Mathematical Programming (2020) 183:41–59
https://doi.org/10.1007/s10107-020-01535-9

FULL LENGTH PAPER

Series B

Extended formulations from communication protocols in output-efficient time

Manuel Aprile[1] · **Yuri Faenza**[2]

Received: 29 May 2019 / Accepted: 17 June 2020 / Published online: 3 July 2020

Abstract

Deterministic protocols are well-known tools to obtain extended formulations, with many applications to polytopes arising in combinatorial optimization. Although constructive, those tools are not output-efficient, since the time needed to produce the extended formulation also depends on the number of rows of the slack matrix (hence, on the exact description in the original space). We give general sufficient conditions under which those tools can be implemented as to be output-efficient, showing applications to e.g. Yannakakis' extended formulation for the stable set polytope of perfect graphs, for which, to the best of our knowledge, an efficient construction was previously not known. For specific classes of polytopes, we give also a direct, efficient construction of extended formulations arising from protocols. Finally, we deal with extended formulations coming from unambiguous non-deterministic protocols.

Keywords Communication protocols · Extended formulations · Perfect graphs

Mathematics Subject Classification 52B11 · 90C05 · 90C27 · 05C17 · 05C69 · 05C85

1 Introduction

Let P, Q be polytopes such that Q linearly projects to P. Q is an *extension* of P, any set of linear inequalities describing Q is an *extended formulation* for P, and the minimum number of inequalities in an extended formulation for P is called the *extension complexity* of P. It is denoted by $\mathrm{xc}(P)$. Computing or bounding the extension

✉ Manuel Aprile
manuelf.aprile@gmail.com

Yuri Faenza
yf2414@columbia.edu

1 Université Libre de Bruxelles, Bruxelles, Belgium

2 IEOR, Columbia University, New York, USA

complexity of polytopes has been a central area of research in optimization, see e.g. [8,16,28].

Most upper bounds on extension complexity are constructive and produce an extended formulation in time polynomial (often linear) in its size. Examples of the latter include Balas' union of polytopes, reflection relations, and branched polyhedral systems (see e.g. [11,20]). The fact that we can construct extended formulations *efficiently* is crucial, since their final goal is to make certain optimization problems (more) tractable. It is interesting to observe that there is indeed a gap between the *existence* of certain extended formulations, and the fact that we can construct them efficiently. For instance, in [6], it is shown that there is an extended formulation for the stable set polytope that is $O(\sqrt{n})$-approximated and has size $O(n)$ (with n being the number of nodes of the graph), but we do not expect to obtain it efficiently because of known hardness results [19]. In another case, a proof of the existence of a subexponential formulation with integrality gap $2+\epsilon$ for min-knapsack [5] predated the efficient construction of a formulation with these properties [15].

In this paper, we investigate the efficiency of an important tool for producing extended formulations: communication protocols. In a striking result, Yannakakis [31] showed that a deterministic communication protocol computing the slack matrix of a polytope $P=\{x : Ax \leq b\} \subseteq \mathbb{R}^n$ can be used to produce an extended formulation for P. The number of inequalities of the latter is at most 2^c, where c is the *complexity* of the protocol (see Sect. 2 for definitions). Hence, deterministic protocols can be used to provide upper bounds on the extension complexity of polytopes. However, *producing* this formulation requires a time at least linear in the number of rows of A (we refer to [31] for details). The main application of Yannakakis' technique is arguably given in his original paper, and it provides a relaxation of the stable set polytope. In particular, it gives an exact formulation for the stable set polytope when G is a perfect graph. This is a class of polytopes that has received much attention in the literature, see e.g. [10,18], and their extension complexity is one of the main open problems in the area. Yannakakis' protocol gives an upper bound of $n^{O(\log n)}$, while the best lower bound is as small as $\Omega(n \log n)$ [3]. On the other hand, a maximum stable set in a perfect graph can be computed efficiently via a *semidefinite* extension of polynomial size known as Lovasz' Theta body [24]. We remark that designing a combinatorial (or at least SDP-free) polynomial-time algorithm to find a maximum stable set in perfect graphs is a main open problem, see [9]. Such an algorithm could also be applied to efficiently and optimally color a perfect graph without using the ellipsoid method, see again [9].

Our results In this paper, we investigate conditions under which we can explicitly obtain an extended formulation from a communication protocol in time polynomial in the size of the formulation itself. We first show a general algorithm that achieves this for any deterministic protocol, given a compact representation of the protocol as a labelled tree and of certain extended formulations associated to leaves of the protocol. The algorithm runs in time linear in the input size and is flexible, in that it also handles non-exact extended formulations. We show applications of our techniques in the context of (not only) perfect graphs. Our most interesting application is to Yannakakis' original protocol, and it leads to an extended formulation of size $n^{O(\log n)}$ that can be constructed in time $n^{O(n \log n)}$. For perfect graphs, this gives a subexponential SDP-free algorithm that computes a maximum stable set (resp. an optimal coloring). For general graphs, this gives a new relaxation of the stable set polytope which is (strictly) contained in the clique relaxation. Finally, we extend our result to construct extended formulations from unambiguous non-deterministic protocols.

Conference version In comparison to the one that appeared in the proceedings of IPCO 2019 [2], this version contains full proofs of results from Sects. 4 and 5, as well as extension of other results, and the completely new Sect. 6.

2 Preliminaries

Communication protocols We start by describing the general setting of communication protocols, referring to [21] for more details. Let M be a non-negative matrix with row (resp. column) set X (resp. Y), and two agents Alice and Bob. Alice is given as input a row index $i \in X$, Bob a column index $j \in Y$, and they aim at determining M_{ij} by exchanging information according to some pre-specified mechanism, that goes under the name of *deterministic protocol*. At each step of the protocol, the actions of Alice (resp. Bob) only depend on her (resp. his) input and on what they exchanged so far. Similarly, the value output by the protocol also depends solely on the exchanged messages between Alice and Bob. The protocol is said to *compute* M if, for any input i of Alice and j of Bob, it returns M_{ij}. Such a protocol can be modelled as a rooted tree (which we assume w.l.o.g. to be full, i.e., each non-leaf node has two children), with each vertex modelling a step where exactly one of Alice or Bob sends a bit (hence the vertex is labelled with A or B), and its children representing subsequent steps of the protocol. The *leaves* of the tree indicate the termination of the protocol and are labelled with the corresponding output. The tree is therefore binary, with each edge representing a 0 or a 1 sent. Hence, a deterministic protocol can be identified by the following parameters: a rooted binary tree τ with node set $\mathcal{V}$; a *type* function $\ell : \mathcal{V} \to \{A, B\}$ ("Alice","Bob"), associating each vertex to its type; for each leaf $v \in \mathcal{V}$, a non-negative number Λ_v output at v; for each $v \in \mathcal{V}$, the set S_v of pairs $(i, j) \in X \times Y$ such that, on input (i, j), the step corresponding to node v is executed during the protocol. We represent this compactly by $(\tau, \ell, \Lambda, \{S_v\}_{v\in\mathcal{V}})$. It can be shown that each S_v is a *rectangle*, i.e., it defines a submatrix of M. For a leaf v of τ, all entries of S_v have the same value Λ_v, i.e. they form a submatrix of M with entries all equal to Λ_v. Such submatrices are called *monochromatic rectangles*, their collection is denoted by $\mathcal{R}$, and we say that $\Lambda_R := \Lambda_v$ is the *value* of the rectangle R associated to the leaf v. We assume that $S_v \neq \emptyset$ for each node v of the protocol.

An *execution* of the protocol is a path of τ from the root to a leaf, whose edges correspond to the bits sent during the execution. The *complexity* of the protocol is given by the height h of the tree τ. A deterministic protocol computing M gives a partition of M in at most 2^h monochromatic rectangles. The *size* of the tree τ, i.e. its number of vertices, is at most $2^{h+1} - 1$. We remark that one can obtain a protocol (and a partition in rectangles) for M^T given a protocol for M by just exchanging the roles of Alice and Bob.

Extended formulations and how to find them We follow here the framework introduced in [25], that extends [31]. Consider a pair of polytopes (P, Q) with $P = \mathrm{conv}(v_1, \ldots, v_n) \subseteq Q = \{x \in \mathbb{R}^d : Ax \leq b\} \subset \mathbb{R}^d$, where A has m rows $a_1, \ldots, a_m$.

