Representations of Monotone Boolean Functions by Linear Programs*

Mateus de Oliveira Oliveira[1] and Pavel Pudlák[2]

1 University of Bergen, Bergen, Norway
mateus.oliveira@uib.no

2 Czech Academy of Sciences, Prague, Czech Republic
pudlak@math.cas.cz

Abstract

We introduce the notion of monotone linear-programming circuits (MLP circuits), a model of computation for partial Boolean functions. Using this model, we prove the following results.

1. MLP circuits are superpolynomially stronger than monotone Boolean circuits.
2. MLP circuits are exponentially stronger than monotone span programs.
3. MLP circuits can be used to provide monotone feasibility interpolation theorems for Lovász-Schrijver proof systems, and for mixed Lovász-Schrijver proof systems.
4. The Lovász-Schrijver proof system cannot be polynomially simulated by the cutting planes proof system. This is the first result showing a separation between these two proof systems.

Finally, we discuss connections between the problem of proving lower bounds on the size of MLPs and the problem of proving lower bounds on extended formulations of polytopes.

1998 ACM Subject Classification F.2.2 [Nonnumerical Algorithms and Problems] Complexity of Proof Procedures

Keywords and phrases Monotone Linear Programming Circuits, Lovász-Schrijver Proof System, Cutting-Planes Proof System, Feasible Interpolation, Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2017.3

1 Introduction

Superpolynomial lower bounds on the size of Boolean circuits computing explicit Boolean functions have only been proved for circuits from some specific families of circuits. A prominent role among these families is played by *monotone Boolean circuits*. Exponential lower bounds on monotone Boolean circuits were proved already in 1985 by Razborov [26]. In 1997 Krajíček discovered that lower bounds on monotone complexity of particular partial Boolean functions can be used to prove lower bounds on resolution proofs [18]. Incidentally, the functions used in Razborov's lower bound were just of the form needed for resolution lower bounds. Exponential lower bounds on resolution proofs had been proved before (coincidentally about at the same time as Razborov's lower bounds). Krajíček came up with a new general method, the so called *feasible interpolation,* that potentially could be used for other proof systems. Indeed, soon after his result, this method was used to prove exponential lower bounds on the cutting-planes proof system [22, 15]. That lower

* Mateus de Oliveira Oliveira acknowledges support from the ERC Advanced Grant 339691 (FEALORA) and from the Bergen Research Foundation. Pavel Pudlák acknowledges support from the ERC Advanced Grant 339691 (FEALORA).

32nd Computational Complexity Conference (CCC 2017).
Editor: Ryan O'Donnell; Article No. 3; pp. 3:1–3:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

COMPUTATIONAL COMPLEXITY CONFERENCE

bound is based on a generalization of Razborov's lower bounds to a more general monotone computational model, the *monotone real circuits*. Another monotone computational model for which superpolynomial lower bounds have been obtained is the *monotone span program* model [2, 11]. An exponential lower bound on the size of monotone span programs have been recently obtained in [7]. For a long time the best known lower bound for this model of computation was of the order of $n^{\Omega(\log n)}$ [2]. Again, superpolynomial lower bounds on the size of monotone span programs can be used to derive lower bounds on the degree of Nullstellensatz proofs, as shown in [23].[1]

The results listed above suggest that proving lower bounds on stronger and stronger models of monotone computation may be a promising approach towards proving lower bounds on stronger proof systems. Indeed, in his survey article [27] Razborov presents the problem of understanding feasible interpolation for stronger systems as one of the most challenging ones.

In this work we introduce several computational models based on the notion of *monotone linear program*. In particular, we introduce the notion of *monotone linear-programming gate* (MLP gate). In its most basic form, an MLP gate is a *partial* function $g : \mathbb{R} \to \mathbb{R}$ of the form $g(y) = \max\{c \cdot x \mid Ax \leq b + By, x \geq 0\}$ where y is a set of input variables, and B is a non-negative matrix. The complexity of such a gate is defined as the number of rows plus the number of columns in the matrix A. For each assignment $\alpha \in \mathbb{R}^n$ of the variables y, the value $g(\alpha)$ is the optimal value of the linear program with objective function $c \cdot x$, and constraints $Ax \leq b + B\alpha$. The requirement that $B \geq 0$ guarantees monotonicity, i.e., that $g(\alpha) \leq g(\alpha')$ whenever $g(\alpha)$ is defined and $\alpha \leq \alpha'$. We note that the value $g(\alpha)$ is considered to be undefined if the associated linear program $\max\{c \cdot x \mid Ax \leq b + B\alpha\}$ has no solution. Other variants of MLP gates are defined in a similar way by allowing the input variables to occur in the objective function, and by allowing the corresponding linear programs to be minimizing or maximizing. We say that an MLP gate is weak if the input variables occur either in the objective function or in the constraints. We say that an MLP gate is strong if the input variables occur in both the objective function and in the constraints.

An MLP circuit is a straightforward generalization of the notion of unbounded-fan-in (monotone) Boolean circuit where MLP gates are used instead of Boolean gates. In Theorem 3 we show that if all gates of an MLP circuit C are weak, then this circuit can be simulated by a single weak MLP gate ℓ_C whose size is polynomial on the size of C. Since the AND and OR gates can be faithfully simulated by weak MLP gates, we have that monotone Boolean circuits can be polynomially simulated by weak MLP circuits (Theorem 4). In contrast, we show that weak MLP gates are super-polynomially stronger than monotone Boolean circuits. On the one hand, Razborov has shown that that any monotone Boolean circuit computing the *bipartite perfect matching function* $\mathrm{BPM}_n : \{0,1\}^{n^2} \to \{0,1\}$ must have size at least $n^{\Omega(\log n)}$. On the other hand, a classical result in linear programming theory [29] can be used to show that the same function can be computed by weak MLP gates of polynomial size.

