Sub-Exponential Lower Bounds for Branch-and-Bound with General Disjunctions via Interpolation

Max Gläser Marc E. Pfetsch

September 12, 2023

Abstract

This paper investigates linear programming based branch-and-bound using general disjunctions, also known as stabbing planes, for solving integer programs. We derive the first sub-exponential lower bound (in the encoding length L of the integer program) for the size of a general branch-and-bound tree for a particular class of (compact) integer programs, namely $2^{\Omega(L^{1/12-\epsilon})}$ for every $\epsilon > 0$. This is achieved by showing that general branch-and-bound admits quasi-feasible monotone real interpolation, which allows us to utilize sub-exponential lower-bounds for monotone real circuits separating the so-called clique-coloring pair. Moreover, this also implies that refuting $\Theta(\log(n))$-CNFs requires size $2^{n^{\Omega(1)}}$ branch-and-bound trees with high probability by considering the closely related notion of infeasibility certificates introduced by Hrubeš and Pudlák [18]. One important ingredient of the proof of our interpolation result is that for every general branch-and-bound tree proving integer-freeness of a product $P \times Q$ of two polytopes P and Q, there exists a closely related branch-and-bound tree for showing integer-freeness of P or one showing integer-freeness of Q. Moreover, we prove that monotone real circuits can perform binary search efficiently.

1 Introduction

In recent years, there has been renewed interest in the proof system associated to branch-and-bound using general disjunctions for solving integer linear programs (ILPs) [4, 13, 2, 8, 12, 15]; the literature sometimes also uses the name "Stabbing Planes" (SP), see [4]. In each node, a general disjunction of the form $\alpha^\top x \leq \delta \vee \alpha^\top x \geq \delta + 1$ for $\alpha \in \mathbb{Z}^n$, $\delta \in \mathbb{Z}$ is used to create two child nodes.

Branching on general disjunctions lies at the core of Lenstra's algorithm for integer programming in fixed dimension [22]. It has also been used, for example, for special ordered sets [3], exploiting flatness [11], achieving feasibility [23], and symmetry handling [24].

Nevertheless, the dominant strategy in practice is to employ variable branching of the form $x_i \leq \delta \vee x_i \geq \delta + 1$ for some variable x_i, which is the special case where α is the ith unit vector. Some reasons for this choice are that the selection of a branching disjunction is easier, the sparsity of the constraint matrix is not increased, and it often allows to fix variables, e.g., if the variables are binary. This variable branching strategy is then usually enhanced by the application of cutting planes like Chvátal-Gomory cuts in a branch-and-cut framework, which has seen a tremendous improvement over the last decades.

As a proof system, branch-and-bound with general disjunctions is not only a generalization of branch-and-bound using variable disjunctions, but also branch-and-cut with variable disjunctions and Chvátal-Gomory cuts, see Beame et al. [4] (in fact it is even equivalent to branch-and-cut with general disjunctions and split-cuts). Hence, lower bounds on the size of a branch-and-bound tree using general disjunctions are also lower bounds on the size of a branch-and-cut tree using Chvátal-Gomory cuts. This fact shows that branch-and-bound using general disjunctions form a quite general and important algorithm class.

It is thus surprising that so far no family of integer linear programs provably requiring branch-and-bound trees using general disjunctions of super-polynomial size (in the encoding length of the program) without some kind of caveat is known. In fact, no super-linear bounds are known. In this paper we close this gap by providing a class of compact integer programs requiring branch-and-bound trees using general disjunctions of size $2^{\Omega(L^{1/12-\epsilon})}$ for every $\epsilon > 0$, where L denotes the encoding length of the ILP. This has been posed as an open problem by Dadush and Tiwari [7].

We briefly survey previous contributions: It is actually relatively easy to give families of ILPs which require branch-and-bound trees of exponential size *in the number of variables of the ILP* (but not the encoding size of the ILP). Here the two main strategies are the following: Dadush and Tiwari [7] argued that an ILP which is barely infeasible (i.e., removal of any constraint makes the ILP feasible) must require large branch-and-bound trees, since it is impossible to construct a certified branch-and-bound tree which does not use every constraint in at least one Farkas-certificate at its leaves. More accurately, the obtained bound is the number of constraints divided by the number of variables, which is only strong for a large number of constraints. This weakness is mitigated by an extended formulation of the ILP they use (with polynomial encoding size in the number of variables). However, this formulation also uses continuous variables. Their strategy was later generalized by Dey et al. [12]. The other strategy, as investigated by Gläser and Pfetsch [15], is based on finding a large set of points which have to be associated to different leaves of some given branch-and-bound trees. Formally, it considers hiding sets, which have been introduced by Kaibel and Weltge [19]. However, it seems impossible to derive bounds on the size of a branch-and-bound tree which exceed the number of the constraints of the ILP via either of these two strategies.

Despite the fact that no strong lower bounds on the size of a branch-and-bound tree have been available prior to this paper, Beame et al. [4] gave a family of unsatisfiable CNF formulas, such that refuting the corresponding ILP requires a branch-and-bound tree of *depth* $\Omega(n/\log^2 n)$. Note that since trees are not necessarily balanced, this does not yield a good lower bound on the size of a tree.

Besides lower bounds, there are some structural insights into branch-and-bound using general disjunctions: Dadush and Tiwari [7] have shown that branch-and-bound using general disjunctions does not become weaker (with respect to polynomial simulation) when restricting the coefficients of the disjunctions to have polynomial *encoding length* (cf. Theorem 1 below), which is crucial for our argument. If we restrict the coefficients of branch-and-bound using general disjunctions to polynomial *size* then branch-and-bound can be quasi-polynomially simulated by the Chvátal-Gomory cutting planes proof system (CG-CP), see Fleming et al. [13]. Thus, known lower bounds for CG-CP [25, 18, 14] can be lifted to branch-and-bound using general disjunctions with polynomially bounded coefficients.

The strategy we employ in this paper is to show that branch-and-bound using general disjunction admits *quasi-feasible real monotone interpolation* and then lift lower bounds for monotone real circuits separating the so-called clique-coloring pair given by Pudlák [25] (cf. Theorem 2) to lower bounds for branch-and-bound trees. This idea is explained in Section 2 and has already been successfully used for other proof systems. Most prominently, [25] derived sub-exponential lower bounds for CG-CP for the same ILP used below by showing that CG-CP admits real feasible monotone interpolation and a similar result for the resolution proof system. Dash showed an analogous result for the cutting plane proof system using lift-and-project cuts [9] and later for split cuts [10], which generalize both Chvátal-Gomory and lift-and-project cuts. The concept of feasible interpolation and the method in which it is used to derive lower bounds has been developed in the sequence of papers [20, 27, 21, 6, 25].

Note however, that feasible monotone interpolation can only be used to obtain lower bounds for problems in a very specific form. This limitation has recently been addressed independently in [18] and [14], where it is shown that random $\Theta(\log(n))$-CNFs are hard for CG-CP. To this end, [18] introduced the concept of infeasibility certificates which are very closely related to the notion of feasible interpolation. We mimic their approach to lift real monotone circuit lower bounds for infeasibility certificates for random CNFs to establish that $\Theta(\log(n))$-CNFs require branch-and-bound trees with general disjunctions of size at least $2^{n^{\Omega(1)}}$ with high probability as well.

The rest of this paper is structured as follows: We first survey some necessary preliminaries in Section 2. Then we explicitly state our results in Section 3 and describe in which way they are related. The proofs are then given in Section 4.

2 Preliminaries

For polyhedra $P \subseteq \mathbb{R}^{n_1}$ and $Q \subseteq \mathbb{R}^{n_2}$, let $P \times Q = \{\binom{x}{y} \in \mathbb{R}^{n_1+n_2} \mid x \in P,\ y \in Q\}$ denote their Cartesian product. For $n \in \mathbb{N}$, we use $[n] \coloneqq \{1, \dots, n\}$.

Systems of Linear Inequalities Let $Ax \leq b$ with $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$ be a system of linear inequalities. By scaling, we can assume that A and b have integral entries. If a polyhedron P is described by $Ax \leq b$ we write $\{Ax \leq b\} \coloneqq \{x \mid Ax \leq b\} = P$ for brevity.

We say $Ax \leq b$ is *integer-feasible*, if there is a point $\hat{x} \in \mathbb{Z}^n$ with $A\hat{x} \leq b$ and *integer-infeasible* otherwise. The polyhedron $\{Ax \leq b\}$ is *integer-free* if $Ax \leq b$ is integer-infeasible. Similarly, $Ax \leq b$ is *LP-feasible*, if there is a point $\hat{x} \in \mathbb{Q}^n$ with $A\hat{x} \leq b$ and *LP-infeasible* otherwise.

A *Farkas-certificate (of infeasibility)* for the system $Ax \leq b$ is a vector $f \in \mathbb{Z}^m_+$, such that $f^\top A = 0$ and $f^\top b < 0$. It is well known that a linear system $Ax \leq b$ is LP-infeasible if and only if it admits a Farkas-certificate of infeasibility. Note that the restriction to integral f is without loss of generality.

Branch-and-Bound Trees To fix notation, we formalize branch-and-bound trees. A *disjunction* is a pair of linear inequalities of the form $(\alpha^\top x \leq \delta,\ \alpha^\top x \geq \delta + 1)$, where $\alpha \in \mathbb{Z}^n$ and $\delta \in \mathbb{Z}$, which we denote $\alpha^\top x \leq \delta \vee \alpha^\top x \geq \delta + 1$. Note that every integer point satisfies exactly one of them. A *branch-and-bound proof (of integer-infeasibility)* or *branch-and-bound tree* for an system of linear inequalities (P) is a rooted binary directed tree T with the following properties:

1. For every non-leaf node N there is a disjunction $\alpha^\top x \leq \delta \vee \alpha^\top x \geq \delta + 1$, such that the *left* outgoing edge of N is labeled with the inequality $\alpha^\top x \leq \delta$ and the *right* edge is labeled with $\alpha^\top x \geq \beta + 1$. The neighbor $N_\leq$ of N incident to the left edge is called the $(\alpha^\top x \leq \delta)$-*child* of N, whereas the neighbor $N_\geq$ incident to the right edge is the $(\alpha^\top x \geq \delta + 1)$-*child*. The $(\alpha^\top x \leq \delta)$-*branch* at a node N is the directed subtree $T(N_\leq)$ rooted at $N_\leq$ and the $(\alpha^\top x \geq \delta + 1)$-*branch* is the subtree $T(N_\geq)$ rooted at $N_\geq$.
2. For a node N in T, the problem $T_N(P)$ *associated to* N *in a branch-and-bound tree* T *for* (P) is (the LP-relaxation) of (P) and all constraints occurring as edge labels on the unique path from the root to N in T. We say N is *feasible* if $T_N(P)$ is LP-feasible and *infeasible* otherwise. We require N to be infeasible for every leaf N of T.

A *branch-and-bound tree for an integer-free polyhedron* P is a branch-and-bound tree for a system $Ax \leq b$ with $P = \{Ax \leq b\}$.

We emphasize that we do not require non-leaf nodes of T to be feasible. This is convenient, since then a branch-and-bound tree T for a polyhedron P is also a branch-and-bound tree for every polyhedron $Q \subseteq P$, see, e.g., Dey et al. [12]. Moreover, this does not alter the minimal size of a branch-and-bound tree for any infeasible integer problem.

A tree labeled as described in Property 1 which does not necessarily satisfy Property 2 will be called *not necessarily valid branch-and-bound tree*. For emphasis, we sometimes call trees that satisfy both properties *valid*. Note that with above definition only integer-infeasible systems of linear inequalities have valid branch-and-bound trees.

Since we consider only binary trees, the number of nodes of a tree will be asymptotically twice the number of its leaves. Hence, we may define the *size* $|T|$ *of* T to denote the number of leaves of T, which turns out to be slightly more convenient. We let $\mathcal{T}(P)$ be the smallest size of a branch-and-bound tree using general disjunctions proving integer-freeness of P. For P containing integral points, we define $\mathcal{T}(P) \coloneqq +\infty$.

An important result about branch-and-bound trees is that they can be recompiled to reduce the encoding length of the coefficients used in the disjunctions at the nodes. This is stated in the following Theorem by Dadush and Tiwari [7]:

Theorem 1. *(Theorem 1 in [7] and its proof) Let $P \subseteq \mathbb{R}^n$ be an integer-free polytope contained in the ball $B_1^n(R) \coloneqq \{x \in \mathbb{R}^n \mid \|x\|_1 \leq R\}$ with radius $R \in \mathbb{N}$ with respect to the ℓ_1-norm. Let T be a branch-and-bound tree showing the integer-freeness of P. Then, there exists a branch-and-bound tree T' for P, such that $|T'| \leq (4n+5)|T|$, and for every disjunction $\alpha^\top x \leq \delta \vee \alpha^\top x \geq \delta+1$ which T' branches on we have $\max\{\|\alpha\|_\infty, |\delta|\} \leq (10nR)^{(n+2)^2}$.*

A *certified branch-and-bound tree (for a system of inequalities/polyhedron)* T is a branch-and-bound tree (for a system of inequalities/polyhedron), where attached to every leaf L is a Farkas-certificate f^L of infeasibility for the problem associated to L.