A polytope $R \in \mathbb{R}^{d'}$ is an *extension* for the pair (P, Q) if there is a projection $\pi : \mathbb{R}^{d'} \to \mathbb{R}^d$ such that $P \subseteq \pi(R) \subseteq Q$. An *extended formulation* for (P, Q) is a set of linear inequalities describing R as above, and the minimum number of inequalities in an extended formulation for (P, Q) is its *extension complexity*. The *slack matrix* $M(P, Q)$ of the pair (P, Q) is the non-negative $m \times n$ matrix with $M(P, Q)_{i,j} = b_i - a_i v_j$, where a_i is the i-th row of A. A *non-negative factorization* of M is a pair of non-negative matrices (T, U) such that $M = TU$. The *non-negative rank* of M is the smallest intermediate dimension in a non-negative factorization of M.

Theorem 1 ([25] Yannakakis' Theorem for pairs of polytopes) *Given a slack matrix M of a pair of polytopes (P, Q) of dimension at least* 1*, the extension complexity of (P, Q) is equal to the non-negative rank of M. In particular, if $M = TU$ is a non-negative factorization of M, then $P \subseteq \{x : \exists\, y \geq 0 : Ax + Ty = b\} \subseteq Q$.*

Hence, a factorization of the slack matrix of intermediate dimension N gives an extended formulation of size N (i.e., with N inequalities). However such formulation has as many equations as the number of rows of A.

Now assume we have a deterministic protocol of complexity c for computing $M = M(P, Q)$. The protocol gives a partition of M into at most 2^c monochromatic rectangles. This implies that $M = R_1 + \cdots + R_N$, where $N \leq 2^c$ and each R_i is a rank 1 matrix corresponding to a monochromatic rectangle of non-zero value. Hence M can be written as a product of two non-negative matrices T, U of intermediate dimension N, where $T_{i,j} = 1$ if the (monochromatic) rectangle R_j contains row index i and 0 otherwise, and $U_{i,j}$ is equal to the value of R_i if R_i contains column index j, and 0 otherwise. As a consequence of Theorem 1, this yields an extended formulation for (P, Q). In particular, let $\mathcal{R}^{\neq 0}$ be the set of monochromatic, non-zero (i.e., of value different from 0) rectangles of M produced by the protocol and, for $i = 1, \ldots, m$, let $\mathcal{R}_i^{\neq 0} \subset \mathcal{R}^{\neq 0}$ be the set of rectangles whose row index set includes i. Then the following is an extended formulation for (P, Q):

$$
\begin{aligned}
a_i x + \sum_{R \in \mathcal{R}_i^{\neq 0}} y_R = b_i \quad \forall\, i = 1, \ldots, m \\
y \geq 0
\end{aligned} \tag{1}
$$

Again, the formulation has as many equations as the number of rows of A, and it is not clear how to get rid of non-redundant equations efficiently. Note that all definitions and facts from this section specialize to those from [31] for a single polytope when $P = Q$.

Stable set polytope and **QSTAB(G)** The stable set polytope STAB(G) is the convex hull of the characteristic vectors of stable (also, independent) sets of a graph G. It has (almost) exponential extension complexity [16,17]. The *clique relaxation* of STAB(G) is:

$$
\mathrm{QSTAB}(G) = \left\{ x \in \mathbb{R}^d_+ : \sum_{v \in C} x_v \leq 1 \text{ for all cliques } C \text{ of } G \right\}. \tag{2}
$$

Note that in (2) one could restrict to maximal cliques, even though in the following we will consider all cliques when convenient. As a consequence of the equivalence between separation and optimization, optimizing over $\mathrm{QSTAB}(G)$ is NP-hard for general graphs, see e.g. [29]. However, the clique relaxation is exact for perfect graphs, for which the optimization problem is polynomial-time solvable using semidefinite programming (see Sect. 1):

Theorem 2 [10] *A graph G is perfect if and only if* $\mathrm{STAB}(G) = \mathrm{QSTAB}(G)$.

A fundamental result from [31] is the following.

Theorem 3 *Let G be a graph with n vertices. There is a deterministic protocol of complexity $O(\log^2 n)$ computing the slack matrix of the pair* $(\mathrm{STAB}(G), \mathrm{QSTAB}(G))$. *Hence, there is an extended formulation of size $n^{O(\log n)}$ for* $(\mathrm{STAB}(G), \mathrm{QSTAB}(G))$.

We remark that, when G is perfect, Theorem 3 gives a quasipolynomial sized extended formulation for $\mathrm{STAB}(G)$. However, as discussed above, it does not give a quasi-polynomial (or even subexponential) time algorithm to obtain such formulation.

3 A general approach

We present here a general technique to explicitly and efficiently produce extended formulations from deterministic protocols. We start with an informal discussion.

It is important to address the issue of what is our input, and what assumptions we need in order to get an "efficient" algorithm. Recall that a deterministic protocol is identified with a tuple $(\tau, \ell, \Lambda, \{S_v\}_{v\in\mathcal{V}})$, and the set of monochromatic rectangles corresponding to leaves of the protocol is denoted by $\mathcal{R}$. We will assume that τ, ℓ and certain information on $\mathcal{R}$ are given to us explicitly, as their size is linearly dependent to the size of the extended formulation we want to produce. On the other hand, the matrix Λ and the sets S_v are not part of our input, since they have in general exponential size.

The natural approach to reduce the size of (1) is to eliminate redundant equations. However, the structure of the coefficient matrix depends both on Λ and on rectangles $\mathcal{R}_i$'s of the factorization, which can have a complex behaviour. The reader is encouraged to try e.g. on the extended formulations obtained via Yannakakis' protocol for $\mathrm{STAB}(G)$, G perfect: the sets $\mathcal{R}_i$'s have very non-trivial relations with each other that depend heavily on the graph, and we did not manage to directly reduce the system (1) for general perfect graphs.

Our algorithm bypasses this problem. It defines, for each node v of the full binary tree describing the protocol, a pair of polytopes $P_v \subseteq Q_v \subseteq \mathbb{R}^d$ such that: an extended formulation for (P_v, Q_v) can be constructed in time linear in the sum of the sizes of extended formulations for $(P_{v'}, Q_{v'})$ and $(P_{v''}, Q_{v''})$, where v', v'' are the children of v (if any); when ρ is the root node, $P_\rho = P$ and $Q_\rho = Q$. This shifts the problem from eliminating redundant equations from the system (1) to describing one extended formulation for each pair (P_R, Q_R) associated to a monochromatic rectangle R (i.e., to a leaf of the protocol). A formal description is given in Algorithm 1 and a proof of correctness in Theorem 5. Moreover, as we observe in Lemma 1, an extended formulation for (P_R, Q_R) is given by $\{A_R x = b_R - \mathbf{1}\Lambda_R\}$, where A_R is an appropriate submatrix of the constraint matrix of Q, and Λ_R is the value of R. Note that this system *does not have any additional variable*, since Λ_R is a real number. Hence, we can expect that it is easier to deal with than the original extended formulation (1), whose number of additional variables is linear in the size of the formulation itself. Although this is not a formal statement, we substantiate it by showing interesting examples where this is the case. We remark that we need to deal with a system as above for *every* monochromatic rectangle R, including those with $\Lambda_R = 0$.

We now switch gears and make the discussion formal, by presenting a well-known theorem by Balas [4], in a version given by Weltge [30, Section 3.1.1].

Theorem 4 *Let $P_1, P_2 \subset \mathbb{R}^d$ be non-empty polytopes, with $P_i = \pi_i(\{y \in \mathbb{R}^{m_i} : A^i y \le b^i\})$, where $\pi_i : \mathbb{R}^{m_i} \to \mathbb{R}^d$ is a linear map, for $i = 1, 2$. Let $P = \mathrm{conv}(P_1 \cup P_2)$. Then we have:*

$$\begin{aligned} P = &\{x \in \mathbb{R}^d : \exists\, y^1 \in \mathbb{R}^{m_1}, y^2 \in \mathbb{R}^{m_2}, \lambda \in \mathbb{R} : x = \pi_1(y^1) + \pi_2(y^2), \\ &A^1 y^1 \le \lambda b^1, A^2 y^2 \le (1-\lambda) b^2, 0 \le \lambda \le 1\}. \end{aligned}$$

Moreover, the inequality $\lambda \ge 0$ ($\lambda \le 1$ respectively) is redundant if P_1 (P_2) has dimension at least 1. Hence $\mathrm{xc}(P) \le \mathrm{xc}(P_1) + \mathrm{xc}(P_2) + |\{i : \dim(P_i) = 0\}|$.