In [2], Babai, Gál and Wigderson showed that there is a function that can be computed by span programs of linear size but which requires superpolynomial-size monotone Boolean circuits. Recently, Cook et al. [7] showed that there is a function that can be computed by polynomial-size monotone Boolean circuits, but that requires exponential-size monotone span programs over the reals. Therefore, monotone span programs (which we will abbreviate by MSPs) and monotone Boolean circuits are incomparable in the sense that neither of these

[1] We note however that strong degree lower bounds for Nullstellensatz proofs can be proved using more direct methods [3, 6, 13, 1].

models can polynomially simulate the other. In Theorem 7 we show that a particular type of weak MLP gate can polynomially simulate monotone span programs over the reals. On the other hand, by combining the results in [7] with Theorem 7, we have that these weak MLP gates are exponentially stronger than monotone span programs over reals. Therefore, while monotone Boolean circuits are incomparable with MSPs, weak MLP-gates are strictly stronger than both models of computation.

Next we turn to the problem of proving a monotone interpolation theorem for Lovász-Schrijver proof systems [20]. Currently, size lower bounds for these systems have been proved only with respect to tree-like proofs [21], and therefore, it seems reasonable that a monotone interpolation theorem for this system may be a first step towards proving size lower bounds for general LS proof systems. Towards this goal we show that MLP circuits which are constituted by strong MLP gates can be used to provide a *monotone* feasible interpolation theorem for LS proof systems. In other words, we reduce the problem of proving superpolynomial lower bounds for the size of LS proofs, to the problem of proving lower bounds on the size of MLP circuits with strong gates.

While circuits with weak MLP gates can be collapsed to a single weak MLP gate, we do not know how to collapse MLP circuits with strong gates into a single strong gate. Nevertheless, in Theorem 10 we show that a single weak MLP gate suffices in a monotone interpolation theorem for *mixed LS proofs*. These are proofs in which, on top of variables representing 0s and 1s, there are also variables that range over real numbers. This interpolation theorem implies two things. First, the cutting-planes proof system cannot polynomially simulate the LS proof system (Corollary 18). Understanding the mutual relation between the power of the cutting-planes proof system and the LS proof system is a longstanding open problem in proof complexity theory. Our result solves one direction of this mutual relation by showing that for some tautologies, LS proofs can be superpolynomially more concise than cutting-planes proofs. Second, using this interpolation theorem, and a size lower bound for monotone real circuits due to Fu [10], we can show that MLP-circuits cannot be polynomially simulated by monotone real circuits (Theorem 19).

Monotone linear programs programs may be regarded as a generalization of both monotone Boolean circuits and monotone span programs. Since superpolynomial lower bounds for these two latter formalisms were proved via rather distinct formalisms, it is reasonable to expect that new lower bound methods will need to be developed in order to prove superpolynomial lower bounds for the size of weak monotone linear programs. A possible approach is to try to strength recent lower bounds obtained for the extension complexity of polytopes whose vertices correspond to minterms of certain monotone Boolean functions [28, 9, 4, 5]. To prove a lower bound on the size of weak MLP gates, it will be necessary to prove lower bounds on the size of extended formulations for all polytopes of a certain form that separate minterms from maxterms. This is clearly a harder problem than proving lower bounds on the extension complexity of a single polytope. Nevertheless, there are certain results that point in this direction [4, 5]. In any case, Theorem 19 suggests that this will not be an easy task. The theorem gives an example of a monotone function whose set of ones requires exponentially large extended formulation, but whose minterms can be separated from a large subset of maxterms by a polynomial size weak MLP gate.

In this extended abstract all proofs are omitted.

2 Monotone Linear-Programming Gates

▶ **Definition 1** (MLP Gate). A *monotone linear-programming gate*, or MLP gate, is a partial function $\ell : \mathbb{R}^n \to \mathbb{R} \cup \{*\}$ whose value at each point $y \in \mathbb{R}^n$ is specified via a monotone linear program. More precisely, we consider the following six types of MLP gates:

MAX-RIGHT: $\ell(y) = \max\{c^T \cdot x \mid Ax \leq b + By,\ x \geq 0\}$

MIN-RIGHT: $\ell(y) = \min\{c^T \cdot x \mid Ax \geq b + By,\ x \geq 0\}$

MAX-LEFT: $\ell(y) = \max\{(c + Cy)^T \cdot x \mid Ax \leq b,\ x \geq 0\}$

MIN-LEFT: $\ell(y) = \min\{(c + Cy)^T \cdot x \mid Ax \geq b,\ x \geq 0\}$

MAX: $\ell(y) = \max\{(c + Cy)^T \cdot x \mid Ax \leq b + By,\ x \geq 0\}$

MIN: $\ell(y) = \min\{(c + Cy)^T \cdot x \mid Ax \geq b + By,\ x \geq 0\}$

where A is a matrix in $\mathbb{R}^{m \times k}$, b is a vector in $\mathbb{R}^m$, c is a vector in $\mathbb{R}^k$, and B and C are nonnegative matrices in $\mathbb{R}^{m \times n}$, i.e., $B \geq 0$ and $C \geq 0$.

Intuitively, the variables y should be regarded as input variables, while the variables x should be regarded as internal variables. If the linear program specifying a gate $\ell(y)$ has no solution when setting y to a particular point $\alpha \in \mathbb{R}^n$, then we set $\ell(\alpha) = *$. In other words, in this case we regard the value $\ell(\alpha)$ as being undefined. We note that the requirement $B \geq 0, C \geq 0$ guarantees that the gates introduced above are monotone. More precisely, if $\alpha \leq \alpha'$, and both $\ell(\alpha)$ and $\ell(\alpha')$ are well defined, then $\ell(\alpha) \leq \ell(\alpha')$. The size $|\ell|$ of an MLP gate ℓ is defined as the number of rows plus the number of columns in the matrix A.

