Representations of Monotone Boolean Functions by Linear Programs

MATEUS DE OLIVEIRA OLIVEIRA, University of Bergen, Norway
PAVEL PUDLÁK, Czech Academy of Sciences, Czech Republic

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].

(1) MLP circuits are superpolynomially stronger than monotone Boolean circuits.
(2) MLP circuits are exponentially stronger than monotone span programs over the reals.
(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.

Finally, we establish connections between the problem of proving lower bounds for the size of MLP circuits and the field of extension complexity of polytopes.

CCS Concepts: • **Theory of computation** → **Circuit complexity**; **Proof complexity**.

Additional Key Words and Phrases: Monotone Linear Programming Circuits, Lovász-Schrijver Proof Systems, Feasible Interpolation

ACM Reference Format:
Mateus de Oliveira Oliveira and Pavel Pudlák. 2019. Representations of Monotone Boolean Functions by Linear Programs. 1, 1 (January 2019), 32 pages. https://doi.org/0000001.0000001

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 for monotone Boolean circuits were proved already in 1985 by Razborov [30]. In 1995 Krajíček showed that lower bounds on the monotone complexity of particular partial Boolean functions can be used to prove lower bounds for Resolution, and for some other proof systems such as cutting-planes with bounded coefficients; these results were published in [21]. A similar idea appeared in the same year in a preprint of Bonet et al. which was later published as [4].[2] 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). However, 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

[1] An extended abstract of this work appeared in the proceedings of CCC 2017 [8].
[2] Another article of Razborov [31] was instrumental for Krajíček, although it did not deal with propositional proofs.

Authors' addresses: Mateus de Oliveira Oliveira, University of Bergen, Norway, mateus.oliveira@uib.no; Pavel Pudlák, Czech Academy of Sciences, Prague, Czech Republic, pudlak@math.cas.cz.

result, this method was used to prove exponential lower bounds on the cutting-planes proof system [17, 25]. That lower 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, 12]. An exponential lower bound on the size of monotone span programs has been recently obtained in [33]. For a long time the best known lower bound for this model of computation was of the order of $n^{\Omega(\log n)}$ [11]. 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 [27].[3]

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 [32] Razborov presents the problem of understanding feasible interpolation for stronger systems as one of the most challenging ones in proof complexity theory.

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 $\ell : \mathbb{R}^n \to \mathbb{R} \cup \{*\}$ of the form $\ell(y) = \max\{c \cdot x \mid Ax \leq b + By, x \geq 0\}$ where y is a string of n input real variables, and B is a nonnegative 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 $\ell(\alpha)$ 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 $\ell(\alpha) \leq \ell(\alpha')$ whenever $\ell(\alpha)$ and $\ell(\alpha')$ are defined and $\alpha \leq \alpha'$. We note that the value $\ell(\alpha)$ is considered to be undefined if the associated linear program $\max\{c \cdot x \mid Ax \leq b + B\alpha\}$ has no solution. In this case, we set $\ell(\alpha) = *$. 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 only in the objective function or only in the constraints. We say that an MLP gate is strong if the input variables occur both in the objective function and in the constraints.

MLP circuits are the straightforward generalization of unbounded-fan-in monotone Boolean circuits where gates are MLP gates, instead of Boolean gates. In Theorem 4.3 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 gates (Theorem 5.1). In contrast, we show that weak MLP gates are superpolynomially stronger than monotone Boolean circuits. On the one hand, Razborov has shown 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 $n^{\Omega(\log n)}$ [29]. On the other hand, a classical result in linear programming theory [35] can be used to show that the same function can be computed by weak MLP gates of polynomial size.

In [2, 11], Babai, Gál and Wigderson, and Gál showed that there is a function that can be computed by monotone span programs of linear size but which require superpolynomial-size monotone Boolean circuits. Recently, Cook et al. [33] 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 none of these models can polynomially simulate the other. In Theorem 5.4 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 [33] with Theorem 5.4,

[3]We note however that strong degree lower bounds for Nullstellensatz proofs can be proved using more direct methods [1, 3, 7, 14].

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 [23]. Currently, size lower bounds for these systems have been proved only with respect to tree-like proofs [24], and to static proofs [15, 20]. 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.

It is worth noting that we do not know how to collapse MLP circuits with strong gates into a single strong gate. Nevertheless, in Theorem 6.2 we show that a single weak MLP gate suffices in a monotone interpolation theorem for LS proofs of unsatisfiable sets of *mixed* inequalities of a certain form. Here, a mixed inequality is an inequality which involves both Boolean variables and real variables. Using this interpolation theorem together with 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 (Corollary 6.11).

We show that the cutting-planes proof system cannot polynomially simulate the LS proof system (Corollary 6.9). 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 unsatisfiable set of inequalities, LS proofs can be superpolynomially more concise than cutting-planes proofs. Concerning the other direction, Pitassi and Segerlind have shown that tree-like LS does not polynomially simulate cutting-planes [24]. The problem whether the LS proof system with DAG-like proofs can polynomially simulate the cutting-planes proof system remains open.

Monotone linear programs may be regarded as a simultaneous generalization of monotone Boolean circuits and monotone span programs. Nevertheless, currently there is no lower bound technique that can be used to prove lower bounds both for the size of monotone Boolean circuits and for the size of monotone span programs. Therefore, proving lower bounds for the size of MLP circuits will likely require the development of substantially new techniques. A possible approach is to strengthen lower bound techniques for the size of extended formulations of explicit polytopes. A lower bound on extended formulations is a lower bound on the number of inequalities needed to define an extension of a polytope to some higher dimension. Such lower bounds have been proved, in particular, for polytopes spanned by the 0-1 vectors representing minterms of certain monotone Boolean functions [5, 6, 9, 34]. Nevertheless, 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 seems to be a much harder problem than proving a lower bound for a given polytope, but there are results on extended formulations that go in this direction [5, 6]. However, Theorem 6.10 suggests that this will surely not be easy. It gives an example of a monotone function where the convex-hull of the set of ones requires exponentially large extended formulation, but where the set of ones can be separated from a large set of maxterms by a weak MLP representation of polynomial size.

2 PRELIMINARIES

Monotone Partial Boolean Functions: A *partial Boolean function* is a mapping of the form $F : \{0,1\}^n \to \{0,1,*\}$. Intuitively, the function F should be regarded as being undefined on each point $p \in \{0,1\}^n$ for which $F(p) = *$. The support of F, which is defined as *support*$(F) = F^{-1}(\{0,1\})$, is the set of all points $p \in \{0,1\}^n$ for which F is defined. If p and p' are Boolean strings in $\{0,1\}^n$, then we write $p \geq p'$ to indicate that $p_i \geq p'_i$ for each $i \in \{1, \ldots, n\}$. We say that a partial Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ is *monotone* if $F(p) = 1$ whenever $p \geq p'$ and $F(p') = 1$.

Let $A \in \mathbb{R}^{m\times k}$ denote that A is a real matrix with m rows and k columns. For vectors x and y, $x \leq y$ means that $x_i \leq y_i$ for all coordinates i; the same for matrices and Boolean strings. As an abuse of notation, we write 0 (1) to denote vectors in which all coordinates are equal to 0 (1). For two vectors x and y, we will denote their scalar product by $x \cdot y$.

Linear Programs. A linear program is an optimization problem of the form

$$\max\{c \cdot x \mid Ax \leq b,\ x \geq 0\}, \tag{1}$$

where $A \in \mathbb{R}^{m\times k}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^k$ for some $m, k \in \mathbb{N}$. The dual of the linear program of (1) is defined as follows.

$$\min\{b \cdot y \mid A^T y \geq c,\ y \geq 0\}. \tag{2}$$

According to *linear programming duality*,

$$\max c \cdot x = \min b \cdot y, \tag{3}$$

provided that the maximum in (1) and the minimum in (2) exist.

3 MONOTONE LINEAR-PROGRAMMING GATES

In this section we define the notion of *monotone linear programming gate*, or breifly *MLP gate*.

Definition 3.1 (MLP Gate). *Let A be a matrix in $\mathbb{R}^{m\times k}$, b be a vector in $\mathbb{R}^m$, c be a vector in $\mathbb{R}^k$, and B and C be matrices in $\mathbb{R}^{m\times n}$ with $B \geq 0$ and $C \geq 0$. An* 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 by a monotone linear program. More precisely, we consider the following six types of MLP gates.*

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

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

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

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

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

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

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 that $B \geq 0$ and $C \geq 0$ 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.

Definition 3.2 (MLP-Gate Representation). *We say that an MLP gate $\ell : \mathbb{R}^n \to \mathbb{R} \cup \{*\}$ represents a partial Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ if the following holds true for each $a \in \{0,1\}^n$.*

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

3.1 Sign Representations

We say that an MLP gate ℓ sign-represents a partial Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ if the following conditions can be verified for each $a \in \{0,1\}^n$.

(1) $\ell(a) > 0$ if $F(a) = 1$.
(2) $\ell(a) < 0$ if $F(a) = 0$.

Proposition 3.3. *Let $F : \{0,1\}^n \to \{0,1,*\}$ be a partial Boolean function and assume that F can be represented by an MLP gate of type τ and size s. Then F can be sign-represented by an MLP gate of type τ and size $O(s)$.*

We leave the proof to the reader as an easy exercise.

3.2 Weak vs Strong Gates

Recall that weak MLP gates are gates where input variables occur either only in the objective function, or only in the constraints. On the other hand, strong MLP gates are gates where input variables are allowed to occur both in the objective function and in the constraints.

The distinction between weak and strong gates is motivated by the fact that while weak gates are only able to compute piecewise-linear monotone real functions, strong gates may compute quadratic monotone real functions.

Proposition 3.4. *Let $\ell : \mathbb{R}^m \to \mathbb{R} \cup \{*\}$ be a weak MLP gate. Then the graph*

$$\{(y, \ell(y)) \mid y \in \mathbb{R}^m, \ell(y) \in \mathbb{R}\}$$

is piecewise linear.

Proof. We show that the proposition is valid for MAX-RIGHT MLP gates. The proof that it is valid for other types of weak gates is analogous. Let $\ell(y) = \max\{c \cdot x \mid Ax \leq b + By,\ x \geq 0\}$ be a MAX-RIGHT MLP gate. This gate can be alternatively represented as $\ell(y) = \max\{x_0 \mid Ax \leq b + By,\ x \geq 0,\ x_0 \leq c \cdot x\}$ where x_0 is a new variable. Let P be the polyhedron on variables x, y and x_0 defined by the inequalities $Ax \leq b + By, x \geq 0$ and $x_0 \leq c \cdot x$. Let P' be the polyhedron obtained by projecting P into the variables y and x_0. Then the graph of ℓ is the set $S = \{(y, x_0) \mid \forall x_0'$ such that $(y, x_0') \in P', x_0' \leq x_0\}$. Since S is a union of faces of P', S is piecewise linear. □

On the other hand, the graph of strong gates may not be piecewise linear even for gates with a unique input variable.

Observation 3.5. *Strong MLP gates may compute functions whose graph is not piecewise linear.*

Proof. Consider the following max MLP gate ℓ and min MLP gate ℓ'.

$$\ell(y) = \max\{y \cdot x \mid x \leq y,\ x \geq 0\} \qquad \ell'(y) = \min\{y \cdot x \mid x \geq y,\ x \geq 0\}. \tag{4}$$

Then we have that for each $y \geq \mathbb{R}^+$, $\ell(y) = y^2 = \ell'(y)$. This shows that the graphs of ℓ and ℓ' are not piecewise linear. □

Proposition 3.4 and Observation 3.5 show that strong MLP gates are a strictly stronger model than weak gates when it comes to defining monotone *real* functions. Therefore proving lower bounds for the size of strong MLP gates computing some specific monotone Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ may be harder than proving such lower bounds for the size of weak MLP gates computing F. We note however that it is still conceivable that every partial monotone Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ that can be represented by strong MLP gates of size s, can be also represented by weak MLP gates of size $s^{O(1)}$.

