by the protocol on input (x,y) has to equal $f(x,y)$ in expectation. For a thorough description of deterministic as well as randomized protocols (with the usual notion of computation) we refer the reader to [11].

3.1 Deterministic protocols

A protocol is best viewed as a rooted binary tree where each internal node is marked either Alice or Bob. The leaves have values associated with them. An execution of the protocol on a particular input is a root-to-leaf path in the tree. 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. In case the protocol is deterministic, to each input $(x,y) \in X \times Y$ corresponds a unique path from the root to one of the leaves, and the value at that leaf is $f(x,y)$. Thus none of the players uses any randomness to decide which bits to send to the other player.

More formally, we define a *deterministic protocol* as a rooted binary tree with some extra information attached to its nodes. Each internal node has a *type*, which is either X or Y. To each node v of type X is attached a function $p_v : X \to \{0,1\}$; to each node v of type Y is attached a function $q_v : Y \to \{0,1\}$; and to each leaf v is attached a number $\lambda_v \in \mathbb{R}_+$, called the *value* of that leaf.

An *execution* of the protocol on input $(x,y) \in X \times Y$ is a root-to-leaf path that starts at the root and descends to a leaf. At any internal node v of type X the execution follows the egde to the left child if $p_v(x) = 0$ and to the right child if $p_v(x) = 1$. Similarly, at any internal node v of type Y, the execution follows the edge to the left child if $q_v(y) = 0$ and to the right child if $q_v(x) = 1$. The *value of the execution* is the value of the leaf attained by the execution.

A deterministic protocol is said to *compute* the function f if for each input pair (x,y) the value of the execution of the protocol is exactly $f(x,y)$.

The *complexity* of a protocol is the height of the corresponding tree.

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.

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 $m \times n$ non-negative matrix $M = M(f) = (m_{ij})$ such that $m_{ij} = f(x_i, y_j)$ for all $(i, j) \in [m] \times [n]$.

Now consider a deterministic protocol computing f. Since the protocol is deterministic, each of its leaves v determines a subset of rows $R = R_v$ and columns $C = C_v$ such that any input (x_i, y_j), the execution of the protocol on (x_i, y_j) ends at leaf v if and only if $i \in R$ and $j \in C$, that is, $(i, j) \in R \times C$. On each of the inputs (x_i, y_j) with $(i, j) \in R \times C$, the function f evaluates to same value, namely the value at leaf v. The set $R \times C$ is called a *rectangle*.

When v varies among the leaves of the protocol, the rectangles $R_v \times C_v$ form a partition of $[m] \times [n]$. It is easy to see that such a partition can be used to write M as a sum of non-negative rank one matrices, one for each leaf. For each leaf v, define a $m \times n$ matrix M_v whose entry in the ith row and jth column is given by $f(x_i, y_j)$ if $i \in R_v$ and $j \in C_v$, and 0 otherwise. Thus the support of M_v is $R_v \times C_v$. Each of these matrices has rank at most one and we have that $M = \sum_{v \in L} M_v$, where L denotes the set of leaves of the protocol.

If M is the slack matrix of a polytope P, it follows from Theorem 1 that P has an extension of size at most $|L| \leqslant 2^c$, where c is the complexity of the protocol. This was first observed by Yannakakis [19]. He proved the existence of a $n^{O(\log n)}$ size extension for the stable set polytope of a n-vertex perfect graph by giving a $O(\log^2 n)$ complexity deterministic protocol for computing its slack matrix. We illustrate this by a related example, that appears to be new.

Example 1. 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., [17], page 1216). Recently Galluccio et al. gave a complete description of the facets of claw free graphs with at least one stable set of size greater than three [7]. Also, recall that for a perfect graph G the facets of $\text{STAB}(G)$ are defined by inequalities (7) and (8), see Section 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 $P := \text{STAB}(G)$ of G.

Because G is perfect, the (non-trivial part of the) slack matrix of P 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[2].

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 its clique K to Bob. Then Bob sends the names of all the vertices of its 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 extended formulation of $\text{STAB}(G)$ of size $O(n^3)$.

3.2 Randomized protocols

Randomized protocols are similar to deterministic ones except the players are allowed to use random bits to decide what to send to the other player. As mentioned earlier, the no-

---

[2]This describes the rows of the slack matrix of P that correspond to the clique inequalities (7). The slack matrix of P has another, shorter set of rows that correspond to non-negativity inequalities (8), and that we may safely ignore (see also Corollary 4 below).

tion of computation "in expectation" that we define here differs from the usual notion of computation "with high probability".