The gates of type MAX-RIGHT, MAX-LEFT, MIN-RIGHT and MIN-LEFT are called weak gates. Note that in these gates, the input variables y occur either only in the objective function, or only in the constraints. The gates of type MAX and MIN are called strong gates. The input variables in strong gates occur both in the constraints and in the objective function.

Circuits that are constituted of MLP gates are called *MLP circuits*.

▶ **Definition 2** (MLP-Circuit Representation). We say that an MLP circuit C represents a partial Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ if the following conditions are satisfied for each $a \in \{0,1\}^n$.

1. $C(a) > 0$ if $F(a) = 1$.
2. $C(a) \leq 0$ if $F(a) = 0$.

We say that an MLP-circuit C *sharply* represents $F : \{0,1\}^m \to \{0,1,*\}$ if $C(a) = 1$ whenever $F(a) = 1$ and $C(a) = 0$ whenever $F(a) = 0$. We define the size of an MLP circuit C as the sum of the sizes of MLP gates occurring in C. The next theorem states that if all gates in an MLP circuit C are weak MLP gates with the same type τ, then this circuit can be polynomially simulated by a *single* MLP gate ℓ_C of type τ.

▶ **Theorem 3** (From Circuits to Gates). *Let C be an MLP circuit of size s where all gates in C are weak MLP gates of type τ. Then there is an MLP gate ℓ_C of type τ and size $O(s)$ such that for each $a \in \mathbb{R}^n$ for which $C(a)$ is defined, $\ell_C(a) = C(a)$.*

3 Weak MLP Gates vs Monotone Boolean Circuits

In this section we show that partial Boolean functions that can be represented by monotone Boolean circuits of size s can also be sharply represented by weak MLP gates of size $O(s)$. On the other hand, we exhibit a partial function that can be represented by polynomial-size max-right MLP gates, but which require Boolean circuits of superpolynomial size.

▶ **Theorem 4.** *Let $F : \{0,1\}^n \to \{0,1,*\}$ be a partial Boolean function, and let C be a Boolean circuit of size s representing F. Then for any weak type τ, F can be sharply represented by an MLP gate of type τ and size $O(s)$.*

Let $BPM_n : \{0,1\}^{n^2} \to \{0,1\}$ be the Boolean function that evaluates to 1 on an input $p \in \{0,1\}^{n^2}$ if and only if p represents a bipartite graph with a perfect matching. The next theorem, whose proof is based on a classical result in linear programming theory (Theorem 18.1 of [29]) states that the function BPM_n has small MAX-RIGHT MLP representations.

▶ **Theorem 5.** *The Boolean function $BPM_n : \{0,1\}^{n^2} \to \{0,1\}$ can be represented by a* MAX-RIGHT *MLP gate of size $n^{O(1)}$.*

In a celebrated result, Razborov proved a lower bound of $n^{\Omega(\log n)}$ for the size of monotone Boolean circuits computing the function BPM_n [26]. By combining this result with Theorem 5, we have the following corollary.

▶ **Corollary 6.** MAX-RIGHT *MLP gates cannot be polynomially simulated by monotone Boolean circuits.*

We note that the gap between the complexity of MAX-RIGHT MLP gates and the complexity of Boolean formulas computing the BPM_n function is even exponential, since Raz and Wigderson have shown a linear lower-bound on the depth of monotone Boolean circuits computing BPM_n [24].

3.1 Monotone Span Programs

Monotone span programs (MSP) were introduced by Karchmer and Wigderson [17]. Such a program, which is defined over an arbitrary field $\mathbb{F}$, is specified by a vector $c \in \mathbb{F}^k$ and a labeled matrix $A^\rho = (A, \rho)$ where A is a matrix in $\mathbb{F}^{m\times k}$, and $\rho : \{1, ..., m\} \to \{p_1, ..., p_n, *\}$ labels rows in A with variables in p_i or with the symbol $*$ (meaning that the row is unlabeled). For an assignment $p := w$, let $A^\rho_{\langle w\rangle}$ be the matrix obtained from A by deleting all rows labeled with variables which are set to 0. A span program (A^ρ, c) represents a partial Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ if the following conditions are satisfied for each $w \in \{0,1\}^n$.

$$F(w) = \begin{cases} 1 \Rightarrow \quad \exists y,\ y^T A^\rho_{\langle w\rangle} = c^T \\ 0 \Rightarrow \neg\exists y,\ y^T A^\rho_{\langle w\rangle} = c^T \end{cases} \tag{1}$$

That is, if $F(p) = 1$ then c is a linear combination of the rows of $A_{\langle w\rangle}$, while if $F(p) = 0$, then c cannot be cast as such linear combination. We define the size of a span program (A^ρ, c) as the number of rows plus the number of columns in the matrix A. The next theorem states that functions that can be represented by small MSPs over the reals can also be represented by small MIN-RIGHT MLP gates.

▶ **Theorem 7.** *Let $F : \{0,1\}^n \to \{0,1\}$ be a Boolean function. If F can be represented by an MSP of size s over the reals, then F can be represented by a* MIN-RIGHT *MLP gate of size $O(s)$.*

It has been recently shown that there is a family of functions $\mathrm{GEN}_n : \{0,1\}^n \to \{0,1\}$ which can be computed by polynomial-size monotone Boolean circuits but which require monotone span programs over the reals of size $\exp(n^{\Omega(1)})$ [7]. On the other hand, since by Theorem 4, monotone Boolean circuits can be polynomially simulated by weak MLP gates of any type, we have that weak MLP gates of size polynomial in n can represent the function $\mathrm{GEN}_n : \{0,1\}^n \to \{0,1\}$. Therefore, we have the following corollary.