Monotone Real Circuits A *monotone real circuit* C is an acyclic directed graph whose vertices are called *gates*, such that every gate has either zero or two incoming edges and the incoming edges at every gate are ordered. The number of incoming edges of a gate is called its *fan-in*. If gate g has fan-in zero, it is called an *input gate* and is labeled with a variable x_i and if g has fan-in two, then g is labeled by a non-decreasing function $f_g\colon \mathbb{R}^2 \to \mathbb{R}$, which is the *function applied at* g. If $x = (x_1, \ldots, x_k)$ are the variables occurring as labels of input gates, we define (slightly abusing notation) the function $g\colon \mathbb{R}^k \to \mathbb{R}$ *computed by* g inductively (along a topological order of the underlying graph) to be $g(x) = x_j$, if g is an input gate labeled by x_j and by $f_g(g_1(x), g_2(x))$, if g is a non-input gate and g_1 and g_2 are the first and second predecessors of g. Finally, there is a designated output gate h, and the value $C(x)$ computed by C on input x is $h(x)$. The *size* $|C|$ of a circuit C is the number of its gates.

Note that bounded fan-in is essential, since monotone real circuits with unbounded fan-in of linear size can compute arbitrary monotone functions (consider the circuit with a unique non-input gate connected to all inputs). Moreover, the term 'real monotone' circuit is slightly misleading: the arithmetical structure of the real numbers $\mathbb{R}$ does not play a role in above definition – instead $\mathbb{R}$ merely plays the role of a sufficiently large linearly ordered domain (this has already been mentioned in [25]).

A circuit C *decides* (the membership problem for) a set $X \subseteq \mathbb{R}^k$, if $C(x) = 1$ for $x \in X$ and $C(x) = 0$ otherwise. Similarly, C separates two sets $Z_1 \subseteq \mathbb{R}^k$ and $Z_2 \subseteq \mathbb{R}^k$, if $C(z) = 1$ for all $z \in Z_1$ and $C(z) = 0$ for all $z \in Z_2$ or vice versa. While modifying a given circuit, *post-composing* the function f_g applied at a gate g with a function $\varphi\colon \mathbb{R} \to \mathbb{R}$ means replacing f_g by $\varphi \circ f_g$, where $\circ$ denotes the composition of functions. Similarly, *pre-composing the first [second] input* means replacing the function $f_g\colon x, y \mapsto f_g(x, y)$ by $x, y \mapsto f_g(\varphi(x), y)$ $[x, y \mapsto f_g(x, \varphi(y))]$. The notion of monotone real circuits was introduced in [25].

Interpolation We briefly translate the notion of interpolation into the language of linear integer programs. For this, we consider integer-infeasible linear inequality systems of the following form:

$$\begin{pmatrix} A \\ 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ B \end{pmatrix} y + \begin{pmatrix} C \\ D \end{pmatrix} z \leq \begin{pmatrix} a \\ b \end{pmatrix}, \qquad x,\ y,\ z \in \{0,1\}^{n_1+n_2+n_3}, \tag{1}$$

where $C \geq 0$ and $D \leq 0$ (entry-wise). Moreover, we let $A \in \mathbb{Z}^{m_1 \times n_1}$, $B \in \mathbb{Z}^{m_2 \times n_2}$, $C \in \mathbb{Z}^{m_1 \times n_3}$, $D \in \mathbb{Z}^{m_2 \times n_3}$, $a \in \mathbb{Z}^{m_1}$, $b \in \mathbb{Z}^{m_2}$ and $n := n_1 + n_2 + n_3$.

Since (1) is integer-infeasible, at least one of the systems $Ax \leq a - Cz$, $x \in \{0,1\}^{n_1}$ or $By \leq b - Dz$, $y \in \{0,1\}^{n_2}$ is infeasible for every fixed $z \in \{0,1\}^{n_3}$. An *interpolant* for (1) is a binary function $I\colon \{0,1\}^{n_3} \to \{0,1\}$, such that $I(z) = 1$ implies that the first system is infeasible and $I(z) = 0$ implies that the second is infeasible. Note that there is a interpolant for every integer-infeasible system of the form (1). We say a proof system $\mathcal{S}$ *admits (monotone/real monotone) [quasi-]feasible interpolation*, if for any proof of infeasibility of (1) in $\mathcal{S}$ of size S there exists a (monotone/real monotone) circuit of size [quasi-]polynomial in S and the encoding size of (1) computing such an interpolant. Then, if every such monotone circuit must be large, also any proof for (1) in $\mathcal{S}$ must be large. This is the case for the examples described in the next paragraph.

The Clique-Coloring Pair and the Broken-Mosquito-Screen Pair The following construction is based on the observation that no r-vertex graph G simultaneously admits both a $(k-1)$-coloring and a k-clique. This is expressed by the integer-infeasibility of:

$$x_{i\ell} + x_{j\ell} \leq 2 - z_{ij} \qquad \forall \{i,j\} \in \binom{[r]}{2},\ \forall \ell \in [k-1], \tag{2a}$$

$$\sum_{\ell \in [k-1]} x_{i\ell} \geq 1 \qquad \forall i \in [r], \tag{2b}$$

$$y_i + y_j \leq 1 + z_{ij} \qquad \forall \{i,j\} \in \binom{[r]}{2}, \tag{2c}$$

$$\sum_{i \in [r]} y_i \geq k, \tag{2d}$$

$$x \in [0,1]^{r \times (k-1)},\ y \in [0,1]^r,\ z \in [0,1]^{\binom{[r]}{2}}, \tag{2e}$$

where we interpret $z \in \{0,1\}^{\binom{[r]}{2}}$ as an encoding of the graph $G = ([r], E)$, with $z_{ij} = 1$ if and only if $\{i,j\} \in E$, $x \in \{0,1\}^{r \times (k-1)}$ as a $(k-1)$-coloring of G, with $x_{i\ell} = 1$ if and only if vertex i gets assigned color ℓ, and $y \in \{0,1\}^r$ as an k-clique of G, with $y_i = 1$ if and only the clique contains vertex i.

If we write (2) in the form of (1), the pair of sets

$$Z_1 := \{z \in \{0,1\}^{\binom{[r]}{2}} \mid \exists x \in \{0,1\}^{r \times (k-1)} \colon Ax \leq a - Cz\} \text{ and}$$
$$Z_2 := \{z \in \{0,1\}^{\binom{[r]}{2}} \mid \exists y \in \{0,1\}^{r} \colon By \leq b - Dz\}$$

is known as the *clique-coloring pair* or *CC-pair*. Note that $n_1 = r(k-1)$, $n_2 = r$, and $n_3 = \binom{r}{2} = (r^2 - r)/2$ for later calculations. As already remarked, we have $Z_1 \cap Z_2 = \emptyset$.

Pudlák [25] gave the following lower-bound for any monotone real circuit separating the CC-pair:

Theorem 2. *Every family of monotone real circuits separating the CC-pair with r vertices and $k := \lfloor \frac{1}{8}(r/\log r)^{2/3} \rfloor$ has size $2^{\Omega((r/\log r)^{1/3})}$.*

Note that log always refers to the logarithm with respect to base 2 in this paper.

Since $n_3 = (r^2 - r)/2$ is the number of inputs to a circuit separating the CC-pair, we note

$$2^{\Omega((r/\log r)^{1/3})} = 2^{\Omega((n_3/\log n_3)^{1/6})} \in 2^{\Omega(n_3^{1/6-\epsilon})} \text{ for every } \epsilon > 0.$$

Theorem 2 is in contrast to the fact that the CC-Pair can be separated in polynomial time by semi-definite programming using Lovász' Theta body (cf., e.g., Remark 9.3.20(b) in [16]).

Another example of a disjoint pair of languages which requires a large monotone real circuit to be separated is the *broken-mosquito-screen pair* or *BMS-pair*. The BMS-pair is polynomially equivalent to the CC-pair [26] and is therefore also polynomial time separable. For the BMS-pair, a slightly more explicit bound is available in the literature: Cook and Haken [17] show that any circuit separating the BMS-pair has at least $2^{Kn_3^{1/8}}$ gates for some $K > 0.32$.

Random CNFs and Infeasibility Certificates In this section we will consider Boolean variables $x_1, \dots, x_n$. A *literal* is a variable x_i or its negation $\neg x_i$. A *clause* is a set of literals. A *CNF* (or formula in *conjunctive normal form*) is a set of clauses. A *k-clause* is a clause with k variables and a *k-CNF* is a CNF containing only k-clauses. The *satisfiability problem* is the problem of deciding whether there exists a satisfying assignment to a given CNF $\mathcal{C}$, i.e., an assignment α of binary values (or truth values) to $x_1, \dots, x_n$, such that every clause $C \in \mathcal{C}$ contains either a literal x_i for a variable which is assigned to be true (i.e., $\alpha(x_i) = 1$) or a literal $\neg x_i$ for a variable which is assigned to be false (i.e., $\alpha(x_i) = 0$).

The satisfiability problem can be cast as an ILP in a straight-forward way:

$$\sum_{x_i \in C} x_i + \sum_{\neg x_i \in C} (1 - x_i) \geq 1 \quad \forall C \in \mathcal{C}, \qquad x_1, \dots, x_n \in \{0, 1\}. \tag{3}$$

Thus we can speak about branch-and-bound trees refuting unsatisfiable CNFs.

A random k-CNF $\mathcal{C}$ in n variables and m clauses is obtained by picking k-clauses uniformly and independently at random from the $\binom{n}{k}2^k$ possible k-clauses. Since we allow repetition, $\mathcal{C}$ may contain less than m clauses. We are interested in choosing the parameters m and k in a way, such that the resulting CNFs are unsatisfiable, but hard to refute with high probability. Moreover, we are interested in choosing these parameters as small as possible. For this we follow the choices made in [18], where this is discussed in more detail. First observe that any random assignment to the variables satisfies $\mathcal{C}$ with probability $(1 - 2^{-k})^m$. Hence, by the union bound there is a satisfying assignment with probability at most $(1 - 2^{-k})^m 2^n \leq e^{-2^{-k}m} 2^n$. Hence, if we choose $m \geq (\ln 2 + \varepsilon) 2^k n$ for some $\varepsilon > 0$, then a random-formula is unsatisfiable with high probability, where ln refers to the natural logarithm. Note that if $k \in O(\log(n))$, then the number of clauses is polynomial in n.

Let $\mathcal{C}$ denote a CNF and $X_0 \cup X_1 = X \coloneqq \{x_1, \dots, x_n\}$ denote a partition of its variables. Every clause C_i in $\mathcal{C}$ can be written as as $C_i = C_i^0 \cup C_i^1$, where C_i^j contains only variables from X_j and their negations. A monotone Boolean function $F : \{0, 1\}^m \to \{0, 1\}$ is an (X_0, X_1)-*certificate (of infeasibility)* for $\mathcal{C}$, if for every $A \subseteq [m]$ we have

$$\begin{aligned} F(A) = 0 &\implies \{C_i^1 : i \in [m] \setminus A\} \text{ is unsatisfiable,} \\ F(A) = 1 &\implies \{C_i^0 : i \in A\} \text{ is unsatisfiable,} \end{aligned}$$

where we identify a set of (indices of) clauses with its characteristic vector. That is, given a subset A of the clauses of $\mathcal{C}$ as input, F determines one of the CNFs $\{C_i^1 : i \in [m] \setminus A\}$ or $\{C_i^0 : i \in A\}$ which is unsatisfiable. It is easy to see that, for any choice of X_0 and X_1, $\mathcal{C}$ admits an (X_0, X_1)-certificate if and only if it is unsatisfiable (Proposition 5 in [18]).

Similarly to interpolants, infeasibility certificates have high monotone real circuit complexity for some problems, among them the interesting case of random CNFs. We say that a sequence of events $(E(n))_{n \in \mathbb{N}}$ *holds with high probability* if $\lim_{n \to \infty} \Pr[E(n)] = 1$. We then have:

Theorem 3 (Theorem 2 in [18])**.** *Let $c > 1$ be a constant and let $n \geq 1$ be given. Let $X_0 \cup X_1$ be a partition of $2n$ variables into two sets of equal size. If $\mathcal{C}$ is a random k-CNF with $O(n2^k)$ clauses, variables $X_0 \cup X_1$, and $k \geq c \log(n)$, then every (X_0, X_1)-certificate for $\mathcal{C}$ requires monotone real circuits of size $2^{n^{\Omega(1)}}$ with high probability.*

Note that the conditions in Theorem 3 do not ensure that $\mathcal{C}$ is unsatisfiable with high probability. Hence, for small values of c, the theorem states that with high probability either every (X_0, X_1)-certificate requires large real monotone circuits or $\mathcal{C}$ is satisfiable (and thus there are no (X_0, X_1)-certificates at all). However, for $c > \ln 2$ almost all random CNFs as in the statement of the theorem are unsatisfiable, cf. the discussion above. A similar remark is true about most results about random CNFs in this paper.

