APPROXIMATION LIMITATIONS OF PURE DYNAMIC PROGRAMMING

STASYS JUKNA[1] AND HANNES SEIWERT[2]

ABSTRACT. We prove the first, even super-polynomial, lower bounds on the size of tropical (min,+) and (max,+) circuits approximating given optimization problems. Many classical dynamic programming (DP) algorithms for optimization problems are pure in that they only use the basic min, max, + operations in their recursion equations. Tropical circuits constitute a rigorous mathematical model for this class of algorithms. An algorithmic consequence of our lower bounds for tropical circuits is that the approximation powers of pure DP algorithms and greedy algorithms are incomparable. That pure DP algorithms can hardly beat greedy in approximation, is long known. New in this consequence is that also the converse holds.

Keywords: dynamic programming, greedy algorithm, approximation, lower bounds

1. INTRODUCTION

A combinatorial optimization problem is specified by a finite set of *ground elements* and a family $\mathcal{F}$ of subsets of these elements, called *feasible solutions*. The problem itself then is, given an assignment of nonnegative real weights to the ground elements, to compute the minimum or the maximum weight of a feasible solution, the latter being the sum of weights of its elements.

The family $\mathcal{F}$ of feasible solutions itself can be described either explicitly, or as the set of 0-1 solutions of a system of linear inequalities (as in linear programming), or by other means. Important is only that $\mathcal{F}$ does *not* depend on the actual input weighting: the family $\mathcal{F}$ is the *same* for *all* arriving input weightings.

For example, in the MST problem (minimum weight spanning tree problem) on a given graph, feasible solutions are spanning trees of this graph (viewed as sets of their edges), and the problem is to compute the minimum weight of a spanning tree of this graph. In the assignment problem, feasible solutions are perfect matchings in a complete bipartite graph, etc.

Dynamic programming (DP) is a fundamental algorithmic paradigm for solving combinatorial optimization problems. Many classical DP algorithms are *pure* in that they only apply the *basic* operations $(\min, +)$ or $(\max, +)$ in their recursion equations. Note that these are the *only* operations used in the definitions of the optimization problems themselves.

Notable examples of pure DP algorithms for combinatorial optimization problems are the well-known Bellman–Ford–Moore shortest *s*-*t* path algorithm [4, 11, 22], the Floyd–Warshall all-pairs shortest paths algorithm [9, 29] (see Fig. 1), the Held–Karp traveling salesman algorithm [14], the Dreyfus–Levin–Wagner Steiner tree algorithm [6, 19]. The Viterbi $(\max, \times)$ DP algorithm [28] is also a pure $(\min, +)$ DP algorithm via the isomorphism $h : (0, 1] \to \mathbb{R}_+$ given by $h(x) = -\ln x$.

This work was funded by the DFG (German Research Foundation) grant JU 3105/1-1.

[1]Faculty of Mathematics and Computer Science, Vilnius University, Lithuania. Email: stjukna@gmail.com, homepage: http://www.thi.cs.uni-frankfurt.de/~jukna/.

[2]Institute of Computer Science, Goethe University Frankfurt, Frankfurt am Main, Germany. Email: seiwert@thi.cs.uni-frankfurt.de.

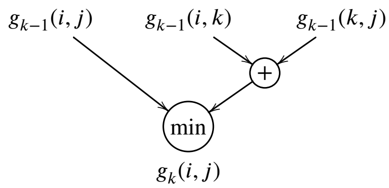

FIGURE 1. A fragment of a tropical $(\min,+)$ circuit of size $O(n^3)$ implementing the Floyd–Warshall DP algorithm for the all-pairs lightest paths on K_n problem. At the gate $g_k(i,j)$, the minimum weight of a path from i to j, which only uses nodes $1,\dots,k$ as inner nodes, is computed.

The main question we ask in this paper is: How many operations are necessary for pure DP algorithms to *approximate* a given combinatorial optimization problem within a given factor? That is, we are interested in proving *lower bounds* on the number of performed operations.

A natural mathematical model for pure $(\min,+)$ and $(\max,+)$ DP algorithms is that of tropical circuits. A *tropical* $(\min,+)$ *circuit* is a directed acyclic graph, whose each indegree-zero node holds either one of the input variables $x_1,\dots,x_n$ or a nonnegative real constant, and every other node (a gate) has indegree two and computes either the minimum or the sum of the values computed at its two predecessors. Tropical $(\max,+)$ circuits are defined similarly. The *size* of a circuit is the total number of its gates. Note that pure $(\min,+)$ and $(\max,+)$ DP algorithms are just special (recursively constructed) tropical circuits (see Fig. 1). So, lower bounds on the size of tropical circuits show limits of these pure DP algorithms.

In this paper, we prove the first non-trivial, even super-polynomial, lower bounds for *approximating* tropical circuits and, hence, also for approximating pure DP algorithms.

Recall that an algorithm *approximates* a given optimization problem f within a *factor* $r \geqslant 1$ (or *r-approximates* f) if for every input weighting x (a vector of n nonnegative real numbers), the output value of the algorithm lies:

- between $f(x)$ and $r \cdot f(x)$, in the case when f is a *minimization* problem;
- between $f(x)/r$ and $f(x)$, in the case when f is a *maximization* problem.

The factor r may depend on the length n of the inputs x, but not on the inputs x themselves. In both cases, the *smaller* the factor r is, the better is the approximation. In particular, factor $r = 1$ means that the problem is solved exactly.

One of our motivations for proving lower bounds on the number of operations performed by *approximating* pure DP algorithms is to compare their approximation power with that of the greedy algorithm; see Appendix A for what we mean by *the* greedy algorithm.

That the greedy algorithm can have much worse approximation behavior than pure DP algorithms is long known. Namely, there are many optimization problems easily solvable by pure DP algorithms using a small number of $(\min,+)$ or $(\max,+)$ operations, but the greedy algorithm cannot achieve any non-trivial approximation factor (smaller than the maximum number of elements in feasible solutions). Such are, for example, the maximum weight independent set in a tree, or the maximum weight simple path in a transitive tournament problem, and many other problems. To give a trivial example, note that the problem $f(x) = \max\{x_1, x_2 + \cdots + x_n\}$ can be solved (within factor $r = 1$) by a trivial pure $(\max,+)$ DP algorithm performing only $n-1$ operations, but the greedy algorithm cannot achieve any smaller than $r = n-1$ approximation factor for this problem (see Proposition A.1 in Appendix A). But what about the *converse* direction: can also pure DP algorithms have worse approximation behavior than greedy?

Apparently, the first indication that greedy can also beat pure DP was given by Jerrum and Snir [15]. They proved that every $(\min,+)$ circuit solving (exactly, within factor $r = 1$) the *directed* MST problem on n-vertex graphs (known also as the *arborescence* problem) requires $2^{\Omega(n)}$ gates. Since the family of feasible solutions of the arborescence problem is an intersection of two matroids, the greedy algorithm can approximate this problem within factor $r = 2$. This result was later improved in [17] by showing that also the *undirected* MST problem, which can already be solved by the greedy algorithm exactly, requires $(\min,+)$ circuits of size $2^{\Omega(\sqrt{n})}$ to be solved exactly.

But what if pure DP algorithms are only required to *approximate* a given optimization problem within some factor $r > 1$? Can greedy algorithms achieve smaller approximation factors than efficient pure DP algorithms? Our lower bounds on the size of approximating tropical circuits answer this question in the *affirmative*.

Below we summarize our main results. Since the approximation behaviors of tropical $(\min,+)$ and $(\max,+)$ circuits turned out to be completely different, we consider minimization and maximization problems separately.

2. Main results

Recall that a combinatorial optimization problem $f(x_1,\ldots,x_n)$ is specified by giving some family $\mathcal{F} \subseteq 2^{[n]}$ of feasible solutions. The problem itself is then, given an input weighting $x \in \mathbb{R}_+^n$, to compute either the minimum or the maximum weight $\sum_{i\in S} x_i$ of a feasible solution $S \in \mathcal{F}$. To indicate the total number n of ground elements, we will also write f_n instead of just f.

Minimization. The *boolean version* of a minimization problem f_n is the monotone boolean function which, given a set of ground elements, decides whether this set contains at least one feasible solution of f_n.