We say that the extended formulation for P as in Theorem 4 is obtained by applying the *Balas operator* to the extended formulations for P_1, P_2.

Algorithm 1 Construction of an extended formulation from a deterministic protocol

Require: A full binary tree τ with nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$ ordered so that every non-leaf node follows both its children, a type function $\ell : \mathcal{V} \to \{A, B\}$. For each leaf R of τ, a linear description (possibly, an extended formulation) T_R of a polytope in $\mathbb{R}^d$.

for $i = 1, \ldots, N$ **do**
  if $v = v_i$ is a leaf of τ corresponding to a rectangle R **then**
    Let $T_v := T_R$.
  else
    Let v', v'' be the children of v.
    if $\ell(v) = A$ **then**
      Let T_v be the juxtaposition of $T_{v'}$ and $T_{v''}$.
    else
      Let T_v be obtained by applying the Balas operator to $T_{v'}$ and $T_{v''}$.
    end if
  end if
end for
Output T_{v_N}.

Theorem 5 *Let (P, Q) be a pair of polytopes with $P = \mathrm{conv}\{x_1^*, \ldots, x_n^*\} \subseteq Q = \{x \in \mathbb{R}^d : a_i x \le b_i$ for $i = 1, \ldots, m\} \subset \mathbb{R}^d$. For $j \in [d]$, let q_j, u_j be reals with $q_j \le x_j \le u_j$ for all $x \in Q$. Let $(\tau, \ell, \Lambda, \{S_v\}_{v \in \mathcal{V}})$ be a deterministic protocol with set of rectangles $\mathcal{R}$ computing the slack matrix of the pair (P, Q). For each $R \in \mathcal{R}$, let T_R be an extended formulation for (P_R, Q_R), where $P_R := \mathrm{conv}\{x_j^* : j$ is a column of $R\}$ and $Q_R := \{x \in \mathbb{R}^d : a_i x \le b_i\ \forall\, i$ row of $R;\ q_j \le x_j \le u_j$ for all $j \in [d]\}$.*

Then Algorithm 1 on input $(\tau, \ell, \{T_R\}_{R\in\mathcal{R}})$ outputs an extended formulation for (P, Q) of size linear in $\sum_{R\in\mathcal{R}} \sigma(T_R)$ in time linear in $\sum_{R\in\mathcal{R}} \sigma_+(T_R)$, where, for every $R \in \mathcal{R}$:

- *$\sigma(T_R)$ is the size (number of inequalities) of T_R;*
- *$\sigma_+(T_R)$ is the total encoding length of the description of T_R (including the number of inequalities, variables and equations).*

Proof Recall that S_v is the (non-necessarily monochromatic) rectangle given by all pairs (i, j) such that, on input (i, j), the execution of the protocol visits node v. Let us define, for any such S_v, a pair (P_v, Q_v) with $P_v = \text{conv}\{x_j^* : j \text{ is a column of } S_v\}$ and

$$Q_v = \{x \in \mathbb{R}^d : a_i x \leq b_i \ \forall\, i \text{ row of } S_v;\ \ q_j \leq x_j \leq u_j \text{ for all } j \in [d]\}.$$

Clearly $P_v \subseteq P \subseteq Q \subseteq Q_v$, and Q_v is a polytope. Moreover, $S_\rho = S$, $P_\rho = P$, and $Q_\rho = Q$ for the root ρ of τ.

We claim that Algorithm 1 correctly computes an extended formulation T_v of (P_v, Q_v) for every node v of τ. This, in particular, implies that the output of Algorithm 1 is correct. The proof is by induction on the distance of v from its descendant in the protocol that is farthest away. If this distance is 0 (i.e., v is a leaf), the claim follows by definition. Else, by induction hypothesis, we have that $T_{v'}$ and $T_{v''}$ are extended formulations for the pairs $(P_{v'}, Q_{v'})$ and $(P_{v''}, Q_{v''})$, respectively.

Assume first that v is labelled A. Then we have $S_v^T = \left[S_{v'}\ S_{v''}\right]^T$ (up to permutation of rows of S_v), since the bit sent by Alice at v splits S_v in two rectangles by rows – those corresponding to rows where she sends 1 and those corresponding to rows where she sends 0. Therefore $P_v = P_{v'} = P_{v''}$ and $Q_v = Q_{v'} \cap Q_{v''}$. Hence we have $P_v \subseteq \pi'(T_{v'}) \cap \pi''(T_{v''}) \subseteq Q_v$, where π' (resp. π'') is a projection from the space of $T_{v'}$ (resp. $T_{v''}$) to $\mathbb{R}^d$. Hence, the juxtaposition T_v of the formulations of $T_{v'}, T_{v''}$ is an extended formulation for $\pi'(T_{v'}) \cap \pi''(T_{v''})$.

Now assume that v is labelled B. Then similarly we have $S_v = \left[S_{v'}\ S_{v''}\right]$ (up to permutations of columns of S_v). Hence, $P_v = \text{conv}\{P_{v'} \cup P_{v''}\}$ and $Q_v = Q_{v'} = Q_{v''}$. This implies $P_v \subseteq \text{conv}\{\pi'(T_{v'}) \cup \pi''(T_{v''})\} \subseteq Q_v$, where π' and π'' are defined as in the previous case. Hence, applying Theorem 4 to the formulations of $T_{v'}, T_{v''}$ gives an extended formulation T_v for $\text{conv}\{\pi'(T_{v'}) \cup \pi_1(T_{v''})\}$, as required.

We now bound the encoding size of formulation. If $T_v = \pi'(T_{v'}) \cap \pi''(T_{v''})$, then $\sigma_+(T_v) \leq \sigma_+(T_{v'}) + \sigma_+(T_{v''})$. Consider now $T_v = \text{conv}\{\pi'(T_{v'}) \cup \pi''(T_{v''})\}$. From Theorem 4 we have $\sigma_+(T_v) \leq \sigma_+(T_{v'}) + \sigma_+(T_{v''}) + O(d)$. The binary tree associated to the protocol has size at most linear in the number of leaves, hence for the final formulation T_ρ we have

$$\sigma_+(T_\rho) \leq \sum_{R\in\mathcal{R}} (\sigma_+(T_R) + O(d)) = O\left(\sum_{R\in\mathcal{R}} \sigma_+(T_R)\right),$$

where the last equation is justified by the fact that we can assume $\sigma_+(T_R) \geq d$ for any $R \in \mathcal{R}$. The bounds on the size of T_ρ and on the time needed to construct the formulation are derived analogously. □

A couple of remarks on Theorem 5 are in order. The reader may recognize similarities between the proof of Theorem 5 and that of the main result in [15], where a technique is given to construct approximate extended formulations for polytopes using Boolean formulas. While similar in flavour, those two results seem incomparable, in the sense that one does not follow from the other. They both fall under a more general framework, in which one derives extended formulations for a polytope by associating a tree to it, starting from simpler formulations and taking intersection and convex hulls following the structure of the tree. However, we are not aware of any other applications of this framework, hence we do not discuss it here.

Algorithm 1 deals with binary protocols. In the applications, however, we will mostly see non-binary protocols. This is without loss of generality, for every non-binary protocol can be turned into a binary one by replacing each node with an appropriate sequence of nodes with the same label. Restricting the theory to binary protocols allows for leaner statements and is customary in the literature, see, e.g., [12, Page 83], [21, Page 4].

The formulation that is produced by Theorem 5 may not have *exactly* the form given by the corresponding protocol. Also, even for the special case $P = Q$, the proof relies on the version of Yannakakis' theorem for pairs of polytopes. On the other hand, it does not strictly require that we reach the leaves of the protocol—a similar bottom-up approach works starting at any node v, as long as we have an extended formulation for (P_v, Q_v). However, if we indeed reach the leaves, there is an extended formulation for each (P_R, Q_R) that has a very special structure, as next lemma shows.