▶ **Corollary 8.** *Weak MLP gates cannot be polynomially simulated by monotone span programs over the reals.*

4 Lovász-Schrijver and Cutting-Planes Proof Systems

4.1 The Lovász-Schrijver Proof System

The Lovász-Schrijver proof system is a refutation system based on the Lovász-Schrijver method for solving integer linear programs [20]. During the past two decades several variants (probably nonequivalent) of this system have been introduced. In this work we will be only concerned with the basic system LS. In Lovász-Schrijver systems the domain of variables is restricted to $\{0,1\}$, i.e., they are Boolean variables. Given an unfeasible set of inequalities Φ, the goal is to use the axioms and rules of inference defined below to show that the inequality $0 \geq 1$ is implied by Φ.

- **Axioms:**
  1. $0 \geq 0$, $1 \geq 0$, $1 \geq 1$,
  2. $0 \leq p_j \leq 1$,
  3. $p_i^2 - p_i = 0$ (integrality).
- **Rules:**
  1. *Positive linear combinations of linear and quadratic inequalities*,
  2. *Multiplication:* Given a linear inequality $\sum_i c_i p_i - d \geq 0$, and a variable p_j, derive

$$p_j(\sum_i c_i p_i - d) \geq 0 \quad \text{and} \quad (1-p_j)(\sum_i c_i p_i - d) \geq 0.$$

Note that using multiplication we may produce quadratic inequalities, but we can only apply the multiplication rule to linear inequalities, hence all inequalities are at most quadratic. Axiom (3) corresponds to two inequalities, but it suffices to use $p_i^2 - p_i \geq 0$, since the other inequality $p_i^2 - p_i \leq 0$ follows from Axiom (2) and Rule (2). We also observe that the inequality $1 \geq 0$ can be derived from the axioms $p_i \geq 0$ and $1 - p_i \geq 0$.

A proof Π of an inequality $\sum_i c_i p_i - d \geq 0$ from Φ is a sequence of inequalities such that every inequality in the sequence is either an element of Φ or is derived from previous ones using some LS rule. We say that Π is a refutation of the set of inequalities Φ, if the last inequality is $-d \geq 0$ for some $d > 0$.

The LS proof system is implicationally complete. This means that if an inequality $\sum_i c_i p_i - d \geq 0$ is semantically implied by an initial set of inequalities Φ, then $\sum_i c_i p_i - d \geq 0$ can be derived from Φ by the application of a sequence of LS-rules [20].

Superpolynomial lower bounds on the size of LS proofs have been obtained only in the restricted case of tree-like proofs [21]. The problem of obtaining superpolynomial lower bounds for the size of DAG-like LS proofs remains a tantalizing open problem in proof complexity theory.

The LS proof system is stronger than Resolution. It can be shown that resolution proofs can be simulated by LS proofs with just a linear blow up in size. Additionally, the Pigeonhole principle has LS proofs of polynomial size, while this principle requires exponentially long resolution proofs [14]. On the other hand, the relationship between the power of the LS proof system and other well studied proof system is still elusive. For instance, previous to this work, nothing was known about how the LS proof system relates to the cutting-planes proof system with respect to polynomial-time simulations. In Subsection 4.4 we will show that there is a family of sets of inequalities which have polynomial-size DAG-like LS refutations, but which require superpolynomial-size cutting-planes refutations. This shows that the cutting-planes proof system cannot polynomially simulate the LS proof system. The converse problem, of determining whether the LS proof system polynomially simulates the cutting-planes proof system, remains open.

4.2 Feasible interpolation

Feasible interpolation is a method that can sometimes be used to translate circuit lower bounds into lower bounds for the size of refutations of Boolean formulas and linear inequalities. Let $\Phi(p, q, r)$ be an unsatisfiable Boolean formula which is a conjunction of formulas $\Phi_1(p, q)$ and $\Phi_2(p, r)$ where q and r are disjoint sets of variables. Since $\Phi(p, q, r)$ is unsatisfiable, it must be the case that for each assignment a of the variables p, either $\Phi_1(a, q)$ or $\Phi_2(a, r)$ is unsatisfiable, or both. Given a proof Π of unsatisfiability for $\Phi(p, q, r)$, an *interpolant* is a Boolean circuit $C(p)$ such that for every assignment a to the variables p:

1. if $C(a) = 0$, then $\Phi_1(x, a)$ is unsatisfiable, and
2. if $C(a) = 1$, then $\Phi_2(y, a)$ is unsatisfiable.

If both formulas are unsatisfiable, then $C(a)$ can be either of the two values. Krajíček has shown that given a resolution refutation Π of a CNF formula, one can construct an interpolant $C(p)$ whose size is polynomial in the size of Π [18]. Krajíček 's interpolation theorem has been generalized, by himself and some other authors, to other proof systems such as the cutting-planes proof system and the Lovász-Schrijver proof system [8].

In principle, such *feasible interpolation* theorems could be used to prove lower bounds on the size of proofs if we could prove lower bounds on circuits computing some particular functions. But since we are not able to prove essentially any lower bounds on general Boolean circuits, feasible interpolation gives us only conditional lower bounds. For instance, the assumption that $\mathbf{P} \neq \mathbf{NP} \cap \mathbf{coNP}$, an apparently weaker assumption than $\mathbf{NP} \neq \mathbf{coNP}$, implies that certain tautologies require superpolynomial-size proofs on systems that admit feasible interpolation.

However, in some cases, one can show that there exist monotone interpolating circuits of polynomial size provided that all variables p appear positively in $\Phi_1(p, q)$, (or negatively in $\Phi_2(p, r)$). In the case of resolution proofs, such circuits are simply monotone Boolean circuits [18, 19]. In the case of cutting-planes proofs, the interpolants are *monotone real circuits* [22]. Monotone real circuits are circuits with Boolean inputs and outputs, but whose gates are allowed to be arbitrary 2-input functions over the reals. Razborov's lower bound on the clique function has been generalized to monotone real circuits [22, 15]. Another proof system for which one can prove superpolynomial lower bounds using monotone feasible interpolation is the Nullstellensatz Proof System [23]. In this proof system, the monotone interpolants are given in terms of monotone span programs[2] [23].