Result 1 (Boolean bound for $(\min,+)$ circuits; Theorem 4.3). If the boolean version of a minimization problem f_n requires monotone boolean $(\vee,\wedge)$ circuits of size $> t$, then no tropical $(\min,+)$ circuit of size $\leqslant t$ can approximate f_n within any finite factor $r = r(n) \geqslant 1$.

That is, if a tropical $(\min,+)$ circuit has fewer than t gates, then regardless of how large approximation factor r we will allow, there will be an input weighting on which the circuit makes an error: the computed value on this input will be either strictly smaller or more than r times larger than the optimal value.

Together with known lower bounds for monotone boolean circuits, Result 1 yields the same lower bounds for tropical $(\min,+)$ circuits approximating the corresponding minimization problems.

Take, for example the assignment problem: given a nonnegative weighting of the edges of the complete bipartite $n \times n$ graph, compute the minimum weight of a perfect matching. Jerrum and Snir [15] have proved that any $(\min,+)$ circuit solving this problem exactly (within the factor $r = 1$) must have $2^{\Omega(n)}$ gates. On the other hand, together with Razborov's monotone circuit lower bound for the logical permanent function [26], Result 1 implies that a polynomial in n number of gates is not sufficient to approximate this problem even when an *arbitrarily* large approximation factor is allowed: for *any* finite approximation factor $r = r(n) \geqslant 1$, at least $n^{\Omega(\log n)}$ gates are necessary to approximate the assignment problem within the factor r.

By combining the boolean bound (Result 1) with counting arguments, we show that the greedy algorithm can beat approximating pure $(\min,+)$ DP algorithms on some minimization problems.

Result 2 (Greedy can beat $(\min,+)$ circuits; Theorem 4.6). There are doubly-exponentially many in n minimization problems f_n such that the greedy algorithm solves f_n exactly, but any $(\min,+)$ circuit approximating f_n within any finite factor $r = r(n) \geqslant 1$ must have $2^{\Omega(n)}$ gates.

Our proof of Result 1 is fairly simple, but it only gives us an "absolute" lower bound on the number of gates, below which no tropical $(\min,+)$ circuit can approximate a given minimization problem within *any* factor. More interesting (and less simple), however, is the fact that, after an appropriate definition of the "semantic degree" of monotone boolean circuits (Section 7.1), also a *converse* of Result 1 holds:

the approximation power of tropical $(\min, +)$ circuits is *captured* (not only lower bounded) by the computational power of monotone boolean circuits of bounded semantic degree.

Result 3 (Converse of the boolean bound; Theorem 7.1). A minimization problem f can be approximated within a factor r by a tropical $(\min, +)$ circuit of size t if and only if the boolean version of f can be computed by a monotone boolean $(\vee, \wedge)$ circuit of size t and semantic degree at most r.

We prove this result in Section 7 using convexity arguments. Yet another consequence of these arguments is (see Remark 7) that, in order to show that the minimization problem on a family $\mathcal{F} \subseteq 2^{[n]}$ of feasible solutions can be r-approximated by a $(\min, +)$ circuit of size t, it is enough to design a monotone *arithmetic* $(+, \times)$ circuit of size $\leqslant t$ such that the polynomial computed by this circuit has the following two properties (where we, as customary, only consider monomials with nonzero coefficients):

(1) for every monomial $\prod_{i \in T} x_i^{d_i}$ there is a set $S \in \mathcal{F}$ with $S \subseteq T$;

(2) for every set $S \in \mathcal{F}$ there is a monomial $\prod_{i \in T} x_i^{d_i}$ with $T = S$ and all $d_i \leqslant r$.

That is, we can approximate minimization problems by designing monotone arithmetic circuits of bounded degree. This is a (rough) *upper* bound on the size of approximating $(\min, +)$ circuits in terms of *arithmetic* circuits. Result 3 gives a *tight* bound, but in terms of *boolean* circuits.

Maximization. It turned out that not only the approximation behaviors of $(\min, +)$ and $(\max, +)$ circuits are different (approximation factors may be unbounded in the former model, while they are always bounded in the latter model), but also the task of proving lower bounds for approximating $(\max, +)$ circuits is by far more difficult than that for $(\min, +)$ circuits.

The point is that for approximating $(\max, +)$ circuits, even Shannon type *counting* arguments fail (see Section 5.1). In particular, there are doubly-exponentially many in n maximization problems f_n such that $(\max, +)$ circuits require $2^{\Omega(n)}$ gates to solve any of them exactly (within the factor $r = 1$), but *one single* $(\max, +)$ circuit of size $O(n^2)$ approximates each of these problems within a just slightly larger than 1 factor $r = 1 + o(1)$ (Proposition 5.2). Such a jump in circuit size occurs also on *random* maximization problems (Proposition 5.3). Moreover, there are also *explicit* maximization problems f_n that require $(\max, +)$ circuits of size at least $2^{n/4}$ to solve them exactly (within factor $r = 1$), but can be approximated within the factor $r = 2$ by using only n gates (Theorem 5.4).

Being warned by these facts, we go much deeper (than in the case of minimization) into the structure of approximating $(\max, +)$ circuits and prove a general "rectangle lower bound" for them.

Let $\mathcal{F}$ be a family of feasible solutions. A *rectangle* is a family of sets specified by a pair $\mathcal{A}, \mathcal{B}$ of families satisfying $A \cap B = \emptyset$ for all $A \in \mathcal{A}$ and $B \in \mathcal{B}$. The rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ itself consists of all sets $A \cup B$ with $A \in \mathcal{A}$ and $B \in \mathcal{B}$. The rectangle $\mathcal{R}$ *lies below* $\mathcal{F}$ if every set of $\mathcal{R}$ is contained in at least one set of $\mathcal{F}$. Given an approximation factor $r \geqslant 1$, we say that a set $F \in \mathcal{F}$ *appears r-balanced* in the rectangle $\mathcal{R}$ if there are sets $A \in \mathcal{A}$ and $B \in \mathcal{B}$ such that F shares $\geqslant |F|/r$ elements with $A \cup B$, and $\geqslant |F|/3r$ elements with both A and B.

Result 4 (Rectangle bound; special case of Theorem 5.9). If in any rectangle lying below $\mathcal{F}$, at most a $1/t$ portion of sets of $\mathcal{F}$ appear r-balanced, then every $(\max, +)$ circuit approximating the maximization problem on $\mathcal{F}$ within the factor r must have at least t gates.

Using the rectangle bound, we show that already a slight decrease of the allowed approximation factors r can make tractable problems intractable, and that this happens for arbitrarily large factors r. In the following result formalizing this phenomenon, $\epsilon > 0$ is an arbitrarily small constant.

Result 5 (Factor hierarchy theorem; Theorem 5.12). For every prime power m and integer $1 \leqslant d \leqslant m$, there is an explicit maximization problem f_n on $n = m^2$ ground elements which can be approximated within the factor $r = m/d$ by a $(\max, +)$ circuit of size $3n$, but any $(\max, +)$ circuit approximating f_n within the factor $(1 - \epsilon)r$ must have at least $n^{\epsilon d/4}$ gates.

Finally, using the rectangle bound, we show that there are explicit maximization problems f_n such that $(\max, +)$ circuits of polynomial in n size cannot achieve even an exponentially larger factor than the factor achieved by the greedy algorithm on f_n.

Result 6 (Greedy can beat $(\max, +)$ circuits; Theorem 5.13). For every integer $r \geqslant 6$, there are explicit maximization problems f_n such that the greedy algorithm approximates f_n within the factor r, but every $(\max, +)$ circuit approximating f_n within the factor $2^r/9$ must have $2^{n^{\Omega(1)}}$ gates.

Families of feasible solutions of the maximization problems f_n in Result 5 are particular combinatorial designs, while those in Result 6 are families of perfect matchings in r-partite r-uniform hypergraphs.

The algorithmic message. As we already mentioned above, it was long known that for some combinatorial optimization problems, greedy algorithms can have much worse approximation behavior than pure DP algorithms. Thus, Results 2 and 6 imply that the approximation powers of greedy and pure DP algorithms are *incomparable*: on some optimization problems, pure DP algorithms can also have much *worse* approximation behavior than greedy.