The connection between infeasibility certificates and interpolation is explained via the observation that an infeasibility certificate for a CNF $\mathcal{C} = \{C_1, \ldots, C_m\}$ can be seen as an interpolant to a closely related CNF: Let $X_0 \cup X_1$ be a partition of the variables in $\mathcal{C}$. We introduce additional variables $Y = \{y_1, \ldots, y_m\}$, one for every clause, and consider the CNF $\mathcal{D}$ with clauses

$$C_1^0 \cup \{\neg y_1\}, \; \ldots, \; C_m^0 \cup \{\neg y_m\}, \; C_1^1 \cup \{y_1\}, \; \ldots, \; C_m^0 \cup \{y_m\}.$$

Let $\mathcal{D}_0$ denote the CNF containing the first m clauses (with variables $X_0 \cup Y$) and $\mathcal{D}_1$ the CNF containing the second m clauses (with variables $X_1 \cup Y$). Let Y_i (where $i \in \{0,1\}$) denote the collection of assignments α to the variables in Y which make $\mathcal{D}_i$ satisfiable, if we fix a variable $y \in Y$ to $\alpha(y)$. It is then easy to see that any interpolant separating Y_0 and Y_1 is an (X_0, X_1)-certificate for $\mathcal{C}$ and vice versa. In particular, this observation implies that interpolation theorems also convert short proofs into infeasibility certificates with low monotone circuit complexity, which can be used to lift the lower bound given by Theorem 3 to a lower bound on the size of proofs, provided $\mathcal{D}$ is not significantly harder to refute than $\mathcal{C}$ (which typically does not seem to be the case).

3 Results

The central result of this work is that branch-and-bound using general disjunctions admits quasi-feasible monotone real interpolation, that is:

Theorem 4. *Given a branch-and-bound tree T for (1), there exists a monotone real circuit of size $50(n+1)^2|T|^2 \cdot [(n+2)^2 \log(10n^3+3)]^{\log((4n+5)|T|)}$ with input z, which separates the sets $Z_1 := \{z \in \{0,1\}^{n_3} \mid \exists x \in \{0,1\}^{n_1} \colon Ax \leq a - Cz\}$ and $Z_2 := \{z \in \{0,1\}^{n_3} \mid \exists y \in \{0,1\}^{n_2} \colon By \leq b - Dz\}$.*

Choosing (1) to be integer linear programs expressing a separation problem for which we have lower bounds for separating monotone real circuits, we obtain lower bounds for branch-and-bound trees for (1). For example, combining Theorems 4 and 2, we immediately obtain the following sub-exponential bound:

Theorem 5. *Every family of branch-and-bound trees for (2), where $k := \lfloor \frac{1}{8}(r/\log r)^{2/3} \rfloor$, has size at least $2^{\Omega(n^{1/6-\epsilon})}$, for every $\epsilon > 0$, where n is the number of variables.*

We note that one could make this bound completely explicit, i.e., for fixed $\epsilon > 0$, we can give $N \in \mathbb{N}$ and $\delta > 0$, such that any branch-and-bound tree for (2) has size at least $2^{\delta n^{1/6-\epsilon}}$ for every $n \geq N$, by tracking the factor hidden in the Ω-notation in Theorem 2 through its proof (see [25] building on [1] as presented in [28]). Alternatively, we can give a similar bound for the BMS-pair, which can then be made explicit by some elementary calculations.

Since the encoding length L of (2) satisfies $L \in \Theta(n^2)$, we immediately obtain:

Corollary 6. *Every family of branch-and-bound trees for (2), where $k := \lfloor \frac{1}{8}(r/\log r)^{2/3} \rfloor$, has size at least $2^{\Omega(L^{1/12-\epsilon})}$, for every $\epsilon > 0$, where L is the encoding length (2).*

We also obtain a similar result for random CNFs:

Theorem 7. *If $\mathcal{C}$ is a random k-CNF with $2n$ variables and $O(n\,2^k)$-clauses, where $k \geq c \log n$ for a constant $c > 1$, then any branch-and-bound tree for (3) for $\mathcal{C}$ has size at least $2^{n^{\Omega(1)}}$ with high probability.*

The remainder of this section outlines how we prove Theorem 4. For this we will require three ingredients.

Let $P \in \mathbb{R}^{n_1}$ and $Q \in \mathbb{R}^{n_2}$ be two polytopes. Our first ingredient is the fact that given a certified branch-and-bound tree T showing the integer-freeness of $P \times Q$, there is a branch-and-bound tree for the integer-freeness of P, which is structurally very close to T, or there is such a tree showing the integer-freeness of Q. This is helpful for showing Theorem 4, since after fixing the variables z, the feasible region of the LP-relaxation of (1) is a product of two lower-dimensional polytopes:

$$\{x \in [0,1]^{n_1} \mid Ax \leq a - Cz\} \times \{y \in [0,1]^{n_2} \mid By \leq b - Dz\}.$$

Given a branch-and-bound tree T for showing integer-freeness of $P \times Q \subseteq \mathbb{R}^{n_1 \times n_2}$, a branch-and-bound tree T' for P *conforms to* T, if their underlying directed graphs (including the ordering of the children as $\leq$- and $\geq$-children) are identical and, if the disjunction used by T at a node N is $\alpha^\top x + \beta^\top y \leq \delta \vee \alpha^\top x + \beta^\top y \geq \delta + 1$, then the disjunction used by T' at N is $\alpha^\top x \leq \delta' \vee \alpha^\top x \geq \delta' + 1$ for some $\delta' \in \mathbb{Z}$.

A precursor to our first ingredient will be the following Lemma.

Lemma 8. *(Structural Interpolation Lemma) For every branch-and-bound tree T for an integer-free product of polytopes $P \times Q$ there exits*

(a) a branch-and-bound tree T^P for P conforming to T or

(b) a branch-and-bound tree T^Q for Q conforming to T.

By applying Lemma 8 to a smallest branch-and-bound tree for $P \times Q$, we immediately obtain the following result, which is of independent interest.

Corollary 9. $\mathcal{T}(P \times Q) = \min(\mathcal{T}(P), \mathcal{T}(Q))$

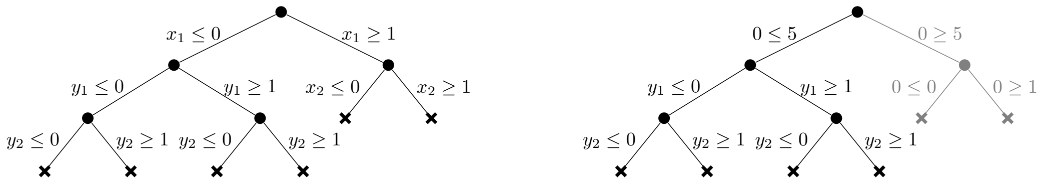
Note that Lemma 8 does not give any information of what the right hand side of the disjunctions used at the nodes of T^P or T^Q should be, while almost every other property of the tree is preserved. The remaining two ingredients for the proof of Theorem 4 are intended to supply this information. To this end, it would be helpful, if Farkas-certificates maintain their validity when passing from T to T^P or T^Q. This is due to the fact that it is not clear how we can decide the LP-feasibility of a system of linear inequalities (even without integrality constraints), since we need to perform computations via monotone real circuits. However, it is easy to check whether a given Farkas-certificate is valid.

Unfortunately, Lemma 8 does not seem to hold for *certified* branch-and-bound trees. Indeed, the naive way to obtain Farkas-certificates for T^P (or T^Q) from Farkas-certificates for T does not work, i.e., for a leaf L in T^P using the projection of the Farkas-certificate f^L attached to L in T onto constraints of P and branching constraints. As a counter example, consider the square of the two-dimensional cross-polytope

$$\begin{aligned} C_x^2 \times C_y^2 \coloneqq \quad & \{x_1, x_2 \in [0,1] \colon \sum_{i \in S} x_i + \sum_{i \notin S} (1 - x_i) \leq \tfrac{3}{2} \quad \forall S \subseteq \{1,2\}\} \\ & \times \{y_1, y_2 \in [0,1] \colon \sum_{i \in S} y_i + \sum_{i \notin S} (1 - y_i) \leq \tfrac{3}{2} \quad \forall S \subseteq \{1,2\}\} \end{aligned}$$

and the branch-and-bound tree shown in Figure 1a for $C_x^2 \times C_y^2$. Here, let all Farkas-certificates at the leaves have the form

$$\begin{array}{llr} & 1 \cdot & (\sum_{i \in S} x_i + \sum_{i \notin S} (1 - x_i) \leq \frac{3}{2}) \\ + \sum_{i \in S} & 1 \cdot & (-x_i \leq -1) \\ + \sum_{i \notin S} & 1 \cdot & (x_i \leq 0) \\ \hline & & (0 \leq -0.5) \end{array}$$



(a) A branch-and-bound tree for $C_x^2 \times C_y^2$.

(b) A conforming tree using quasi-Farkas-certificates.

Figure 1: A branch-and-bound tree for $C_x^2 \times C_y^2$ showing that Lemma 8 does not hold for certified branch-and-bound trees. (b) shows a tree for C_y^2 conforming to the one in (a) using quasi-Farkas-certificates. For the gray part of the tree, the second case from the definition of quasi-Farkas-certificates holds, i.e., we do not require the validity of Farkas-certificates there.

for some $S \subseteq \{1, 2\}$ or the analogous form for variables from C_y^2. For any leaf, there is only one such choice.

It is not hard to see that no choice of new right-hand-sides $\hat{\delta}$ can make every Farkas-certificate $\hat{f}^L$ obtained as described above valid for all leaves simultaneously, when we try to obtain a conforming branch-and-bound tree for either C_x^2 or C_y^2: For C_x^2, $\hat{\delta}_r \leq -1$ has to hold for the right-hand-side $\hat{\delta}_r$ used in the disjunction at the root r, if all Farkas-certificates in the $\leq$-branch at the root are to be valid. But then it is impossible for both Farkas-certificates at leaves in the $\geq$-branch to be valid simultaneously. The situation for C_y^2 is similar.

However, if we relax the notion of a Farkas-certificate very slightly, then Lemma 8 holds also for certified trees: Let P be a polytope and U an arbitrary set with $P \subseteq U$. A *quasi-certified branch-and-bound tree* T for P relative to U is a branch-and-bound tree for P, such that to every leaf L there is an attached *quasi-Farkas-certificate* f^L *relative to* U, i.e., a vector $f^L \in \mathbb{Z}_+^{m_L}$ indexed by the m_L constraints of the problem $T_L(P)$ associated to L, such that

1. f^L is a valid Farkas-certificate for $T_L(P)$ or
2. an edge on the unique root-leaf path in T to L is labeled with a constraint $\alpha^\top x \leq \gamma$ ($\alpha^\top x \geq \gamma + 1$), such that $U \cap \{\alpha^\top x \leq \gamma\} = \emptyset$ ($U \cap \{\alpha^\top x \geq \gamma + 1\} = \emptyset$).

Note that f^L does not appear in the second condition. A quasi-certified branch-and-bound tree T' for P *conforms* to a quasi-certified branch-and-bound tree T for $P \times Q$, if it does so as (uncertified) branch-and-bound tree and moreover the quasi-Farkas-certificate $(f')^L$ attached to a leaf L of T' is the projection of the quasi-Farkas-certificate f^L attached to the leaf L in T onto variables corresponding to constraints of P and branching constraints.

In the above example, there exists a quasi-certified branch-and-bound relative to $[0, 1]^2$ conforming to the previously problematic tree, which is shown in Figure (1b). More generally, we have:

Lemma 10. *(Certified Structural Interpolation Lemma) For every quasi-certified branch-and-bound tree T for an integer-free product of polytopes $P \times Q$ relative to $U = U_P \times U_Q$, where $P \subseteq U_P$, $Q \subseteq U_Q$ and U_P and U_Q are bounded, there exists*

(a) a quasi-certified branch-and-bound tree T^P for P relative to U_P conforming to T or

(b) a quasi-certified branch-and-bound tree T^Q for Q relative to U_Q conforming to T.

Passing from certified trees to quasi-certified trees allows us to choose not to branch at a node N of T during construction of T^P or T^Q and instead immediately proceed as in one of the subtrees rooted at the children of N. Conceptually, it might seem cleaner to consider certified branch-and-bound trees which embed into T for some suitable notion of embedding. However, this does not appear to work well with certificates. Moreover, we do not want to be forced to enumerate over all possible embedded trees later.

As previously mentioned, the last two ingredients for the proof of Theorem 4 instruct us how to reconstruct the right-hand-sides of the disjunctions in the tree T^P or T^Q given by Lemma 10. The first ingredient is Theorem 1, which establishes that the set of possible right-hand-sides is not too large (if we recompile T beforehand).