[2] In the context of polynomial calculus, alternative methods (e.g. [1, 16]) yield stronger lower bounds than the monotone interpolation technique.

The results mentioned above suggest that if a proof system has the feasible interpolation property, then it may also have monotone feasible interpolation property for a suitable kind of monotone computation. The next theorem states that LS-proofs can be interpolated using MLP circuits constituted of MAX MLP gates.

▶ **Theorem 9.** *Let $\Phi(p,q) \cup \Gamma(p,r)$ be an unsatisfiable set of inequalities such that the variables $p = (p_1, ..., p_n)$ occur in Φ only with negative coefficients. Let Π be an LS refutation of $\Phi(p,q) \cup \Gamma(p,r)$. Then one can construct an MLP circuit C containing only* MAX *MLP gates which represents a Boolean function $F : \{0,1\}^n \to \{0,1\}$ such that for each $a \in \{0,1\}^n$:*

1. *if $F(a) = 1$, then $\Phi(a,q)$ is unsatisfiable,*
2. *if $F(a) = 0$, then $\Gamma(a,r)$ is unsatisfiable,*

and the size of the circuit C is polynomial in the size of Π.

4.3 Lovász-Schrijver Refutations of Mixed LP Problems

While proof systems for integer linear programming have been widely studied, very little is known about proof systems for mixed linear programming. In mixed linear programming part of variables range over integers and part of them range over reals. The Lovász-Schrijver system can naturally be adapted for mixed linear programming by disallowing the use of axioms and of the multiplication rule for variables ranging over reals. One can easily prove that the such a system is a complete refutation system (i.e., a family of inequalities is unsatisfiable iff a contradiction is derivable).

We will prove a monotone interpolation theorem for systems of mixed linear inequalities (inequalities with both types of variables) of a particular form. The advantage of this theorem, compared with the previous one, is that is uses a *single* MAX-LEFT MLP *gate* (or, by linear-programming duality, a *single* MIN-RIGHT MLP *gate*). While proving lower bounds on the size of general MLP circuits may be beyond the reach of current methods, proving a lower bound on the size of single weak MLP gate seems to be feasible, because this problem is closely related to lower bounds on extended formulations.

▶ **Theorem 10.** *Let $\Phi(p,q) \cup \Gamma(p,r)$ be a set of inequalities where p,q range over 0s and 1s, r range over reals, and the common variables $p = (p_1, ..., p_n)$ occur in Φ only with negative coefficients. Let Π be an LS-refutation of $\Phi(p,q) \cup \Gamma(p,r)$. Then there exists a* MAX-LEFT *MLP gate ℓ that represents a Boolean function $F : \{0,1\}^n \to \{0,1\}$ such that for every $a \in \{0,1\}^n$:*

1. *if $F(a) = 1$, then $\Phi(a,q)$ is unsatisfiable, and*
2. *if $F(a) = 0$, then $\Gamma(a,r)$ is unsatisfiable,*

and the size of the MLP gate ℓ is polynomial in the size of Π.

In the next subsection we will give a natural example of a set of inequalities of the form used in the theorem. We will show that it has polynomial-size mixed LS-refutations, but it requires superpolynomial-size cutting-plane refutations.

4.4 Cutting-Planes vs. Lovász-Schrijver Refutations and Monotone Real Circuits vs MLP Gates

In this subsection we will define a family Ψ_n of unsatisfiable sets of inequalities that admit polynomial-size mixed LS-refutations, but which require superpolynomial refutations in the cutting-planes proof system.

▶ **Definition 11** (Monotone Real Circuit). A monotone real circuit is a circuit C whose gates are monotone real functions of at most two variables. The size of the circuit is the number of the gates.

The following theorem can be used to translate superpolynomial lower bounds on the size of monotone real circuits computing certain partial Boolean functions into superpolynomial lower bounds for the size of cutting plane proofs.

▶ **Theorem 12** (Monotone Interpolation for the cutting-planes Proof System [22]). *Let $\Psi(p, q, r) \equiv \Phi(p, q) \cup \Gamma(p, r)$ be an unsatisfiable set of inequalities where the common variables $p = (p_1, ..., p_n)$ occur in Φ only with negative coefficients. Let Π be a cutting-planes refutation for Ψ. Then one can construct a monotone real circuit C such that for every $a \in \{0,1\}^n$:*

1. *if $C(a) = 1$ then $\Phi(p, q)$ is unsatisfiable,*
2. *if $C(a) = 0$ then $\Gamma(p, q)$ is unsatisfiable,*

and the size of the circuit is at most constant time the size of the proof.

Let $K_n = \{\{i, j\} \mid 1 \leq i < j \leq n\}$ be the complete undirected graph with vertex set $[n] = \{1, ..., n\}$. We say that a subgraph $X \subseteq K_n$ is a perfect matching if the edges in X are vertex-disjoint and each vertex $i \in [n]$ belongs to some edge of X. We say that a subgraph $B \subseteq K_n$ is an *unbalanced complete bipartite graph* if there exist sets $V, U \subseteq [n]$ with $V \cap U = \emptyset$, $|V| > |U|$, and $B = \{\{i, j\} \mid i \in V,\ j \in U\}$. Let $W \subseteq K_n$ be a graph. We let $\mathcal{V}(W) = \{i \mid \exists j \in [n],\ \{i, j\} \in W\}$ be the vertex set of W. For each vertex $i \in \mathcal{V}(W)$, we let $\mathcal{N}(i) = \{j \mid \{i, j\} \in W\}$ be the set of neighbors of i in W. For a subset $V \subseteq \mathcal{V}(W)$, we let $\mathcal{N}(V) = \bigcup_{v \in V} \mathcal{N}(v)$ be the set of neighbors of vertices in $\mathcal{N}(V)$. We say that W is *unbalanced* if there exists $V, U \subseteq \mathcal{V}(W)$ such that $\mathcal{N}(V) \subseteq U$ and $|V| > |U|$. Note that such an unbalanced graph W cannot contain a perfect matching X, since the existence of such a perfect matching would imply the existence of an injective mapping from V to U. We also note that unbalanced complete bipartite graphs are by definition a special case of unbalanced graphs.