Why "only" pure DP?. In this paper, we only consider pure $(\min, +)$ and $(\max, +)$ DP algorithms. Non-pure DP algorithms may use other arithmetic operations, rounding, as well as very powerful operations like conditional branchings (via if-then-else constraints), argmin, argmax, etc. The presence of such operations makes the corresponding circuit models no longer amenable for analysis using known mathematical tools. In particular, such DP algorithms have the full power of arithmetic circuits as well as of unrestricted boolean $(\vee, \wedge, \neg)$ circuits (for example, $\neg x$ is a simple conditional branching operation if $x = 0$ then 1 else 0). Let us stress that our goal is to prove (unconditional) *lower bounds*. In the context of this task, even proving lower bounds for exactly solving $(\min, +, -)$ circuits (tropical circuits with subtraction operation allowed), remains a challenge (see Section 8.4).

Organization. In Section 3, we recall the concept of sets *produced* by circuits, and show that when approximating combinatorial optimization problems, we can safely assume that tropical circuits are *constant-free*, that is, contain no constants as inputs (Lemma 3.2). Sections 4 to 7 are devoted to the proofs of our main results. Results 1 and 2 are proved in Section 4, and Results 4 to 6 are proved in Section 5. In Section 6, we use convexity arguments (Farkas' lemma) to give a *tight* structural connection between the sets of feasible solutions of optimization problems to be *approximated* and the sets of feasible solutions *produced* by approximating tropical circuits. In Section 7, we prove the *converse* of our boolean lower bound for approximating $(\min, +)$ circuits (Result 3). The concluding section (Section 8) contains some open problems. In Appendix A, we recall greedy algorithms. In Appendix B, we exhibit an exponential (almost maximal possible) decrease in the size of $(\max, +)$ circuits on *explicit* maximization problems when going from the approximation factor $r = 1$ (exact solution) to factor $r = 2$.

Notation. Through the paper, $\mathbb{N} = \{0, 1, 2, \dots\}$ will denote the set of all nonnegative integers, $[n] = \{1, 2, \dots, n\}$ the set of the first n positive integers, $\mathbb{R}_+$ the set of all nonnegative real numbers, and 2^E the family of all subsets of a set E. Also, $\vec{0}$ will denote the all-0 vector, $\vec{e}_i$ will denote the 0-1 vector with exactly one 1 in the ith position. For sets $A, B \subseteq \mathbb{R}^n$ of vectors, their *Minkowski sum* (or *sumset*) is the set of vectors $A + B = \{a + b \colon a \in A, b \in B\} \subseteq \mathbb{R}^n$, where $a + b = (a_1 + b_1, \dots, a_n + b_n)$ is the componentwise sum of vector a and b. That is, we add every vector of B to every vector of A. For a real vector $a = (a_1, \dots, a_n)$ and a scalar $\lambda \in \mathbb{R}$, $\lambda \cdot a$ stands for the vector $(\lambda a_1, \dots, \lambda a_n)$. If $A \subseteq \mathbb{R}^n$ is a set of vectors, then $\lambda \cdot A$ stands for the set of vectors $\{\lambda \cdot a : a \in A\}$. The *support* of vector a is the set $S_a = \{i : a_i \neq 0\}$ of its nonzero positions.

As customary, a family $\mathcal{F}$ of sets is an *antichain* if none of its sets is a proper subset of another set of $\mathcal{F}$. For two vectors $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$ we write $a \leqslant b$ if $a_i \leqslant b_i$ holds for all positions $i = 1, \ldots, n$. A set A of vectors is an *antichain* if $a \leqslant a'$ holds for no two distinct vectors $a \neq a' \in A$. The *characteristic vector* of a set $S \subseteq [n]$ is the vector $a \in \{0,1\}^n$ with $a_i = 1$ if and only if $i \in S$.

3. Preliminaries

Every finite set $A \subset \mathbb{N}^n$ of *feasible solutions* defines a *discrete optimization problem* of the form $f(x) = \min_{a\in A}\langle a, x\rangle$ or of the form $f(x) = \max_{a\in A}\langle a, x\rangle$, where here and in what follows, $\langle a, x\rangle = a_1x_1 + \cdots + a_nx_n$ stands for the scalar product of vectors $a = (a_1, \ldots, a_n)$ and $x = (x_1, \ldots, x_n)$.

We will refer to such problems as *problems defined by* A, or as *problems on* A. Such a problem is a 0-1 *optimization problem* if the set $A \subseteq \{0,1\}^n$ of feasible solutions consists of only 0-1 vectors. These latter problems are exactly what we called "combinatorial optimization" problems on *families* $\mathcal{F} \subseteq 2^{[n]}$ of feasible solutions, where $\mathcal{F}$ consists of all sets $S_a = \{i : a_i = 1\}$ for vectors $a \in A$.

To avoid trivialities, we will throughout assume that the all-0 vector (or the empty set) is *not* a feasible solution, that is, we will always assume that $\vec{0} \notin A$ and $\emptyset \notin \mathcal{F}$.

3.1. Circuits over semirings.

Recall that a (commutative) *semiring* is a set R closed under two associative and commutative binary operations "addition" ($\oplus$) and "multiplication" ($\otimes$), where multiplication distributes over addition: $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$. That is, in a semiring, we can "add" and "multiply" elements, but neither "subtraction" nor "division" are necessarily possible. Besides of being commutative, we will assume that the semiring contains a multiplicative identity element $\mathsf{1}$ with $\mathsf{1} \otimes x = x \otimes \mathsf{1} = x$.

A *circuit* over a semiring R is a directed acyclic graph; parallel edges joining the same pair of nodes are allowed. Each indegree-zero node (an *input* node) holds either one of the variables $x_1, \ldots, x_n$ or a semiring element. Every other node, a *gate*, has indegree two and performs one of the semiring operations. One of the gates is designated as the output gate. The *size* of a circuit is the total number of gates in it. A circuit is *constant-free* if it has no semiring elements as inputs.

Since in any semiring $(R, \oplus, \otimes)$, multiplication distributes over addition, each circuit Φ over R computes (at the output gate) some polynomial

$$\Phi(x_1, \ldots, x_n) = \sum_{b\in B} \lambda_b X^b \quad \text{with} \quad X^b = \prod_{i=1}^{n} x_i^{b_i} \tag{1}$$

over R in a natural way, where $B \subset \mathbb{N}^n$ is some set of *exponent vectors*, and x_i^k stands for $x_i \otimes x_i \otimes \cdots \otimes x_i$ k-times. Since we only consider semirings with multiplicative identity, coefficients $\lambda_b \in R$ are semiring elements. To see why this assumption is necessary, consider the semiring $(R, +, \times)$, where R is the set of all positive even integers. Then the coefficient 3 of the monomial x in the polynomial $x + x + x$ is not a semiring element.

In this paper, we will mainly consider circuits over three commutative and idempotent semirings $(R, \oplus, \otimes)$. In the *boolean* $(\vee, \wedge)$ semiring, we have $R = \{0, 1\}$, $x \oplus y := x \vee y$ and $x \otimes y := x \wedge y$. In the *tropical* $(\min, +)$ semiring, we have $R = \mathbb{R}_+$, $x \oplus y := \min(x, y)$ and $x \otimes y := x + y$. Similarly, in the *tropical* $(\max, +)$ semiring, we have $R = \mathbb{R}_+$, $x \oplus y := \max(x, y)$ and $x \otimes y := x + y$. The multiplicative identity element in the boolean semiring is $\mathsf{1} = 1$, and is $\mathsf{1} = 0$ in both tropical semirings.

Over the boolean semiring, the polynomial Eq. (1) computes the monotone boolean function

$$\Phi(x) = \bigvee_{b\in B} \bigwedge_{i\,:\,b_i\neq 0} x_i\,.$$

Over the tropical semirings, every monomial $X^b = \prod_{i=1}^n x_i^{b_i}$ turns into the scalar product $X^b = \sum_{i=1}^n b_i x_i = \langle b, x\rangle$ of vectors b and x. Hence, the polynomial Eq. (1) solves one of the two optimization problems with linear objective functions:

$$\Phi(x) = \min_{b\in B}\, \langle b, x\rangle + \lambda_b \ \text{ or } \ \Phi(x) = \max_{b\in B}\, \langle b, x\rangle + \lambda_b\,. \tag{2}$$

Note that if a tropical circuit Φ is constant-free, then $\lambda_b = 0$ holds for all $b \in B$.