It then remains to show that if a tree T^P as in Lemma 10 exits, we can search the space of possible right-hand-side – whose size is limited by Theorem 1 – efficiently via binary search, even in the quite restricted computational model of monotone real circuits.

More concretely, assume we have chosen values for the right-hand-sides of all disjunctions used at a descendant of a node N in T and are now choosing a value γ_N for the disjunction used at N. If we consider (say) the $(\alpha_N^\top x \geq \gamma_N + 1)$-branch at N, then we want to choose γ_N as small as possible (i.e., we want to choose the inequality $\alpha_N^\top x \geq \gamma_N + 1$ as weak as possible), such that we still obtain the validity of all quasi-Farkas-certificates in the $(\alpha_N^\top x \geq \gamma_N + 1)$-branch (depending on the choices for the right-hand-sides for all disjunctions used at ancestors of N). Assume that we have already expressed the simultaneous validity of the quasi-Farkas-certificates in the $(\alpha_N^\top x \geq \gamma_N + 1)$-branch via a monotone real circuit computing values in $\{0, 1\}$. The following result states we can efficiently compute the smallest possible value of γ_N making these certificates valid via a monotone real circuit:

Lemma 11. *(Oblivious Binary Search Lemma) For a monotone real circuit C computing values in $\{0, 1\}$ with k inputs, $\Lambda_{\max} \in \mathbb{R}$ such that $C(x_1, \ldots, x_{k-1}, \Lambda_{\max}) = 1$ for all $x_1, \ldots, x_{k-1} \in \mathbb{R}$, and any $q \in \mathbb{N}$, there exists a monotone real circuit $\tilde{C}$ of size $|C| \cdot q$ which computes*

$$b\colon (x_1, \ldots, x_{k-1}) \mapsto \max \left\{\lambda \in \{0, \ldots, 2^q - 1\} \mid C(x_1, \ldots, x_{k-1}, \Lambda_{\max} - \lambda) = 1\right\}.$$

Note that $\tilde{C}$ uses q invocations of C to find the maximal λ among the 2^q candidates in $\{0, \ldots, 2^q - 1\}$ causing C to accept on input $(x_1, \ldots, x_{k-1}, \Lambda_{\max} - \lambda)$, which is the best we can reasonably expect while treating C as a black box. In fact, the concrete form of the set of candidate values is irrelevant – we can search over any set of this cardinality with a circuit of the same size. This reflects the fact that the definition of monotone circuits does not make use of the arithmetical structure of the real numbers. Finally, we note that the assumption $C(x_1, \ldots, x_{k-1}, \Lambda_{\max}) = 1$ for all $x_1, \ldots, x_{k-1} \in \mathbb{R}$ is purely for convenience and could be replaced by one more invocation of C and a slightly less elegant definition of b.

A key observation is that we are able to choose the right-hand-side γ_N of the disjunction at a node N by querying the circuit corresponding to the smaller child subtree. We then have to invoke the circuit corresponding to the larger subtree only once, to ensure that this choice of γ_N works for both subtrees. This will ensure the efficiency of our construction. Formally, we have:

Corollary 12. *For κ, $\Lambda_{\min}$, $\Lambda_{\max} \in \mathbb{Z}$ and monotone real circuit C_1 and C_2 computing values in $\{0, 1\}$ with k inputs each, such that $C_1(x_1, \ldots, x_{k-1}, \Lambda_{\max}) = 1$ for all $x_1, \ldots, x_{k-1} \in \mathbb{R}$ and $C_2(x_1, \ldots, x_{k-1}, \kappa - \Lambda_{\min}) = 1$ for all $x_1, \ldots, x_{k-1} \in \mathbb{R}$, there exists a monotone real circuit $\tilde{C}$ with $k - 1$ inputs which decides whether there exist integral values x_k, $x'_k \in \mathbb{Z}$ with $x_k + x'_k = \kappa$, such that $C_1(x_1, \ldots, x_k) = C_2(x_1, \ldots, x_{k-1}, x'_k) = 1$ of size $|C_1| \cdot \lceil \log(L+1) \rceil + |C_2|$, where $L \coloneqq \Lambda_{\max} - \Lambda_{\min}$.*

Note that above lemma and its corollary are the reason we consider monotone real circuit complexity. We do not see a way in which binary monotone circuits can achieve a similar result (when the inputs are provided in binary encoding).

Finally, for the proof of Theorem 4, we will construct a monotone real circuit which on input $z \in Z_1 \cup Z_2$ decides whether after fixing the variables z in (1), there exists a branch-and-bound tree $T^{P(z)}$ for $P(z) = \{x \in \{0, 1\}^{n_1} \mid Ax \leq a - Cz\}$ as in Lemma 10 by efficiently searching the space of possible right-hand-sides using Corollary 12. Clearly, if such right-hand-sides exist, we have $z \in Z_2$ and $z \in Z_1$ otherwise.

4 Proofs

4.1 Proof of Lemmas 8 and 10

To showcase our technique, we begin by showing Lemma 8:

Proof of Lemma 8. Let $P \coloneqq \{x \mid Ax \leq a\}$ and $Q \coloneqq \{y \mid By \leq b\}$. We will proceed by induction on $|T|$. In case $|T| = 1$, we have $P \times Q = \emptyset$, i.e., $P = \emptyset$ or $Q = \emptyset$, say $P = \emptyset$. Then there exists a branch-and-bound tree T' for P with $|T'| = 1$ which conforms to T, since their common underlying directed graph does not contain any internal nodes.

Assume $|T| > 1$ and that $\alpha^\top x + \beta^\top y \leq \delta \vee \alpha^\top x + \beta^\top y \geq \delta + 1$ is the topmost disjunction of T. Let $N_\leq$ denote the $\leq$-child of the root of T and $T(N_\leq)$ the subtree of T rooted at $N_\leq$. For any $\gamma \in \mathbb{Z}$ the subtree $T(N_\leq)$ is a valid branch-and-bound tree for

$$(P \times Q)_{\leq\gamma} \coloneqq (P \cap \{\alpha^\top x \leq \delta + \gamma\}) \times (Q \cap \{\beta^\top y \leq -\gamma\}),$$

since the two added constraints imply $\alpha^\top x + \beta^\top y \leq \delta$.

The induction hypothesis then implies that either (a) or (b) holds for $T(N_\leq)$ and $(P \times Q)_{\leq\gamma}$. This allows us to define $(\mu_\leq(\gamma))_{\gamma\in\mathbb{Z}}$ by

$$\mu_\leq(\gamma) \coloneqq \begin{cases} -1, & \text{if case (a) holds for } T(N_\leq) \text{ and } (P \times Q)_{\leq\gamma}, \text{ but case (b) does not,} \\ 0, & \text{if cases (a) and (b) both hold for } T(N_\leq) \text{ and } (P \times Q)_{\leq\gamma}, \\ 1, & \text{if case (b) holds for } T(N_\leq) \text{ and } (P \times Q)_{\leq\gamma}, \text{ but case (a) does not.} \end{cases}$$

Indeed, by the induction hypothesis, we have defined $\mu_\leq$ for all possible cases. It is easy to see that $\mu_\leq$ is non-decreasing. Moreover, $\mu_\leq$ is neither identically -1 nor identically 1, since sufficiently extreme values of γ can render both $(P \cap \{\alpha^\top x \leq \delta + \gamma\})$ and $(Q \cap \{\beta^\top y \leq -\gamma\})$ empty (and hence any branch-and-bound tree is valid for them).

Similarly, we let $N_\geq$ denote the $\geq$-child of the root of T and $T(N_\geq)$ the subtree of T rooted at $N_\geq$. We define $(P \times Q)_{\geq\gamma} \coloneqq (P \cap \{\alpha^\top x \geq \delta + \gamma + 1\}) \times (Q \cap \{\beta^\top y \geq -\gamma\})$ and then $(\mu_\geq(\gamma))_{\gamma\in\mathbb{Z}}$ by

$$\mu_\geq(\gamma) \coloneqq \begin{cases} -1, & \text{if case (a) holds for } T(N_\geq) \text{ and } (P \times Q)_{\geq\gamma}, \text{ but case (b) does not,} \\ 0, & \text{if cases (a) and (b) both hold for } T(N_\geq) \text{ and } (P \times Q)_{\geq\gamma}, \\ 1, & \text{if case (b) holds for } T(N_\geq) \text{ and } (P \times Q)_{\geq\gamma}, \text{ but case (a) does not.} \end{cases}$$

Note that $\mu_\geq(\gamma)$ is non-increasing and neither identically -1 nor identically 1. We remark that this definition is not symmetric in P and Q (the '+1' goes with P).

The noted properties of $\mu_\leq$ and $\mu_\geq$ imply that at least one of the following cases hold: Either (i) there is $\gamma \in \mathbb{Z}$, such that $\mu_\leq(\gamma) \leq 0$ and $\mu_\geq(\gamma) \leq 0$, or (ii) there is $\gamma \in \mathbb{Z}$, such that $\mu_\leq(\gamma) = \mu_\geq(\gamma - 1) = 1$.

If there exists γ as in (i), we construct T^P as desired by branching on $\alpha^\top x \leq \delta + \gamma \vee \alpha^\top x \geq \delta + \gamma + 1$ and attaching to the resulting children the trees for $(P \cap \{\alpha^\top x \leq \delta + \gamma\})$ and $(P \cap \{\alpha^\top x \geq \delta + \gamma + 1\})$ conforming to $T(N_\leq)$ and $T(N_\geq)$ for which existence is guaranteed by $\mu_\leq(\gamma) \leq 0$ and $\mu_\geq(\gamma) \leq 0$.

Otherwise, there exists γ as in (ii) and we construct T^Q as desired by branching on $\beta^\top y \leq -\gamma \vee \beta^\top y \geq -\gamma + 1$ and attaching to the resulting children the trees for $(Q \cap \{\beta^\top y \leq -\gamma\})$ and $(Q \cap \{\beta^\top y \geq -\gamma + 1\})$ conforming to $T(N_\leq)$ and $T(N_\geq)$ for which existence is guaranteed by $\mu_\leq(\gamma) = \mu_\geq(\gamma - 1) = 1$. □

The proof of Lemma 10 follows the same arguments as Lemma 8, but we now need to track that the condition on the quasi-Farkas certificates holds as well.

Proof of Lemma 10. Let $P := \{x \mid Ax \le a\}$ and $Q := \{y \mid By \le b\}$, i.e.,

$$P \times Q = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \;\middle|\; \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \le \begin{pmatrix} a \\ b \end{pmatrix} \right\}.$$

We will proceed by induction on $|T|$. If we have $|T| = 1$, then the root r of T is the unique node of T and T does not branch on any disjunctions. Hence, there is a valid Farkas-certificate f^r attached to r. Let f^r_P denote the projection of f^r onto constraints belonging to P and f^r_Q the projection of f^r onto constraints belonging to Q. We then have

$$\begin{pmatrix} f^r_P \\ f^r_Q \end{pmatrix}^\top \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} = 0, \quad \begin{pmatrix} f^r_P \\ f^r_Q \end{pmatrix}^\top \begin{pmatrix} a \\ b \end{pmatrix} = (f^r_P)^\top a + (f^r_Q)^\top b < 0.$$

It follows that $(f^r_P)^\top A = 0$, $(f^r_Q)^\top B = 0$, and that $(f^r_P)^\top a < 0$ or $(f^r_Q)^\top b < 0$. Hence, f^r_P is a Farkas-certificate for the infeasibility of P or f^r_Q is a Farkas-certificate for the infeasibility of Q. Thus, the branch-and-bound tree with a single leaf and certificate f^r_P for this leaf is a quasi-certified branch-and-bound tree for P relative to U_P conforming to T or the branch-and-bound tree with a single leaf and certificate f^r_Q is a quasi-certified branch-and-bound tree for Q relative for U_Q conforming to T.