3.3 Boolean Duality vs Linear-Programming Duality

In this section we clarify some relationships between linear-programming duality and MLP representations. Towards this goal, it will be convenient to define the notions of a dual of a given type of gate. More precisely, we say that the type max is dual to min, that max-right is dual to min-left, and that max-left is dual to min-right. If τ is a type of gate we let τ^d denote its dual type. The following observation states that MLP gates of type τ can be simulated by MLP gates of type τ^d of similar complexity.

Observation 3.6. *If a partial real monotone function $f : \mathbb{R}^n \to \mathbb{R} \cup \{*\}$ can be represented by an MLP gate of type τ and size s, then f can be also represented by an MLP gate of type τ^d and size s.*

Proof. We prove the proposition with respect to max-right MLP gates. The proof for other types of gates is analogous. Let $\ell(y) = \max\{c \cdot x \mid Ax \leq b + By,\ x \geq 0\}$ be a max-right MLP gate such that $f(y) = \ell(y)$ for every $y \in \mathbb{R}^n$. Consider the following min-left MLP gate: $\ell(y) = \min\{(b + By) \cdot x \mid A^T x \geq c,\ x \geq 0\}$. Then by linear programming duality, for each $\alpha \in \mathbb{R}^n$, $\ell(\alpha)$ is defined if and only if $\ell'(\alpha)$ is defined and $\ell'(\alpha) = \ell(\alpha)$. □

We say that the types max-right and min-right are *semi-dual* to each other. Analogously, the types max-left and min-left are semi-dual to each other. If τ is a type of gate, we let τ^{sd} be its semi-dual type. It is not clear whether functions that can be represented by weak gates of a given type τ may be also represented by gates of type τ^{sd} without a superpolynomial increase in complexity. However, we will see next that if F is a partial Boolean function which can be represented by an MLP gate of type τ and size s, then the *Boolean-dual* of F can be represented by an MLP gate of type τ^{sd} and size $O(s)$.

We say that a partial monotone Boolean function $F : \{0,1\}^n \to \{0,1,*\}$ is *dualizable* if $F(\neg p_1, \ldots, \neg p_n)$ is well defined whenever $F(p_1, \ldots, p_n)$ is well defined. If F is dualizable, then the *Boolean dual* of F is the partial Boolean function $F^d : \{0,1\}^n \to \{0,1,*\}$ which is obtained by setting $F^d(p) = *$ for each point $p \notin support(F)$, and by setting $F^d(p_1, \ldots, p_n) := \neg F(\neg p_1, \ldots, \neg p_n)$ for each $p \in support(F)$.

Proposition 3.7. *Let $F : \{0,1\}^n \to \{0,1,*\}$ be a dualizable partial Boolean function. If F can be represented by an MLP gate of type τ and size s, then F^d can be represented by an MLP gate of type τ^{sd} and size $O(s)$.*

Proof. We will show that if a function F can be represented by max-right MLP gate of size s, then F^d can be represented by a max-left MLP gate of size $O(s)$. The proof for other types of gates follows an analogous reasoning.

Assume that F can be represented by a MAX-RIGHT MLP gate ℓ. Then by Proposition 3.3, F can be represented by a MAX-RIGHT gate ℓ' such that for each $p \in \{0,1\}^n$, $\ell'(p) > 0$ whenever $F(p) = 1$ and $\ell'(p) < 0$ whenever $F(p) = 0$. In other words, $\ell'(p)$ sign-represents F. Let

$$\ell'(p) = \max\{c \cdot x \mid Ax \leq b + Bp,\ x \geq 0\}$$

be such gate. Then, clearly, the function F^d can be represented by the following MIN-RIGHT MLP gate, where $\bar{1}$ denotes the all-ones vector.

$$\begin{aligned} \ell''(p) &= -\ell'(\bar{1} - p) \\ &= \min\{-c \cdot x \mid Ax \leq b + B(\bar{1} - p),\ x \geq 0\} \\ &= \min\{-c \cdot x \mid -Ax \geq -b - B\bar{1} + Bp,\ x \geq 0\} \end{aligned}$$

□

4 MONOTONE LINEAR PROGRAMMING CIRCUITS

Monotone linear programming circuits (MLP circuits) may be defined as the straightforward generalization of unbounded fan-in monotone Boolean circuits where monotone linear programming gates are used instead of Boolean gates. Formally, it will be convenient for us to define MLP circuits using the notation of straight-line programs, i.e., as a sequence of instructions of a suitable form.

DEFINITION 4.1 (MLP CIRCUIT). *An MLP circuit is a sequence of instructions $C = (I_1, I_2, \ldots, I_r)$ where each instruction I_i has one of the following forms:*

(1) $I_i \equiv Input(y_i)$, where y_i is a variable.
(2) $I_i \equiv y_i \leftarrow c_i$, where y_i is a variable and $c_i \in \mathbb{R}$.
(3) $I_i \equiv y_i \leftarrow \ell_i(y_{i_1}, \ldots, y_{i_{n_i}})$ where y_i is a variable and $\ell_i(y_{i_1}, \ldots, y_{i_{n_i}})$ is an MLP gate with input variables $y_{i_1}, \ldots, y_{i_{n_i}}$ such that $i_j < i$ for each $j \in [n_i]$.

We say that instructions of the third form are MLP instructions. We assume that the last instruction, I_r, is an MLP instruction. We say that the variable y_r, which occurs in the left-hand side of I_r is the output variable of C. For each i such that $I_i \equiv Input(y_i)$, we say that y_i is an input variable.

Let $\mathbf{y} = (y_{j_1}, y_{j_2}, \ldots, y_{j_n})$ be the input variables of C, and let $a \in \mathbb{R}^n$ be an assignment of the variables in $\mathbf{y}$, where $y_{j_l} = a_l$ for each $l \in \{1, \ldots, n\}$. For each $i \in \{1, \ldots, r\}$, the value induced by a on variable y_i, which is denoted by $val_a(y_i)$, is inductively defined as follows.

(1) If $i = j_l$ for some $l \in [n]$, then y_i is an input variable ($I_i \equiv Input(y_i)$). In this case we set $val_a(y_i) = a_l$.
(2) If $I_i \equiv y_i \leftarrow c_i$, then $val_a(y_i) = c_i$.
(3) If $I_i \equiv y_i \leftarrow \ell_i(y_{i_1}, \ldots, y_{i_{n_i}})$, and $val_a(y_{i_j}) \in \mathbb{R}$ for each $j \in \{1, \ldots, r\}$, then $val_a(y_i) = \ell_i(val(y_{i_1}), \ldots, val(y_{i_{n_i}}))$. Otherwise, $val_a(y_i) = *$.

For each assignment $a \in \mathbb{R}^n$ of the variables input variables of C, we let $C(a) = val_a(y_r)$ be the value induced by a on the output variable of C. Intuitively, the values of the variables y_i are computed instruction after instruction. If at step i, the value of the variable y_i is set to $*$ ($val_a(y) = *$), meaning that the linear program associated with the instruction I_i has no solution, then the value $*$ is propagated until the last instruction, and the circuit will output $*$.

Definition 4.2 (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 of type τ.

Theorem 4.3 (From Circuits to Gates). *Let $C = (I_1, \ldots, I_r)$ be an MLP circuit 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, $\ell_C(a) = C(a)$.*

Proof. First, we will prove the theorem with respect to max-right MLP gates. Let $C = (I_1, I_2, \ldots, I_r)$ be an MLP circuit in which all gates are max-right MLP gates. For each $i \in \{1, \ldots, r\}$ if I_i is an MLP instruction, then we let

$$I_i \;\equiv\; y_i \leftarrow \ell_i(y^i) = \max\{c^i \cdot x^i \mid A^i x^i \leq b^i + B^i y^i\},$$

where $y^i = (y_{i_1}, \ldots, y_{i_{n_i}})$ are the input variables of ℓ_i and $x^i = (x^i_1, \ldots, x^i_{k_i})$ are the internal variables of ℓ_i. We let $M = \{i \mid I_i \text{ is an MLP instruction}\}$ be the set of all i's such that I_i is an MLP instruction. We let $\mathbf{y} = (y_{j_1}, \ldots, y_{j_n})$ be the input variables of C, and $\mathbf{x} = x^{i_1} x^{i_2} \ldots x^{i_{|M|}}$ with $i_j \in M$ and $i_1 < i_2 < \ldots < i_{|M|}$ be a tuple containing all internal variables of MLP gates occurring in C. For each $i \in M$, let $\mathbf{A}^i\mathbf{x} \leq \mathbf{b}^i + \mathbf{B}^i\mathbf{y}$ be the system of inequalities obtained from $A^i x^i \leq b^i + B^i y^i$ by replacing each variable $y_{i_j} \in y^i$ which is not an input variable of C, with the value c_{i_j} if $I_{i_j} \equiv y_{i_j} \leftarrow c_{i_j}$, and with the expression $c^{i_j} \cdot x^{i_j}$ if I_{i_j} is an MLP instruction. Now, for $i \in M$, consider the following max-right MLP gate.

$$\boldsymbol{\ell}_i(\mathbf{y}) = \max\{c^i \cdot x^i \mid \mathbf{A}^j\mathbf{x} \leq \mathbf{b}^j + \mathbf{B}^j\mathbf{y},\ j \in M,\ j \leq i\} \tag{5}$$

In other words, the objective function of $\boldsymbol{\ell}_i(\mathbf{y})$ is the same as the objective function of the gate ℓ_i, but the constraints of $\boldsymbol{\ell}_i(\mathbf{y})$ are formed by all inequalities $\mathbf{A}^j\mathbf{x} \leq \mathbf{b}^j + \mathbf{B}^j\mathbf{y}$ corresponding to constraints of gates ℓ_j for $j < i$. If u is an assignment of the tuple of variables $\mathbf{x}$, then for each $j \in M$, we let $u^j \in \mathbb{R}^{k_j}$ be the assignment induced by u on the internal variables $x^j = (x_{j_1}, \ldots, x_{j_{k_i}})$ of gate ℓ_j. Let a be an assignment of the input variables $\mathbf{y}$, and u be an assignment of the internal variables $\mathbf{x}$. Then we say that the pair (a, u) is consistent with $\boldsymbol{\ell}_i$ if (a, u) satisfies all constraints of $\boldsymbol{\ell}_i$.

The following claim implies that for each $a \in \mathbb{R}^n$ such that $C(a)$ is defined, the value $C(a)$ is equal to the value $\boldsymbol{\ell}_r(a)$.

Claim 4.4. *Let $a \in \mathbb{R}^n$. If $C(a)$ is defined then the following conditions are satisfied for each $i \in M$.*

(1) There exists an assignment u of the variables $\mathbf{x}$, such that (a, u) is consistent with $\boldsymbol{\ell}_i$ and for each $j \in M$ with $j \leq i$, $c^j \cdot u^j = val_a(y_j)$.
(2) For each assignment u of the variables $\mathbf{x}$, such that (a, u) is consistent with $\boldsymbol{\ell}_i$, and each $j \in M$ with $j \leq i$, $c^j \cdot u^j \leq val_a(y_j)$.
(3) $\boldsymbol{\ell}_i(a) = val_a(y_i)$.