3.2. **Sets of vectors produced by circuits.** A simple, but important in our later analysis, observation is that every circuit of n variables over a semiring $(R, \oplus, \otimes)$ not only *computes* some polynomial over R, but also *produces* (purely syntactically) a finite set of vectors in $\mathbb{N}^n$ in a natural way.

At each input node holding a semiring element, the same set $\{\vec{0}\}$ is produced. At an input node holding a variable x_i, the set $\{\vec{e}_i\}$ is produced. At an "addition" ($\oplus$) gate, the union of sets produced at its inputs is produced. Finally, at a "multiplication" ($\otimes$) gate, the Minkowski sum of sets produced at its inputs is produced. The set produced by the entire circuit is the set produced at its output gate.

It is clear that the same circuit Φ with only "addition" ($\oplus$) and "multiplication" ($\otimes$) gates may *compute* different functions over different semirings. It is, however, important to note that the set $B \subset \mathbb{N}^n$ of vectors *produced* by Φ is always the same—it only depends on the circuit itself, not on the underlying semiring.

On the other hand, up to coefficients, the polynomial function *computed* by the circuit Φ is determined by the set of *produced* vectors.

Proposition 3.1. *If $B \subset \mathbb{N}^n$ is the set of vectors produced by a circuit Φ over a semiring R, then Φ computes some polynomial over R whose set of exponent vectors coincides with B.*

Proof. Simple induction on the size of a circuit Φ. Let $B \subset \mathbb{N}^n$ be the set of vectors produced, and $f : R^n \to R$ the polynomial function computed by Φ.

If the circuit Φ consists of a single input node holding a semiring element $\lambda \in R$, then $f(x) = \lambda$ is a constant polynomial with a single exponent vector $\vec{0}$. If Φ consists of a single input node holding a variable x_i, then $f(x) = x_i$ is a degree-1 polynomial with the single exponent vector $\vec{e}_i$.

Now, the set of exponent vectors of a sum of two polynomials is just the union of the sets of exponent vectors of these polynomials. Finally, when multiplying two polynomials, we multiply each monomial of the first polynomial with all monomials of the second polynomial. The exponent vector of a product of two monomials is the sum of exponent vectors of these monomials. □

Remark 1. In general, Proposition 3.1 has no converse, even for constant-free circuits: if a circuit Φ computes some polynomial f, then Φ does *not* need to produce the set of exponent vectors of f; a simple example for tropical circuits is given in Fig. 2. Monotone arithmetic $(+, \times)$ circuits, that is, circuits over the arithmetic semiring $(\mathbb{R}_+, +, \times)$, are here an exception: for them, also the converse of Proposition 3.1 holds. Namely, if such a circuit *computes* a polynomial f, then the set of vectors *produced* by the circuit is exactly the set of exponent vectors of this polynomial f. This holds because, if two arithmetic polynomials coincide on sufficiently many (with respect to the number of variables and the degrees of these polynomials) inputs, then these polynomials must *syntactically* coincide (even up to coefficients).

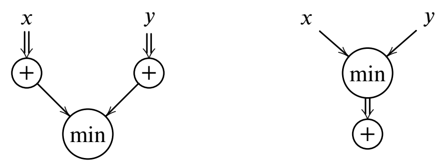

FIGURE 2. Two constant-free $(\min, +)$ circuits solving the minimization problem $f(x, y) = \min\{2x, 2y\}$ whose set of feasible solutions is $A = \{(2, 0), (0, 2)\}$. The first circuit produces the set A itself, whereas the second saves one gate by producing a *different* set $B = \{(2, 0), (1, 1), (0, 2)\}$. Here $\Downarrow$ stands for two parallel edges.

3.3. **Eliminating constant inputs.** Recall that an optimization problem on a set $A \subset \mathbb{N}^n$ of feasible solutions is of the form $f(x) = \min_{a\in A}\langle a, x\rangle$ or of the form $f(x) = \max_{a\in A}\langle a, x\rangle$. To avoid trivialities, we always assume that $A \neq \emptyset$ and $\vec{0} \notin A$.

These problems are "constant-free" in that they are *completely* specified by their sets A of feasible solutions: there are no additional constant terms. In contrast, since tropical circuits can have constant inputs, the optimization problems actually solved by such circuits (exactly) may be not constant-free: they may have additional constant terms; see Eq. (2).

However, as the following lemma shows, when dealing with tropical circuits approximating (constant-free) optimization problems, we can safely restrict ourselves to constant-free circuits. Recall that constant-free circuits only use the variables $x_1, \dots, x_n$ as inputs.

Lemma 3.2 (Eliminating constant inputs)**.** *If an optimization problem on a set $A \subset \mathbb{N}^n$ can be r-approximated by a tropical circuit of size t, then this problem can also be r-approximated by a constant-free tropical circuit of size t.*

Proof. Let Φ be a tropical $(\max, +)$ or $(\min, +)$ circuit, and $B \subset \mathbb{N}^n$ the set of vectors produced by Φ. By Proposition 3.1, the circuit computes the maximum or the minimum, over all vectors $b \in B$, of linear functions $\langle b, x\rangle + \lambda_b$, where $\lambda_b \in \mathbb{R}_+$ are some constants.

We obtain the *constant-free version* Φ^* of Φ as follows. First, replace every constant input by 0. Then eliminate zeros by repeatedly replacing gates $u + 0$ and $\max(u, 0)$ by the gate u, and a gate $\min(u, 0)$ by an input node holding 0. Since $\Phi(x) \neq 0$ must hold for at least one $x \in \mathbb{R}^n_+$, the constant 0 input also disappears at the end of this replacement. Since constant inputs can only affect the additive constant terms λ_b, the constant-free version Φ^* computes the maximum or the minimum of linear functions $\langle b, x\rangle$ *without* any constant terms. Our goal is to show that Φ^* still r-approximates our optimization problem f on the set A.

Case 1: Φ is a $(\max, +)$ circuit; hence, $f(x) = \max_{a\in A}\langle a, x\rangle$. In this case, we have that $\Phi^*(x) = \max_{b\in B}\langle b, x\rangle$, and $\Phi(x) = \max_{b\in B}\langle b, x\rangle + \lambda_b$ for some nonnegative constants $\lambda_b \in \mathbb{R}_+$. Since Φ approximates f, $\Phi(x) \leqslant f(x)$ must hold for all input weightings $x \in \mathbb{R}^n_+$. Taking $x = \vec{0}$, we obtain $\Phi(\vec{0}) \leqslant f(\vec{0}) = 0$ and, hence, $\lambda_b = 0$ for all $b \in B$. Thus, in the case of maximization, the constant-free version of the circuit solves just the same problem as the original circuit, and we are done.

Case 2: Φ is a $(\min, +)$ circuit; hence, $f(x) = \min_{a\in A}\langle a, x\rangle$. Since Φ r-approximates f, we know that the inequalities $f(x) \leqslant \Phi(x) \leqslant r \cdot f(x)$ must hold for all $x \in \mathbb{R}^n_+$. We have to show that Φ^* also satisfies these inequalities. We know that $\Phi^*(x) = \min_{b\in B}\langle b, x\rangle$, and $\Phi(x) = \min_{b\in B}\langle b, x\rangle + \lambda_b$ for some nonnegative constants $\lambda_b \in \mathbb{R}_+$.

Since the constants λ_b are nonnegative, we clearly have $\Phi^*(x) \leqslant \Phi(x)$ and, hence, also $\Phi^*(x) \leqslant r \cdot f(x)$ for all $x \in \mathbb{R}_+^n$. So, it remains to show that $\Phi^*(x) \geqslant f(x)$ holds for all $x \in \mathbb{R}_+^n$, as well. We know that $\Phi(x) \geqslant f(x)$ holds for all $x \in \mathbb{R}_+^n$.