Assume $|T| > 1$ and that $\alpha^\top x + \beta^\top y \le \delta \vee \alpha^\top x + \beta^\top y \ge \delta + 1$ is the topmost disjunction of T. Let $N_\le$ denote the $\le$-child of the root of T and $T(N_\le)$ the subtree of T rooted at $N_\le$. For any $\gamma \in \mathbb{Z}$ the subtree $T(N_\le)$ is a valid branch-and-bound tree for

$$(P \times Q)_{\le\gamma} := (P \cap \{\alpha^\top x \le \delta + \gamma\}) \times (Q \cap \{\beta^\top y \le -\gamma\}),$$

since the two added constraints imply $\alpha^\top x + \beta^\top y \le \delta$. In order to use the induction hypothesis, we must turn $T(N_\le)$ into a quasi-certified branch-and-bound tree for $(P \times Q)_{\le\gamma}$ relative to U. For every leaf L in T which is a descendant of $N_\le$, there is a quasi-Farkas-certificate f^L for the associated subproblem $T_L(P \times Q)$. To obtain the problem $T(N_\le)_L((P \times Q)_{\le\gamma})$ associated to L as a leaf of the branch-and-bound tree $T(N_\le)$ for $(P \times Q)_{\le\gamma}$, we have to replace the constraint $\alpha^\top x + \beta^\top y \le \delta$, which we denote by η, by $\alpha^\top x \le \delta + \gamma$ and $\beta^\top y \le -\gamma$, which we denote by η_P and η_Q, respectively. We define $\tilde{f}^L$ indexed by the constraints of $T(N_\le)_L((P \times Q)_{\le\gamma})$ by

$$\tilde{f}^L_\nu = \begin{cases} f^L_\nu, & \text{if } \nu \notin \{\eta_P, \eta_Q\}, \\ f^L_\eta, & \text{if } \nu \in \{\eta_P, \eta_Q\}. \end{cases}$$

We claim that $\tilde{f}^L$ is a Farkas-certificate for $T(N_\le)_L((P \times Q)_{\le\gamma})$, if f^L is one for $T_L(P \times Q)$. Indeed, let us consider the system $M\binom{x}{y} \le m$, with

$$M := \begin{pmatrix} A & 0 \\ 0 & B \\ E_1 & E_2 \end{pmatrix} \text{ and } m := \begin{pmatrix} a \\ b \\ e \end{pmatrix},$$

such that with $E := [E_1, E_2]$ we have that $E\binom{x}{y} \le e$ are the constraints of $T_L(P \times Q)$ which come from branching decisions in T and additionally η_P and η_Q. Moreover, extend f^L and $\tilde{f}^L$ with zeros, so that the following computations are well-defined:

$$\begin{aligned} (f^L - \tilde{f}^L)^\top M &= f^L_\eta M_\eta - \tilde{f}^L_{\eta P} M_{\eta_P} - \tilde{f}^L_{\eta_Q} M_{\eta_Q} \\ &= f^L_\eta M_\eta - f^L_\eta M_{\eta_P} - f^L_\eta M_{\eta_Q} \\ &= f^L_\eta (M_\eta - M_{\eta_P} - M_{\eta_Q}) = f^L_\eta \cdot 0 = 0 \end{aligned}$$

and

$$\begin{aligned} (f^L - \tilde{f}^L)^\top m &= f^L_\eta m_\eta - \tilde{f}^L_{\eta P} m_{\eta_P} - \tilde{f}^L_{\eta_Q} m_{\eta_Q} \\ &= f^L_\eta m_\eta - f^L_\eta m_{\eta_P} - f^L_\eta m_{\eta_Q} = 0. \end{aligned}$$

Hence, we have

$$(\tilde{f}^L)^\top M = (f^L)^\top M = 0 \text{ and } (\tilde{f}^L)^\top m = (f^L)^\top m < 0.$$

Thus, $\tilde{f}^L$ is a valid Farkas-certificate for the infeasibility of $T(N_\leq)_L((P \times Q)_{\leq\gamma})$, if f^L is one for $T_L(P \times Q)$. Furthermore, note that if there is an edge e in the path from the root to L in T labeled with an inequality $\tilde{\alpha}^\top x + \tilde{\beta}^\top y \leq \tilde{\delta}$, such that $U \cap \{\tilde{\alpha}^\top x + \tilde{\beta}^\top y \leq \tilde{\delta}\} = \emptyset$ and e is not the edge between the root and $N_\leq$, then e is also contained in the path between the root and L in $T(N_\leq)$.

Hence, if $U \cap \{\alpha^\top x + \beta^\top y \leq \delta\} \neq \emptyset$, then $T(N_\leq)$ with the quasi-Farkas-certificates constructed above is indeed a valid quasi-certified branch-and-bound tree for $(P \times Q)_{\leq\gamma}$ relative to U and thus, the induction hypothesis implies that (a) or (b) holds for $T(N_\leq)$ as a branch-and-bound tree for $(P \times Q)_{\leq\gamma}$ relative to U. Moreover, if $U \cap \{\alpha^\top x + \beta^\top y \leq \delta\} = \emptyset$, then we have $U_P \cap \{\alpha^\top x \leq \delta + \gamma\} = \emptyset$ or $U_Q \cap \{\beta^\top y \leq -\gamma\} = \emptyset$.

Define the following cases:

(a′) Case (a) holds for $T(N_\leq)$ and $(P \times Q)_{\leq\gamma}$ or $U_P \cap \{\alpha^\top x \leq \delta + \gamma\} = \emptyset$.

(b′) Case (b) holds for $T(N_\leq)$ and $(P \times Q)_{\leq\gamma}$ or $U_Q \cap \{\beta^\top y \leq -\gamma\} = \emptyset$.

This allows us to define $(\mu_\leq(\gamma))_{\gamma\in\mathbb{Z}}$ by

$$\mu_\leq(\gamma) := \begin{cases} -1, & \text{if case (a′) holds, but case (b′) does not,} \\ 0, & \text{if cases (a′) and (b′) both hold,} \\ 1, & \text{if case (b′) holds, but case (a′) does not.} \end{cases}$$

Indeed, by the induction hypothesis and our previous considerations, we have defined $\mu_\leq$ for all possible cases. It is easy to see that $\mu_\leq$ is non-decreasing. Moreover, $\mu_\leq$ is neither identically -1 nor identically 1, since sufficiently extreme values of γ can render both $(U_P \cap \{\alpha^\top x \leq \delta + \gamma\})$ and $(U_Q \cap \{\beta^\top y \leq -\gamma\})$ empty.

Similarly, one defines $(P \times Q)_{\geq\gamma} := (P \cap \{\alpha^\top x \geq \delta + \gamma + 1\}) \times (Q \cap \{\beta^\top y \geq -\gamma\})$ and then the following cases:

(a″) Case (a) holds for $T(N_\geq)$ and $(P \times Q)_{\geq\gamma}$ or $U_P \cap \{\alpha^\top x \geq \delta + \gamma + 1\} = \emptyset$.

(b″) Case (b) holds for $T(N_\geq)$ and $(P \times Q)_{\geq\gamma}$ or $U_Q \cap \{\beta^\top y \geq -\gamma\} = \emptyset$,

and finally $(\mu_\geq(\gamma))_{\gamma\in\mathbb{Z}}$ by

$$\mu_\geq(\gamma) := \begin{cases} -1, & \text{if case (a″) holds, but case (b″) does not,} \\ 0, & \text{if cases (a″) and (b″) both hold,} \\ 1, & \text{if case (b″) holds, but case (a″) does not.} \end{cases}$$

One checks easily that $\mu_\geq(\gamma)$ is non-increasing and neither identically -1 nor identically 1.

The noted properties of $\mu_\leq$ and $\mu_\geq$ imply that at least one of the following cases hold: (i) There is $\gamma \in \mathbb{Z}$, such that $\mu_\leq(\gamma) \leq 0$ and $\mu_\geq(\gamma) \leq 0$, or (ii) there is $\gamma \in \mathbb{Z}$, such that $\mu_\leq(\gamma) = \mu_\geq(\gamma - 1) = 1$.

If there exists γ as in case (i), i.e., cases (a′) and (a″) hold for this γ, we construct T^P as desired by branching on $\alpha^\top x \leq \delta + \gamma \vee \alpha^\top x \geq \delta + \gamma + 1$ and attaching to the resulting children $N_\leq$ and $N_\geq$ the following trees: If (a) holds for $T(N_\leq)$ and $(P \times Q)_{\leq\gamma}$, we attach the quasi-certified branch-and-bound tree $T(N_\leq)^P$ for $P \cap \{\alpha^\top x \leq \delta + \gamma\}$ relative to U_P conforming to $T(N_\leq)$ given by (a). While doing so, we can keep the quasi-Farkas-certificates attached to leaves in $T(N_\leq)^P$. For this, note that the set of constraints describing the problem $(T(N_\leq)^P)_L(P \cap \{\alpha^\top x \leq \gamma + \delta\})$ associated to a leaf L in the branch-and-bound tree $T(N_\leq)^P$ for $P \cap \{\alpha^\top x \leq \gamma + \delta\}$ is identical to the constraints describing the problem $(T^P)_L(P)$ associated to L as a leaf of the branch-and-bound tree T^P for P. If (a) does not hold for $T(N_\leq)$ and $(P \times Q)_{\leq\gamma}$, we have $U_P \cap \{\alpha^\top x \leq$

$\delta + \gamma\} = \emptyset$, and we attach an arbitrary, not necessarily valid quasi-certified branch-and-bound tree conforming to $T(N_{\leq})$.

Similarly, if (a) holds for $T(N_{\geq})$ and $(P \times Q)_{\geq\gamma}$, then we attach to $N_{\geq}$ the quasi-certified branch-and-bound tree $T(N_{\geq})^P$ relative to U_P conforming to $T(N_{\geq})$ given by (a). Otherwise, we have $U_P \cap \{\alpha^\top x \geq \delta + \gamma \ + 1\} = \emptyset$, and we attach an arbitrary, not necessarily valid quasi-certified branch-and-bound tree for P conforming to $T(N_{\geq})$. Choose quasi-Farkas-certificates as in the previous case.

It is then easy to see that T^P is conforming to T. It remains to check that T^P is a valid quasi-certified branch-and-bound tree for P relative to U_P. For this consider a leaf L of T^P in the subtree rooted at $N_{\leq}$. If we have $U_P \cap \{\alpha^\top x \leq \delta + \gamma\} = \emptyset$, there is nothing to check for f^L. Similarly, if there is an edge in $T(N_{\leq})^P$ on the path from the root to L labeled with an inequality $\tilde{\alpha}^\top x \leq \tilde{\delta}$, such that $U_P \cap \{\tilde{\alpha}^\top x \leq \tilde{\delta}\} = \emptyset$, then the same inequality appears on the path from the root to L in T^P. The only remaining case is that the quasi-Farkas-certificate f^L attached to L in $T(N_{\leq})^P$ is a Farkas-certificate for the associated problem $(T(N_{\leq})^P)_L(P \cap \{\alpha^\top x \leq \delta + \gamma\})$. But then f^L is also a Farkas-certificate for the problem $(T^P)_L(P)$ associated to L in T^P, which is the same problem. For leaves in the subtree rooted at $N_{\geq}$ we proceed similarly.

In case (ii) we proceed analogously. □

4.2 Proofs of Lemma 11 and Corollary 12

Proof of Lemma 11. For ease of notation, we write $p := q-1$ instead. Thus we have to compute

$$b\colon (x_1, \dots, x_{k-1}) \mapsto \max\left\{\lambda \in \{0, \dots, 2^{p+1} - 1\} \mid C(x_1, \dots, x_{k-1}, \Lambda_{\max} - \lambda) = 1\right\}.$$

We will construct a monotone real circuit $\tilde{C}$ of the desired size which works in $p+1$ phases $0, \dots, p$. For each phase i, there will be a gate h_i in $\tilde{C}$ representing the state of computation after phase i. The gate h_i will compute the function

$$b_i(x) := \lfloor b(x) \rfloor_{2^{p-i}},$$

where $x = (x_1, \dots, x_{k-1})$ and $\lfloor\cdot\rfloor_{2^{p-i}}$ denotes rounding down to the nearest integer divisible by 2^{p-i}. Clearly, this suffices, since h_p then computes the desired function. Moreover, b_0 can be computed by a copy C_0 of C, which receives $(x, \Lambda_{\max} - 2^p)$ where the output gate h_0 is modified to compute $b_0(x) = 2^p C(x, \Lambda_{\max} - 2^p)$.

To construct the part of $\tilde{C}$ belonging to phase $i > 0$, we will rely on the recurrence

$$b_i(x) = b_{i-1}(x) + 2^{p-i} C(x, \Lambda_{\max} - b_{i-1}(x) - 2^{p-i}).$$

Unfortunately, this recurrence is not necessarily monotone in x and b_{i-1} due to the sign on the second occurrence of $b_{i-1}(x)$. However, since it is immediate from the definition that b_{i-1} is divisible by $2^{p-(i-1)}$, we might as well consider the recurrence

$$b_i = \lfloor b_{i-1} \rfloor_{2^{p-(i-1)}} + 2^{p-i} C(x, \Lambda_{\max} - \lfloor b_{i-1} \rfloor_{2^{p-(i-1)}} - 2^{p-i}). \tag{4}$$

Note that we drop the dependence of b_i and b_{i-1} on x to improve readability.

Unfortunately for us, the latter recurrence – while clearly monotone in x and b_{i-1} – still does not provide an obvious monotone real circuit, since the last summand applies a non-monotone function to b_{i-1}.

Thus, we have to give a version of C, which passes along the old bound b_{i-1} from the k-th input gate to the output gate in the higher order bits, together with how C behaves on input $(x, \Lambda_{\max} - b_{i-1} - 2^{p-i})$ in the lower order bits, in order to make our computation monotone.

We begin by assuming that every non-input gate in C applies a function f with range $f \subseteq (0, \frac{1}{2})$. This can be achieved by post-composing the function applied at every gate with the monotone bijection

$$\varphi\colon \mathbb{R} \to (0, \tfrac{1}{2}), \quad y \mapsto (\arctan(y) + \tfrac{\pi}{2})/2\pi$$

and pre-composing every function applied at a gate, which takes as input such a modified gate, with φ^{-1} for the respective input. Let C' be the monotone real circuit obtained from C this way. We then have $C'(x) = \varphi(C(x))$.