Lemma 1 *Using the notation from Theorem 5, let $R \in \mathcal{R}$ and denote by $\Lambda_R \geq 0$ its value. Then we have that $P_R \subseteq T^*_R \subseteq Q_R$, where*

$$T^*_R := \{A_R x + \mathbf{1}\Lambda_R = b_R,\ b_j \leq x_j \leq u_j \text{ for all } j \in [d]\}, \tag{3}$$

$\mathbf{1}$ is the all-1 vector of appropriate length, and A_R (resp. b_R) is the submatrix (resp. subvector) of the constraint matrix (resp. of the right-hand side) describing Q corresponding to rows of R.

Proof By construction, every vertex of P_R satisfies $A_R x + \mathbf{1}\Lambda_R = b_R$, hence $P_R \subseteq T^*_R$. Moreover, since $\Lambda_R \geq 0$, $A_R x \leq b_R$ is clearly valid for T^*_R, showing $T^*_R \subseteq Q_R$. □

We remark that, in case the value of a monochromatic rectangle R is not known, one can just replace in (3) Λ_R with a variable y_R, together with the constraint $y_R \geq 0$: similarly as in the proof above, one checks that the resulting T^*_R is an extended formulation of (P_R, Q_R).

4 Applications

(STAB(G), QSTAB(G)). We now describe how to apply Theorem 5 to the protocol from Theorem 3 as to obtain an extended formulation for (STAB(G), QSTAB(G)) in output-efficient time. In particular we show:

Theorem 6 *Let G be a graph on n vertices. Then there is an algorithm that, on input G, outputs an extended formulation for* $(\mathrm{STAB}(G), \mathrm{QSTAB}(G))$ *of size $n^{O(\log n)}$ in time $n^{O(\log n)}$. If moreover G is also perfect, then the output is an extended formulation of* $\mathrm{STAB}(G)$.

We first give a modified version of the protocol from [31], stressing a few details that will be important in the following. The reader familiar with the original protocol can immediately verify its correctness. Let $v_1, \ldots, v_n$ be the vertices of G in any order. At the beginning of the protocol, Alice is given a clique C of G as input and Bob a stable set S, and they want to compute the entry of the slack matrix of $\mathrm{STAB}(G)$ corresponding to C, S, i.e. to establish whether C, S intersect or not.

At each stage of the protocol, the vertices of the current graph $G = (V, E)$ are partitioned between *low degree* L (i.e., at most $|V|/2$) and *high degree* H. Suppose first $|L| \geq |V|/2$. Alice sends (i) the index of the low degree vertex of smallest index in C, or (ii) 0 if no such vertex exists. In case (i), if $v_i \in S$, then $C \cap S \neq \emptyset$ and the protocol ends; else, G is replaced by $G[N(v_i) \setminus \{v_j \in L : j < i\}]$, where $G[U]$ denotes the subgraph of G induced by U. In case (ii), if Bob has no high degree vertex, then $C \cap S = \emptyset$ and the protocol ends; else, G is replaced by $G[H]$.

Now suppose conversely that $|L| < |V|/2$, then the protocol proceeds symmetrically to above: Bob sends (i) the index of the high degree vertex of smallest index in S, or (ii) 0 if no such vertex exists. In case (i), if $v_i \in C$, then $C \cap S \neq \emptyset$ and the protocol ends; else, G is replaced by $G[V \setminus (N(v_i) \cup \{v_j \in H : j < i\})]$. In case (ii), if Alice has no low degree vertex, then $C \cap S = \emptyset$ and the protocol ends, else, G is replaced by $G[L]$. Note that at each step the number of vertices of the graph is decreased by at least half, and C and S do not intersect in any of the vertices that have been removed.

Now, let S be the slack matrix of the pair $(\mathrm{STAB}(G), \mathrm{QSTAB}(G))$. Each monochromatic rectangle $R \in \mathcal{R}$ in which the protocol from Theorem 3 partitions S is univocally identified by the list of cliques and of stable sets corresponding to its rows and columns. With a slight abuse of notation, for a clique C (resp. stable set S) whose corresponding row is in R, we write $C \in R$ (resp. $S \in R$), and we also write $(C, S) \in R$. We let P_R be the convex hull of characteristic vectors of stable sets $S \in R$ and Q_R the set of points satisfying all clique inequalities corresponding to cliques $C \in R$, together with the unit cube constraints.

We need a fact on the structure of $\mathcal{R}$, for which we introduce some more notation. For a (monochromatic) rectangle $R \in \mathcal{R}$, f R has value 1, let C_R be the set of vertices sent by Alice and S_R the set of vertices sent by Bob during the corresponding execution of the protocol; if R has value 0, i.e. it corresponds to pairs (C, S) sharing a vertex v_i, then S_R, C_R are defined as above, except that we include v_i in both S_R, C_R. Note that C_R is a clique and S_R is a stable set.

Observation 1 *For each $R \in \mathcal{R}$, there is exactly one clique C and one stable set S of G such that $C = C_R$ and $S = S_R$. Conversely, given a clique C and a stable set S, there is at most one rectangle $R \in \mathcal{R}$ such that $C = C_R$ and $S = S_R$. Notice that* $|C_R| + |S_R| \leq \lceil \log_2 n \rceil$ *for any $R \in \mathcal{R}$.*

Recall that, to apply Theorem 5 we need to have a description of τ, ℓ and the extended formulations for (P_R, Q_R), for each $R \in \mathcal{R}$. These are computed as follows.

1. *τ, ℓ, and (C_R, S_R) for all $R \in \mathcal{R}$.* Enumerate all cliques and stable sets of G of combined size at most $\lceil \log_2 n \rceil$ and run the protocol on each of those input pairs. Each of those inputs gives a path in the tree (with the corresponding ℓ), terminating in a leaf v, corresponding to a rectangle R. By Observation 1, τ is given by the union of those paths. Moreover, observe that, for each $R \in \mathcal{R}$, C_R (resp. S_R) is contained in all cliques C (resp. stable sets S) such that, on input (C, S), the protocol terminates in the leaf v corresponding to R. In particular, on input (C_R, S_R), the protocol terminates in v. Hence, the inclusion-wise minimal such C and S give C_R, S_R.
2. *For each leaf of τ corresponding to a rectangle $R \in \mathcal{R}$, give a compact extended formulation T_R for the pair (P_R, Q_R).* We follow an approach inspired by Lemma 1. We first need the following fact on the structure of the rectangles.

Lemma 2 *Let $R = (C_R, S_R) \in \mathcal{R}$ and $(C, S) \in R$. Then $(C', S') \in R$ for any C' such that $C_R \subseteq C' \subseteq C$ and S' such that $S_R \subseteq S' \subseteq S$.*

Proof At each step of the protocol the vertex to be sent from Alice (or Bob) is chosen as the first node of low degree in their current subsets. Vertices in $C \setminus C_R$ (or $S \setminus S_R$) are not sent during the protocol on input (C, S), hence they are always lower in order than the vertices in C_R (or S_R), or do not have the appropriate degree. Thus, deleting vertices from $C \setminus C_R$ and $S \setminus S_R$ does not change the protocol. □

Now, let R be a rectangle of value 1. We claim that $P_R \subseteq T_R \subseteq Q_R$, with

$$T_R := \{x \in \mathbb{R}^d : \ x_v = 0 \ \ \forall \ v \in V : C_R + v \in R, \ 0 \leq x \leq 1\},$$

where we write $C_R + v$ for $C_R \cup \{v\}$. The first inclusion is due to Lemma 2 and to the fact that, since R has value 1, all vertices of P_R correspond to stable sets that are disjoint from cliques of R. For the second inclusion, given a clique $C \in R$ and $v \in C$, using again Lemma 2 we have that $C_R + v \in R$, so for $x \in T_R$ we have $x(C) = \sum_{v \in C} x_v = 0 \leq 1$. Observe that, in order to decide if $C_R + v \in R$, it suffices to run the protocol on input $(C_R + v, S_R)$.