Assume contrariwise that $\Phi^*(x_0) < f(x_0)$ holds for some input weighting $x_0 \in \mathbb{R}_+^n$. Then the difference $d = f(x_0) - \Phi^*(x_0)$ is positive. We also know that $\lambda := \max_{b\in B} \lambda_b$ is positive, for otherwise, there would be nothing to prove. So, take the constant $c := 2\lambda/d > 0$, and consider the input weighting $z := c \cdot x_0$. Since $\Phi^*(x_0) = f(x_0) - d$, and since $\Phi(x) \leqslant \Phi^*(x) + \lambda$ holds for all weightings $x \in \mathbb{R}_+^n$, the desired contradiction follows:

$$\begin{aligned}\Phi(z) &= \Phi(c \cdot x_0) \leqslant \Phi^*(c \cdot x_0) + \lambda = c \cdot \Phi^*(x_0) + \lambda = c \cdot [f(x_0) - d] + \lambda \\ &= c \cdot f(x_0) - c \cdot d + \lambda = f(c \cdot x_0) - \lambda = f(z) - \lambda < f(z)\,.\end{aligned}$$

□

4. Approximation limitations of (min,+) circuits

In this section, we first prove a general "boolean bound" for approximating $(\min, +)$ circuits: if the boolean (decision) version of a minimization problem requires monotone boolean $(\vee, \wedge)$ circuits of size at least t, then no $(\min, +)$ circuit of size $< t$ can approximate the problem within any finite factor (Theorem 4.3). Together with known lower bounds on the monotone boolean circuit complexity, this gives us explicit minimization problems which are hard to approximate by $(\min, +)$ circuits and, hence, by pure DP algorithms; three selected examples are given in Section 4.2. Then, in Section 4.3, we combine the boolean bound (Theorem 4.3) with counting arguments to show that greedy algorithms can "hardly" beat pure DP algorithms: there exist many minimization problems solvable by the greedy algorithm exactly, while polynomial-size $(\min, +)$ circuits cannot approximate any of them within any finite factor.

4.1. **The boolean bound for approximating** $(\min, +)$ **circuits.** Recall that the *support* of a vector $a \in \mathbb{N}^n$ is the set $S_a = \{i : a_i \neq 0\}$ of its nonzero positions. Every finite set $A \subset \mathbb{N}^n$ of vectors *defines* the monotone boolean function

$$f_A(x) = \bigvee_{a\in A} \bigwedge_{i\in S_a} x_i\,.$$

Note that, for every input $x \in \{0, 1\}^n$, we have

$$f_A(x) = 1 \text{ if and only if } S_x \supseteq S_a \text{ for some } a \in A. \tag{3}$$

For example, if A is the set of characteristic 0-1 vectors of perfect matchings in $K_{m,m}$, then f accepts a subgraph G of $K_{m,m}$ if and only if G contains a perfect matching.

Two sets $A, B \subseteq \mathbb{N}^n$ are *similar* if the support of every vector $b \in B$ contains the support of at least one vector $a \in A$, and vice versa. That is, A and B are similar if and only if

$$\forall b \in B\ \exists a \in A :\ S_b \supseteq S_a \text{ and } \forall a \in A\ \exists b \in B :\ S_a \supseteq S_b. \tag{4}$$

Observation Eq. (3) immediately yields the following.

Proposition 4.1. *Two sets of vectors define the same boolean function if and only if these sets are similar.*

The main connection between approximating $(\min, +)$ circuits and monotone boolean circuits is given by the following lemma. The *boolean version* of a constant-free tropical $(\min, +)$ circuit is the monotone boolean $(\vee, \wedge)$ circuit obtained by replacing each min-gate by an $\vee$-gate, and each $+$-gate by an $\wedge$-gate.

Lemma 4.2. *If a constant-free* $(\min,+)$ *circuit* Φ *approximates the minimization problem on a set* $A \subset \mathbb{N}^n$ *within a finite factor* $r = r(n) \geqslant 1$*, then the boolean version of* Φ *computes the boolean function defined by* A.

Proof. Let $B \subset \mathbb{N}^n$ be the set of vectors produced by Φ. Since the circuit Φ is constant-free, it solves the minimization problem $\Phi(x) = \min_{b\in B}\langle b, x\rangle$ defined by this set B. The minimization problem on A is $f(x) = \min_{a\in A}\langle a, x\rangle$. We know that $f(x) \leqslant \Phi(x) \leqslant r \cdot f(x)$ must hold for all input weightings $x \in \mathbb{R}^n_+$. The boolean version ϕ of Φ also produces the same set B. By Proposition 4.1, it remains to show that the set B is similar to A; see Eq. (4).

For the sake of contradiction, suppose first that there is a vector $b \in B$ such that $S_a \setminus S_b \neq \emptyset$ holds for all vectors $a \in A$. Consider the assignment $x \in \{0, 1\}^n$ of weights such that $x_i = 0$ for $i \in S_b$, and $x_i = 1$ for $i \notin S_b$. On this weighting, we have $\Phi(x) \leqslant \langle b, x\rangle = 0$. But since every vector $a \in A$ has a position $i \notin S_b$ with $a_i \neq 0$, $\langle a, x\rangle \geqslant 1$ holds for all $a \in A$ and, hence, also $f(x) \geqslant 1$, contradicting the inequality $f(x) \leqslant \Phi(x)$.

Now suppose that there is a vector $a \in A$ such that $S_b \setminus S_a \neq \emptyset$ holds for all vectors $b \in B$. Let $M = \max\{\langle a, a\rangle : a \in A\}$, and consider the weighting $x \in \{1, rM + 1\}^n$ such that $x_i = 1$ for all $i \in S_a$ and $x_i = rM + 1$ for all $i \notin S_a$ (note that $rM + 1$ is a finite number, because both the approximation factor r and the set A are finite. Then $f(x) \leqslant \langle a, x\rangle = \langle a, a\rangle \leqslant M$. But since every vector $b \in B$ has a position $i \notin S_a$ such that $b_i \geqslant 1$, we have $\Phi(x) \geqslant rM + 1 > r \cdot f(x)$, contradicting the inequality $\Phi(x) \leqslant r \cdot f(x)$. □

For a set $A \subseteq \mathbb{N}^n$ of vectors, let $\mathsf{Bool}(A)$ denote the minimum size of a monotone boolean $(\vee, \wedge)$ circuit computing the boolean function f_A defined by A. Let also $\mathsf{Min}_r(A)$ denote the minimum size of a tropical $(\min,+)$ circuit approximating the minimization problem on A within the factor r.

Theorem 4.3 (Boolean bound)**.** *For every finite set* $A \subset \mathbb{N}^n$ *and every finite factor* $r = r(n) \geqslant 1$*, we have* $\mathsf{Min}_r(A) \geqslant \mathsf{Bool}(A)$.

Proof. Take a $(\min,+)$ circuit Φ of size $t = \mathsf{Min}_r(A)$ approximating the minimization problem on A within the factor r, and let $B \subset \mathbb{N}^n$ be the set of vectors produced by Φ. By Lemma 3.2, we can assume that the circuit Φ is constant-free. Hence, by Lemma 4.2, the boolean version ϕ of Φ (which has the same size) computes the boolean function defined by the set A, as desired. □

Remark 2. Note that Theorem 4.3 does *not* exclude that, using *more* than $\mathsf{Bool}(A)$ gates, $(\min,+)$ circuits could achieve finite (and even small) approximation factors. The boolean bound $\mathsf{Bool}(A)$ is just an "absolute" lower bound below which no approximation is possible at all.

Remark 3. The proof of Theorem 4.3 is so direct and elementary, because it totally ignores the given approximation factor r: it only must be finite and, hence, can be used in input weightings to fool too small $(\min,+)$ circuits. Using more involved arguments (based on Farkas' lemma), we will show in Section 7 (Theorem 7.1) that, under an appropriate definition of the "semantic degree" of monotone boolean circuits, Theorem 4.3 has also a *converse*: a minimization problem can be approximated within a factor r by a tropical $(\min,+)$ circuit of size t if and only if the boolean version of this problem can be computed by a monotone boolean $(\vee, \wedge)$ circuit of size t and semantic degree at most r. Thus, the *approximation* power of tropical $(\min,+)$ circuits is *captured*, not only lower bounded, by the *computational* power of monotone boolean circuits.