Next, we introduce a gate $\tilde{g}_k$, which provides both the k-th input (which is supposed to be b_{i-1}) and the transformed value of $\Lambda_{\max} - b_{i-1} - 2^{p-i}$, i.e, both inputs to $\tilde{g}_k$ are the k-th input gate g_k of C' and the function applied at $\tilde{g}_k$ is

$$f^i(g_k, g_k) \coloneqq \lfloor g_k \rfloor_{2^{p-(i-1)}} + \varphi\big(\Lambda_{\max} - \lfloor g_k \rfloor_{2^{p-(i-1)}} - 2^{p-i}\big).$$

Then, for every gate g which uses g_k as input in C', we let g use $\tilde{g}_k$ as input instead. We note f^i is non-decreasing: If an increase of g_k would cause the summand involving φ to decrease, then it decreases by at most $\frac{1}{2}$, but the other summand then increases by at least 1.

Let S denote the set of gates in C' which are a descendant of g_k (hence now of $\tilde{g}_k$) and consider a gate g in S with predecessors g_1 and g_2, such that $g_1 \in S$, but $g_2 \notin S$. We then replace the function $f_g(g_1, g_2)$ applied at g by the function

$$f_g^i(g_1, g_2) \coloneqq \lfloor g_1 \rfloor_{2^{p-(i-1)}} + f_g\big(\{g_1\}_{2^{p-(i-1)}},\, g_2\big),$$

where $\{g_1\}_{2^{p-(i-1)}} \coloneqq g_1 - \lfloor g_1 \rfloor_{2^{p-(i-1)}}$. We check that f_g^i is non-decreasing: If an increase of g_1 would cause $\{g_1\}$ to decrease, then $f_g(\{g_1\}_{2^{p-(i-1)}},\, g_2)$ decreases by at most $\frac{1}{2}$, since range $f_g \subseteq (0, \frac{1}{2})$, while $\lfloor g_1 \rfloor_{2^{p-(i-1)}}$ increases by at least 1.

If g is a gate with both predecessor g_1 and g_2 in S, we replace the function f_g applied at g by

$$f_g^i(g_1, g_2) \coloneqq (\lfloor g_1 \rfloor_{2^{p-(i-1)}} + \lfloor g_2 \rfloor_{2^{p-(i-1)}})/2 + f_g\big(\{g_1\}_{2^{p-(i-1)}},\, \{g_2\}_{2^{p-(i-1)}}\big),$$

Once again, f_g^i is non-decreasing by an analogous argument.

Let C_i denote the monotone real circuit obtained by applying these modifications to C'. For a gate g' in C', let g^i denote the corresponding gate of C_i. It is then easy to show by induction along a topological order on the gates of C' (or equivalently C_i) that

$$g^i(x, b_{i-1}) = \begin{cases} g'\big(x,\, \Lambda_{\max} - b_{i-1} - 2^{p-i}\big), & \text{if } g \notin S, \\ b_{i-1} + g'\big(x,\, \Lambda_{\max} - b_{i-1} - 2^{p-i}\big) & \text{if } g \in S. \end{cases}$$

We note that this holds for the input gates $g_1, \dots g_{k-1}$ and $\tilde{g}_k$. For a non-input gate g' in C', consider for example the case where g has predecessors g'_1 and g'_2, such that $g'_1 \in S$ and $g'_2 \notin S$. Then a straight-forward computation yields

$$\begin{aligned} g^i(x, b_{i-1}) &= f_g^i\big(g_1^i(x, b_{i-1}), g_2^i(x, b_{i-1})\big) \\ &= \lfloor g_1^i(x, b_{i-1}) \rfloor_{2^{p-(1-i)}} + f_g\big(\{g_1^i(x, b_{i-1})\}_{2^{p-(1-i)}},\, g_2^i(x, b_{i-1})\big) \\ &= \lfloor b_{i-1} + g'_1\big(x,\, \Lambda_{\max} - b_{i-1} - 2^{p-i}\big) \rfloor_{2^{p-(1-i)}} \\ &\quad + f_g\big(\{b_{i-1} + g'_1(x,\, \Lambda_{\max} - b_{i-1} - 2^{p-i})\}_{2^{p-(i-1)}},\, g'_2(x, \Lambda_{\max} - b_{i-1} - 2^{p-i})\big) \\ &= b_{i-1} + f_g\big(g'_1(x,\, \Lambda_{\max} - b_{i-1} - 2^{p-i}),\, g'_2(x, \Lambda_{\max} - b_{i-1} - 2^{p-i})\big) \\ &= b_{i-1} + g'\big(x,\, \Lambda_{\max} - b_{i-1} - 2^{p-i}\big) \end{aligned}$$

as desired, where the first and last identity are due to the definition of the computed function, the second due to the definition of f_g^i, the third due to the induction hypothesis and the fourth due to $b_{i-1} = \lfloor b_{i-1} \rfloor_{2^{p-(i-1)}}$ and range $g \in (0, \frac{1}{2})$. The other cases are analogous.

By considering this identity for the output gate h^i of C_i (note that we can assume to be in the second case), we obtain

$$\begin{aligned} C_i(x, b_{i-1}) &= b_{i-(i-1)} + C'(x, \Lambda_{\max} - b_{i-1} - 2^{p-i}) \\ &= b_{i-(i-1)} + \varphi(C(x, \Lambda_{\max} - b_{i-1} - 2^{p-i})). \end{aligned}$$

Thus, by our recurrence (4), if we post-compose the function applied at h_i with

$$y \mapsto \begin{cases} \lfloor y \rfloor_{2^{p-(i-1)}} & \text{if } \{y\}_{2^{p-(i-1)}} < \varphi(1), \\ \lfloor y \rfloor_{2^{p-(i-1)}} + 2^{p-i} & \text{if } \{y\}_{2^{p-(i-1)}} \geq \varphi(1), \end{cases}$$

we obtain a monotone real circuit which on input (x, b_{i-1}) computes b_i. For this, note that $\{C_i(x, b_{i-1})\}_{2^{p-(i-1)}} = \varphi(C(x, \Lambda_{\max} - b_{i-1} - 2^{p-i})) < \varphi(1)$ is equivalent to $C(x, \Lambda_{\max} - b_{i-1} - 2^{p-i}) = 0$. Then $\tilde{C}$ can be constructed in the obvious way, i.e., by sequentially using the constructed circuits $C_0, \ldots, C_p$ to compute the values $b_0(x), \ldots, b_p(x) = b(x)$.

For the size bound, observe that we have used $q = p+1$ copies of C and that the introduced auxiliary gates $\tilde{g}_k$ can be eliminated from the circuit, since the functions applied at the children of $\tilde{g}_k$ can instead be pre-composed with the function applied at $\tilde{g}_k$. □

Proof of Corollary 12. Choose $q \coloneqq \lceil \log(L+1) \rceil$ and apply Lemma 11 to C_1 to construct $\tilde{C}_1$, such that the output gate h of $\tilde{C}_1$ computes

$$b\colon (x_1, \ldots, x_{k-1}) \mapsto \max \left\{ \lambda \in \{0, \ldots, 2^q - 1\} \;\middle|\; C(x_1, \ldots, \Lambda_{\max} - \lambda) = 1 \right\}.$$

and use a copy of C_2 to compute $C_2(x_1, \ldots, x_{k-1}, \kappa - (\Lambda_{\max} - b(x_1, \ldots, x_{k-1})))$.

Clearly, if the output gate of this copy of C_2 computes 1, then $x_k = \Lambda_{\max} - b(x_1, \ldots, x_{k-1})$ and $x'_k = \kappa - x_k$ satisfy $C_1(x_1, \ldots, x_{k-1}, x_k) = C_2(x_1, \ldots, x_{k-1}, x'_k) = 1$ and $x_k + x'_k = \kappa$. Otherwise, there are no possible such choices for x_k and x'_k, since for $x_k < \Lambda_{\max} - b(x_1, \ldots, x_{k-1})$ we have $C_1(x_1, \ldots, x_{k-1}, x_k) = 0$ and for $x_k \geq \Lambda_{\max} - b(x_1, \ldots, x_{k-1})$ and $x'_k = \kappa - x_k$ we have $C_2(x_1, \ldots, x_{k-1}, x'_k) = 0$. Hence, the constructed circuit decides the question posed in the corollary. □

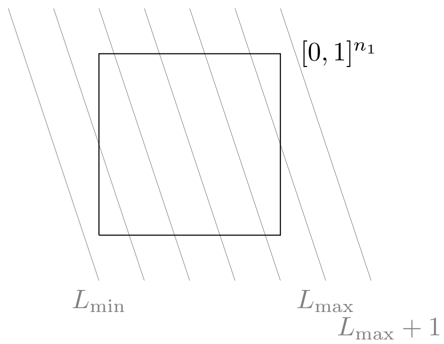
4.3 Proof of Theorem 4

For ease of notation, we assume that the variable bounds in (1) are incorporated into the constraints. Then the LP-relaxation of (1) is given by:

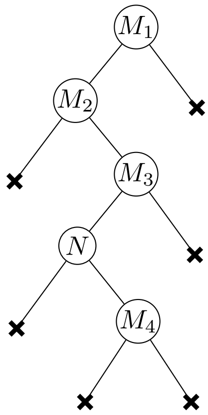
$$\begin{pmatrix} A \\ 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ B \end{pmatrix} y + \begin{pmatrix} C \\ D \end{pmatrix} z \leq \begin{pmatrix} a \\ b \end{pmatrix},$$

The basic structure of the proof is as follows: Given $z \in Z_1 \cup Z_2$, where $Z_1 = \{z \in \{0,1\}^{n_3} \mid \exists x \in \{0,1\}^{n_1} \colon Ax \leq a - Cz\}$ and $Z_2 = \{z \in \{0,1\}^{n_3} \mid \exists y \in \{0,1\}^{n_2} \colon By \leq b - Dz\}$, and a branch-and-bound tree T for the infeasibility of (1), we compute Farkas-certificates for the leaves of T and then obtain a certified branch-and-bound tree $\tilde{T}$ for $P(z) \times Q(z) \coloneqq \{Ax \leq a - Cz\} \times \{Bx \leq b - Dz\}$ by plugging in the values for z in the disjunctions used in T. Then at least one of the alternatives in Lemma 10 holds. However, since $z \in Z_1 \cup Z_2$, exactly one of $P(z)$ and $Q(z)$ is integer-feasible, and thus at most one of the alternatives in Lemma 10 holds. Clearly, we have $z \in Z_2$ if and only if there exists a quasi-certified branch-and-bound tree $(\tilde{T})^P$ for $P(z)$ conforming to $\tilde{T}$. Thus, if we construct a monotone real circuit C that given values for the z variables decides whether there exists such a tree $(\tilde{T})^P$, then C separates Z_1 and Z_2. We work out the details below:

Proof of Theorem 4. We again assume that variable bounds in (1) are incorporated into the constraints as above. We begin by applying Theorem 1 to our branch-and-bound tree T for (1) to obtain a certified branch-and-bound tree T' for (1) with bounded coefficients: Note that the linear programming relaxation of (1) is contained in the ball $B_1^n(n) = \{x \in \mathbb{R}^n \mid \|x\|_1 \leq n\}$, where $n \coloneqq n_1 + n_2 + n_3$. Hence, we can assume that for every disjunction $d^\top w \leq \delta \vee d^\top w \geq \delta + 1$ used in T', we have $\max\{\|d\|_\infty, |\delta|\} \leq (10n^2)^{(n+2)^2}$ and moreover we have $|T'| \leq (4n+5)|T|$. Then we fix some Farkas-certificates for T' which thus becomes a certified branch-and-bound tree.



(a) The hyperplanes $\alpha^\top x = \gamma$ for different values of γ.



(b) An example for T' for which $\mathcal{V}^+(N) = \{\gamma^+_{M_2}\}$, $\mathcal{V}^-(N) = \{\gamma^-_{M_1}, \gamma^-_{M_3}\}$ and $\mathcal{U}(N) = \{\gamma^-_N, \gamma^+_N, \gamma^-_{M_4}, \gamma^+_{M_4}\}$.

Figure 2: Illustrations of $L_{\min}$, $L_{\max}$, $\mathcal{U}(N)$ and $\mathcal{V}(N)$.

By fixing the values of z in the disjunctions used in T', we obtain a certified branch-and-bound tree $\tilde{T}$ for $P(z) \times Q(z) := \{Ax \le a - Cz\} \times \{By \le b - Dz\}$. Since $P(z) \times Q(z) \subseteq [0,1]^{n_1} \times [0,1]^{n_2}$, we may also consider $\tilde{T}$ as a quasi-certified branch-and-bound tree for $P(z) \times Q(z)$ relative to $[0,1]^{n_1} \times [0,1]^{n_2}$.