We note that if $|M| = 1$ then the circuit has a unique MLP gate and the claim is trivial. Therefore, we assume that $|M| \geq 2$. Let $a \in \mathbb{R}^n$ be an assignment of the input variables $\mathbf{y}$ such that $C(a)$ is defined. The proof of Claim 4.4 is by induction on i. In the base case, let i be the smallest number in M. In this case, $y_i \leftarrow \ell_i(y^i)$ is the first MLP gate occurring in C, and therefore the gate $\boldsymbol{\ell}_i(\mathbf{y})$ has precisely the same objective function and constraints as $\ell_i(y^i)$. This implies that the value $\boldsymbol{\ell}_i(a)$ is equal to the value induced by a on y_i. Therefore, the claim is valid in the base case. Now, let l be an arbitrary number in M and let i be the greatest number in M which smaller than l. Let $I_l \equiv\ y_l \leftarrow \ell_l(y^l)$, where $y^l = (y_{l_1}, ..., y_{l_{n_l}})$. Then the objective function of $\boldsymbol{\ell}_l(\mathbf{y})$ is $c^l \cdot x^l$, and the constraints of $\boldsymbol{\ell}_l(\mathbf{y})$ contain all constraints of $\boldsymbol{\ell}_i(\mathbf{y})$ together with the constraints $\mathbf{A}^l\mathbf{x} \leq \mathbf{b}^l + \mathbf{B}^l\mathbf{y}$ which are obtained from $A^l x^l \leq b^l + B^l y^l$ by making the substitution $y_{l_j} \leftarrow c^{l_j} \cdot x^{l_j}$ for each $j \in \{1, ..., n_l\}$. By the induction hypothesis, Conditions 1, 2 and 3 are satisfied with respect to $\boldsymbol{\ell}_i$. Therefore by Condition 1, there is an assignment u of $\mathbf{x}$ such that $c^{l_j} \cdot u^{l_j} = val_a(y^{l_j})$ for each $j \in \{1, ..., n_l\}$. Now, since the internal variables x^l of gate ℓ_l do not occur with non-zero coefficient in the constraints of $\boldsymbol{\ell}_i$, we may assume that when restricted to these variables, the assignment u^l is the one that maximizes the objective function $c^l \cdot x^l$ of the linear program which defines $\ell_l(y^{l_1}, ..., y^{l_{n_l}})$ when each variable y^{l_j} is set to $c^{l_j} \cdot u^{l_j} = val_a(y^{l_j})$. When assigning this particular u to the variables $\mathbf{x}$, we have that $c^l \cdot x^l = val_a(y^l)$. This implies that Condition 1 is also satisfied with respect to $\boldsymbol{\ell}_l$. Additionally, we have that $\boldsymbol{\ell}_l(a)$ is at least $val_a(y^l)$. Now, by Condition 2, $c^{l_j} \cdot u^{l_j} \leq val_a(y_{l_j})$ for each $j \in \{1, ..., n_l\}$. Therefore, since ℓ_l is monotone, we also have that $c^l \cdot x^l \leq val_a(y^l)$. This implies that Condition 2 is also satisfied with respect to $\boldsymbol{\ell}_l$. Additionally, this shows that $\boldsymbol{\ell}_l(a)$ is at most $val_a(y^l)$. By combining the two bounds obtained for $\boldsymbol{\ell}_l(a)$, we have that $\boldsymbol{\ell}_l(a) = val_a(y^l)$. This shows that Condition 3 is also satisfied with respect to $\boldsymbol{\ell}_l$.

The proof that the theorem holds for circuits consisting of MIN-RIGHT MLP gates is analogous to the proof for the case of MAX-RIGHT MLP gates established above. If C is a circuit containing only MIN-LEFT MLP gates, then we first transform this circuit into a circuit C' consisting only of MAX-RIGHT gates using linear program duality. In other words, we replace each MIN-LEFT MLP gate in C with an equivalent MAX-RIGHT MLP gate. Then applying the proof described above, we construct a MAX-RIGHT MLP gate $\boldsymbol{\ell}_{C'}(\mathbf{y})$. Once this is done, we apply linear-programming duality one more time to convert $\boldsymbol{\ell}_{C'}(\mathbf{y})$ into an equivalent MIN-LEFT GATE. Analogously, if C is a circuit with MAX-LEFT MLP gates, then we first convert it into an equivalent circuit consisting of MIN-RIGHT gates, then transform it into a single MIN-RIGHT MLP gate in analogy with the proof described above, and finally, convert this gate back to an equivalent MAX-LEFT MLP gate. □

While weak MLP gates define piecewise linear functions, strong MLP gates define piecewise quadratic functions. The composition of piecewise quadratic functions is not piecewise quadratic in general. Therefore a similar theorem does not hold true for strong gates.

5 WEAK MLP GATES VS MONOTONE BOOLEAN CIRCUITS

We say that an MLP gate ℓ *sharply* represents a partial Boolean function $F : \{0, 1\}^n \to \{0, 1, *\}$ if $\ell(a) = 1$ whenever $F(a) = 1$, and $\ell(a) = 0$ whenever $F(a) = 0$. In this section we show that partial Boolean functions that can be represented by monotone Boolean circuits of size s may 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 monotone Boolean circuits of superpolynomial size.

THEOREM 5.1. *Let $F : \{0, 1\}^n \to \{0, 1, *\}$ be a partial Boolean function, and let C be a monotone Boolean circuit of size s representing F. Then for any type τ, F can be sharply represented by an MLP gate of type τ and size $O(s)$.*

Proof. Clearly, it is enough to prove the theorem with respect to weak gates, since strong gates are at least as powerful as weak gates. The $\wedge$ gate can be sharply represented by the following max-right and min-right MLP gates respectively.

(1) $\ell_{\wedge}^{max-right}(p_1, p_2) = \max\{x \mid x \leq p_1,\ x \leq p_2, x \geq 0\}$.
(2) $\ell_{\wedge}^{min-right}(p_1, p_2) = \min\{x \mid x \geq p_1 + p_2 - 1, x \geq 0\}$.

Therefore, by linear-programming duality, the $\wedge$ gate can be sharply represented by constant size min-left and max-left MLP gates $\ell_{\wedge}^{min-left}$ and $\ell_{\wedge}^{max-left}$ respectively.

Analogously, the $\vee$ gate can be sharply represented by the following max-right and min-right MLP gates respectively.

(1) $\ell_{\vee}^{max-right}(p_1, p_2) = \max\{x_1 + x_2 \mid x_1 \leq p_1,\ x_2 \leq p_2,\ x_1 + x_2 \leq 1\}$.
(2) $\ell_{\vee}^{min-right}(p_1, p_2) = \min\{x \mid x \geq p_1,\ x \geq p_2, x \geq 0\}$.

Again, by linear-programming duality, the $\vee$ gate can also be sharply represented by suitable min-left and max-left MLP gates $\ell_{\vee}^{min-left}$ and $\ell_{\vee}^{max-left}$ of constant size.

Now let C be a Boolean circuit representing F. Then for each type τ we can construct an MLP circuit C^{τ} which sharply represents F as follows. Replace each $\wedge$ gate of C by the corresponding MLP gate $\ell_{\wedge}^{\tau}$ of type τ, and each $\vee$ gate by the corresponding MLP gate $\ell_{\vee}^{\tau}$. Then C^{τ} has size $O(s)$, and that C^{τ} sharply simulates F. Since all gates in C^{τ} have type τ, by Theorem 4.3, there is an MLP gate ℓ^{τ} of type τ and size $O(s)$ that sharply represents F. □

Let $BPM_n : \{0,1\}^{n^2} \rightarrow \{0,1\}$ 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 [35]) states that the function BPM_n has small max-right MLP representations.

Theorem 5.2. *The Boolean function $BPM_n : \{0,1\}^{n^2} \rightarrow \{0,1\}$ can be represented by a* max-right *MLP gate of size $n^{O(1)}$.*

Proof. Let $[n] = \{1, ..., n\}$, and $E \subseteq [n] \times [n]$ be a bipartite graph. We represent a subgraph of E as a 0/1 vector with n^2 coordinates, which has a 1 at position M_{ij} if and only if (i, j) is an edge of E. The bipartite perfect matching polytope associated with E, which is denoted by $P(E)$, is the convex-hull of all vectors $M \in \{0,1\}^{n^2}$ which correspond to a perfect matching in E. Note that if E has no perfect matching then $P(E)$ is simply empty. It can be shown (Schrijver [35], Theorem 18.1) that the polytope $P(E)$ is determined by the following system of inequalities.

System 1:

(1) $x \geq 0$.
(2) $\sum_{(i,j)\in E} x_{ij} = 1$, for each $i \in [n]$.
(3) $\sum_{(i,j)\in E} x_{ij} = 1$, for each $j \in [n]$.

In other words, if $u \in \mathbb{R}^{n^2}$ is a 0/1 vector representing a perfect matching in E, then all inequalities of System 1 are satisfied if we set $x = u$. Conversely, each vector $u \in \mathbb{R}^{n^2}$ that satisfies all inequalities in System 1 is a convex combination of 0/1 vectors corresponding to perfect matchings in E.

Now, consider the following system of inequalities.

System 2:

(1) $x \geq 0$.

(2) $\sum_j x_{ij} = 1$, for each $i \in [n]$.
(3) $\sum_i x_{ij} = 1$, for each $j \in [n]$.
(4) $x \leq p$.

If a 0/1 vector $w \in \mathbb{R}^{n^2}$ represents a graph $E \subseteq [n] \times [n]$ containing a perfect matching, then some $u \leq w$ represents a perfect matching in E. Therefore, by setting $p = w$ and $x = u$, all inequalities of System 2 are satisfied.

Now let $w \in \mathbb{R}^{n^2}$ be a 0/1 vector such that for some $u \in \mathbb{R}^{n^2}$, the assignment $p = w$ and $x = u$ satisfies all inequalities of System 2. Then the graph represented by w has a perfect matching according to the theorem cited above. □

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. By combining this result with Theorem 5.2, we have the following corollary.

Corollary 5.3. 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 [28]; see also Corollary 6.11 for a stronger result.

5.1 Monotone Span Programs

Monotone span programs (MSP) were introduced by Karchmer and Wigderson [19]. 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 $w \in \{0, 1\}^n$.

$$F(w) = \begin{cases} 1 \;\Rightarrow\; \exists y,\; y^T A^\rho_{\langle w\rangle} = c^T \\ 0 \;\Rightarrow\; \neg\exists y,\; y^T A^\rho_{\langle w\rangle} = c^T \end{cases} \tag{6}$$

That is, if $F(w) = 1$ then c is a linear combination of the rows of $A^\rho_{\langle w\rangle}$, while if $F(w) = 0$, 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, which will be proved in Subsection 5.2, states that functions that can be represented by small MSPs over the reals can also be represented by small min-right MLP gates.

Theorem 5.4. *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 $\exp(n^{\Omega(1)})$ [33]. On the other hand, since by Theorem 5.1, 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 5.5. *Weak MLP gates cannot be polynomially simulated by monotone span programs over the reals.*

5.2 Proof of Theorem 5.4

In this section we prove Theorem 5.4. As an intermediate step we define the notion of *nonnegative monotone span program* (NONNEGATIVE-MSP). Such a NONNEGATIVE-MSP is specified by a pair $(A^\rho, c)^+$ consisting of a labeled matrix $A^\rho = (A, \rho)$, and a vector c, just as in the case of monotone span programs. The only difference is in the way in which such programs are used to represent functions. We say that a NONNEGATIVE-MSP $(A^\rho, c)^+$ represents a partial Boolean function $F : \{0,1\}^n \rightarrow \{0,1,*\}$ if the following conditions are satisfied for each $w \in \{0,1\}^n$.

$$F(w) = \begin{cases} 1 \Rightarrow & \exists y \geq 0,\ y^T A^\rho_{\langle w \rangle} = c^T \\ 0 \Rightarrow & \neg\exists y \geq 0,\ y^T A^\rho_{\langle w \rangle} = c^T \end{cases} \tag{7}$$

Note that while MSP representations are defined in terms of linear combinations of rows of A^ρ, NONNEGATIVE-MSP representations are defined in terms of *nonnegative* linear combinations of rows of A^ρ.