Let X and Y be finite sets. A *randomized protocol with private random bits* (or shortly, a *randomized protocol*) is a rooted binary tree with some extra information attached to the nodes. Each internal node has a *type*, which is either X or Y. To each node v of type X is attached a function $p_v : X \to [0,1]$; to each node v of type Y is attached a function $q_v : Y \to [0,1]$; and to each leaf v is attached a non-negative number $\lambda_v \in \mathbb{R}_+$, called the *value* of that leaf. The functions p_v and q_v define *transition probabilities*.

An *execution* of the protocol on input $(x, y) \in X \times Y$ is a random root-to-leaf path that starts at the root and descends to the left child of an internal node v with probability $p_v(x)$ if v is of type X and $q_v(y)$ if v is of type Y, and to the right child of v with the complementary probability $1 - p_v(x)$ if v is of type X and $1 - q_v(y)$ if v is of type Y. The *value* of the execution is the value of leaf attained by the execution.

For each fixed input $(x, y) \in (X, Y)$, the value of an execution on input (x, y) is a random variable. 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.

As observed in Section 3.1, we can regard a function $f : X \times Y \to \mathbb{R}_+$ as a non-negative matrix $M = M(f)$ with $m = |X|$ rows and $n = |Y|$ columns. Below, as is natural, we will not make a distinction between these two types of objects.

4 Factorizations from protocols

Theorem 2. *If there exists a randomized protocol of complexity c computing a matrix $M \in \mathbb{R}_+^{X \times Y}$ in expectation, then the non-negative rank of M is at most 2^c.*

Proof. 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. We claim that P_v is always a rank one matrix.

We prove this by induction on the depth of a node, starting from the root. When v is the root, P_v is an all-one matrix because all executions start at the root. Thus $P_v = \mathbf{1}\mathbf{1}^T$ is a rank one matrix in this case.

Next, consider a node u of depth at least one and its parent v. Without loss of generality, we assume that v is of type X, that is, v is assigned to Alice. Assume that $P_v = pq^T$ for some non-negative vectors $p \in \mathbb{R}^X$ and $q \in \mathbb{R}^Y$. Then we have $P_u = p'q^T$ where $p'(x) = p(x)p_v(x)$ for $x \in X$ in case u is the left child of v, and $p'(x) = p(x)(1 - p_v(x))$ for $x \in X$ in case u is the right child of v. This proves the claim.

Finally, let L be the set of all leaves of the protocol and λ_v be the value at leaf v. Because the protocol computes M in expectation, for all inputs $(x, y) \in X \times Y$ we have $M(x, y) = \sum_{v \in L} \lambda_v P_v(x, y)$. Thus, $M = \sum_{v \in L} \lambda_v P_v$. Since the claim holds, each term in this last sum is a non-negative rank one matrix. The theorem follows. $\qquad\square$

Recall that the polytopes considered in this paper have some facet-defining inequalities enforcing non-negativity of the variables along with other facet-defining inequalities. The next lemma and its corollary will allow us to ignore the rows corresponding to non-negativity inequalities, and focus on the non-trivial parts of the slack matrices considered here.

Lemma 3. *Let M be a non-negative matrix. Let R_1, R_2 be a partition of the rows of M defining partition of M into M_1 and M_2. If there exist randomized protocols computing M_1 and M_2 in*

expectation with complexity c_1 and c_2 respectively, then there exists a randomized protocol complexity computing M with complexity $1 + \max\{c_1, c_2\}$.

Proof. When Alice gets a row of M she sends a bit to Bob to indicate whether her row lies in R_1 or R_2. Now that both Alice and Bob know whether they want to compute an entry in M_1 or M_2, they use the protocol for that particular submatrix. $\quad\square$

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 non-negativity 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 non-negativity 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 non-negativity inequalities, this protocol has complexity $\lceil \lg d \rceil$. The corollary thus follows from Lemma 3. $\quad\square$

For the protocols constructed here, we will always have $c \geqslant \lceil \lg d \rceil$. Because of Corollary 4, we can thus ignore the non-negativity inequalities without blowing up the size of any extension by more than a factor of two. Moreover, in terms of lower bounds, it is always safe to ignore inequalities because the non-negative rank of a matrix cannot increase when rows are deleted.