Razborov showed that any monotone Boolean circuit which decides whether a graph has a perfect matching must have size at least $n^{\Omega(\log n)}$ [25]. This lower bound was generalized by Fu to the context of monotone real circuits [10]. More precisely, Fu proved that any monotone real circuit distinguishing graphs with a perfect matching from unbalanced complete bipartite graphs must have size at least $n^{\Omega(\log n)}$.

▶ **Theorem 13** ([10]). *Let $F : \{0,1\}^{\binom{n}{2}} \to \{0, 1, *\}$ be a partial Boolean function such that for each $w \in \{0,1\}^{\binom{n}{2}}$:*

- *$F(w) = 1$ if w encodes a graph with a perfect matching,*
- *$F(w) = 0$ if w encodes an unbalanced complete bipartite graph.*

Then any monotone real circuit computing F must have size at least $n^{\Omega(\log n)}$.

Since unbalanced complete bipartite graphs are a special case of unbalanced graphs, monotone real circuits distinguishing graphs with a perfect matching from unbalanced graphs must have size at least $n^{\Omega(\log n)}$ gates.

▶ **Corollary 14.** *Let $g : \{0,1\}^{\binom{n}{2}} \to \{0, 1, *\}$ be a partial Boolean function such that for each $w \in \{0,1\}^{\binom{n}{2}}$:*

- *$g(w) = 1$ if w has a perfect matching.*
- *$g(w) = 0$ if w is unbalanced.*

Then any monotone real circuit computing g must have size at least $n^{\Omega(\log n)}$.

Below we will define a set Ψ_n of unsatisfiable inequalities on variables:

$$p = \{w_{i,j} \mid 1 \leq i < j \leq n\},$$
$$q = \{u_i, v_i \mid i \in [n]\},$$
$$r = \{x_{ij} \mid 1 \leq i < j \leq n\}.$$

Intuitively each assignment of the variables in p defines a graph $W \subseteq K_n$ such that $\{i,j\} \in W$ if and only if $w_{ij} = 1$. Each assignment to the variables in q defines subsets $U, V \subseteq [n]$ where $i \in U$ if and only if $u_i = 1$, and $i \in V$ if and only if $v_i = 1$. Finally, each assignment to the variables in r defines a subset of edges X in such a way that $\{i,j\} \in X$ if and only if $x_{ij} = 1$. The set of inequalities Ψ_n would be satisfiable by an assignment α of the variables in p,q and r only if α defined a graph $W \subseteq K_n$ which contained, at the same time, a perfect matching X and a pair of subsets of vertices $V, U \subseteq \mathcal{V}(W)$ certifying that W is unbalanced. Since no such graph exists, the set Ψ_n is unsatisfiable.

▶ **Definition 15** (Unbalanced Graphs vs Perfect Matching Inequalities)**.** Let $\Psi_n(p,q,r) = \Phi_n(p,q) \cup \Gamma_n(p,r)$ be a set of inequalities on variables $p = \{w_{ij}\}$, $q = \{u_i, v_i\}$ and $r = \{x_{ij}\}$ defined as follows.

Inequalities in $\Phi(p,q)$:	W is unbalanced.
1) $u_j - v_i - w_{ij} + 1 \geq 0$	$\mathcal{N}(V) \subseteq U$. If $i \in V \wedge \{i,j\} \in W \Rightarrow j \in U$.
2) $\sum_j v_j - \sum_i u_i - 1 \geq 0$	$\lvert V\rvert > \lvert U\rvert$.
Inequalities in $\Gamma(p,r)$:	Existence of a perfect matching.
3) $w_{ij} - x_{ij} \geq 0$	X is a subset of edges of W.
4) $\sum_{i,i \neq j} x_{ij} - 1 = 0$	X defines a perfect matching.

Note that we can interpret the system in two alternative ways. First, all variables range over 0s and 1s. Second, variables p, q range over 0s and 1s, while r range over reals. In the second case the meaning of $\Gamma(p,q)$ is that X defines a *fractional* perfect matching. The system is, clearly, unsatisfiable also in the second case.

Note also that the variables in $w_{ij} \in p$, which occur both in $\Phi_n(p,q)$ and in $\Gamma_n(p,r)$, only occur negatively in $\Phi_n(p,q)$. A combination of Fu's size lower-bound for monotone real circuits (Theorem 13) with the monotone interpolation theorem for cutting-planes (Theorem 12) was used in [10] to show that a suitable unsatisfiable set of inequalities Ψ'_n requires cutting-planes refutations of size $n^{\Omega(\log n)}$. The next theorem states that a similar lower bound can be proved with respect to the inequalities introduced in Definition 15.

▶ **Theorem 16.** *Let $\Psi(p,q,r)$ be the set of inequalities of Definition 15 . Then any cutting-planes refutation of $\Psi(p,q,r)$ must have size at least $n^{\Omega(\log n)}$.*

On the other hand, the following theorem states that the set inequalities $\Psi_n(p,q,r)$ has LS-refutations of size polynomial in n. In fact, these refutations are for the case where variables r are real (which means that axioms and the multiplication rule is not used for variables r).