Let $\mathcal{N}(T')$ denote the set of internal nodes of T' and let $\gamma \in \mathbb{Z}^{\mathcal{N}(T')}$. Consider the not necessarily valid quasi-certified branch-and-bound tree $\tilde{T}^P(\gamma)$ for $P(z)$ relative to $[0,1]^{n_1}$, which has the same underlying directed tree as $\tilde{T}$, and at a node N branches on the disjunction $\alpha_N^\top x \le \gamma_N \vee \alpha_N^\top x \ge \gamma_N + 1$, when $\tilde{T}$ branches at N on the disjunction $\alpha_N^\top x + \beta_N^\top y \le \delta \vee \alpha_N^\top x + \beta_N^\top y \ge \delta_N + 1$. Similarly, the Farkas-certificate at a leaf L of $\tilde{T}^P(\gamma)$ is the Farkas-certificate at leaf L of $\tilde{T}$ with the entries corresponding to constraints from $Q(z)$ removed. We are interested in whether there exists a choice for γ for which $\tilde{T}^P(\gamma)$ is a valid quasi-certified branch-and-bound tree for $P(z)$ relative to $[0,1]^{n_1}$.

For any candidate disjunction $\alpha_N^\top x \le \gamma_N \vee \alpha_N^\top x \ge \gamma_N + 1$ to be used at a node N in $\tilde{T}^P(\gamma)$, the slab $\{x \in \mathbb{R}^{n_1} : \gamma_N \le \alpha_N^\top x \le \gamma_N + 1\}$ has width $\frac{1}{\|\alpha_N\|_2} \ge \frac{1}{\sqrt{n_1}\|\alpha_N\|_\infty}$. Since

$$\max\left\{\left(\frac{\alpha_N}{\|\alpha_N\|_2}\right)^\top x \mid x \in [0,1]^{n_1}\right\} - \min\left\{\left(\frac{\alpha_N}{\|\alpha_N\|_2}\right)^\top x \mid x \in [0,1]^{n_1}\right\} \le \sqrt{n}_1,$$

we have that at most $\sqrt{n_1} \cdot \sqrt{n_1}\|\alpha_N\|_\infty + 2 \le n(10n^2)^{(n+2)^2} + 2$ of our slabs intersect $[0,1]^{n_1}$. Let $L^N_{\min}$ denote the maximal value for γ for which $[0,1]^{n_1} \cap \{\alpha_N^\top x \le \gamma\} = \emptyset$ (cf. Figure 2a). Similarly, let $L^N_{\max}$ denote the minimal γ for which $[0,1]^{n_1} \cap \{\alpha_N^\top x \ge \gamma + 1\} = \emptyset$. Moreover, let $L^N := L^N_{\max} - L^N_{\min}$ and $L := \max\{L^N \mid N \text{ internal node of } T'\} \le n(10n^2)^{(n+2)^2} + 2$.

Then, for every internal node N of T', we introduce two variables γ^+_N and γ^-_N. Variable γ^+_N represents how far the right-hand-side of the disjunction at N is chosen away from the lower bound $L^N_{\min}$, while γ^-_N represents how far the right-hand-side of the disjunction at N is chosen away from the upper bound $L^N_{\max}$. Thus, $\gamma_N = L^N_{\min} + \gamma^+_N = L^N_{\max} - \gamma^-_n$. Hence, in order for the pair (γ^+_N, γ^-_N) to represent a valid right-hand-side for the node N, we must have $\gamma^+_N + \gamma^-_N = L^N$. Note that this representation of γ_N allows us to work with the usual definition of monotone real circuits and not deal with the case where a function is non-increasing in an input variable.

For every node N of T', let $\mathrm{anc}(N)$ denote the set of proper ancestors of N (i.e., excluding N).

Then define

$$\mathcal{V}(N) \coloneqq \mathcal{V}^+(N) \cup \mathcal{V}^-(N) \coloneqq \left\{ \gamma_M^+ \;\middle|\; \begin{array}{l} M \in \operatorname{anc}(N) \text{ and } N \text{ is in the subtree} \\ \text{rooted at the } \alpha_M^\top x \geq \gamma_M + 1\text{-child of } M \end{array} \right\}$$
$$\cup \left\{ \gamma_M^- \;\middle|\; \begin{array}{l} M \in \operatorname{anc}(N) \text{ and } N \text{ is in the subtree} \\ \text{rooted at the } \alpha_M^\top x \leq \gamma_M\text{-child of } M \end{array} \right\}$$

and

$$\mathcal{U}(N) \coloneqq \{\gamma_M^-, \gamma_M^+ \mid M \text{ internal node of } T' \text{ and descendant of } N\}.$$

For this definition, we consider N as a descendant of N. See Figure 2b for an example.

Then, for the sake of induction, we strengthen the statement of the theorem to:

Claim. *For every node N of T', there exists a monotone real circuit C_N of size $|T'(N)| \cdot 2(|T'| + n_3) \cdot \log(L+1)^{\log|T'(N)|}$, which receives as inputs values for the variables $\mathcal{Z} \cup \mathcal{V}(N)$, where $\mathcal{Z} \coloneqq \{z_1, \ldots, z_{n_3}\}$ and decides whether there exist values for the variables $\mathcal{U}(N)$, which obey $\gamma_N^+ + \gamma_N^- = L^N$ and choosing*

$$\gamma_N \coloneqq L_{\min}^N + \gamma_N^+ \qquad \forall\, \gamma_N^+ \in \mathcal{U}(N) \cup \mathcal{V}^+(N) \quad \text{and}$$
$$\gamma_N \coloneqq L_{\max}^N - \gamma_N^- \qquad \forall\, \gamma_N^- \in \mathcal{V}^-(N)$$

turns every f^L attached to a leaf L in the corresponding subtree $\tilde{T}^P(\gamma)(N)$ rooted at N in $\tilde{T}^P(\gamma)$ into a valid quasi-Farkas-certificate for L in the branch-and-bound tree $\tilde{T}^P(\gamma)$ for $P(z)$ relative to $[0,1]^{n_1}$.

Note that γ is only partially defined by the definition given in the claim; however, all entries of γ which are relevant for the validity of the quasi-Farkas-certificates in $\tilde{T}^P(\gamma)(N)$ are defined.

It suffices to show the claim, since then $C \coloneqq C_r$ (where r denotes the root of T') decides whether there exists a branch-and-bound tree $(\tilde{T})^P$ for $P(z)$ conforming to $\tilde{T}$ and hence separates Z_1 and Z_2.

We begin by noting that for every input $\gamma_M^\pm$ of such a circuit C_N corresponding to the right-hand-side of the disjunction used at a node M in $\mathcal{V}(N)$, we have $C_N(z, \tilde{\gamma}, L^M) = 1$ for all $(z, \tilde{\gamma}) \in \mathbb{R}^{\mathcal{Z} \cup (\mathcal{V}(N) \setminus \gamma_M^\pm)}$, since then the side of the disjunction at M which corresponds to the branch containing N does not intersect $[0,1]^{n_1}$ and hence all vectors attached to leaves in $\tilde{T}^P(\gamma)(N)$ are valid quasi-Farkas-certificates.

We prove the claim via induction on $|\tilde{T}(N)|$. If $|\tilde{T}(N)| = 1$, then N is a leaf. Hence, $\mathcal{U}(N) = \emptyset$ and since we are given values for all variables from $\mathcal{Z} \cup \mathcal{V}(N)$, we are given all right-hand-sides to the subproblem $\tilde{T}^P(\gamma)_N(P(z)) =: \{x \in \mathbb{R}^{n_1} \mid Ex \leq e\}$ associated to the leaf N of the tree $\tilde{T}^P(\gamma)$ for $P(z)$. We have to test whether f^N is a valid quasi-Farkas-certificate. To this end, it suffices to test if $(f^L)^\top e < 0$, since we have $(f^L)^\top E = 0$, because T' is a valid certified branch-and-bound tree for $P(z) \times Q(z)$ (see the proof of Lemma 10). Note that $\tilde{T}^P(\gamma)_N(P(z))$ contains constraints which are also contained in $P(z)$ and are indexed with numbers $i \in [m_1]$ and constraints of the form $\alpha_M^\top x \geq \gamma_M + 1$ or $\alpha_M^\top x \leq \gamma_M$ coming from branching, which we will index with the node M at which they appear in a disjunction. We then calculate:

$$\begin{aligned}(f^L)^\top e &= \sum_{i \in [m_1]} f_i^N (a_i - C_i z) + \sum_{\gamma_M^+ \in \mathcal{V}^+(N)} f_M^N(-L_{\min}^M - \gamma_M^+ - 1) + \sum_{\gamma_M^- \in \mathcal{V}^-(N)} f_M^N(L_{\max}^M - \gamma_M^-) \\ &=: k^N - \sum_{\tau \in \mathcal{Z} \cup \mathcal{V}(N)} s_\tau^N \cdot \tau,\end{aligned}$$

where C_i is the i-th row of C and the second line is defined by aggregating variables and constants. We note that the resulting s_τ^N are non-negative (recall C is non-negative). Evidently, the sum in the second line can be computed by a monotone real circuit with inputs corresponding to the elements of $\mathcal{Z} \cup \mathcal{V}(N)$ and $|\mathcal{Z} \cup \mathcal{V}(N)| - 1$ further gates by iteratively adding summands.

Note that adding s_τ^N times the first input to the second input is a monotone operation, since s_τ^N is non-negative. By post-composing the function applied at the output gate with the function sending numbers larger than k^N to 1 and numbers at most k^N to 0, we obtain a monotone real circuit $\hat{C}_N$ that decides whether f^N is a Farkas-certificate.

We modify $\hat{C}_N$ to obtain a monotone real circuit C_N which decides whether f^N is a quasi-Farkas-certificate as follows: For every gate g in $\hat{C}_N$ which adds s_τ^N-times the value of an input $\gamma_M^+ \in \mathcal{V}(N)$ (or γ_M^-) to our sum, we modify the function applied at this gate such that it adds a very large constant K^N instead, if $\gamma_M^+ \geq L_M$ ($\gamma_M^- \geq L_M$). By our definition of L_M and γ_M, this is the case if and only if the side of the disjunction at the node M corresponding to the subtree of M containing N, does not intersect $[0,1]^{n_1}$, which makes f^N a valid quasi-Farkas-certificate relative to $[0,1]^{n_1}$ by definition. Hence, if we choose K^N sufficiently large, such that C_N will certainly accept in this case, for example $K^N \coloneqq k^N + 1$, then C_N correctly decides whether f^N is a valid quasi-Farkas-certificate. Moreover, C_N satisfies the claimed bound on its size.

If $|T'(N)| > 1$, we appeal to Corollary 12: Let $N_\leq$ and $N_\geq$ denote the children of N. Then, by the induction hypothesis, there exist circuits $C_{N_\leq}$ and $C_{N_\geq}$ for these nodes as in the claim. Since $\mathcal{V}(N_\leq) = \mathcal{V}(N) \cup \{\gamma_N^-\} = (\mathcal{V}(N_\geq) \setminus \{\gamma_N^+\}) \cup \{\gamma_N^-\}$ and $C_{N_\leq}(z, \tilde{\gamma}, L^N) = C_{N_\geq}(z, \tilde{\gamma}, L^N) = 1$ for all $(z, \tilde{\gamma}) \in \mathbb{R}^{\mathcal{Z} \cup \mathcal{V}(N)}$, we may apply Corollary 12 to $C_{N_\leq}$ and $C_{N_\geq}$ (with $\Lambda_{\max} = L^N$, $\Lambda_{\min} = 0$ and $\kappa = L^N$), in a way which invokes the larger circuit only once.