PROPOSITION 5.6. *Let $F : \{0,1\}^n \rightarrow \{0,1\}$ 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* NONNEGATIVE-*MSP of size $O(s)$ over the reals.*

PROOF. Let $A^\rho = (A, \rho)$ be a labeled matrix over $\mathbb{R}$, and let (A^ρ, c) be a span program over $\mathbb{R}$. Let $B = \begin{bmatrix} A \\ -A \end{bmatrix}$. In other words, for each row a_i of A, the matrix B has a row a_i, and a row $-a_i$. Now let ρ' be the function that labels the rows of B in such a way that the rows corresponding to a_i and $-a_i$ in B are labeled with the same label as row i of A. Then for each $w \in \{0,1\}^n$, c is equal to a linear combination of rows of $A^\rho_{\langle w \rangle}$ if and only if c is equal to a nonnegative linear combination of rows of $B^{\rho'}_{\langle w \rangle}$. Therefore, $(B^{\rho'}, c)^+$ is a NONNEGATIVE-MSP of size $O(s)$ representing F. □

Therefore, it is enough to show that any partial Boolean function that can be represented by NONNEGATIVE-MSPs of size s can also be represented by MIN-RIGHT MLP gates of size $O(s)$. Consider the condition

$$\exists y \geq 0,\ y^T A^\rho_{\langle w \rangle} = c^T. \tag{8}$$

In other words, the formula in Equation (8) is satisfied if and only if the row vector c^T is a nonnegative linear combination of the rows of $A^\rho_{\langle w \rangle}$. Let $y \geq 0$ be a nonnegative vector such that $y^T A^\rho_{\langle w \rangle} = c^T$. Then we have that for each $x \in \mathbb{R}^k$ (where k is the number of columns in A), the fact that $A^\rho_{\langle w \rangle} x \geq 0$ implies that $c \cdot x = (y^T A^\rho_{\langle w \rangle})x = y^T (A^\rho_{\langle w \rangle} x) \geq 0$. In particular $c \cdot x \geq 0$ whenever $x \geq 0$ and $A^\rho_{\langle w \rangle} x \geq 0$. Conversely, assume that for some $x \geq 0$ and for some $b \geq 0$, we have that $Ax^\rho_{\langle w \rangle} = b$ and $c \cdot x \geq 0$. Then by linear programming duality, we have that $\min\{y^T b \mid y^T A^\rho_{\langle w \rangle} = c^T,\ y \geq 0\} \geq c \cdot x \geq 0$. This implies that there exists some $y \geq 0$ such that $y^T A^\rho_{\langle w \rangle} = c^T$. In summary, we have argued about the validity of the following equivalence.

$$\exists y \geq 0,\ y^T A^\rho_{\langle w \rangle} = c^T \ \Leftrightarrow\ \min\{c \cdot x \mid A^\rho_{\langle w \rangle} x \geq 0, x \geq 0\} \geq 0. \tag{9}$$

Now let $\{p_1, ..., p_n, *\}$ be the codomain of the row labeling function ρ, $p = (p_1, ..., p_n)$, and let $A'x \geq Bp$ be the system of inequalities obtained from the labeled matrix A^ρ as follows. For each i, let a_i be the i-th row of A. If this row is unlabeled (meaning that $\rho(i) = *$), the system $A'x \geq Bp$ has the inequality $a_i x \geq 0$. On the other hand, if this row is labeled with variable p_j (meaning that $\rho(i) = p_j$), then $A'x \geq Bp$ has the inequality $a_i x \geq \alpha(p_j - 1)$ where $\alpha \in \mathbb{R}^+$ is a positive number that is large enough to make the inequality irrelevant when p_j is set to 0. Then for each assignment

$w \in \{0,1\}^n$ of the variables p,

$$\min\{c \cdot x \mid A'x \geq Bw, x \geq 0\} = \min\{c \cdot x \mid A^{\rho}_{\langle w \rangle} x \geq 0, x \geq 0\}. \tag{10}$$

Now, consider the MIN-RIGHT MLP gate $\ell(p) = \min\{c \cdot x \mid A'x \geq Bp, x \geq 0\}$. Then for each $w \in \{0,1\}^n$, we have that

$$F(w) = \begin{cases} 1 & \Rightarrow \ell(w) \geq 0 \\ 0 & \Rightarrow \ell(w) < 0. \end{cases} \tag{11}$$

Finally, let $\varepsilon = \min_{w \in \{0,1\}^n} \{|\ell(w)| \mid \ell(w) < 0\}$ be the minimum absolute value of $\ell(w)$ where the minimum is taken over all inputs $w \in \{0,1\}^n$ which evaluate to a number strictly less than zero, and let

$$\ell'(w) = \min\{c \cdot x + x' \mid x' = \varepsilon/2,\ A'x \geq Bw, x \geq 0\}.$$

Then $\ell'(w) = \ell(w) + \varepsilon/2$ and therefore, for each $w \in \{0,1\}^n$, we have that

$$F(w) = \begin{cases} 1 & \Rightarrow \ell'(w) \geq \varepsilon/2 > 0 \\ 0 & \Rightarrow \ell'(w) < -\varepsilon/2 < 0. \end{cases} \tag{12}$$

In other words, ℓ' is a MIN-RIGHT MLP representation of F.

6 LOVÁSZ-SCHRIJVER AND CUTTING-PLANES PROOF SYSTEMS

6.1 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 [23]. 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 over variables $p_1, \ldots, p_n$, 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) $0 \leq p_j \leq 1$
  (2) $p_i^2 - p_i = 0$ (integrality).
- Rules:
  (1) *Positive linear combinations of inequalities.*
  (2) *Multiplication:* given a linear inequality $\sum_i c_i p_i - d \geq 0$, and a variable p_j, derive

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

  (3) *Weakening rule:*

$$\text{from } \textstyle\sum_i c_i p_i - d \geq 0, \text{ derive } \sum_i c_i p_i - d' \geq 0 \text{ for any } d' < d.$$

In these inequalities, p_i are variables representing Boolean values, and c_i, d, d' are real constants.

We note that positive linear combinations may involve both linear and quadratic inequalities, but the multiplication rule can only be applied to linear inequalities. Hence, all inequalities occurring in a proof are at most quadratic. Axiom (2) 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 (1) and Rule (2). We also observe that the inequality $1 \geq 0$ can be derived from the axioms $p_i \geq 0$ and $1 - p_i \geq 0$. Therefore the weakening rule can be simulated by an application of these axioms together with linear combinations.

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 [23].

Superpolynomial lower bounds on the size of LS proofs have been obtained only in the restricted case of tree-like proofs [24]. 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 [16]. 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 6.5 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. A partial result in this direction was obtained by Pitassi and Segerlind, who showed that tree-like LS does not polynomially simulate cutting-planes [24].

In this paper we will consider general (i.e., DAG-like) proofs. Thus, a sequence of inequalities Π is a derivation of an inequality $\sum_i c_i p_i - d \geq 0$ from a set of inequalities Φ if every inequality in Π 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 $d > 0$.

6.2 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 $\Psi(p, q, r)$ be an unsatisfiable Boolean formula which is a conjunction of formulas $\Phi(p, q)$ and $\Gamma(p, r)$ where q and r are disjoint sets of variables. Since $\Psi(p, q, r)$ is unsatisfiable, it must be the case that for each assignment a of the variables p, either $\Phi(a, q)$ or $\Gamma(a, r)$ is unsatisfiable, or both. Given a proof Π of unsatisfiability for $\Psi(p, q, r)$, an *interpolant* is a Boolean circuit $C(p)$ such that for every assignment a to the variables p,

(1) if $C(a) = 1$, then $\Phi(a, q)$ is unsatisfiable,
(2) if $C(a) = 0$, then $\Gamma(a, r)$ 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 [21]. 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 [25, 26].

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 some kind) of polynomial size (in the size of the proof) provided that all variables p appear negatively in $\Phi(p, q)$, (or positively in $\Gamma(p, r)$). In the case of resolution proofs, the interpolating circuits are simply *monotone Boolean circuits* [21, 22]. In the case of cutting-planes proofs, the interpolants are *monotone real circuits* [25]. 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 [17, 25]. Another proof system for which one can prove lower bounds (although only on the degree of refutations) using monotone feasible interpolation is the Nullstellensatz Proof System [27]. In this proof system, the monotone interpolants are given in terms of monotone span programs[4] [27].

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. We will show that the Lovász-Schrijver proof system has the monotone feasible interpolation property with the interpolants computed by MLP circuits with strong gates.

6.3 Feasible Interpolation for the Lovász-Schrijver System

Let $F_1(q) - c_1 \geq 0, F_2(q) - c_2 \geq 0, \ldots, F_m(q) - c_m \geq 0$ be a sequence of linear inequalities over a set of variables q. We say that a linear inequality $F(q) - c \geq 0$ is obtained from this sequence in one *lift-and-project* step, or simply *lap*-step for short, if

$$\begin{aligned} F(q) - c \quad = \quad & \textstyle\sum_{ij} \alpha_{ij} q_i (F_j(q) - c_j) + \\ & \textstyle\sum_{ij} \beta_{ij} (1 - q_i)(F_j(q) - c_j) + \\ & \textstyle\sum_i \gamma_i (q_i - q_i^2) \end{aligned} \tag{13}$$

for some $\alpha_{ij}, \beta_{ij}, \gamma_j \geq 0$. A refutation in the LS proof system for an unsatisfiable set of inequalities $\Phi(q)$ can naturally be regarded as a sequence $L_1 \geq 0, \ldots, L_m \geq 0$ of linear inequalities where for each $i \in \{1, \ldots, m\}$, the inequality $L_i \geq 0$ is either in $\Phi(q)$, or is obtained from $L_1 \geq 0, \ldots, L_{i-1} \geq 0$ by the application of one *lap*-step. Intuitively, inequalities involving quadratic terms, obtained as instances of the integrality axiom or by the application of the multiplication rule, are regarded as intermediate steps towards the derivation of new linear inequalities. [5]

Let p, q and r be tuples of Boolean variables. We say that an unsatisfiable set of inequalities $\Phi(p, q) \cup \Gamma(p, r)$ is *monotonically separable* if all p-variables occurring in inequalities of $\Phi(p, q)$ have negative coefficients. The next theorem states that LS-proofs for monotonically separable unsatisfiable sets of inequalities can be interpolated using MLP circuits constituted of MAX MLP gates.

Theorem 6.1. *Let $\Phi(p, q) \cup \Gamma(p, r)$ be a monotonically separable unsatisfiable set of inequalities, and let $p = (p_1, \ldots, p_n)$. Let Π be an LS refutation of $\Phi(p, q) \cup \Gamma(p, r)$. Then one can construct in polynomial time 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 $a \in \{0, 1\}^n$,*

[4] In the context of polynomial calculus, alternative methods (e.g. [1, 18]) yield stronger lower bounds than the monotone interpolation technique.

[5] Note that pure linear combinations can be easily simulated by a lap-step with $\alpha_{ij} = \beta_{ij}$ and $\gamma_i = 0$.

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

In particular, the size of the circuit C is polynomial in the size of Π.

Proof. The proof is divided into three parts. We start by recalling the idea of feasible interpolation for LS in the non-monotone case as presented in [26]. Then we explain what is needed to obtain monotone gates. Finally we define explicitly the gate simulating one lap step of the given Lovász-Schrijver proof.