Finally, let R be a rectangle of value 0. We have $C_R \cap S_R = \{u\}$ for some vertex u. One can conclude, in a similar way as before, that $P_R \subseteq T_R \subseteq Q_R$, where

$$T_R := \{x \in \mathbb{R}^d : \ x_u = 1, x_v = 0 \ \ \forall \ v \in V - u : C_R + v \in R, \ 0 \leq x \leq 1\}.$$

We conclude by observing that the approach described above proceeds by obtaining the leaves of τ through an enumeration of all cliques and stable sets of combined size $\lceil \log_2 n \rceil$, and then reconstructing τ. This takes time $n^{\Theta(\log n)}$. However, one could instead try to construct τ from the root, by distinguishing cases for each possible bit sent by Alice or Bob. This intuition is the basis for a "top-down" approach that decomposes the input graph and gives an alternative formulation for $(\mathrm{STAB}(G), \mathrm{QSTAB}(G))$ in output-efficient time. For a detailed description of this approach, we refer to [1,2].

Threshold-free graphs A threshold graph is a graph for which there is an ordering of the vertices $v_1, \ldots, v_n$, such that for each i, v_i is either complete or anticomplete to $v_{i+1}, \ldots, v_n$. Fix a threshold graph H on t vertices, where t is a constant. We say that a graph is H-free if it does not contain H as an induced subgraph. A deterministic protocol for the clique-stable set incidence matrix of H-free graphs is known [22]. It implies a polynomial size extended formulation for (STAB(G), QSTAB(G)) if G is H-free. We will sketch how one can adapt the results of the previous section in order to apply Theorem 5 and write down such formulation efficiently, obtaining an analogous of Theorem 6.

Before doing so, we briefly describe the protocol. Fix a threshold graph H, and let $V(H) = \{u_1, \ldots, u_t\}$ with u_i either complete or anticomplete to $u_{i+1}, \ldots, u_t$, for $i = 1, \ldots, t-1$. Let G be any H-free graph. The protocol is a simple variant of Yannakakis' protocol [31], described at the beginning of the previous section, where at each step either Alice or Bob sends a vertex of their sets. In this version, whether it is the turn of Alice or Bob to speak at a given step does not depend on G and on the degree on its vertices, but exclusively on the structure of H: at step i, for $i = 1, \ldots, t$, if u_i is complete to $u_{i+1}, \ldots, u_t$ Alice sends any vertex of her clique, and otherwise Bob sends any vertex of his stable set. If the vertex sent is in both the stable set and the clique, the protocol ends with output 0, otherwise the players restrict the current graph exactly as in the classical protocol and proceed onto the next step (if the current graph is empty, the protocol ends with output 1). Notice that the vertices sent mimic the structure of H: if a vertex v is sent by Alice, the current graph is restricted to the neighbourhood of v, hence v is complete to any vertex that will be sent afterwards; similarly, if v is sent by Bob, then v is anticomplete to any vertex that will be sent afterwards. Hence, the protocol must end before step t, as otherwise the t vertices sent would form a subgraph of G that is isomorphic to H, a contradiction.

Now, in order to efficiently obtain the desired extended formulation, one only needs to efficiently obtain the tree τ corresponding to this protocol. This can be done by enumerating all the possible executions of the protocol, similarly to the way it was described in the previous section, with a key difference: since, during any execution of the protocol, at most t vertices are communicated, it suffices to enumerate all the cliques and stable sets of combined size t (instead of $\lceil \log_2 n \rceil$). Hence, obtaining τ takes time $O(n^t)$, and the same applies to the formulations T_R, $R \in \mathcal{R}$, as in the previous section. By feeding this input to Algorithm 1 and applying Theorem 5, we obtain the following:

Theorem 7 *Let H be a fixed threshold graph, and let G be an H-free graph on n vertices. Then there is an algorithm that, on input G, outputs an extended formulation for* (STAB(G), QSTAB(G)) *of size $O(n^{|V(H)|})$ in time $O(n^{|V(H)|})$. If moreover G is also perfect, then the output is an extended formulation of* STAB(G).

We conclude the section mentioning that it is not known whether a compact extended formulation exists for STAB(G), when G belongs to the the related class of *H-free graphs*, with H being a fixed *split graph*. Some work on this direction has appeared in Bousquet et al. [7], who showed that for such graphs there exists a CS-separator of polynomial size. We refer to [7] for details.

Min-up/min-down polytopes We give here another application of Theorem 5. Min-up/min-down polytopes were introduced in [23] to model scheduling problems with machines that have a physical constraint on the frequency of switches between the operating and not operating states. For a vector $x \in \{0, 1\}^T$ and an index i with

$1 \leq i \leq T-1$, let us call i a *switch-on index* if $x_i = 0$ and $x_{i+1} = 1$, a *switch-off index* if $x_i = 1$ and $x_{i+1} = 0$. For $L, \ell, T \in \mathbb{N}$ with $\ell \leq T$, $L \leq T$, the min-up/min-down polytope $P_T(L, \ell)$ is defined as the convex hull of vectors in $\{0,1\}^T$ satisfying the following: for any i, j switch indices, with $i < j$, we have $j - i \geq L$ if i is a switch-on index and j a switch-off index and $j - i \geq \ell$ if viceversa i is a switch-off index and j a switch-on index. In other words, in these vectors (seen as strings) each block of consecutive zeros (resp. ones) has length at least ℓ (resp. L), apart from the first and the last. In [23], the following is shown:

Theorem 8 *The following is a complete, non-redundant description of $P_T(L, \ell)$:*

$$\sum_{j=1}^{k}(-1)^j x_{i_j} \leq 0 \quad \forall \{i_1, \dots, i_k\} \subseteq [T] : k \text{ odd and } i_k - i_1 \leq \ell$$
$$\sum_{j=1}^{k}(-1)^{j-1} x_{i_j} \leq 1 \quad \forall \{i_1, \dots, i_k\} \subseteq [T] : k \text{ odd and } i_k - i_1 \leq L.$$

When ℓ, L are not constant, $P_T(L, \ell)$ has an exponential number of facets. An extended formulation for $P_T(L, \ell)$ of size linear in T is given in [26]. We now provide a simple deterministic protocol for its slack matrix $S^{\ell, L, T}$ and use Theorem 5, to efficiently construct a compact extended formulation. This depends linearly in T as the one from [26], but also on L, ℓ. It is however, we believe, an example of how good formulations can be obtained very easily using the machinery developed in this paper.

For simplicity we only consider the part of the slack matrix indexed by the first set of inequalities, as the protocol for the second part can be derived in an analogous way. Hence, assume that Alice is given an index set $I = \{i_1, \dots, i_k\} \subseteq [T]$ with k odd and $i_k - i_1 \leq \ell$, and Bob is given a vertex v of $P_T(L, \ell)$, which is univocally determined by its switch indices. Alice sends to Bob the index i_1. Assume that $v_{i_1} = 0$, the other case being analogous. Then v can have at most one switch-on index and at most one switch-off index in $\{i_1, \dots, i_1 + \ell\}$, hence in particular in I. Bob sends to Alice 1 bit to signal $v_{i_1} = 0$, and the coordinates of such indices. Hence, Alice now knows exactly the coordinates of v on I and can output the slack corresponding to I, v. In total, it is easy to see that the protocol has complexity $\lceil \log T \rceil + 2 \max(\lceil \log L \rceil, \lceil \log \ell \rceil)$), and can be modeled by a tree τ of size $O(T \cdot (L + \ell)^2)$.

Finally, in order to apply Theorem 5 we would need to obtain a compact extended formulation for the pair (P_R, Q_R) for each rectangle R corresponding to a leaf of τ. While this can be achieved by applying Lemma 1 with a tedious case distinction, we proceed in a simpler, alternative way. We exploit the fact that, as remarked above, when applying Theorem 5, we do not necessarily need to give formulations to the leaves τ, but we can start from their ancestors as well. Consider a step of the protocol, corresponding to a node v of τ, where the following communication has taken place: Alice sent an index i_1, Bob sent a bit to signal that $v_{i_1} = 1$ and the switch-off index i^* (the other cases are dealt with similarly). This determines a rectangle $S_v = R$ that is not monochromatic (as its value depends on Alice's input) and has as columns all the vertices x of $P_T(L, \ell)$ with

$$x_{i_1} = \cdots = x_{i^*} = 1, \; x_{i^*+1} = \cdots = x_{i_1+\ell} = 0 \tag{4}$$

and as rows all the inequalities corresponding to subsets I whose smallest index is i_1. Consider the polytope

$$T_R := \{0 \leq x \leq 1 : \; x \text{ satisfies } (4)\}.$$

Clearly, we have $P_R \subseteq T_R \subseteq Q_R$. Now, it is easy to see that a similar formulation can be given to all nodes v of τ whose children are leaves. Hence we can conclude the following.