To see that C_N is no larger than claimed, assume the subtree $T'(N_\leq)$ of T' rooted at $N_\leq$ is smaller than the one rooted at $N_\geq$, the other case is analogous. Hence, $T'(N_\leq)$ has size at most $|T'(N)|/2$ while $T'(N_\geq)$ has size at most $|T'(N)| - 1$. Then, compute

$$\begin{aligned}
|C_N| &\leq |C_{N_\leq}| \cdot (\lceil \log(L^N + 1) \rceil) + |C_{N_\geq}| \\
&\leq |T'(N_\leq)| \cdot 2(|T'| + n_3) \cdot (\lceil \log(L+1) \rceil)^{\log(|T'(N)|/2)} \cdot (\lceil \log(L^N + 1) \rceil) \\
&\quad + |T'(N_\geq)| \cdot 2(|T'| + n_3) \cdot (\lceil \log(L+1) \rceil)^{\log(|T'(N)|-1)} \\
&\leq |T'(N_\leq)| \cdot 2(|T'| + n_3) \cdot (\lceil \log(L+1) \rceil)^{\log|T'(N)|} \\
&\quad + |T'(N_\geq)| \cdot 2(|T'| + n_3) \cdot (\lceil \log(L+1) \rceil)^{\log|T'(N)|} \\
&\leq |T'(N)| \cdot 2(|T'| + n_3) \cdot (\lceil \log(L+1) \rceil)^{\log|T'(N)|}.
\end{aligned}$$

Finally, set $C \coloneqq C_r$ for the circuit C_r given by the claim for the root node r of T' and note $\lceil \log(L+1) \rceil = \lceil \log(n(10n^2)^{(n+2)^2} + 3) \rceil$ as well as $|T'| \leq (4n+5)|T|$. Hence C_r has size at most

$$\begin{aligned}
&|T'(r)| \cdot 2(|T'| + n_3) \cdot (\lceil \log(L+1) \rceil)^{\log|T'(r)|} \\
&\leq (4n+5)|T| \cdot 2[(4n+5)|T| + n] \cdot [(n+2)^2 \log(10n^3 + 3)]^{\log((4n+5)|T|)} \\
&\leq 2(5n+5)^2 |T|^2 \cdot [(n+2)^2 \log(10n^3 + 3)]^{\log((4n+5)|T|)} \\
&\leq 50(n+1)^2 |T|^2 \cdot [(n+2)^2 \log(10n^3 + 3)]^{\log((4n+5)|T|)}.
\end{aligned}$$

□

The computations bounding the circuit size in the recursive step are taken from Fleming et al. [13] where they are used to show that branch-and-bound with really small coefficients can be quasi-polynomially simulated by cutting planes. A very similar recursive formula already appears in [5], where it is used to show that branch-and-bound for variable disjunctions is quasi-automatizable.

4.4 Proof of Theorems 5 and 7

Proof of Theorem 5. Assume that we have a family of branch-and-bound trees T for (2), one for each r, such that $|T| \in 2^{O(n^{1/6-\epsilon})}$ for some $\epsilon > 0$. Then Theorem 4 gives rise to a family of circuits C_n separating the CC-pair of size

$$\begin{aligned}|C_n| &= 50(n+1)^2|T|^2 \cdot [(n+2)^2 \log(10n^3+3)]^{\log((4n+5)|T|)} \\ &= 50(n+1)^2 2^{2O(n^{1/6-\epsilon})} \cdot \left(2^{\log[(n+2)^2\log(10n^3+3)]}\right)^{\log(4n+5)O(n^{1/6-\epsilon})} \\ &= 2^{O(n^{1/6-\epsilon})} \cdot 2^{\log[(n+2)^2\log(10n^3+3)]\cdot\log(4n+5)O(n^{1/6-\epsilon})} \\ &= 2^{O(n^{1/6-\epsilon})} \cdot 2^{O(n^{\epsilon/2})\cdot O(n^{1/6-\epsilon})} = 2^{O(n^{1/6-\epsilon})} \cdot 2^{O(n^{1/6-\epsilon/2})} = 2^{O(n^{1/6-\epsilon/2})}.\end{aligned}$$

Since we have $n = n_1 + n_2 + n_3 = r\lfloor \frac{1}{8}(r/\log r)^{2/3} \rfloor + r + (r^2 - r)/2$ and $n_3 = r + (r^2 - r)/2$ we have

$$1 \leq \frac{n}{n_3} = 1 + \frac{O(r^{5/3})}{\Omega(r^2)}$$

Since $n \to \infty$ implies $r \to \infty$, we have $n_3 \in \Theta(n)$. But then we have

$$|C_n| \in 2^{O(n^{1/6-\epsilon/2})} = 2^{O(n_3^{1/6-\epsilon/2})},$$

which contradicts Theorem 2. ☐

For the proof of Theorem 7, we require an analog of Theorem 8 in [18].

Given an (unsatisfiable) CNF $\mathcal{C} = \{C_1, \ldots, C_m\}$ and a partition of its variables $X_0 \cup X_1$, let Y_1, Y_2 and $\mathcal{D}$ be defined as in Section 2.

Observation 13. *Every branch-and-bound tree for the ILP (3) for $\mathcal{C}$ and any partition $X_0 \cup X_1$ is also a branch-and-bound tree for the ILP (3) for $\mathcal{D}$.*

Proof. It suffices to note that linear constraints corresponding to the original clauses of $\mathcal{C}$ are valid inequalities for the LP-relaxation of (3) for $\mathcal{D}$. ☐

Lemma 14. *For every branch-and-bound tree T for (3) for $\mathcal{C}$ and a partition $X_0 \cup X_1$ of its variables, there is a monotone real circuit computing an (X_0, X_1)-infeasibility certificate for $\mathcal{C}$ of size quasi-polynomial in n, m and $|T|$, i.e., size at most* $\text{poly}(n + m + f(n))^{\log(n+m+f(n))}$.

Proof. By Observation 13, we can consider T as a branch-and-bound tree for (3) for $\mathcal{D}$ and hence can apply Theorem 4 to obtain a monotone real circuit separating Y_0 and Y_1 of size

$$50(n'+1)^2|T|^2 \cdot [(n'+2)^2 \log(10n'^3+3)]^{\log((4n'+5)|T|)} \in \text{poly}(n+m+T)^{\log(n+m+|T|)},$$

where $n' = 2m + n$. Since a monotone function separating Y_0 and Y_1 is an (X_0, X_1)-infeasibility certificate for $\mathcal{C}$, the lemma is shown. ☐

Finally, combining Theorem 3 with Lemma 14, we obtain a proof for Theorem 7:

Proof of Theorem 7. Assume that there exists a function $f \in O(2^{n^{o(1)}})$ such that for a random k-CNF $\mathcal{C}$ with $O(n2^k)$ clauses and $2n$ variables there exists a branch-and-bound tree T refuting (3) for $\mathcal{C}$ of size at most $f(n)$ with non-negligible probability, i.e., the probability of this occurring does not tend to 0 for $n \to \infty$. Then, due to Lemma 14, for any fixed partition $X_0 \cup X_1$ of the variables with $|X_0| = |X_1| = n$ there is a monotone real circuit computing an (X_0, X_1)-certificate for $\mathcal{C}$ of size $\text{poly}(n+m+f(n))^{\log(n+m+f(n))}$ with non-negligible probability. We may assume that $f(n) \geq \max(n, m)$ for simplicity, hence $\mathcal{C}$ has size at most $g \in \text{poly}(f(n))^{\log(f(n))}$. However, clearly $\text{poly}(O(2^{n^{o(1)}})) = O(2^{n^{o(1)}})$ and

$$O(2^{n^{o(1)}})^{\log(O(2^{n^{o(1)}}))} = O(2^{n^{o(1)}})^{O(n^{o(1)})} = O(2^{n^{o(1)}\cdot O(n^{o(1)})}) = O(2^{n^{o(1)}}).$$

Hence, for a random k-CNF with $O(n2^k)$ clauses and $2n$ variables with partition $X_0 \cup X_1$ such that $|X_0| = |X_1| = n$ and $k \geq c\log(n)$ there is an (X_0, X_1)-certificate with size at most $g(n)$ with non-negligible probability which contradicts Theorem 3. ☐

References

[1] Alon, N., Boppana, R.B.: The monotone circuit complexity of Boolean functions. Combinatorica **7**, 1–22 (1987). doi:10.1007/bf02579196

[2] Basu, A., Conforti, M., Di Summa, M., Jiang, H.: Complexity of branch-and-bound and cutting planes in mixed-integer optimization – II. Combinatorica **42**(1), 971–996 (Dec 2022). doi:10.1007/s00493-022-4884-7

[3] Beale, E., Forrest, J.: Global optimization using special ordered sets. Mathematical Programming **10**, 52–69 (1976). doi:10.1007/bf01580653

[4] Beame, P., Fleming, N., Impagliazzo, R., Kolokolova, A., Pankratov, D., Pitassi, T., Robere, R.: Stabbing planes. In: Karlin, A.R. (ed.) 9th Innovations in Theoretical Computer Science Conference (ITCS 2018). Leibniz International Proceedings in Informatics (LIPIcs), vol. 94, pp. 10:1–10:20. Schloss Dagstuhl, Germany, Dagstuhl, Germany (2018). doi:10.4230/LIPIcs.ITCS.2018.10

[5] Beame, P., Pitassi, T.: Simplified and improved resolution lower bounds. In: Proceedings of 37th Conference on Foundations of Computer Science (FOCS). pp. 274–282. IEEE (1996). doi:10.1109/sfcs.1996.548486

[6] Bonet, M., Pitassi, T., Raz, R.: Lower bounds for cutting planes proofs with small coefficients. The Journal of Symbolic Logic **62**(3), 708–728 (1997). doi:10.2307/2275569

[7] Dadush, D., Tiwari, S.: On the complexity of branching proofs. In: Saraf, S. (ed.) 35th Computational Complexity Conference (CCC 2020). Leibniz International Proceedings in Informatics (LIPIcs), vol. 169, pp. 34:1–34:35. Schloss Dagstuhl, Germany (2020). doi:10.4230/LIPIcs.CCC.2020.34

[8] Dantchev, S., Galesi, N., Ghani, A., Martin, B.: Depth lower bounds in stabbing planes for combinatorial principles. In: Berenbrink, P., Monmege, B. (eds.) 39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 219, pp. 24:1–24:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). doi:10.4230/LIPIcs.STACS.2022.24

[9] Dash, S.: Exponential lower bounds on the lengths of some classes of branch-and-cut proofs. Mathematics of Operations Research **30**(3), 678–701 (2005). doi:10.1287/moor.1050.0151

[10] Dash, S.: On the complexity of cutting-plane proofs using split cuts. Operations Research Letters **38**(2), 109–114 (2010). doi:10.1016/j.orl.2009.10.010

[11] Derpich, I., Vera, J.R.: Improving the efficiency of the branch and bound algorithm for integer programming based on "flatness" information. European Journal of Operational Research **174**(1), 92–101 (2006). doi:10.1016/j.ejor.2005.02.051

[12] Dey, S.S., Dubey, Y., Molinaro, M.: Lower bounds on the size of general branch-and-bound trees. Mathematical Programming pp. 1–21 (2022). doi:10.1007/s10107-022-01781-z

[13] Fleming, N., Göös, M., Impagliazzo, R., Pitassi, T., Robere, R., Tan, L.Y., Wigderson, A.: On the power and limitations of branch and cut. In: Kabanets, V. (ed.) 36th Computational Complexity Conference (CCC 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 200, pp. 6:1–6:30. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). doi:10.4230/LIPIcs.CCC.2021.6

[14] Fleming, N., Pankratov, D., Pitassi, T., Robere, R.: Random $\theta(\log n)$-CNFs are hard for cutting planes. J. ACM **69**(3) (jun 2022). doi:10.1145/3486680

[15] Gläser, M., Pfetsch, M.E.: On computing small variable disjunction branch-and-bound trees. Mathematical Programming pp. 1–29 (2023). doi:10.1007/s10107-023-01968-y, to appear

[16] Grötschel, M., Lovász, L., Schrijver, A.: Geometric algorithms and combinatorial optimization. Springer, Berlin (1988)

[17] Haken, A., Cook, S.A.: An exponential lower bound for the size of monotone real circuits. Journal of Computer and System Sciences **58**(2), 326–335 (1999). doi:10.1006/jcss.1998.1617

[18] Hrubeš, P., Pudlák, P.: Random formulas, monotone circuits, and interpolation. In: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS). pp. 121–131. IEEE (2017). doi:10.1109/focs.2017.20

[19] Kaibel, V., Weltge, S.: Lower bounds on the sizes of integer programs without additional variables. Mathematical Programming **154**(1), 407–425 (2015). doi:10.1007/s10107-014-0855-0

[20] Krajiček, J.: Lower bounds to the size of constant-depth propositional proofs. The Journal of Symbolic Logic **59**(1), 73–86 (1994). doi:10.2307/2275250

[21] Krajíček, J.: Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. The Journal of Symbolic Logic **62**(2), 457–486 (1997). doi:10.2307/2275541

[22] Lenstra, H.W.: Integer programming with a fixed number of variables. Mathematics of Operations Research **8**(4), 538–548 (1983). doi:10.1287/moor.8.4.538

[23] Mahmoud, H., Chinneck, J.W.: Achieving MILP feasibility quickly using general disjunctions. Computers & Operations Research **40**(8), 2094–2102 (2013). doi:10.1016/j.cor.2013.03.001

[24] Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S.: Orbital branching. Mathematical Programming **126**, 147–178 (2011). doi:10.1007/s10107-009-0273-x

[25] Pudlák, P.: Lower bounds for resolution and cutting plane proofs and monotone computations. The Journal of Symbolic Logic **62**(3), 981–998 (1997). doi:10.2307/2275583

[26] Pudlák, P.: On reducibility and symmetry of disjoint NP pairs. Theoretical Computer Science **295**(1-3), 323–339 (2003). doi:10.1016/s0304-3975(02)00411-5

[27] Razborov, A.A.: Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. Izvestiya: mathematics **59**(1), 205 (1995). doi:10.1070/im1995v059n01abeh000009

[28] Wegener, I.: The complexity of Boolean functions. John Wiley & Sons, Inc. (1987)