4.2. **Explicit lower bounds.** Together with lower bounds on the monotone boolean circuit complexity, the boolean bound (Theorem 4.3) immediately yields the same lower bounds on the size of approximating $(\min,+)$ circuits. Let us mention some examples.

In the *lightest triangle* problem, we are given an assignment of nonnegative weights to the edges of K_n, and the goal is to compute the minimum weight of a triangle.

Corollary 4.1. *The lightest triangle problem in K_n can be solved by a* $(\min,+)$ *circuit using only n^3 gates, but no* $(\min,+)$ *circuit with $n^{3-\Omega(1)}$ gates can approximate this problem within any finite factor.*

Proof. Since we only have $\binom{n}{3}$ triangles, a trivial $(\min,+)$ circuit of size at most n^3 (taking the minimum over all triangles) solves this problem exactly. On the other hand, it is known ([1, Lemma 3.14]) that the decision version of this problem requires monotone boolean circuits with $\Omega(n^3/\log^4 n)$ gates. Theorem 4.3 gives the same lower bound for approximating $(\min,+)$ circuits. □

Recall that the *n-assignment problem* is: given an assignment of nonnegative real weights to the edges of the complete bipartite $n \times n$ graph, compute the minimum weight of a perfect matching in this graph. The corresponding family of feasible solutions is here the family of all perfect matchings, viewed as sets of their edges.

Corollary 4.2. *Every* $(\min,+)$ *circuit approximating the n-assignment problem within any finite factor must have at least $n^{\Omega(\log n)}$ gates.*

Proof. The boolean function defined by the family of feasible solutions of the assignment problem is the boolean permanent function which, as proved by Razborov [26], requires monotone boolean circuits of size $n^{\Omega(\log n)}$. □

Let n be a prime power, and $1 \leqslant d \leqslant n$ an integer. The *polynomial* (n,d)*-design* is the family of all $|\mathcal{F}| = n^d$ n-element subsets $\{(a,p(a)) : a \in \mathrm{GF}(n)\}$ of the grid $\mathrm{GF}(n) \times \mathrm{GF}(n)$, where $p = p(x)$ ranges over all n^d univariate polynomials of degree at most $d-1$ over $\mathrm{GF}(n)$.

Corollary 4.3. *If $d \leqslant (n/4\ln n)^{1/2}$, then every* $(\min,+)$ *circuit approximating the minimization problem on the polynomial (n,d)-design within any finite factor must have at least $n^{\Omega(d)}$ gates.*

Proof. By (numerically) improving the earlier result of Andreev [2], Alon and Boppana [1] have shown that, at least for such values of d, any monotone boolean circuit computing the boolean function defined by the corresponding family of feasible solutions requires $n^{\Omega(d)}$ gates. □

4.3. **Greedy can beat approximating (min,+) circuits.** Our goal now is to show that there *exist* many (combinatorial) minimization problems which are solvable by the greedy algorithm exactly (within the factor $r = 1$), but no $(\min,+)$ circuit with a polynomial in the number n of ground elements number of gates can approximate any of these problems within *any* finite factor $r = r(n)$.

We identify matroids with their families of bases. Under this proviso, a family $\mathcal{F}$ is a matroid if and only if $\mathcal{F}$ is uniform (all sets have the same cardinality) and the *basis exchange axiom* holds: if $A \neq B \in \mathcal{F}$, then for every $a \in A \setminus B$ there is a $b \in B \setminus A$ such that the set $(A \setminus \{a\}) \cup \{b\}$ belongs to $\mathcal{F}$.

It is well known (see, for example, [23, Theorem 1.8.4]) that an optimization problem on an antichain $\mathcal{F}$ can be solved by the greedy algorithm exactly if and only if $\mathcal{F}$ is a matroid. This fact is usually called the Rado–Edmonds theorem [25, 7]. In contrast, we will now show that most matroids require $(\min,+)$ circuits of exponential size to be even only approximated within any finite factor. We will do this by counting, so we need a lower bound on the number of matroids.

The following simple construction of matroids was implicit in several papers, starting from those of Piff and Welsh [24], and Knuth [18], and was made explicit by Bansal, Pendavingh and Van der Pol [3, Lemma 8]. Let $\binom{[n]}{m}$ denote the family of all m-element subsets of $[n] = \{1,\dots,n\}$. The Hamming distance between two sets A and B is $\mathrm{dist}(A,B) = |A \setminus B| + |B \setminus A|$. A family $\mathcal{H}$ is *separated* if $\mathrm{dist}(A,B) > 2$ holds for all $A \neq B \in \mathcal{H}$.

Proposition 4.4. *If $\mathcal{H} \subseteq \binom{[n]}{m}$ is separated, then $\mathcal{F} = \binom{[n]}{m} \setminus \mathcal{H}$ is a matroid.*

Proof. Suppose contrariwise that $\mathcal{F}$ is not a matroid. Since the family $\mathcal{F}$ is uniform, there must be two sets $A \neq B \in \mathcal{F}$ violating the basis exchange axiom: there is an $a \in A \setminus B$ such that $(A \setminus \{a\}) \cup \{b\} \notin \mathcal{F}$ for all $b \in B$. Observe that $B \setminus A$ must have at least two elements: held $B \setminus A = \{b\}$ then, since both A and B have the same cardinality, the set $(A \setminus \{a\}) \cup \{b\}$ would coincide with B and, hence, would belong to $\mathcal{F}$. So, take $b \neq c \in B \setminus A$ and consider the sets $S = (A \setminus \{a\}) \cup \{b\}$ and $T = (A \setminus \{a\}) \cup \{c\}$. Since the basis exchange axiom fails for A and B, neither S nor T can belong to $\mathcal{F}$; hence, *both* sets S and T belong to the family $\binom{[n]}{m} \setminus \mathcal{F} = \mathcal{H}$. But $\mathrm{dist}(S, T) = |\{b, c\}| = 2$, a contradiction with the family $\mathcal{H}$ being separated. □

Proposition 4.5. *There are $2^{\binom{n}{m}/n}$ matroids $\mathcal{F} \subseteq \binom{[n]}{m}$ such that $\binom{[n]}{m} \setminus \mathcal{F}$ is separated.*

Proof. Since subfamilies of separated families are also separated, it is enough, by Proposition 4.4 to show that a separated family $\mathcal{H} \subseteq \binom{[n]}{m}$ of size $|\mathcal{H}| \geqslant \binom{n}{m}/n$ exists.

The following amazingly simple argument was suggested by Graham and Sloane [13]. For $l \in \{0, 1, \ldots, n-1\}$, let $\mathcal{H}_l$ be the family of all sets $S \in \binom{[n]}{m}$ such that $\sum_{i \in S} i = l \mod n$. We claim that each such family $\mathcal{H}_l$ is separated. Suppose contrariwise that $\mathrm{dist}(S, T) = 2$ holds for some two sets $S \neq T$ of $\mathcal{H}_l$. Then $S = A \cup \{s\}$ and $T = A \cup \{t\}$ for some $(m-1)$-element set A, and $s \neq t$ are distinct numbers in $[n] \setminus A$. But then for $a = \sum_{i \in A} i$, we have $a + s = l \mod n$ and $a + t = l \mod n$, which is impossible because both numbers s and t are at most n. Thus, every family $\mathcal{H}_l$ is separated. Since there are only n such families, and they exhaust the entire family $\binom{[n]}{m}$, there must be an l for which $|\mathcal{H}_l| \geqslant \binom{n}{m}/n$ holds. □

Theorem 4.6. *There are at least $2^{2^n/n^3}$ matroids $\mathcal{F} \subseteq 2^{[n]}$ such that every $(\min, +)$ circuit approximating the minimization problem on any of them within any finite factor $r = r(n) \geqslant 1$ must have at least $2^n/n^3$ gates.*

Proof. The number of monotone boolean $(\vee, \wedge)$ circuits of size t on n input variables is at most $L(n, t) = 2^t (t+n)^{2t}$. This is, actually, an upper bound on the number of constant-free circuits over any semiring $(R, \oplus, \otimes)$. Indeed, each gate in such a circuit is assigned a semiring operation (two choices) and acts on two previous nodes. Each previous node can either be a previous gate (at most t choices) or an input variable (n choices). Thus, each single gate has at most $N = 2(t+n)^2$ choices, and the number of choices for a circuit is at most N^t.