(1) For the sake of simplicity, we will assume that the inequalities $0 \le q_i \le 1$ and $0 \le r_i \le 1$ are included in Φ and Γ. Let

$$E_1(p) + F_1(q) + G_1(r) - e_1 \ge 0, \ \ldots \ , E_m(p) + F_m(q) + G_m(r) - e_m \ge 0 \tag{14}$$

be the linear inequalities of an LS refutation of $\Phi(p, q) \cup \Gamma(p, r)$. Since the last inequality is a contradiction, the linear forms E_m, F_m, G_m are zeros and $e_m > 0$. Let $a \in \{0, 1\}^n$ be an assignment to variables p. Substituting a for p into the proof we get a refutation

$$F_1(q) + G_1(r) + E_1(a) - e_1 \ge 0, \ \ldots \ , F_m(q) + G_m(r) + E_m(a) - e_m \ge 0 \tag{15}$$

of $\Phi(a, q) \cup \Gamma(a, r)$ (note that the last inequality is $-e_m \ge 0$ as in the proof above). Our aim now is to split the restricted proof into two proofs

$$F_1(q) - c_1 \ge 0, \ \ldots \ , F_m(q) - c_m \ge 0 \quad \text{and} \quad G_1(r) - d_1 \ge 0, \ \ldots \ , G_m(r) - d_m \ge 0 \tag{16}$$

in such a way that the first sequence of inequalities is a potential refutation of $\Phi(a, q)$, the second sequence of inequalities is a potential refutation of $\Gamma(a, r)$, and

$$c_j + d_j \ge e_j - E_j(a) \quad \text{for } j \in \{1, \ldots, m\}. \tag{17}$$

Since (15) is a refutation of $\Phi(a, q) \cup \Gamma(a, r)$, we have that $e_m - E_m(a) > 0$. Therefore, (17) implies that $c_m > 0$ or $d_m > 0$. Hence, in (16), either the left sequence is a refutation of $\Phi(a, q)$, or the right sequence is a refutation of of $\Gamma(a, r)$, or both sequences are refutations of their respective sets of inequalities.

We now describe how such a splitting can be constructed. First, suppose $E_j(p) + F_j(q) + G_j(r) - e_j \ge 0$ is an inequality in $\Phi(p, q)$. Then $G_j(r) = 0$, and we split $E_j(a) + F_j(q) + G_j(r) - e_j \ge 0$ into

$$F_j(q) + E_j(a) - e_j \ge 0 \quad \text{and} \quad 0 \ge 0. \tag{18}$$

It is important to note that since $\Phi(p, q) \cup \Gamma(r, q)$ is monotonically separable, all p-variables occurring in the linear form $E_j(p)$ have negative coefficients. Therefore, the function $e_j - E_j(p)$ is monotone in p. Additionally, this function can be computed using a single MAX MLP gate (or even by a MAX-LEFT MLP gate).

Now, if $E_j(p) + F_j(q) + G_j(r) \ge e_j$ is an inequality in $\Gamma(p, r)$, we split the inequality into

$$0 \ge 0 \quad \text{and} \quad G_j(r) + E_j(a) - e_j \ge 0. \tag{19}$$

We note that in this case, the function $e_j - E_j(p)$ is not necessarily monotone in p, since some coefficients in the linear form $E_j(p)$ may be positive. Nevertheless, this is not important, because the monotone interpolant circuit we want to construct will only take into consideration inequalities concerning the q part part of the splitting.

Now suppose that $E_t(p) + F_t(q) + G_t(r) \geq e_t$ follows from previous inequalities and suppose we have already split the previous part of the proof. Substituting a for p in the t-th *lap*-step we obtain an equality of the following form.

$$\begin{gathered}
F_t(q) + G_t(r) + E_t(a) - e_t = \\
\sum_{ij} \alpha_{ij} a_i (F_j(q) + G_j(r) + E_j(a) - e_j) + \sum_{ij} \beta_{ij}(1 - a_i)(F_j(q) + G_j(r) + E_j(a) - e_j) + \\
\sum_{ij} \alpha'_{ij} q_i (F_j(q) + G_j(r) + E_j(a) - e_j) + \sum_{ij} \beta'_{ij}(1 - q_i)(F_j(q) + G_j(r) + E_j(a) - e_j) + \\
\sum_{ij} \alpha''_{ij} r_i (F_j(q) + G_j(r) + E_j(a) - e_j) + \sum_{ij} \beta''_{ij}(1 - r_i)(F_j(q) + G_j(r) + E_j(a) - e_j) + \\
\sum_i \gamma_i (a_i - a_i^2) + \\
\sum_i \gamma'_i (q_i - q_i^2) + \sum_i \gamma''_i (r_i - r_i^2)
\end{gathered} \tag{20}$$

In the sums, we have $j < t$ and the indices i range over the sets of indices of the corresponding variables p, q, r. All these linear combinations are nonnegative, i.e., the coefficients α_{ij}, α'_{ij}, α''_{ij}, β_{ij}, β'_{ij}, β''_{ij}, γ_i,γ'_i, and γ''_i are nonnegative. Note that the term $\sum_i \gamma_i(a_i - a_i^2)$ is always zero, since by assumption $a_i \in \{0, 1\}$. By setting $\delta_j = \sum_i (\alpha_{ij} a_i + \beta_{ij}(1 - a_j))$, for each j, and by noting that δ_j is nonnegative, (20) can be simplified as follows.

$$\begin{gathered}
F_t(q) + G_t(r) + E_t(a) - e_t = \\
\sum_{ij} \alpha'_{ij} q_i (F_j(q) + G_j(r) + E_j(a) - e_j) + \sum_{ij} \beta'_{ij}(1 - q_i)(F_j(q) + G_j(r) + E_j(a) - e_j) + \\
\sum_{ij} \alpha''_{ij} r_i (F_j(q) + G_j(r) + E_j(a) - e_j) + \sum_{ij} \beta''_{ij}(1 - r_i)(F_j(q) + G_j(r) + E_j(a) - e_j) + \\
\sum_i \gamma'_i (q_i - q_i^2) + \sum_i \gamma''_i (r_i - r_i^2) + \\
\sum_j \delta_j (F_j(q) + G_j(r) + E_j(a) - e_j).
\end{gathered} \tag{21}$$

By substituting $-c_j - d_j$ for $E_j(a) - e_j$ in (21) and rearranging terms, we get the following inequality.

$$\begin{array}{ccc} & F_t(q) + G_t(r) + E_t(a) - e_t \;\; \geq & \\ \sum_{ij} \alpha'_{ij} q_i (F_j(q) - c_j) & + & \sum_{ij} \alpha''_{ij} r_i (G_j(r) - d_j) + \\ \sum_{ij} \beta'_{ij} (1 - q_i)(F_j(q) - c_j) & + & \sum_{ij} \beta''_{ij} (1 - r_i)(G_j(r) - d_j) + \\ \sum_i \gamma'_i (q_i - q_i^2) & + & \sum_i \gamma''_i (r_i - r_i^2) + \\ \sum_j \delta_j (F_j(q) - c_j) & + & \sum_j \delta_j (G_j(r) - d_j) + \\ \sum_{ij} \alpha'_{ij} q_i (G_j(r) - d_j) & + & \sum_{ij} \beta'_{ij} (1 - q_i)(G_j(r) - d_j) + \\ \sum_{ij} \alpha''_{ij} r_i (F_j(q) - c_j) & + & \sum_{ij} \beta''_{ij} (1 - r_i)(F_j(q) - c_j). \end{array} \tag{22}$$

It is important to realize what is going on here. We want to modify the proof so that it can be split into two parts and right-hand side of (22) should be a step towards this goal. Therefore we need two conditions to be satisfied:

(1) the inequality *"right-hand side of (22)* ≥ 0*"* is derivable in the Lovász-Schrijver system by a single lap step from inequalities $F_j(q) - c_j \geq 0$, $G_j(r) - d_j \geq 0$, and
(2) it is at least as strong as $F_t(q) + G_t(r) + E_t(a) - e_t \geq 0$.

First we observe that the substitution does not change the coefficients at quadratic terms of the right-hand side of (21). Hence quadratic terms cancel each other also in (22). Thus the expression has the form of a lap step, which verifies the first condition. For the second, we have to check that the coefficients at variables in $F_t(q) + G_t(r) + E_t(a) - e_t \geq 0$ are at least as large as in the right-hand side of (22) and so is the constant term. This can also be easily verified by inspecting the terms.

To sum up, we should view the formal inequality (22) as a system of inequalities, one for each variable and one for the constant terms.

Next we note that each line in the right-hand side of (22), except for the last two, splits into expressions involving only q variables and another one involving only r variables. Let

$$\begin{aligned} P(q,r) \quad = \quad & \sum_{ij} \alpha'_{ij} q_i (G_j(r) - d_j) + \sum_{ij} \beta'_{ij} (1 - q_i)(G_j(r) - d_j) + \\ & \sum_{ij} \alpha''_{ij} r_i (F_j(q) - c_j) + \sum_{ij} \beta''_{ij} (1 - r_i)(F_j(q) - c_j) \end{aligned} \tag{23}$$

be the polynomial corresponding to the two last lines of (22). The key observation is that, since the inequality $F_t(q) + G_t(r) + E_t(a) - e_t \geq 0$ is linear, all quadratic terms $q_i r_j$ in the polynomial $P(q,r)$ must cancel. Hence $P(q,r)$ is a linear polynomial. Clearly $P(q,r) \geq 0$ whenever $q_i \geq 0$, $G_j(r) - d_j \geq 0$, $1 - q_i \geq 0$, $r_i \geq 0$, $F_j(q) - c_j \geq 0$, $1 - r_i \geq 0$ and $F_j(q) - c_j \geq 0$ for all i and all $j < t$. Hence, by Farkas' Lemma, $P(q,r)$ is a positive linear combination of these linear polynomials. Since the inequalities $q_i \geq 0$, $1 - q_i \geq 0$, $r_i \geq 0$, $1 - r_i \geq 0$ are included among the initial inequalities, we have

$$P(q,r) = \sum_{j<t} \xi_j (F_j(q) - c_j) + \sum_{j<t} \xi'_j (G_j(r) - d_j), \tag{24}$$

for some $\xi_j, \xi'_j \geq 0$. Thus, (22) can be rewritten as follows.

$$\begin{array}{ccc} & F_t(q) + G_t(r) + E_t(a) - e_t \ \geq & \\ \sum_{ij} \alpha'_{ij} q_i (F_j(q) - c_j) & + & \sum_{ij} \alpha''_{ij} r_i (G_j(r) - d_j) + \\ \sum_{ij} \beta'_{ij} (1 - q_i)(F_j(q) - c_j) & + & \sum_{ij} \beta''_{ij} (1 - r_i)(G_j(r) - d_j) + \\ \sum_i \gamma'_i (q_i - q_i^2) & + & \sum_i \gamma''_i (r_i - r_i^2) + \\ \sum_j \delta_j (F_j(q) - c_j) & + & \sum_j \delta_j (G_j(r) - d_j) + \\ \sum_j \xi_j (F_j(q) - c_j) & + & \sum_j \xi'_j (G_j(r) - d_j) \end{array} \tag{25}$$

Now, based on the assumption that the inequalities $F_j(q) + G_j(r) + E_j(a) - e_j \geq 0$, for $j < t$, have been split into inequalities $F_j(q) - c_j \geq 0$ and $G_j(r) - d_j \geq 0$, our goal is to split the inequality $F_t(q) + G_t(r) + E_t(a) - e_t \geq 0$ into inequalities $F_t(q) - c_t \geq 0$ and $G_t(r) - d_t \geq 0$. To accomplish this goal, it is enough to find constants c_t and d_t such that

$$c_t + d_t \geq e_t - E_t(a), \tag{26}$$

and such that the following inequalities are satisfied.

$$\begin{array}{cccc} & F_t(q) - c_t \ \geq & & G_t(r) - d_t \ \geq \\ & \sum_{ij} \alpha'_{ij} q_i (F_j(q) - c_j) + & & \sum_{ij} \alpha''_{ij} r_i (G_j(r) - d_j) + \\ (a) & \sum_{ij} \beta'_{ij} (1 - q_i)(F_j(q) - c_j) + & (b) & \sum_{ij} \beta''_{ij} (1 - r_i)(G_j(r) - d_j) + \\ & \sum_i \gamma'_i (q_i - q_i^2) + & & \sum_i \gamma''_i (r_i - r_i^2) + \\ & \sum_j \delta_j (F_j(q) - c_j) + & & \sum_j \delta_j (G_j(r) - d_j) + \\ & \sum_j \xi_j (F_j(q) - c_j) & & \sum_j \xi'_j (G_j(r) - d_j). \end{array} \tag{27}$$

The meaning of these inequalities is as explained after inequality (22). The only unknown coefficients are ξ_j and ξ'_j; all other coefficients are fixed by the proof. The constant terms c_j and d_j are given from previous computations. To compute suitable c_t and d_t, it is enough to find the maximum c_t that satisfies inequality (27).(a), and the maximum d_t that satisfies inequality (27).(b). It turns out that computing c_t reduces to the problem of solving a linear program whose constraints can be extracted from inequality (27).(a). Analogously, computing d_t reduces to the problem of solving a linear program whose constraints are extracted from inequality (27).(b).