To conclude this section, we give two illustrative examples. The first one is a reinterpretation of a well-known $O(n^3)$ size extended formulation for the spanning tree polytopes due to Martin [14]. The second one concerns the perfect matching polytopes and is implicit in Kaibel, Pashkovich and Theis [9].

Example 2. Let P denote the spanning tree polytope of the complete graph K^n, see Section 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 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, she 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 $3 \lg n + O(1)$. 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 directed edge (v, w) that will lead Alice to output a non-zero value, see Figure 1 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 $T[U]$.

The corresponding extended formulation has size $O(n^3)$.

For the next example, we will need the fact that one can cover the complete graph 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. See Lemma 12 and its proof in the appendix.
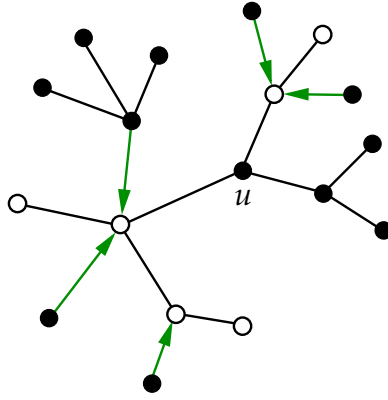
Figure 1: Illustration of the protocol in Example 2. The black vertices are those in U. The green directed edges are those for which Alice outputs a non-zero value. The number of such edges is the number of connected components of $T[U]$ minus one.

Example 3. Assume that n is even and let P denote the perfect matching polytope of the complete graph K^n with vertex set $[n]$, see Section 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 $\bar{U}$, 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 12. Then Alice checks which of X and $\bar{X}$ contains the least number of vertices of her odd set U. Without loss of generality, assume it is X. 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. 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 is

$$
\begin{aligned}
& (|U| - 1)\frac{|U \cap X| - |E[U] \cap M|}{|U \cap X|} + (|U| - 1 - 2|U \cap X|)\frac{|E[U] \cap M|}{|U \cap X|} \\
= & \ |U| - 1 - 2|U \cap X|\frac{|E(U) \cap M|}{|U \cap X|} \\
= & \ |U| - 2|E(U) \cap M| - 1 \\
= & \ |\delta(U) \cap M| - 1 \,.
\end{aligned}
$$

The resulting extension has size at most $2^{(1/2+\varepsilon)n} \leqslant (1.42)^n$ (for ε small enough, and n large enough), whereas the main result of Yannakakis [19] gives a lower bound of $\binom{n}{n/4} \geqslant (1.74)^n$ for the size of any *symmetric* extension.

5 Protocols from factorizations

Theorem 5. *If the non-negative rank of matrix $M \in \mathbb{R}_+^{m \times n}$ has a rank r non-negative factorization, then there exists a randomized protocol computing M in expectation, whose complexity is at most $\lg r + O(1)$.*

Proof. Let $A \in \mathbb{R}_+^{m \times r}$ and $B \in \mathbb{R}_+^{r \times n}$ be non-negative matrices such that $M = AB$. Let Δ denote the maximum row sum of A. Thus, $M = (A/\Delta)(\Delta B)$. Let $\widehat{A}$ denote the $m \times (r+1)$ matrix obtained from A/Δ by appending a column whose entries are chosen so that each row-sum of $\widehat{A}$ is precisely 1. Thus $\widehat{A}$ is row-stochastic. Let $\widehat{B}$ denote the $(r+1) \times n$ matrix obtained from ΔB by appending a zero row. Notice that $M = \widehat{A}\widehat{B}$.

The protocol is as follows: Alice knows a row index i, and Bob knows a column index j. Together they want to compute $M(i, j)$ in expectation, by exchanging as few bits as possible. They proceed as follows: Alice selects a column index $k \in [r+1]$ according to the probabilities found in row i of matrix $\widehat{A}$, sends this index to Bob, and Bob outputs the entry of $\widehat{B}$ in row k and column j.