Theorem 9 *Let L, ℓ, T be positive integers with $\ell \leq T$, $L \leq T$. The min-up/min-down polytope $P_T(L, \ell)$ has an extended formulation of size $O(T \cdot (L + \ell)^2)$ that can be written down in time $O(T \cdot (L + \ell)^2)$.*

5 Stable set polytope of claw-free graphs: a direct derivation

In this section we show a case in which we can construct an extended formulation directly from the protocol, without resorting to Theorem 5.

Let $P = \text{STAB}(G)$, where G is claw-free, i.e., it does not contain the complete bipartite graph $K_{1,3}$ as induced subgraph. Extended formulations for this polytope are known [13,14], but as the stable set polytope of claw-free graphs generalizes the matching polytope, it follows from [28] that it has no extended formulation of polynomial size. The row submatrix of the slack matrix of P corresponding to clique constraints can however be computed by the following simple protocol from [12]. Fix an arbitrary order $v_1, \ldots, v_n$ of V. Alice, who has a clique C as input, sends vertex $v_i \in C$ with smallest i to Bob, who has a stable set S. If $v \in S$, Bob outputs 0. Else, since G is claw-free, we have $|N(v) \cap S| \leq 2$, and clearly $C \subset N(v)$, hence Bob sends $N(v) \cap S$ and Alice can compute and output $1 - |C \cap S|$. The protocol has complexity at most $3 \log n + 1$, hence by applying Theorem 1 we get the following formulation of size $O(n^3)$:

$$\begin{aligned} & x(C) + \sum_{R \in \mathcal{R}_C} y_R = 1 \quad \forall\, C \text{ clique of } G \\ & x, y \geq 0. \end{aligned} \tag{5}$$

Here, $\mathcal{R}^1$ is the set of monochromatic, non-zero rectangles generated by the protocol (see the discussion after Theorem 1). Notice that $\mathcal{R}^1$ contains a 1-rectangle for each couple (v, U), where $v \in V$ and $U \subseteq N(v)$ with U stable, i.e., $|U| \leq 2$. Moreover, for a clique C, $\mathcal{R}_C$ denotes all rectangles (v_i, U) from $\mathcal{R}^1$ with $i = \arg\min\{j \in [n] : v_j \in C\}$ and $C \cap U = \emptyset$.

We now remove from (5) many redundant equations. Before, we notice that the above protocol can be easily generalized to $K_{1,t}$-free graphs for $t \geq 3$: in this case sets $\mathcal{R}^1, \mathcal{R}_C$ are defined similarly as before, except that now we have rectangles (v, U) with $|U| \leq t - 1$. This yields a formulation of size $O(n^t)$. We state our result for this more general class of graphs: informally, the only clique equations that we keep are coming from singletons and edges, obtaining a formulation with only $O(n^2)$ many equations.

Theorem 10 *Let $G = (V, E)$ be a $K_{1,t}$-free graph. Let $\mathcal{R}^1, \mathcal{R}_C$ as above. Then the following is an extended formulation for* $(\mathrm{STAB}(G), \mathrm{QSTAB}(G))$*:*

$$\begin{aligned} & x(v) + \sum_{R \in \mathcal{R}_v} y_R = 1 \quad \forall\, v \in V \\ & x(e) + \sum_{R \in \mathcal{R}_e} y_R = 1 \quad \forall\, e \in E \\ & x, y \geq 0, \end{aligned} \tag{6}$$

where we abbreviated $\mathcal{R}_v = \mathcal{R}_{\{v\}}$ and $x(e) = x(u) + x(v)$ for $e = uv$. In particular, if G is also perfect, (6) *is an extended formulation for* $\mathrm{STAB}(G)$.

Proof Thanks to the above discussion, we only need to show that, for any clique C of G with $|C| = k \geq 3$, the equation $x(C) + \sum_{R \in \mathcal{R}_C} y_R = 1$ is implied by the equations in (6). In fact, since equations from (6) are valid for (5) (they are a subset of valid equations), it will be enough to prove that the *left-hand side* of any equation from (5) is a linear combination of the left-hand sides of equations from (6). It is easy to verify that the same holds for the right-hand side. From now on, fix such C, let $v \in C$ be the first vertex of C (in the order fixed by the protocol) and consider the following expression, obtained by summing the left-hand side of the equations relative to $e = uv$, for every $u \in C - v := C \setminus \{v\}$:

$$\begin{aligned} & \sum_{\substack{e=uv: \\ u \in C-v}} \left(x(u) + x(v) + \sum_{R \in \mathcal{R}_e} y_R \right) \\ & \quad = (k-2)x(v) + x(C) + \sum_{\substack{e=uv: \\ u \in C-v}} \left(\sum_{R \in \mathcal{R}_e \cap \mathcal{R}_C} y_R + \sum_{R \in \mathcal{R}_e \setminus \mathcal{R}_C} y_R \right) \end{aligned}$$

Now, consider a rectangle $R = (v, U) \in \mathcal{R}^1$, and let (C, S) be an entry of R (hence, $C \cap S = \emptyset$). Then v is the first vertex of C and $U = N(v) \cap S$, with $U \cap C = \emptyset$. Hence, we can derive $\mathcal{R}_e = \{(v, U) : U \subset N(v), U \in \mathcal{S}, u \notin U\}$ for $u \in C$ and $e = uv$, and $\mathcal{R}_C = \{(v, U) : U \subset N(v), U \in \mathcal{S}, U \cap C = \emptyset\}$, where $\mathcal{S}$ denotes the family of the stable sets of G. Hence $\mathcal{R}_C \subseteq \mathcal{R}_e$ for $e \subset C$. We can then rewrite the above expression as:

$$
\begin{aligned}
&(k-2)x(v) + x(C) + (k-1) \sum_{\substack{U \subseteq N(v),\\ U \in \mathcal{S}, U \cap C = \emptyset}} y_{(v,U)} + \sum_{u \in C \setminus \{v\}} \sum_{\substack{U \subseteq N(v) - u\\ U \in \mathcal{S}, U \cap C \neq \emptyset}} y_{(v,U)} \\
&\quad = (k-2)x(v) + x(C) + (k-1) \sum_{\substack{U \subseteq N(v),\\ U \in \mathcal{S}, U \cap C = \emptyset}} y_{(v,U)} + (k-2) \sum_{\substack{U \subseteq N(v)\\ U \in \mathcal{S}, U \cap C \neq \emptyset}} y_{(v,U)},
\end{aligned} \tag{7}
$$

where we used that $U \in \mathcal{S}$, $U \cap C \neq \emptyset$ implies that $|U \cap C| = 1$, hence $y_{(v,U)}$ will appear in all summations, except the one corresponding to $\{u\} = U \cap C$.

Now, consider the left-hand side of the equation corresponding to $C = \{v\}$:

$$
x(v) + \sum_{R \in \mathcal{R}_v} y_R = x(v) + \sum_{U \subset N(v), U \in \mathcal{S}} y_{(v,U)}.
$$

Subtracting $k-2$ times the latter from (7) we obtain as required

$$
x(C) + \sum_{U \subset N(v), U \in \mathcal{S}, U \cap C = \emptyset} y_{v,U} = x(C) + \sum_{R \in \mathcal{R}_C} y_R.
$$

□

6 Extended formulations from non-deterministic, unambiguous protocols

Extended formulations can be obtained also via more general classes of communication protocols, namely *unambiguous non-deterministic* and *randomized* communication protocols. The latter are very powerful, as they allow to obtain *any* extended formulation (see [12]), and it is not clear how to extend the results in this paper to deal with them. In this section, we deal with the former.