In this way we can split a proof of contradiction $-e_m \geq 0$ from $\Phi(a,q) \cup \Gamma(a,r)$ into two proofs: one is a proof of $-c_m \geq 0$ from $\Phi(a,q)$ and the other is a proof of $-d_m \geq 0$ from $\Gamma(a,r)$. Since $c_m + d_m \geq e_m - E_m(a) = e_m > 0$ we thus get a proof a contradiction from $\Phi(a,q)$ or from $\Gamma(a,r)$.

(2) Now we would like to show that not only we can split the proof into a q part and an r part, but we also can decide which of the two sets $\Phi(a,q)$ or $\Gamma(a,r)$ is contradictory *using a circuit built from* MAX *MLP gates*. As it will be argued in the final steps of the proof, this decision process will actually only depend on the computation of the quantities c_m. The fact that the original set of inequalities $\Phi(p,q) \cup \Gamma(p,r)$ is monotonically separable guarantees that we can compute the numbers $c_1, c_2, \ldots$ gradually using only MAX MLP gates.

We have sketched how to construct a linear program with the goal of computing c_t in terms of c_j (for $j < t$). However, if we only use (27).(a), the linear program may be not monotone. This is because, from the first two sums, we get terms of the form

$$q_i \sum_j (-\alpha'_{ij} + \beta'_{ij}) c_j.$$

In this sum, $-\alpha'_{ij} + \beta'_{ij}$ may be positive, negative, or zero; we do not know. Hence, in order to obtain an interpolant circuit constituted only of *monotone* gates, we will consider the process of maximizing a constant c_t satisfying the following relaxed version of inequality (27).(a).

$$\begin{aligned} F_t(q) - c_t \quad \geq \quad & \textstyle\sum_{ij} \alpha'_{ij} q_i (F_j(q) - \eta_{ij}) + \\ & \textstyle\sum_{ij} \beta'_{ij} (1-q_i)(F_j(q) - \eta'_{ij}) + \\ & \textstyle\sum_i \gamma'_i (q_i - q_i^2) + \\ & \textstyle\sum_j \delta_j (F_j(q) - c_j) + \\ & \textstyle\sum_j \xi_j (F_j(q) - c_j), \end{aligned} \tag{28}$$

where the new variables η_{ij}, η'_{ij} will be constrained by $\eta_{ij} \leq c_j$ and $\eta'_{ij} \leq c_j$ for each i and each $j < t$. We note that if $\eta_{ij} \leq c_j$ and $\eta'_{ij} \leq c_j$, then we can obtain inequalities $F_j(q) - \eta_{ij} \geq 0$ and $F_j(q) - \eta'_{ij} \geq 0$ from inequalities $F_j(q) - c_j \geq 0$ by applying the weakening rule. Additionally, in the same way that inequality (27).(a) is obtained from inequalities $F_j(q) - c_j \geq 0$ (for $j < t$) in one lap-step, we have that inequality (28) is obtained from inequalities $F_j(q) - \eta_{ij} \geq 0$, $F_j(q) - \eta'_{ij} \geq 0$ and $F_j(q) - c_j \geq 0$ (for $j < t$) in one lap-step.

We also note that the maximum value that c_t can attain under the constraints (28) is at least as large as the maximum value that c_t can attain under the constraints (27).(a), since we can always set $\eta_{ij} = \eta'_{ij} = c_j$ for each $j < t$.

Note that again the substitution of ηs has no effect on quadratic terms, so they cancel each other and we do not have to worry about them.

(3) We shall now write down the monotone linear program explicitly. For each $j \le t$, let $F_j(q) = \sum_k f_{kj} q_k$. The constraints of the program are:

$$\eta_{ij} \le c_j \qquad\qquad \eta'_{ij} \le c_j$$

$$f_{kt} \quad \ge \quad \textstyle\sum_{ij} \beta'_{ij} f_{kj} + \sum_j \delta_j f_{kj} + \gamma'_k + \tag{29}$$

$$\textstyle\sum_j -\alpha'_{kj}\eta_{kj} + \sum_j \beta'_{kj}\eta'_{kj} + \sum_j f_{kj}\xi_j.$$

The inequalities with f_{kt} express that the homogeneous part of the right-hand side of inequality (28) is less than or equal to $F_t(q)$. Under these constraints, we want to maximize the following linear function.

$$c_t = \max \sum_{ij} \beta'_{ij} c_j + \sum_j \delta_j c_j + \sum_j \xi_j c_j. \tag{30}$$

In this linear program the variables are $\eta_{kj}, \eta'_{kj}, \xi_j$ and the maximized variable is c_t.[6] The indices k run over the indices of variables q and $j = 1, \ldots, t-1$. We interpret this program as a MAX MLP gate with input variables c_j for $j < t$, and internal variables $\xi_j, \eta_{ij}, \eta'_{ij}$. Note that the program is monotone in the input variables c_j, $j < t$, and that the input variables occur both in the constraints and in the objective function.

We now construct an interpolant circuit C. For each $t \in \{1, \ldots, m\}$, if $E_t(p) + F_t(q) + G_t(r) - e_t \ge 0$ is an inequality in $\Phi(p,q)$, then we create a MAX MLP gate ℓ_t with inputs p and output c_t. For an assignment $a \in \{0,1\}^n$ of the variables p, the gate ℓ_t computes the value $e_t - E_t(a)$ as already discussed in the paragraph following inequality (18). If $E_t(p) + F_t(q) + G_t(r) - e_t \ge 0$ is an inequality in $\Gamma(p,q)$, then we set $c_t := 0$. If $E_t(p) + F_t(q) + G_t(r) - e_t \ge 0$ is obtained from previous inequalities by the application of one *lap*-step, then we create a MAX MLP gate ℓ_t with inputs p and $c_1, \ldots, c_{t-1}$ and output c_t. The value of c_t is computed according to the linear program described above.[7] One can easily check that the coefficients of the variables in this linear program can be computed in polynomial time from the given LS refutation of $\Phi(p,q) \cup \Gamma(p,r)$.

It remains to check that C interpolates $\Phi(p,q) \cup \Gamma(p,r)$. Let an assignment $a \in \{0,1\}^n$ to the variables p be given. In the process of constructing circuit C, we have also constructed a Lovász-Schrijver proof of $-c_m \ge 0$ from $\Phi(a,q)$. If $C(a) > 0$, then $c_m > 0$, since c_m is the value of the output gate. Hence we have a proof of contradiction, which means that $\Phi(a,q)$ is unsatisfiable. Otherwise, if $C(a) \le 0$, then $c_m \le 0$ and therefore $d_m > 0$, (recall that $c_m + d_m \ge e_m > 0$). Since we can also construct a proof of $-d_m \ge 0$ from $\Gamma(a,r)$, this implies that $\Gamma(a,r)$ is unsatisfiable. ◻

6.4 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 the multiplication rule for variables ranging over reals. One can easily prove that this system is complete with respect to refutations (i.e., a family of inequalities is unsatisfiable if and only if a contradiction is derivable).

---

[6] The objective function does not have the usual form, but it can be put to this form by introducing new variables x_j and adding equalities $x_j = \xi_j + \delta_j + \sum_i \beta'_{ij}$.

[7] The internal variables x_j, ξ_j, η_{ij} and η'_{ij} are distinct for each two distinct gates.

We say that an unsatisfiable set of mixed inequalities $\Phi(p, q) \cup \Gamma(p, r)$ is *strongly monotonically separable* if p and r are tuples of Boolean variables, q is a tuple of *real* variables, and variables in p occur in $\Phi(p, q)$ only with negative coefficients. Although this may seem as a very special set up, we will give later a natural example of a mixed LS refutation of such a set of inequalities.

Next, we will show that LS proofs for strongly monotonically separable unsatisfiable sets of mixed inequalities can be interpolated in terms of a *single* MAX-LEFT MLP *gate* (or, using linear-programming duality, by a *single* MIN-RIGHT MLP *gate*). The advantage of this interpolation theorem compared with Theorem 6.1 is that while proving lower bounds on the size of strong MLP circuits may be beyond the reach of current methods, proving a lower bound on the size of a single weak MPL gate seems to be feasible, because this problem is closely related to lower bounds on extended formulations (see Section 7).

Theorem 6.2. *Let $\Phi(p, q) \cup \Gamma(p, r)$ be a strongly monotonically separable unsatisfiable set of mixed inequalities, and let $p = (p_1, ..., p_n)$. 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.

Additionally, the size of the MLP gate ℓ is polynomial in the size of Π.

Proof. It is enough to construct a circuit C consisting of MAX-LEFT gates representing a function $F : \{0, 1\}^n \to \{0, 1\}$ such that for each $a \in \{0, 1\}^n$, $\Phi(a, q)$ is unsatisfiable whenever $F(a) = 1$, and $\Gamma(a, r)$ is unsatisfiable whenever $F(a) = 0$. By Theorem 4.3, from the circuit C, one can construct a *single* MAX-LEFT MLP *gate* representing F whose size is linear in the size of C.

The construction of C is done in a similar way to the construction of the circuit with MAX MLP gates constructed in Theorem 6.1. The difference is that, by assuming that the LS refutation Π is mixed, the gates used in the circuit can be restricted to MAX-LEFT MLP gates, instead of MAX MLP gates. It is enough to observe that, since the multiplication rule and integrality axioms cannot be used with respect to the real variables q, inequality (28) can be simplified to the following inequality.

$$F_t(q) - c_t \geq \sum_j \delta_j (F_j(q) - c_j) + \sum_j \xi_j (F_j(q) - c_j). \tag{31}$$

From inequality (31), one can extract the following constraints, where as in inequality (29), f_{kj} denotes the coefficient of q_k in the linear form $F_j(q)$.

$$f_{kt} = \sum_j \delta_j f_{kj} + \sum_j \xi_j f_{kj}. \tag{32}$$

Finally, the objective function given in inequality (30) is simplified to

$$c_t = \max \sum_j \delta_j c_j + \sum_j \xi_j c_j. \tag{33}$$

Equivalently, by creating a variable x_j for each $j < t$ and by setting

$$x_j = \delta_j + \xi_j, \tag{34}$$

the maximization in Equation (33) is equivalent to the following maximization.

$$c_t = \max \sum_j c_j x_j. \tag{35}$$

Together, Equation (32), Equation (34) and Equation (35) define an MLP gate with input variables c_j for $j < t$, and internal variables x_j, ξ_j. Note that the input variables c_j only appear in the objective function, and not in the constraints. Therefore, this gate is a MAX-LEFT MLP gate.

The rest of the construction of the circuit C is completely analogous to the construction in the proof of Theorem 6.1. □

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 this set of inequalities has polynomial-size mixed LS refutations, but it requires superpolynomial-size cutting-plane refutations.

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

In this subsection we will define an unsatisfiable set of inequalities $\Phi_n(p, q) \cup \Gamma_n(p, r)$, which has polynomial-size LS refutations, but which requires superpolynomial size refutations in the cutting-planes proof system. Additionally, we define a function $g_n : \{0, 1\}^n \to \{0, 1, *\}$ that has polynomial-size MLP representations, but which require superpolynomial size monotone real circuits.

We recall that the cutting-planes proof systems is defined by the following axioms and rules.

- Axioms:
  $0 \leq p_j \leq 1$.
- Rules:

(1) *Positive linear combinations;*
(2) *Rounding rule:* Suppose that all c_i are integers. Then

$$\text{from } \textstyle\sum_i c_i p_i \geq d, \text{ derive } \sum_i c_i p_i \geq \lceil d \rceil.$$

A monotone real circuit is a circuit C whose gates are monotone real functions of at most two variables. The size of C is the number of gates in C. 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-planes proofs.