When applied with $m = \lfloor n/2 \rfloor$, Proposition 4.5 gives us at least $M(n) = 2^{\binom{n}{m}/n} \geqslant 2^{2^n/2n^{3/2}}$ matroids $\mathcal{F} \subseteq \binom{[n]}{m}$. On the other hand, at most $L(n, t)$ families $\mathcal{F} \subseteq 2^{[n]}$ can have monotone boolean circuit complexity at most t. For $t := 2^n/n^3$, we have $\log L(n, t) = 2^n/n^3 + (2^{n+1}/n^3) \log(n + 2^n/n^3) = O(2^n/n^2) \ll \log M(n) = 2^n/2n^{3/2}$. Since every circuit computes only one function, at least $M(n) - L(n, t) \geqslant L(n, t)$ matroids require monotone boolean circuits of size at least $t = 2^n/n^3$. Theorem 4.3 yields the same lower bound for approximating $(\min, +)$ circuits. □

5. Approximation limitations of (max,+) circuits

Given a family $\mathcal{F} \subseteq 2^{[n]}$ of feasible solutions, and an approximation factor $r \geqslant 1$, we will denote by $\mathsf{Max}_r(\mathcal{F})$ the minimum number of gates in a $(\max, +)$ circuit approximating the maximization problem $f(x) = \max_{S \in \mathcal{F}} \sum_{i \in S} x_i$ on $\mathcal{F}$ within the factor r.

In Section 4, we have shown that there are (even explicit) families $\mathcal{F} \subseteq 2^{[n]}$, the *minimization* problems on which cannot be approximated by small (polynomial in n) size $(\min, +)$ circuits within any finite factor $r = r(n)$. On the other hand, in the case of *maximization* problems, the approximation factor is always finite. Namely, we always have $\mathsf{Max}_n(\mathcal{F}) \leqslant n-1$: since the weights are nonnegative, we can just use the trivial $(\max,+)$ circuit $\max\{x_1,\dots,x_n\}$.

5.1. **Counting fails for approximating** $(\max,+)$ **circuits.** There is an even more substantial difference between approximating $(\min,+)$ and $(\max,+)$ circuits than just the "bounded versus unbounded approximation factors" phenomenon: unlike for $(\min,+)$ circuits, even *counting* arguments are unlikely to yield large lower bounds on the size of approximating $(\max,+)$ circuits, even for very small approximation factors $r = 1+o(1)$.

Say that a family $\mathcal{F} \subseteq 2^{[n]}$ is *k-dense* if every k-element subset of $[n]$ is contained in at least one set of $\mathcal{F}$. The *top k-of-n selection* problem $f_{n,k}(x_1,\dots,x_n)$ outputs the sum of the k largest input numbers.

Proposition 5.1. *The top k-of-n selection problem $f_{n,k}$ can be solved by a* $(\max,+)$ *circuit of size $2kn$, and this circuit approximates the maximization problem on every k-dense family $\mathcal{F} \subseteq \binom{[n]}{m}$ within the factor $r = m/k$.*

Proof. The family of feasible solutions of $f_{n,k}$ consists of all k-element subsets of $[n] = \{1,\dots,n\}$. In particular, $f_{n,1}(x) = \max\{x_1,\dots,x_n\}$ and $f_{n,n}(x) = x_1+\dots+x_n$. The Pascal identity $\binom{n+1}{k} = \binom{n}{k}+\binom{n}{k-1}$ for binomial coefficients gives us the recursion

$$f_{n+1,k}(x_1,\dots,x_{n+1}) = \max\{f_{n,k}(x_1,\dots,x_n), f_{n,k-1}(x_1,\dots,x_n)+x_{n+1}\}\,.$$

So, $f_{n,k}$ can be solved by a $(\max,+)$ circuit with only $2kn$ $(\max,+)$ gates.

Now let $\mathcal{F} \subseteq \binom{[n]}{m}$ be a k-dense family. The maximization problem on $\mathcal{F}$ is $f(x) = \max_{S\in\mathcal{F}} \sum_{i\in S} x_i$. Since the weights are nonnegative, the k-denseness of $\mathcal{F}$ ensures that $f(x) \geqslant f_{n,k}(x)$. On the other hand, since no solution has more than m elements, the optimal weight $f(x)$ of a feasible solution cannot exceed m/k times the sum of weights of k heaviest elements in this solution. Hence, $f(x) \leqslant (m/k)\cdot f_{n,k}(x)$, as desired. □

Proposition 5.2. *There exist doubly-exponentially many in n families $\mathcal{F} \subseteq 2^{[n]}$ such that* $\mathsf{Max}_1(\mathcal{F}) = 2^{\Omega(n)}$ *but* $\mathsf{Max}_{1+o(1)}(\mathcal{F}) \leqslant n^2$.

The families $\mathcal{F}$ are here *matroids*, and the upper bound is achieved by *one single* $(\max,+)$ circuit.

Proof. Let n be a sufficiently large even integer, and $m = n/2$. Proposition 4.5 gives us at least $M := 2^{\binom{n}{m}/n}$ families $\mathcal{F} \subseteq \binom{[n]}{m}$ (which are matroids) with the property that the Hamming distance between any two distinct sets $A \neq B \in \binom{[n]}{m} \setminus \mathcal{F}$ is > 2. We claim that each such family $\mathcal{F}$ is k-dense for $k := m-1$. To see this, take any set $T \in \binom{[n]}{m-1}$, any two distinct elements $a \neq b$ outside T, and consider the m-element sets $A = T\cup\{a\}$ and $B = T\cup\{b\}$. Since the Hamming distance between A and B is 2, they cannot *both* lie outside the family $\mathcal{F}$. So, at least one of them must belong to $\mathcal{F}$, as desired.

We thus have at least M families $\mathcal{F} \subseteq \binom{[n]}{m}$ which are k-dense for $k = m-1$. By Proposition 5.1, one $(\max,+)$ circuit of size at most $2kn \leqslant n^2$ for the top k-of-n problem $f_{n,k}$ approximates the maximization problem on *each* of these M families within the factor $r = m/(m-1) = 1+1/(m-2) = 1+o(1)$. On the other hand, by Lemma 3.2, we can consider only constant-free $(\max,+)$ circuits, and the same counting argument as in the proof of Theorem 4.6 yields the lower bound $\mathsf{Max}_1(\mathcal{F}) = 2^{\Omega(n)}$ for doubly-exponentially many of these families $\mathcal{F}$. □

Remark 4 (Boolean bound fails for approximating (max, +)). The standard counting (as in the proof of Theorem 4.6) shows that the boolean function defined by some of the families $\mathcal{F}$ given by Proposition 5.2 (actually, by many of these families) requires monotone boolean circuits of size $2^{\Omega(n)}$, but (by Proposition 5.2) $\mathsf{Max}_r(\mathcal{F}) \leqslant n^2$ holds already for a factor $r = 1 + o(1)$.

Actually, small (max, +) circuits can approximate even *random* maximization problems quite well. For an even integer $n \geqslant 4$ and $m = n/2$, let $\mathfrak{F}$ be a random family of m-element subsets of $[n]$ with each m-element subset being included in $\mathfrak{F}$ independently with probability $1/2$.

Proposition 5.3. *With probability* $1 - o(1)$, $\mathsf{Max}_{1+o(1)}(\mathfrak{F}) \leqslant n^2$ *holds.*

Proof. Let $k := m - 2$. Since each k-element set is contained in $l = \binom{n-k}{2} = \Omega(n^2)$ sets of $\binom{[n]}{m}$, the probability that a fixed k-element set will be contained in *none* of the sets of $\mathfrak{F}$ is $(1/2)^l = 2^{-\Omega(n^2)}$. So, by the union bound, the family $\mathfrak{F}$ is *not* k-dense with probability at most $\binom{n}{m} \cdot 2^{-\Omega(n^2)} = 2^{-\Omega(n^2)}$. That is, the family $\mathfrak{F}$ is k-dense with probability at least $1 - 2^{-\Omega(n^2)}$. By Proposition 5.1, with this probability, the (max, +) circuit for the top k-of-n selection problem (see Proposition 5.1) approximates the maximization problem on a random family $\mathfrak{F} \subseteq \binom{[n]}{n/2}$ within the factor $r = m/k = 1 + o(1)$. □

Propositions 5.2 and 5.3 only show the mere *existence* of maximization problem that are hard to solve by (max, +) circuits exactly (with factor $r = 1$), but can be approximated by small (max, +) circuits within a slightly larger factor $r > 1$. Still, there are also *explicit* maximization problems exhibiting a similar gap.