▶ **Theorem 17.** *Let $\Psi_n(p,q,r) = \Phi_n(p,q) \cup \Gamma_n(p,r)$ be the set of inequalities of Definition 15. Then $\Psi_n(p,q,r)$ has a mixed LS-refutation of size polynomial in n where r are the real variables.*

By combining Theorem 16 with Theorem 17 we have the following corollary separating cutting-planes from LS proof systems.

▶ **Corollary 18.** *The cutting-planes proof system does not polynomially simulate the Lovász-Schrijver proof system.*

Previous to our work, the problem of determining whether the cutting-planes proof system can polynomially simulate the LS-proof system had been open for almost two decades. We note that to the best of our knowledge, the converse problem, of determining whether the LS-proof system can polynomially simulate the cutting-planes proof system remains open.

By combining Theorem 17 with Theorem 10, we have that MAX-LEFT MLP gates can separate graphs with a perfect matching from unbalanced graphs superpolynomially faster than monotone real circuits. In other words, monotone real circuits cannot polynomially simulate MAX-LEFT MLP gates. We leave open the question of whether MLP gates (of any type) can polynomially simulate monotone real circuits.

▶ **Theorem 19.** *Let $g_n : \{0,1\}^{\binom{n}{2}} \rightarrow \{0,1,*\}$ be the partial Boolean function of Corollary 14. Then g_n can be represented by a* single *MAX-LEFT MLP gate of size polynomial in n.*

5 Monotone Linear Programs and Extended Formulations

A *polytope* is the convex hull of a nonempty finite set of vectors in $\mathbb{R}^n$; in particular, a polytope is *nonempty and bounded*. If a polytope $P \subseteq \mathbb{R}^n$ is given by a polynomial number of inequalities[3], then we can easily decide whether a vector $v \in \mathbb{R}^n$ belongs to P. An important observation is that even if P requires an exponential number of inequalities to be defined, we may still be able to test whether $v \in P$ efficiently if we can find a polytope $R \subseteq \mathbb{R}^{n+m}$ in a higher dimension with $m = n^{O(1)}$ such that P is a projection of P' and P' can be described by a polynomial number of inequalities[3].

More precisely, let $P \subseteq \mathbb{R}^n$ be a polytope, and let $P' \subseteq R^{n+m}$ be a polytope defined via a system of inequalities[4] $A(v,y) \leq b$. Then we say that the system $A(v,y) \leq b$ is an extended formulation of P if for each $v \in \mathbb{R}^n$, $v \in P \Leftrightarrow \exists y \in \mathbb{R}^m, A(v,y) \leq b$. We define the size of such extended formulation as the number of rows plus the number of columns in A. For instance, it can be shown that the permutahedron polytope $P_n \subseteq \mathbb{R}^n$, which is defined as the convex-hull of all permutations of the set $[n] = \{1,...,n\}$, requires exponentially many inequalities to be defined. Nevertheless, P_n has extended formulations of size $O(n \log n)$ [12]. On the other hand, it has been shown that for some polytopes, such as the cut polytope, the TSP polytope, etc., even extended formulations require exponentially many inequalities [9, 28].

The process of defining partial Boolean functions via linear programs is closely related, but not equivalent, to the process of defining polytopes via extended formulations. For a partial Boolean function F, let $\mathit{Ones}(F)$, and $\mathit{Zeros}(F)$ denote the set of all inputs $a \in \{0,1\}^n$ such that $F(a) = 1$, and $F(a) = 0$ respectively. Let P_F^1 denote the convex hull of $\mathit{Ones}(F)$ and P_F^0 denote the convex hull of $\mathit{Zeros}(F)$. Defining F via a linear program is equivalent to finding an extended formulation of some polyhedron Q^1 that contains P_F^1 and is disjoint

[3] With coefficients specified by $n^{O(1)}$ bits.

[4] For column vectors $v \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, (v,y) denotes the column vector $(v_1,...,v_n,y_1,...,y_m)$.

from $Zeros(F)$, or an extended formulation of some polyhedron Q^0 that contains P_F^0 and is disjoint from $Ones(F)$. Finding such an extended formulation for Q^1 (resp. Q^0) with a small number of inequalities is clearly, a simpler task than finding a small extended formulation for the polyhedron P_F^1 (resp. P_F^0) itself. For instance, if F is the matching function for general graphs, then F is computable by a polynomial-size Boolean circuit (containing negation gates), and hence this function can be defined via (not necessarily monotone) linear programs of polynomial size[5]. Nevertheless, the corresponding polytope P_F^1 requires extended formulations of exponential size [28].

Let $F : \{0,1\}^n \to \{0,1,*\}$ be a partial monotone Boolean function. A minterm of F is a vector $v \in \{0,1\}^n$ such that $F(v) = 1$ and such that $F(v') \neq 1$ for each $v' \leq v$. Intuitively, a minterm is a minimal vector which causes F to evaluate to 1. Analogously, a maxterm is a vector $v \in \{0,1\}^n$ such that $F(v) = 0$ and $F(v') \neq 0$ for each $v \geq 0$. Intuitively, a maxterm is a maximal vector that causes F to evaluate to 0. We let $\hat{P}_F^1$ be the convex-hull of minterms of F, and let $\hat{P}_F^0$ be the convex-hull of maxterms of F. Let H^1 be a hyperplane containing $\hat{P}_F^1$. For each maxterm v we define the set $S_v^1 = H^1 \cap \{u \mid u \leq v\}$. Analogously, let H^0 be an hyperplane containing $\hat{P}_F^0$. We define the set $S_v^0 = H^0 \cap \{u \mid u \geq v\}$.