THEOREM 6.3 (MONOTONE INTERPOLATION FOR THE CUTTING-PLANES PROOF SYSTEM [25]). *Let $\Phi(p, q) \cup \Gamma(p, r)$ be a monotonically separable unsatisfiable set of inequalities, and let $p = (p_1, \ldots, p_n)$. Let Π be a cutting-planes refutation for $\Phi(p, q) \cup \Gamma(p, r)$. 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, and
(2) if $C(a) = 0$ then $\Gamma(p, r)$ is unsatisfiable.

Additionally the size of the circuit C is at most a constant times the size of the refutation Π.

Let $K_n = \{\{i, j\} \mid 1 \leq i < j \leq n\}$ be the complete undirected graph with vertex set $[n] = \{1, \ldots, n\}$. 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 neighbours 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 neighbours 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 proved that any monotone Boolean circuit which decides whether a graph has a perfect matching must have size at least $n^{\Omega(\log n)}$ [29]. 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 6.4 ([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 6.5. *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

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

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 if and 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 6.6 (Unbalanced Graphs vs Perfect Matching Inequalities). *Let $\Phi_n(p,q) \cup \Gamma_n(p,r)$ be a set of inequalities on variables $p = \{w_{ij}\}$, $q = \{u_i, v_i\}$ and $r = \{x_{ij}\}$ defined as follows.*

Inequalities in $\Phi_n(p,q)$:	*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 $\Gamma_n(p,r)$:	*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 for each j, the equalities in 4. consist of two inequalities. 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 $\Phi_n(p,q)$. Therefore, $\Phi_n(p,q) \cup \Gamma_n(p,r)$ is monotonically separable.

A combination of Fu's size lower-bound for monotone real circuits (Theorem 6.4) with the monotone interpolation theorem for cutting-planes (Theorem 6.3) 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 6.6.

Theorem 6.7. *Let $\Phi_n(p,q) \cup \Gamma_n(p,r)$ be the set of inequalities of Definition 6.6. Then any cutting-planes refutation of $\Phi_n(p,q) \cup \Gamma_n(p,r)$ must have size at least $n^{\Omega(\log n)}$.*

Proof. If $a \in \{0,1\}^n$ represents a graph containing a perfect matching, then $\Gamma_n(a,r)$ is satisfiable, and consequently $\Phi_n(a,q)$ is unsatisfiable. Analogously, if a represents an unbalanced graph, then $\Phi_n(a,q)$ is satisfiable and consequently, $\Gamma_n(a,r)$ is unsatisfiable. Let Π be a refutation of $\Phi_n(p,q) \cup \Gamma_n(p,r)$. Then, by the interpolation theorem for monotone real circuits (Theorem 6.3), there is a monotone real circuit C of size polynomial in the size of Π such that $C(a) = 1$ if the graph represented by a has a perfect matching, and such that $C(a) = 0$ if the graph represented by a is an unbalanced graph. But by Corollary 6.5, any such circuit must have size at least $n^{\Omega(\log n)}$. Therefore, the proof Π must also have size at least $n^{\Omega(\log n)}$. □

On the other hand, the following theorem states that the set inequalities $\Phi_n(p,q) \cup \Gamma_n(p,r)$ has LS refutations of size polynomial in n. In fact, in these refutations, the integrality axiom and multiplication rules are never used with respect to the variables q.

Theorem 6.8. *Let $\Phi_n(p,q) \cup \Gamma_n(p,r)$ be the set of inequalities of Definition 6.6, Then $\Phi_n(p,q) \cup \Gamma_n(p,r)$ has an LS refutation of size polynomial in n.*

Proof. Consider the following polynomial-size LS refutation of $\Phi_n(p,q) \cup \Gamma_n(p,r)$.

5. $u_j - v_i - x_{ij} + 1 \geq 0$	From 3. and 1. (Definition 6.6).
6. $x_{ij}u_j - x_{ij}v_i - x_{ij}^2 + x_{ij} \geq 0$	Multiplying 5. by x_{ij}
7. $x_{ij}u_j - x_{ij}v_i \geq 0$	Applying $x_{ij}^2 - x_{ij} = 0$ to 6.

8. $\sum_{ij} x_{ij}u_j - \sum_{ij} x_{ij}v_i \geq 0$ Sum of 7. over every i, j with $i \neq j$

9. $\sum_j u_j \sum_{i;i \neq j} x_{ij} - \sum_i v_i \sum_{j;i \neq j} x_{ij} \geq 0$ Rewriting 8.

10. $\sum_j u_j - \sum_i v_j \geq 0$ From 9. and 4. (Definition 6.6).

11. $-1 \geq 0$ From 2. (Definition 6.6) and 10.

□

By combining Theorem 6.7 with Theorem 6.8 we have the following corollary separating cutting-planes from LS proof systems.

Corollary 6.9. *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.

We observe that in the LS refutation of $\Phi_n(p,q) \cup \Gamma_n(p,r)$ described in the proof of Theorem 6.8, the use of integrality axioms and multiplication rules is restricted to variables in r. Therefore, if we regard the variables in q as being real-valued variables, then $\Phi_n(p,q) \cup \Gamma_n(p,r)$ may be regarded as strongly separable unsatisfiable set of mixed inequalities. Therefore, by combining Theorem 6.8 with Theorem 6.2, we have the following theorem.

Theorem 6.10. *Let $g_n : \{0,1\}^{\binom{n}{2}} \to \{0,1,*\}$ be the partial Boolean function of Corollary 6.5. Then g_n can be represented by a* single max-left *MLP gate of size polynomial in n.*

Proof. Let $\Phi_n(p,q) \cup \Gamma_n(p,r)$ be the set of inequalities of Definition 6.6. If we regard the variables q as ranging over the reals, then $\Phi_n(p,q) \cup \Gamma_n(p,r)$ is a strongly monotonically separable set of mixed inequalities, and the refutation in Theorem 6.8 may be regarded as a mixed LS refutation of $\Phi_n(p,q) \cup \Gamma_n(p,r)$. By Theorem 6.2, there is a max-left MLP gate ℓ_n of size $n^{O(1)}$ such that for each $a \in \{0,1\}^{\binom{n}{2}}$, $\ell_n(a) > 0$ implies that $\Phi_n(p,a)$ is unsatisfiable, and $\ell_n(a) \leq 0$ implies that $\Gamma(p,a)$ is unsatisfiable. Therefore, the MLP gate ℓ_n represents the partial function g_n. □

Theorem 6.10 in conjunction with Corollary 6.5 imply that max-left MLP gates can separate graphs with a perfect matching from unbalanced graphs superpolynomially faster than monotone real circuits. Therefore, we have the following corollary.

Corollary 6.11. max-left *MLP gates cannot be polynomially simulated by monotone real circuits.*

We leave open the question of whether MLP gates (of any type) can polynomially simulate monotone real circuits.

7 MONOTONE LINEAR PROGRAMS AND EXTENDED FORMULATIONS

In this section we establish connections between monotone linear programs and the theory of extended formulations for polytopes. In particular, we define the notion of *monotone extension complexity* of a polytope and show that this complexity measure can be used to characterize the size of weak monotone representations of monotone Boolean functions. Since such representations can be used to interpolate mixed Lovász-Schrijver proofs, we may regard the task of proving superpolynomial lower bounds on the monotone extension complexity of polytopes as a first step towards proving lower bounds for the size of mixed Lovász-Schrijver proofs.

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[8], 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 $P' \subseteq \mathbb{R}^{n+m}$ 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[8]. More precisely, let $P \subseteq \mathbb{R}^n$ be a polytope, and let $P' \subseteq \mathbb{R}^{n+m}$ be a polytope defined by a system of inequalities[9] $A(v, y) \leq b$. 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 $[n] = \{1, \ldots, n\}$, requires exponentially many inequalities to be defined. Nevertheless, P_n has extended formulations of size $O(n \log n)$ [13]. 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, 34].

7.1 Existential MLP Representations

The notion of existential MLP representations defined below will be used as a bridge between weak MLP gates and extended formulations for polytopes.

Definition 7.1 (Existential MLP Representations). *Let A be a matrix in $\mathbb{R}^{m\times k}$, b be a vector in $\mathbb{R}^m$, and B be a matrix in $\mathbb{R}^{m\times n}$ with $B \geq 0$. Let $F : \{0, 1\}^n \to \{0, 1, *\}$ be a partial Boolean function. We say that the triple (A, B, b) is a* max-existential *MLP representation of F if the following conditions are satisfied for each $p \in \{0, 1\}^n$.*

$$F(p) = \begin{cases} 1 \;\Rightarrow\; \quad \exists x \geq 0,\; Ax \leq b + Bp, \\ 0 \;\Rightarrow\; \neg\exists x \geq 0,\; Ax \leq b + Bp. \end{cases} \tag{36}$$

We say that (A, B, b) is a min-existential *representation of F if the following conditions are satisfied for each $p \in \{0, 1\}^n$.*

$$F(p) = \begin{cases} 1 \;\Rightarrow\; \neg\exists x \geq 0,\; Ax \geq b + Bp, \\ 0 \;\Rightarrow\; \quad \exists x \geq 0,\; Ax \geq b + Bp. \end{cases} \tag{37}$$

As in the case of MLP gates, the size of existential representations is measured as the number of rows plus the number of columns in the matrix A. We note that there are two differences between max-existential and min-existential MLP representations. First, while the former is defined in terms of inequalities $Ax \leq b + Bp$, the latter is defined in terms of inequalities $Ax \geq b + Bp$. It is not obvious how to transform a system of inequalities in the first form into a system of inequalities in the second form because of the requirement that $B \geq 0$. Second, when considering max-existential representations, $F(p) = 1$ implies the existence of a solution x to the corresponding system of inequalities. On the other hand, when considering min-existential representations, $F(p) = 1$ implies that no solution for the corresponding system of inequalities exists. The min and max prefixes in existential MLP representations come from the following lemma.

Lemma 7.2. *Let $F : \{0, 1\}^n \to \{0, 1, *\}$ be a partial Boolean function. Then F has a* max-existential *(resp.* min-existential*) MLP representation of size $O(s)$ if and only if F can be represented by a* max-right *(resp.* min-right*) MLP gate of size $O(s)$.*

---

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

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

We leave out the proof since it uses the same ideas as similar simulation considered before.

7.2 Monotone Extension Complexity

The process of defining partial Boolean functions by linear programs is closely related, but not equivalent, to the process of defining polytopes by extended formulations. For a partial Boolean function F, let $Ones(F)$, and $Zeros(F)$ 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 $Ones(F)$ and P_F^0 denote the convex hull of $Zeros(F)$. Defining F by a linear program is equivalent to finding an extended formulation of some polyhedron Q^1 that contains P_F^1 and is disjoint 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 such a polyhedron Q^1 (resp. Q^0) with a small number of inequalities is 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 by (not necessarily monotone) linear programs of polynomial size[10]. Nevertheless, the corresponding polytope P_F^1 requires extended formulations of exponential size [34].

Let us now turn to *monotone* linear programs. From the discussion in the last paragraph, in order to have some chance of proving lower bounds for MLP representations, we need to use the fact that these representations are monotone. We define the following complexity measures for monotone functions.

Definition 7.3 (Monotone Extension Complexity). *Let $F : \{0,1\}^n \longrightarrow \{0,1,*\}$ 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

$$(P_F^1 + \mathbb{R}_+^n) \subseteq Q^1, \quad \text{and} \;\; Q^1 \cap Zeros(F) = \emptyset. \tag{38}$$

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

$$P_F^0 \subseteq Q^0, \quad \text{and} \;\; Q^0 \cap (Ones(F) + \mathbb{R}_+^n) = \emptyset. \tag{39}$$

The following theorem establishes an equivalence between the monotone extension complexities $mxc_1(F)$ and $mxc_0(F)$ of a function F and the minimum size of max-existential and min-existential representations for F respectively.