This randomized protocol computes the matrix M in expectation. Indeed, the expected value on input (i, j) is $\sum_{k=1}^{r+1} \widehat{A}(i, k)\widehat{B}(k, j) = M(i, j)$. Moreover, the number of bits exchanged is $\lceil \lg(r+1) \rceil$, thus the complexity of the protocol is at most $\lg r + O(1)$. $\square$

Although the above protocol is defined literally, it is quite easy to obtain a protocol tree for it. The tree is balanced. All of its internal nodes are assigned to Alice, except those in the last layer which are assigned to Bob. There is one such node $v(k)$ for each possible index $k \in [r+1]$. The node $v = v(k)$ has two children which are both leaves: the left leaf has value $\max\{\widehat{B}(k, \ell) : \ell \in [n]\}$ and the right leaf 0. The transition probability $q_v(j)$ is defined in such a way that the expected value output is correct, that is,

$$q_v(j) \cdot \max\{\widehat{B}(k, \ell) : \ell \in [n]\} = \widehat{B}(k, j) \ .$$

Another property of the protocol is that it is *one-way*: one of the player (Bob in this case) does not send any bit to the other player. In some sense, the protocol can be regarded as being in "normal form". (Always putting a protocol in "normal form" does not seem desirable, for instance, when the protocol is deterministic.)

6 New lower bound for perfect matching polytopes

We have seen that every extension of a polytope P corresponds to a randomized protocol computing its slack matrix $S(P)$ in expectation and vice-versa. Now we show in particular that for the perfect matching polytope if we restrict ourselves only to those extensions that can determine with a constant probabilty whether or not an entry in the slack matrix is zero, then every extension has an exponential size. We then observe that the same holds for the spanning tree polytope.

6.1 A reduction from the set disjointness problem

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 function* 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 $[n]$. 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 bounded by a constant strictly less than 1) has $\Omega(n)$ complexity, see, e.g., Kushilevitz and Nisan [11], Babai et. al [1], Kalyanasundaram and Schnitger [10], and Razborov [15].

To each matrix $M \in \mathbb{R}_+^{X \times Y}$, we associate the following communication problem, that we call the *support problem*: Alice is given a row x of M and Bob a column y of M. They wish to determine whether $M(x, y) = 0$ or $M(x, y) > 0$. In the first case, they output 0 and in the second case they output 1.

Lemma 6. *There is a reduction from the set disjointness problem for subsets of $[n]$ to the support problem for the slack matrix of the perfect matching polytope for perfect matchings of K^ℓ, where $\ell \leqslant 3n + 8$, that uses $O(1)$ extra communication.*

Proof. Let A and B be the sets respectively given to Alice and Bob. After sending 1 bit with Alice, Bob and Alice can make sure that both B and its complement $[n] - B$ contain an even number of elements. They do this by adding dummy elements to the initial ground set $[n]$, without adding them to A.

Let $k \leqslant n + 2$ denote the number of elements currently in the ground set, and let $\ell := 3k + 2 \leqslant 3n + 8$. We define an odd set U and a perfect matching M as follows. First, we let

$$U := \{i : i \in A\} \cup \{i + k : i \in A\} \cup \{3k + 1\}.$$

Second, M is obtained 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 Figure 2 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. The theorem follows. $\qquad\qquad\square$
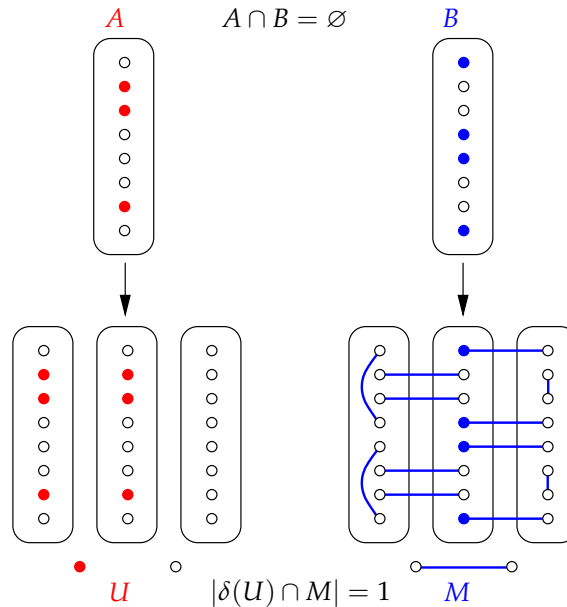

Figure 2: Constructing an odd set and a perfect matching from a set disjointness instance.