▶ **Definition 20** (Monotone Extension Complexity). Let $F : \{0,1\}^n \to \{0,1,*\}$ be a partial monotone Boolean function. Below we define two notions of monotone extension complexity (mxc) for F.

1. We let $mxc_1(F)$ denote the minimum size of an extended formulation for a polytope Q^1 such that

$$\hat{P}_F^1 \subseteq Q^1, \qquad \text{and} \qquad Q \cap \bigcup_v S_v^1 = \emptyset. \tag{2}$$

2. We let $mxc_0(F)$ denote the minimum size of an extended formulation for a polytope Q^0 such that

$$\hat{P}_F^0 \subseteq Q^0, \qquad \text{and} \qquad Q \cap \bigcup_v S_v^0 = \emptyset. \tag{3}$$

The next theorem relates the monotone extension complexity of a partial monotone Boolean function F to the size of MLP representations for F.

▶ **Theorem 21.** *Let $F : \{0,1\}^n \to \{0,1,*\}$ be a partial monotone Boolean function. Then $mxc_1(F)$ is up to a constant factor equal to the minimum size of a* MAX-RIGHT *MLP representation of F. Analogously, the $mxc_0(F)$ is up to a constant factor equal to the minimum size of a* MIN-LEFT MLP *representation of F.*

6 Conclusion

In this work we introduced several models of computation based on the notion of monotone linear programs. In particular, we introduced the notions of weak and strong MLP gates. We reduced the problem of proving lower bounds for the size of LS proofs to the problem of proving lower bounds for the size of MLP circuits with strong gates, and the problem of proving lower bounds on the size of mixed LS proofs to the problem of proving lower bounds on the size of single weak MLP gates.

[5] Note that any function in PTIME can be defined by polynomial-size non-monotone LP programs, due to the fact that linear programming is PTIME complete.

When it comes to comparing MLP gates with other models of computation, we have shown that weak MLP gates are strictly more powerful than monotone Boolean circuits and monotone span programs. Additionally, these gates cannot be polynomially simulated by monotone real circuits. Finally, by combining some results mentioned above, we proved that the cutting-planes proof system is not powerful enough to polynomially simulate the LS proof system. This is the first result showing a separation between the power of these two systems.

Acknowledgments. We would like to thank Pavel Hrubeš and Massimo Lauria for discussion and their valuable suggestions.

References

1 Michael Alekhnovich and Alexander A. Razborov. Satisfiability, branch-width and Tseitin tautologies. In *Proc. of the 43rd Symposium on Foundations of Computer Science*, pages 593–603, 2002.

2 László Babai, Anna Gál, and Avi Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.

3 Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 3(1):1–26, 1996.

4 Gábor Braun, Samuel Fiorini, Sebastian Pokutta, and David Steurer. Approximation limits of linear programs (beyond hierarchies). *Mathematics of Operations Research*, 40(3):756–772, 2015.

5 Mark Braverman and Ankur Moitra. An information complexity approach to extended formulations. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 161–170. ACM, 2013.

6 Samuel R. Buss and Toniann Pitassi. Good degree bounds on Nullstellensatz refutations of the induction principle. *Journal of Computer and System Sciences*, 57(2):162–171, 1998.

7 Stephen A. Cook, Toniann Pitassi, Robert Robere, and Benjamin Rossman. Exponential lower bounds for monotone span programs. *ECCC*, TR16-64, 2016.

8 Sanjeeb Dash. Exponential lower bounds on the lengths of some classes of branch-and-cut proofs. *Mathematics of Operations Research*, 30(3):678–700, 2005.

9 Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald De Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *Journal of the ACM (JACM)*, 62(2):17, 2015.

10 Xudong Fu. Lower bounds on sizes of cutting planes proofs for modular coloring principles. *Proof Complexity and Feasible Arithmetics*, pages 135–148, 1998.

11 Anna Gál and Pavel Pudlák. A note on monotone complexity and the rank of matrices. *Information Processing Letters*, 87(6):321–326, 2003.

12 Michel X. Goemans. Smallest compact formulation for the permutahedron. *Mathematical Programming*, 153(1):5–11, 2015.

13 Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613–622, 2001.

14 Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

15 Armin Haken and Stephen A. Cook. An exponential lower bound for the size of monotone real circuits. *Journal of Computer and System Sciences*, 58(2):326–335, 1999.

16 Russell Impagliazzo, Pavel Pudlák, and Jiri Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

17 M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the 8th Annual Structure in Complexity Theory Conference*, pages 102–111. IEEE CS, 1993.

18 Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(02):457–486, 1997.

19 Jan Krajíček. Interpolation and approximate semantic derivations. *Mathematical Logic Quarterly*, 48(4):602–606, 2002.

20 László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

21 Toniann Pitassi and Nathan Segerlind. Exponential lower bounds and integrality gaps for tree-like lovasz-schrijver procedures. *SIAM Journal on Computing*, 41(1):128–159, 2012.

22 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *The Journal of Symbolic Logic*, 62(03):981–998, 1997.

23 Pavel Pudlak and Jiri Sgall. Algebraic models of computation and interpolation for algebraic proof systems. In *Proc. of Feasible Arithmetic and Proof Complexity, DIMACS Series in Discrete Math. and Theoretical Comp. Sci.*, volume 39, pages 279–295, 1998.

24 Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM (JACM)*, 39(3):736–744, 1992.

25 Alexander A. Razborov. Lower bounds on monotone complexity of the logical permanent. *Mathematical Notes*, 37(6):485–493, 1985.

26 Alexander A. Razborov. Lower bounds for monotone complexity of boolean functions. *American Mathematical Society Translations*, 147:75–84, 1990.

27 Alexander A. Razborov. Proof complexity and beyond. *ACM SIGACT News*, 47(2):66–86, 2016.

28 Thomas Rothvoß. The matching polytope has exponential extension complexity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 263–272. ACM, 2014.

29 Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.