Recall that a deterministic protocol for a matrix M implies a partition of M into monochromatic rectangles. Unambiguous non-deterministic are strictly more general in that they are exactly *equivalent* to such partitions. Hence, we do not formally define those protocols here (we refer the interested reader to [21]), and assume we are given the corresponding partition. Following the arguments in Sect. 2, when M is the slack matrix of a pair of polytopes (P, Q), a partition of M in t monochromatic rectangles implies a non-negative factorization of M, hence an extended formulation of (P, Q), of size t. It is not clear how to efficiently obtain this formulation in general. There is however a known technique that reduces unambiguous non-deterministic protocols to deterministic ones, at the cost of increasing the size of the corresponding partition. In this section, we show how to extend Theorem 5 to deal with this case. The machinery we develop can be applied, for instance, to the extended formulation of STAB(G) when G is a *comparability graph* proposed in [31]. We refer the interested reader to [1] for details.

Reducing non-deterministic to deterministic protocols Let $P \subseteq Q \subset \mathbb{R}^d$ be a pair of polytopes and M be the slack matrix of (P, Q). Assume that we are given a partition of M in monochromatic rectangles $\mathcal{R} = \{R_1, \ldots, R_t\}$. Yannakakis [31] showed how to deduce from $\mathcal{R}$ a deterministic protocol for M of complexity $O(\log^2 t)$, via what we term *Yannakakis' reduction* and describe below, following [27].

In this setting, Alice has as input a row index r of M, Bob a column index c, and the goal of the protocol is to determine the unique rectangle of $R_{r,c} \in \mathcal{R}$ containing the entry (r, c). We say that a rectangle R is *horizontally good* if r is a row of R, and R intersects horizontally at most half of the rectangles in $\mathcal{R}$, *vertically good* if c is a column of R and R intersects vertically at most half of the rectangles in $\mathcal{R}$, and *good* if it is horizontally or vertically good. Notice that, since two rectangles in $\mathcal{R}$ cannot intersect both vertically and horizontally, $R_{r,c}$ is good. We now proceed in stages: in each stage, Alice, who has input r, sends a horizontally good rectangle R (breaking ties arbitrarily), or the information that there is none: in the former case Alice and Bob can delete from $\mathcal{R}$ all the rectangles that do not horizontally intersect R, as they cannot be $R_{r,c}$; in the latter case, there must be a vertically good rectangle R, which Bob sends to Alice (again, breaking ties arbitrarily), and again Alice and Bob can delete from $\mathcal{R}$ all the rectangles that do not vertically intersect R. At the end of the stage, the size of $\mathcal{R}$ is decreased by at least half. Note that $R_{r,c}$ is never deleted from $\mathcal{R}$ during the protocol. Hence, there will be at most $\lceil \log t \rceil$ stages before $R_{r,c}$ is the only remaining rectangle, in which case it is output. The protocol has complexity $O(\log^2 t)$. Therefore, the protocol partitions M into a family $\mathcal{R}'$ of $t^{O(\log t)}$ monochromatic rectangles, each of which is contained in a unique rectangle of $\mathcal{R}$, which is the one that is output at the end of the protocol.

In the following, we describe how to obtain a tree modeling Yannakakis' reduction and how to efficiently obtain an extended formulation of size $t^{O(\log t)}$ of (P, Q). In particular we will use the tree obtained, plus extended formulations T_R for each $R \in \mathcal{R}$, defined as in Theorem 5, as input to Algorithm 1.

An efficient version of the reduction The inputs to our algorithm are formulations T_R for $R \in \mathcal{R}$, together with two graphs G_H, G_V on vertex set $\mathcal{R}$, where two rectangles are adjacent in G_H if they intersect horizontally, and in G_V if they intersect vertically. We call G_H and G_V the *horizontal* and *vertical adjacency graphs for* $\mathcal{R}$, respectively. The latter information allows us to construct a tree τ that contains as subtrees *all* deterministic protocols obtained via Yannakakis' reduction applied to a non-deterministic protocol whose rectangles satisfy G_H and G_V. In particular, it contains as a subtree the deterministic protocol produced via Yannakakis' reduction for our input, while some vertices of the tree (including some leaves) may not be reached by any execution of the protocol for (P, Q). Such leaves do not correspond to any submatrix of M: we call the associated rectangles *empty* rectangles. We cannot identify which leaves correspond to empty rectangles using our input only (as this depends on the specific structure of the matrix M). However, we are able to label each leaf v of τ with a rectangle R_v of $\mathcal{R}$, with the propertiy that its corresponding rectangle in M, if not empty, is contained in R_v. This will be enough to show that applying Algorithm 1 with inputs τ and T_{R_v} for each leaf v gives the desired formulation. Note that this does not follow from the proof of Theorem 5, since, as just remarked, τ does not satisfy a basic assumption made in Sect. 2: that each vertex of τ should correspond to a non-empty rectangle of the slack matrix of (P, Q), or equivalently, that $S_v \neq \emptyset$ for each node v of τ. Hence we need to provide a more general argument.

Our approach is formalized in Algorithm 2 below. We remark that, while Algorithm 1 deals with binary trees, it is straightforward to adapt it to general (rooted) trees, hence for ease of exposition we do not make τ binary. In the description of the algorithm, we write $N_H[R]$ (resp. $N_V[R]$) to denote the inclusive neighbourhood of vertex R in the graph G_H (resp. G_V). We will write $d(R)$ to denote the degree of a vertex R in G_H or G_V, specifying the graph each time.

Algorithm 2 Construction of an extended formulation from an unambiguous non-deterministic protocol

Require: Graphs G_H, G_V with vertex set $\mathcal{R}$. For $R \in \mathcal{R}$, a linear description (possibly, in an extended space) T_R of a polytope in $\mathbb{R}^d$.

Set $\tau = \{\rho\}$, $\mathcal{R}_\rho = \mathcal{R}$, $\ell(\rho) = A$.
while there exists $v \in \tau$ not marked as leaf **do**
    if $\mathcal{R}_v = \{R_v\}$ **then**
        Mark v as leaf, set $T'_v = T_{R_v}$.
    else if $\ell(v) = A$ **then**
        for $R \in \mathcal{R}_v$ with $d(R) < |\mathcal{R}_v|/2$ in $G_H[\mathcal{R}_v]$ **do**
            Attach to v a child v', set $\mathcal{R}_{v'} = N_H[R] \cap \mathcal{R}_v$, $\ell(v') = A$.
        end for
        if $\exists R \in \mathcal{R}_v : d(R) \geq |\mathcal{R}_v|/2$ in $G_H[\mathcal{R}_v]$ **then**
            Attach to v a child u, set $\mathcal{R}_u = \{R \in \mathcal{R}_v : d(R) \geq |\mathcal{R}_v|/2$ in $G_H[\mathcal{R}_v]\}$, $\ell(u) = B$.
        end if
    else if $\ell(v) = B$ **then**
        for $R \in \mathcal{R}_v$ with $d(R) < |\mathcal{R}_v|/2$ in $G_V[\mathcal{R}_v]$ **do**
            Attach to v a child v', set $\mathcal{R}_{v'} = N_V[R] \cap \mathcal{R}_v$, $\ell(v') = A$.
        end for
    end if
end while
Run Algorithm 1 on input $(\tau, \ell, \{T'_v\}_{v \text{ leaf of } \tau})$.

Theorem 11 *Let (P, Q) be a pair of polytopes with slack matrix M, $\mathcal{R}$ be a family of rectangles partitioning M, and G_H, G_V be the horizontal and vertical adjacency graphs for $\mathcal{R}$. For each $R \in \mathcal{R}$, assume T_R is an extended formulation for (P_R, Q_R) (defined as in Theorem 5). Let $t := |\mathcal{R}|$, let σ be an upper bound on the size of any T_R, and σ_+ on its total encoding length. Then Algorithm 2 on input $(G_V, G_H, \{T_R : R \in \mathcal{R}\})$, outputs an extended formulation for (P, Q) of size $t^{O(\log(t))}$poly(σ) in time $t^{O(\log(t))}$poly(σ_+).*

Proof We follow the notation of Algorithm 2: in particular, let τ be the tree constructed by the algorithm. We only need to argue about the correctness of the extended formulation output by the algorithm, as the bounds on its size and on the running time follow exactly as in Theorem 5, by noticing that τ has size $t^{O(\log t)}$ and that all the steps in Algorithm 2 can be performed efficiently in the input size.