6.2 The new lower bound

Theorem 7. *Consider an extended formulation for the perfect matching polytope of K^n and a corresponding randomized protocol computing the slack matrix of this polytope in expectation. If the probability that the protocol outputs a non-zero value, given a pair (U, M) with positive slack, is at least $p(n) \gg 1/n$, then the protocol has complexity $\Omega(np(n))$ and the extended formulation has size $2^{\Omega(np(n))}$.*

Proof. Let c be the complexity of the randomized protocol computing the (non-trivial part of the) slack matrix of the perfect matching polytope of K^n in expectation. From this protocol, we obtain a new randomized protocol for the corresponding support problem by $\lceil 1/p(n) \rceil$ independent executions of the given protocol, and outputting 1 if at least one of the executions led to a non-zero value or 0 otherwise. The new protocol is such that, for all pairs (U, M) with a positive slack, the probability of outputting a zero value is at most

$$(1 - p(n))^{\frac{1}{p(n)}} \leqslant \frac{1}{e} ,$$

where e is the Euler's number. Thus, there is constant probability that the value returned by the algorithm is positive. This gives a protocol of complexity $O(c/p(n))$ for the support problem for the slack matrix of the perfect matching polytope. The theorem follows directly from Lemma 6 and from the fact that the set disjointness problem has randomized communication complexity $\Omega(n)$. $\qquad \square$

For instance, deterministic protocols for computing the slack matrix of the perfect matching polytope give rise to exponential size extended formulations. The same holds if $p(n)$ is a positive constant. For $p(n) = \Omega((\lg^2 n)/n)$, the size of the extended formulation is $n^{\Omega(\lg n)}$.

6.3 Implications for spanning tree polytopes

Incidently, an analogous statement holds for the spanning tree polytope of K^n as well, even though for this polytope an extended formulation of size $O(n^3)$ exists. To show this we first prove a lemma analogous to Lemma 6.

Lemma 8. *There is a reduction from the set disjointness problem for subsets of $[n]$ to the support problem for the slack matrix of the spanning tree polytope of K^ℓ, where $\ell = 2n + 1$, that uses no extra communication.*

Proof. For computing the slack matrix of the spanning tree polytope, Alice and Bob are given a set of vertices U and a spanning tree T respectively. They have to compute the number of connected components in $T[U]$ minus one in expectation. For computing the support of the slack matrix they have to decide whether T remains connected after throwing away the vertices that are not in U.

Given an instance of the set disjointness problem with sets $A, B \subseteq [n]$, Bob creates a spanning tree T on $\ell := 2n + 1$ vertices 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 Figure 3 for an example.

Alice lets $U := \{n + i : i \in A\} \cup \{2n + 1\}$. 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. Also, Alice and Bob do not need to communicate in order to construct the spanning tree T and the subset U. $\qquad \square$
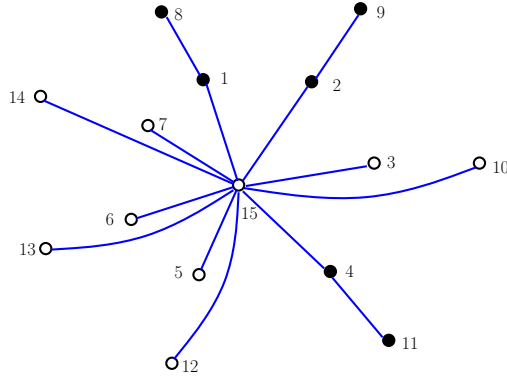
Figure 3: The spanning tree T for $B = \{1, 2, 4\}$ and $n = 7$. Black vertices are those of the form i or $n + i$ where $i \in B$.

Therefore, we have the following theorem.

Theorem 9. *Consider an extended formulation for the spanning tree polytope of K^n and a corresponding protocol computing the slack matrix of this polytope. If the probability that the protocol outputs a non-zero value, given a pair (U, T) with positive slack, is at least $p(n) \gg 1/n$, then the protocol has complexity $\Omega(np(n))$ and the extended formulation has size $2^{\Omega(np(n))}$.*

The proof is identical to that of Theorem 7 and so we omit it.

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.

Theorem 10. *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 $M \setminus \{e\}$ and sends it to Alice. She outputs $n - 2 = |M| - 1$ if $e' \in \delta(U)$ and zero otherwise. The expected value of the protocol is $(n - 2) \cdot (|\delta(U) \cap M| - 1)/(n - 2) = |\delta(U) \cap M| - 1$, as required. Bob needs to send the endpoints of the edge e' to Alice and this requires $2 \lg n + O(1)$ bits. $\square$

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 extended formulation for the perfect matching polytope of subexponential size. We leave it as an open question to settle the existence of such a protocol.

References

[1] László Babai, Péter Frankl, and János Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science (FOCS '86)*, pages 337–347, Los Angeles, Ca., USA, October 1986. IEEE Computer Society Press.