Theorem 7.4. *Let $F : \{0,1\}^n \longrightarrow \{0,1,*\}$ be a partial monotone Boolean function.*

(1) $mxc_1(F)$ is up to a constant factor equal to the minimum size of a max-existential *MLP computing F.*
(2) $mxc_0(F)$ is up to a constant factor equal to the minimum size of a min-existential *MLP computing F.*

Proof.

(1) Let (A, B, b) be a max-existential MLP representation for F. Then for each $p \in \{0,1\}^n$ such that $F(p) = 1$, there exists an $y \geq 0$ such that all inequalities in the system $Ay \leq b + Bp$ are satisfied. Additionally, if $F(p) = 0$, then no such $y \geq 0$ exists. Therefore, the system of inequalities $Ay \leq b + Bx$ is an extended formulation for a polytope Q^1 such that $(P_F^1 + \mathbb{R}_+^n) \subseteq Q^1$ and $Q^1 \cap Zeros(F) = \emptyset$.

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

For the converse, assume that the system of inequalities $A(x, y) \le b$ defines an extended formulation for a polytope Q^1 such that $(P_F^1 + \mathbb{R}_+^n) \subseteq Q^1$ and $Q^1 \cap zeros(F) = \emptyset$. Then the inequalities $A(x, y) \le b,\ x \le p$ define a MAX-EXISTENTIAL MLP representation for F.

(2) Now, let (A, B, b) be a MIN-EXISTENTIAL MLP representation for F. Then for each $p \in \{0, 1\}^n$ such that $F(p) = 0$, there exists an $y \ge 0$ such that all inequalities in the system $Ay \ge b + Bp$ are satisfied. Additionally, if $F(p) = 1$, then no such $y \ge 0$ exists. Therefore, the system of inequalities $Ay \ge b + Bx$ is an extended formulation for a polytope Q^0 such that $P_F^0 \subseteq Q^0$ and $Q^0 \cap (Ones(F) + \mathbb{R}_+^n) = \emptyset$.

For the converse, assume that the system of inequalities $A(x, y) \ge b$ defines an extended formulation for a polytope Q^0 such that $P_F^0 \subseteq Q^0$ and $Q^0 \cap (Ones(F) + \mathbb{R}_+^n) = \emptyset$. Then the inequalities $A(x, y) \ge b,\ x \ge p$ define a MIN-EXISTENTIAL MLP representation for F.

□

A possible approach for proving size lower bounds for weak MLP representations is suggested by a combination of Theorem 7.4 with Lemma 7.2. More precisely, a possible approach to prove superpolynomial lower bounds on the size of MAX-RIGHT MLPs is to come up with a hard monotone function F such that any polytope Q^1 separating $P_F^1 + \mathbb{R}_+^n$ from $zeros(F)$ is close to $P_F^1 + \mathbb{R}_+^n$. If such a function exists, one could try to apply techniques from the theory of approximate extended formulations to show that any polytope sufficiently close to $P_F^1 + \mathbb{R}_+^n$ must have superpolynomial extended formulations. Analogously, in order to prove superpolynomial lower bounds on the size of MIN-RIGHT MLPs, one could first try to come up with a function F such that that any polytope Q^0 separating P_F^0 from $ones(F) + \mathbb{R}^+$ is close to P_F^0.

We note however that lifting lower bound techniques from the theory of extended formulations to the setting of MLP representations will not be an easy task. For instance, the polytope obtained as the convex-hull of points corresponding to graphs with a perfect-matching can only be described by extended formulations of exponential size. Nevertheless, Theorem 6.10 (together with Observation 3.6) shows that MIN-RIGHT MLP gates of polynomial size can be used to separate points corresponding to perfect matchings from points corresponding to unbalanced graphs.

7.3 Some Refinements

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 $F(v') \ne 1$ for each $v' \le v$. Intuitively, a minterm is a minimal vector which causes F to evaluate to 1. We let $MinTerms(F)$ be the set of minterms of F, and $\hat{P}_F^1$ be the convex-hull of minterms of F. Analogously, a maxterm is a vector $v \in \{0, 1\}^n$ such that $F(v) = 0$ and $F(v') \ne 0$ for each $v' \ge v$. Intuitively, a maxterm is a maximal vector that causes F to evaluate to 0. We let $MaxTerms(F)$ be the set of maxterms of F.

All monotone Boolean functions for which lower bounds have been proved have the property that maxterms have essentially larger weight[11] than minterms. Additionally for these functions it is often the case that all minterms have the same weight, and therefore, lie in a hyperplane. For instance, let F be the partial monotone Boolean function where minterms are k-cliques in a graph on n vertices and maxterms are complete $(k-1)$-partite graphs. Suppose $k = n^\alpha$ for some $0 < \alpha < 1$. Then all minterms of F have weight $\binom{k}{2} \approx \frac{1}{2} n^{2\alpha}$, while maxterms have weight at least $(\frac{1}{2} - o(1))n^2$.

We note that, we can always replace P_F^1 in (38) by the convex hull $\hat{P}_F^1$ of the minterms of F. Additionally, if F is a total function, then we can replace $Zeros(F)$ by $MaxTerms(F)$.

If $\hat{P}_F^1$ lays on a hyperplane, we may reduce the task of separating $P_F + \mathbb{R}^n$ from $Zeros(F)$ to the task of separating $\hat{P}_F^1$ from some other polytopes. Let H be a hyperplane such that $\hat{P}_F^1 \subseteq H$. We project the zeros of F to H by applying the

[11] The weight of a vector $v \in \{0, 1\}^n$ is the number of times that 1 occurs in v.

following map for each v such that $F(v) = 0$:

$$v \mapsto S_v := H \cap \{u \mid u \leq v\}. \tag{40}$$

If the weights of maxterms are bigger than the weights of minterms, then each S_v is an $(n-1)$-dimensional simplex (because $\{u \mid u \leq v\}$ is a cone spaned by n lines). The task is now to separate $\hat{P}^1_F$ from $\bigcup_v S_v$ where the union is over the maxterms of F. Therefore, in this case we have the following proposition.

Proposition 7.5. *Let $F : \{0,1\}^n \to \{0,1\}$ be a total Boolean function such that the set of minterms lie on a hyperplane. Then $mxc_1(F)$ is up to a constant factor equal to the minimum size of an extended formulation of a polytope Q^1 such that*

$$\hat{P}^1_F \subseteq Q^1, \quad \text{and } Q^1 \cap \bigcup_{v \in MaxTerms(F)} S_v = \emptyset \tag{41}$$

8 CONCLUSION

In this work we have 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.

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.

The results mentioned above indicate that the study of monotone models of computation based on linear programming has the potential to shed new light on deep questions in circuit complexity and in proof complexity. We note however, that when proposing a new model of monotone computation, there is always a danger that the model is too strong. So strong that proving size lower bounds on this model for explicit Boolean functions would imply a major breakthrough in computational complexity. For instance, a *nondeterministic monotone circuit* for a Boolean function $F(p)$ is a monotone circuit $C(p,q,r)$, where q and r are strings of variables of equal length such that

$$F(p) = 1 \Leftrightarrow \exists q\, C(p,q,\neg q) = 1.$$

Note that this is a fully syntactic definition—the form of the circuit ensures that the function it computes is monotone. Yet this kind of circuits are equivalent to general nondeterministic circuits.

Nevertheless, we conjecture that the models we have introduced in this work do not suffer from this excess of computational power.

We conclude this work by stating some open problems whose solution could lead to the development of more powerful techniques for proving explicit size lower bounds for monotone models of computation and proof systems.

(1) Prove superpolynomial lower bounds for the size of weak MLP gates representing an explicit partial function F.
(2) Since proving superpolynomial lower bounds on the size of MLP circuits seems extremely difficult, the tantalizing question is: Is it possible to interpolate Lovász-Schrijver refutations by a *single* monotone LP gate (similarly as it

is in Theorem 6.2 for a set of mixed inequalities)? We believe that it should be possible to improve Theorem 6.1, because, for instance, our proof does not use the property that quadratic terms with variables p must cancel.

(3) Is it possible to bound the coefficients occurring in MLP gates without increasing too much the size of representations? More specifically, given an MLP gate ℓ of polynomial size representing a function F, can one modify it in such a way that all coefficients in the inequalities and objective function defining ℓ are integers of polynomial magnitude? Note that a similar question is open in the context of monotone span programs.

Acknowledgments. We would like to thank Anna Gál, Mika Göös, Pavel Hrubeš and Massimo Lauria for valuable suggestions. This project was supported by the ERC Advanced Grant 339691 (FEALORA). Mateus de Oliveira Oliveira also acknowledges support from the Bergen Research Foundation.

REFERENCES

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

[2] L. Babai, A. Gál, and A. Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.

[3] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák. Lower bounds on hilbert's nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 3(1):1–26, 1996.

[4] M. L. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62(3):708–728, 1997.

[5] G. Braun, S. Fiorini, S. Pokutta, and D. Steurer. Approximation limits of linear programs (beyond hierarchies). *Mathematics of Operations Research*, 40(3):756–772, 2015.

[6] M. Braverman and A. Moitra. An information complexity approach to extended formulations. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 161–170. ACM, 2013.

[7] S. R. Buss and T. Pitassi. Good degree bounds on nullstellensatz refutations of the induction principle. *Journal of computer and system sciences*, 57(2):162–171, 1998.

[8] M. de Oliveira Oliveira and P. Pudlák. Representations of monotone boolean functions by linear programs. In *Proceedings of the 32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *LIPIcs*, pages 3:1–3:14, 2017.

[9] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. D. Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *Journal of the ACM (JACM)*, 62(2):17, 2015.

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

[11] A. Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, 2001. Preliminary version in STOC 1998.

[12] A. Gál and P. Pudlák. A note on monotone complexity and the rank of matrices. *Information Processing Letters*, 87(6):321–326, 2003.

[13] M. X. Goemans. Smallest compact formulation for the permutahedron. *Mathematical Programming*, 153(1):5–11, 2015.

[14] D. Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613–622, 2001.

[15] D. Grigoriev, E. A. Hirsch, and D. V. Pasechnik. Complexity of semi-algebraic proofs. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 419–430. Springer, 2002.

[16] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

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

[18] R. Impagliazzo, P. Pudlák, and J. Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[19] M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference (San Diego, CA, 1993)*, pages 102–111. IEEE Comput. Soc. Press, Los Alamitos, CA, 1993.

[20] A. Kojevnikov and D. Itsykson. Lower bounds of static lovász-schrijver calculus proofs for tseitin tautologies. In *International Colloquium on Automata, Languages, and Programming*, pages 323–334. Springer, 2006.

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

[22] J. Krajíček. Interpolation and approximate semantic derivations. *Mathematical Logic Quarterly*, 48(4):602–606, 2002.

[23] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[24] T. Pitassi and N. Segerlind. Exponential lower bounds and integrality gaps for tree-like lovasz-schrijver procedures. *SIAM Journal on Computing*, 41(1):128–159, 2012.

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

[26] P. Pudlák. On the complexity of the propositional calculus. *London Mathematical Society Lecture Note Series*, pages 197–218, 1999.

[27] P. Pudlák and J. Sgall. Algebraic models of computation and interpolation for algebraic proof systems. In *Proceedings of Feasible Arithmetic and Proof Complexity, DIMACS Series in Discrete Math. and Theoretical Comp. Sci.*, volume 39, pages 279–295, 1998.

[28] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM (JACM)*, 39(3):736–744, 1992.

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

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

[31] A. A. Razborov. Unprovability of circuit size lower bounds in certain fragments of bounded arithmetic. *Izvestia of the RAN*, 59(1):201–224, 1995.

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

[33] R. Robere, T. Pitassi, B. Rossman, and S. A. Cook. Exponential lower bounds for monotone span programs. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 406–415. IEEE, 2016.

[34] T. 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.

[35] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

Received Month 2099; revised Month 2099; accepted Month 2099