We claim that formulation output by the algorithm is an extended formulation for (P, Q). We argue by contradiction.

First, we define iteratively a set A_v of row indices of M and a set B_v of column indices of M for each node v of τ. For the root ρ of τ, A_ρ, B_ρ are respectively the row set and the column set of M. For an Alice node v, with children $v_1, \ldots, v_k$, we let $B_{v_i} = B_v$ for any i, and define $A_{v_1}, \ldots, A_{v_k}$ to be a partition of A_v defined as follows: $r \in A_v$ is in A_{v_i} if, for a column c of B_v, on input r, c the execution of the protocol obtained via Yannakakis' reduction starting from $\mathcal{R}$ traverses node v_i. Notice that this partitions A_v. Also, some of the A_{v_i} can be empty (this corresponds to a node of τ that is not part of our deterministic protocol). We proceed similarly for a Bob node v with children $v_1, \ldots, v_k$, letting $A_{v_i} = A_v$ for any i, and $B_{v_1}, \ldots, B_{v_k}$ be the partition of B_v corresponding to the information that Bob sends (where some B_{v_i} can be empty).

Now, assume by contradiction that the output formulation T_ρ is not an extended formulation for (P, Q), where T_v for each node v of τ is defined as in Algorithm 1. First, assume that $P \not\subseteq \pi(T_\rho)$, in particular that there is vertex x^* of P, corresponding to a column c^* of M, with $x^* \notin \pi(T_\rho)$ (where π is, as usual, the projection on the original space). Notice that $c^* \in B_\rho$. We color red some vertices of τ according to the following rules: ρ is red; for a red node v and a child u of v, if $x^* \notin \pi(T_u)$, color u red. Notice that a red Alice node v has at least a red child since, otherwise, we would have that x^* is in the intersection of the $\pi(T_u)$'s, which is equal to $\pi(T_v)$ (see Algorithm 1), a contradiction to v being red. Similarly, all children of a red Bob node will be red. Now, color blue some vertices of τ according to the following rules: ρ is blue; for a blue node v and a child of u with $c^* \in B_u$, color u blue. Notice that a blue Bob node will have at least a blue child, and all children of a blue Alice node will be blue. Now, this implies that there is a path from ρ to a leaf v whose nodes are both red and blue. Let $R = R_v$ be the rectangle of $\mathcal{R}$ corresponding to v, as in Algorithm 2. Since v is blue, we have that $c^* \in B_v$, hence $c^* \in R$, but since v is red, $x^* \notin \pi(T_v) = \pi(T_R)$, a contradiction to the fact that T_R is an extended formulation of (P_R, Q_R). Hence we proved $P \subseteq \pi(T_\rho)$. Proceeding in an analogous way one proves that $\pi(T_\rho) \subseteq Q$, concluding the proof. $\square$

Acknowledgements We thank Mihalis Yannakakis for inspiring discussions and Samuel Fiorini for useful comments on [1], where preliminary versions of certain results presented here had appeared. Manuel Aprile would also like to thank Aurélie Lagoutte and Nicolas Bousquet for useful discussions. We are also indebted to the anonymous referees for their suggestions and corrections, which helped us to improve the structure and the presentation of the paper, and in particular to one of them for pointing us to the reference [7].

References

1. Aprile, M.: On some problems related to 2-level polytopes. PhD thesis, École Polytechnique Fédérale de Lausanne (2018)
2. Aprile, M., Faenza, Y.: Extended formulations from communication protocols in output-efficient time. In: International Conference on Integer Programming and Combinatorial Optimization, pp. 43–56. Springer (2019)
3. Aprile, M., Faenza, Y., Fiorini, S., Huynh, T., Macchia, M.: Extension complexity of stable set polytopes of bipartite graphs. In: CONF, vol. 10520, pp. 75–87. Springer, Cham (2017)
4. Balas, E.: Disjunctive programming. Ann. Discret. Math. **5**, 3–51 (1979)
5. Bazzi, A., Fiorini, S., Huang, S., Svensson, O.: Small extended formulation for knapsack cover inequalities from monotone circuits . In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 2326–2341. SIAM (2017)

6. Bazzi, A., Fiorini, S., Pokutta, S., Svensson, O.: No small linear program approximates vertex cover within a factor $2 - \epsilon$. Math. Oper. Res. **44**(1), 147–172 (2018)
7. Bousquet, N., Lagoutte, A., Thomassé, S.: Clique versus independent set. Eur. J. Comb. **40**, 73–92 (2014). https://doi.org/10.1016/j.ejc.2014.02.003
8. Chan, S.O., Lee, J.R., Raghavendra, P., Steurer, D.: Approximate constraint satisfaction requires large LP relaxations. J. ACM (JACM) **63**(4), 34 (2016)
9. Chudnovsky, M., Trotignon, N., Trunck, T., Vušsković, K.: Coloring perfect graphs with no balanced skew-partitions. J. Comb. Theory Ser. B **115**, 26–65 (2015)
10. Chvátal, V.: On certain polytopes associated with graphs. J. Comb. Theory Ser. B **18**, 138–154 (1975)
11. Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. 4OR **8**(1), 1–48 (2010)
12. Faenza, Y., Fiorini, S., Grappe, R., Tiwary, H.R.: Extended formulations, nonnegative factorizations, and randomized communication protocols. Math. Program. **153**(1), 75–94 (2015)
13. Faenza, Y., Oriolo, G., Stauffer, G.: Separating stable sets in claw-free graphs via Padberg-Rao and compact linear programs. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1298–1308. Society for Industrial and Applied Mathematics (2012)
14. Faenza, Y., Oriolo, G., Stauffer, G.: Separation routine and extended formulations for the stable set problem in claw-free graphs. In: Mathematical Programming (2020)
15. Fiorini, S., Huynh, T., Weltge, S.: Strengthening convex relaxations of 0/1-sets using Boolean formulas. arXiv:1711.01358 (2017)
16. 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 Forty-Fourth Annual ACM Symposium on Theory of Computing, pp. 95–106. ACM (2012)
17. Göös, M., Jain, R., Watson, T.: Extension complexity of independent set polytopes. SIAM J. Comput. **47**(1), 241–269 (2018)
18. Grötschel, M., Lovász, L., Schrijver, A.: Polynomial algorithms for perfect graphs. In: Berge, C., Chvátal, V. (eds.) North-Holland Mathematics Studies, vol. 88, pp. 325–356. Elsevier, Amsterdam (1984)
19. Håstad, J.: Some optimal inapproximability results. J. ACM (JACM) **48**(4), 798–859 (2001)
20. Kaibel, V.: Extended formulations in combinatorial optimization. OPTIMA **85**, 2–7 (2011)
21. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press, Cambridge (1996)
22. Lagoutte, A.: Personal communication (2018)
23. Lee, J., Leung, J., Margot, F.: Min-up/min-down polytopes. Discret. Optim. **1**(1), 77–85 (2004)
24. Lovász, L.: On the Shannon capacity of a graph. IEEE Trans. Inf. Theory **25**(1), 1–7 (1979)
25. Pashkovich, K.: Extended formulations for combinatorial polytopes. PhD thesis, Otto-von-Guericke-Universität Magdeburg (2012)
26. Rajan, D., Takriti, S.: Minimum up/down polytopes of the unit commitment problem with start-up costs. IBM Res. Rep. RC **23628**, 1–14 (2005)
27. Rao, A., Yehudayoff, A.: Communication Complexity: and Applications. Cambridge University Press (2020)
28. Rothvoß, T.: The matching polytope has exponential extension complexity. J. ACM (JACM) **64**(6), 41 (2017)
29. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, vol. 24. Springer, Berlin (2002)
30. Weltge, S.: Sizes of linear descriptions in combinatorial optimization. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Fakultät für Mathematik (2015)
31. Yannakakis, M.: Expressing combinatorial optimization problems by linear programs. J. Comput. Syst. Sci. **43**, 441–466 (1991)