[2] Joel E. Cohen and Uriel G. Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and Its Applications*, 190:149–168, 1993.

[3] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010.

[4] Jack Edmonds. Maximum matching and a polyhedron with 0, 1 vertices. *Journal of Research National Bureau of Standards*, 69B:125–130, 1965.

[5] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.

[6] Samuel Fiorini, Volker Kaibel, Kanstantsin Pashkovich, and Dirk Oliver Theis. Combinatorial bounds on nonnegative rank and extended formulations. working paper, 2011.

[7] Anna Galluccio, Claudio Gentile, and Paolo Ventura. The stable set polytope of claw-free graphs with large stability number. *Electronic Notes in Discrete Mathematics*, 36:1025–1032, 2010.

[8] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993.

[9] Volker Kaibel, Kanstantsin Pashkovich, and Dirk Oliver Theis. Symmetry matters for the sizes of extended formulations. In *IPCO*, pages 135–148, 2010.

[10] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discrete Math.*, 5(4):545–557, 1992.

[11] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, Cambridge, 1997.

[12] László Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383 – 390, 1975.

[13] László Lovász. Communication complexity: a survey. In Bernhard Korte, László Lovász, Hans Jünger Prömel, and Alexander Schrijver, editors, *Paths, flows, and VLSI-Layout*, pages 235–265. Springer, 1990.

[14] R. Kipp Martin. Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10(3):119 – 128, 1991.

[15] Alexander A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

[16] Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Paths, flows, matchings, Chapters 1–38.

[17] Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. B*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Matroids, trees, stable sets, Chapters 39–69.

[18] Vijay V. Vazirani. *Approximation algorithms*. Springer, 2001.

[19] Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *J. Comput. System Sci.*, 43(3):441–466, 1991.

[20] Günter M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1995.

A Other proofs

Lemma 11. *Let $P = \{x \in \mathbb{R}^d : Ex \leqslant g\}$ be a polytope, let $S(P) = FV$ be a rank r non-negative factorization of the slack matrix of P with $r := \operatorname{rank}_+(S(P))$, and let $Q := \{(x, y) \in \mathbb{R}^{d+r} : Ex + Fy = g, y \geqslant 0\}$. Then Q is bounded if and only if P is bounded.*

Proof. Clearly, if P is unbounded then Q must be unbounded too. Now assume that P is bounded. The polyhedron Q is unbounded if and only if its recession cone $\operatorname{rec}(Q) = \{(x, y) \in \mathbb{R}^{d+r} : Ex + Fy = 0, y \geqslant 0\}$ contains some nonzero point. 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 $(x, y) \in \operatorname{rec}(Q)$. Therefore, Q is unbounded if and only if the system $Fy = 0, y \geqslant 0$ 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 non-negative, this is only possible if one of the columns of F is identically zero, which would contradict the minimality of r. $\qquad\square$

Lemma 12. *Let n be an even positive integer. There exists a collection of $k = O(2^{n/2}\sqrt{n}\ln n)$ subsets $X_1, \ldots, X_k$ of size $n/2$ of $[n]$ such that for every perfect matching M of K^n at least one of the subsets X_i is compatible with M, that is, all the edges of M have one end in X_i and the other in $\bar{X}_i = [n] \setminus X_i$.*

Proof. Consider the following set covering instance: the universe U is the set of all $\frac{n!}{2^{n/2}(n/2)!}$ matchings of K^n and there is a set $S = S(X)$ for each $(n/2)$-subset X of $[n]$ that contains all the perfect matchings M that are compatible with X. See, e.g., Vazirani [18] for a detailed description of the set cover problem and approximation algorithms for it.

A feasible fractional solution of this set covering instance takes each set X to an extent of $1/2^{n/2}$. This gives a feasible fractional solution because each given perfect matching M is compatible with exactly $2^{n/2}$ subsets X. (By symmetry considerations, it is possible to argue that this solution is actually optimal.) The value of this fractional solution is

$$\frac{1}{2^{n/2}}\binom{n}{n/2} \leqslant \frac{2^{n/2}}{\sqrt{n}},$$

at least for n sufficiently large. Thus the feasible integer solution given by the greedy algorithm [12] is of size at most

$$\left(1 + \ln\frac{n!}{2^{n/2}(n/2)!}\right)\frac{2^{n/2}}{\sqrt{n}} = O(2^{n/2}\sqrt{n}\lg n).$$

$\qquad\square$