A family $\mathcal{F} \subseteq 2^{[n]}$ is a *Sidon family* if the set $A \subseteq \{0,1\}^n$ of the characteristic 0-1 vectors of sets in $\mathcal{F}$ has the following property for all vectors $a, b, c, d \in A$: if $a + b = c + d$, then $\{a, b\} = \{c, d\}$ (the addition is over the reals, not over GF(2)). That is, knowing the sum $a + b \in \{0,1,2\}^n$ of two vectors $a, b \in A$, we know which vectors were added.

Theorem 5.4 (Explicit gaps)**.** *Let m be an odd integer, and $n = 4m$. Then there is an explicit Sidon family* $\mathcal{F} \subseteq 2^{[n]}$ *such that* $\mathsf{Max}_1(\mathcal{F}) \geqslant 2^{n/4}$ *but* $\mathsf{Max}_2(\mathcal{F}) \leqslant n$.

The lower bound here follows from known lower bounds on the monotone arithmetic circuit complexity of polynomials whose sets of exponent vectors are Sidon sets, but the proof of the upper bound is somewhat technical. So, since we are mainly interested in proving *lower* bounds, we postpone the entire proof of Theorem 5.4 to Appendix B.

The message of Propositions 5.2 and 5.3 is: while *most* problems are hard to solve exactly, they are trivially approximable by just *one* small (max, +) circuit within a small factor $r = 1 + o(1)$. Together with Theorem 5.4 (and Remark 4), this serves as a serious indication that the task of proving lower bounds on the size of approximating (max, +) circuits is by far more difficult than for (min, +) circuits: monotone *boolean* circuits cannot help then, and even counting arguments are unlikely to work against (max, +) circuits.

Still, by looking more carefully into the structure of vectors *produced* by approximating (max, +) circuits (Lemma 5.5), and using structural restrictions of such sets given by a "decomposition lemma" (Lemma 5.6), we will be able to derive a general "rectangle bound" for approximating (max, +) circuits (Theorem 5.9).

5.2. **Structure of approximating** (max, +) **circuits.** Since we are interested in the structure of sets of *vectors* produced by (approximating) circuits, it will be convenient to turn to the language of *vectors*.

Lemma 5.5. *If* Φ *is a* (max, +) *circuit approximating the maximization problem on a set* $A \subseteq \{0,1\}^n$ *within a factor r, then the set* $B \subset \mathbb{N}^n$ *of vectors produced by* Φ *has the following two properties:*

(i) *if* $b \in B$, *then* $b \leqslant a$ *for some* $a \in A$*;*
(ii) *if* $a \in A$, *then* $\langle a, b\rangle \geqslant \frac{1}{r}\langle a, a\rangle$ *for some* $b \in B$*.*

Proof. By Lemma 3.2, we can assume that the circuit Φ is constant-free. By Proposition 3.1, the circuit Φ solves the maximization problem $\Phi(x) = \max_{b\in B}\langle b, x\rangle$. The maximization problem on A is of the form $f(x) = \max_{a\in A} \langle a, x\rangle$. Since the circuit r-approximates the maximization problem on A, we know that $\frac{1}{r} \cdot f(x) \leqslant \Phi(x) \leqslant f(x)$ must hold for all input weightings $x \in \mathbb{R}^n_+$.

Had some vector $b \in B$ a position i with $b_i > 1$, then on the input $x = \vec{e}_i$, we would have $\Phi(x) \geqslant \langle b, x\rangle = b_i > 1$ but $f(x) \leqslant 1$, since all vectors in A are 0-1 vectors. So, $B \subseteq \{0, 1\}^n$, that is, the set B also consists of only 0-1 vectors.

To show item (i), suppose contrariwise that there is a vector $b \in B$ such that $b \nleqslant a$ holds for all vectors $a \in A$. Since (as we have just shown) b is a 0-1 vector, this means that, for every vector $a \in A$, there is a position i where $b_i = 1$ but $a_i = 0$. Hence, on the weighting $x := b$, we have $\Phi(x) \geqslant \langle b, x\rangle = \langle b, b\rangle$, but $\langle a, x\rangle = \langle a, b\rangle \leqslant \langle b, b\rangle - 1$ for all $a \in A$, a contradiction with $\Phi(x) \leqslant f(x)$.

To show item (ii), assume contrariwise that there is some vector $a \in A$ such that $\langle a, b\rangle < m/r$ holds for all vectors $b \in B$, where $m = \langle a, a\rangle$. Then, on the input $x := a$, we have $\Phi(x) < m/r$ but $f(x) \geqslant \langle a, a\rangle = m$, a contradiction with $\frac{1}{r} \cdot f(x) \leqslant \Phi(x)$. □

Remark 5. Note that Lemma 5.5 holds even when the circuit Φ is only required to r-approximate the given minimization problem on input weightings $x \in \{0, 1\}^n$. Indeed, to eliminate constant inputs from $(\max, +)$ circuits in Lemma 3.2 we only used the input weighting $x = \vec{0}$, and the proof of Lemma 5.5 itself also uses only boolean 0-1 weightings. This implies that the rectangle bound (Theorem 5.9), whose proof will use Lemma 5.5, holds also when the $(\max, +)$ circuits must approximate a given problem only on boolean 0-1 weightings.

5.3. **Minkowski circuits.** As we already mentioned in Section 3.2, unlike the function *computed* by a circuit Φ over a semiring $(R, \oplus, \otimes)$, the set $B \subset \mathbb{N}^n$ of vectors *produced* by Φ does not depend on the underlying semiring—it only depends on the circuit itself. That is, B depends only on what the underlying graph of Φ is, and what of the two semiring operations are associated with gates. This independence of produced sets from actual semirings is captured by the model of “Minkowski circuits.” These circuits allow one to analyze the *structure* of sets produced by circuits over arbitrary semirings in a uniform and mathematically clean way.

A *Minkowski circuit* Φ is a directed acyclic graph with $n + 1$ input (indegree zero) nodes holding single-element sets $\{\vec{0}\}, \{\vec{e}_1\}, \dots, \{\vec{e}_n\}$. Every other node, a *gate*, has indegree two, and performs either the set-theoretic union ($\cup$) or the Minkowski sum ($+$) operation on its two inputs.

The sets $X_v \subset \mathbb{N}^n$ of vectors *produced* at the gates v of Φ are obtained as follows. If v is an input node, then X_v is one of the single-element sets $\{\vec{0}\}, \{\vec{e}_1\}, \dots, \{\vec{e}_n\}$, depending on which of these sets is held by the node v. Then $X_v = X_u \cup X_w$ if $v = u \cup w$ is a union gate, and $X_v = X_u + X_w$ if $v = u + w$ is a Minkowski sum gate. The set $B \subset \mathbb{N}^n$ produced by the entire circuit Φ is the set X_v produced at the output gate v.

The *Minkowski version* of a circuit Φ over an arbitrary semiring $(R, \oplus, \otimes)$ is obtained by replacing each input constant $\lambda \in R$ by the singleton $\{\vec{0}\}$, each input variable x_i by the singleton $\{\vec{e}_i\}$, each “addition” ($\oplus$) gate by the union ($\cup$) gate, and each “multiplication” ($\otimes$) gate by the Minkowski sum ($+$) gate.

The model of Minkowski circuits is justified by the following trivial observation, which follows directly from the definition of sets produced by circuits over semirings: the set produced by a circuit over *any* semiring is the set produced by the Minkowski version of this circuit.

5.4. **Decomposition lemma for Minkowski circuits.** We will prove lower bounds for approximating $(\max, +)$ circuits using a general "decomposition lemma" for Minkowski circuits. The *sumset* defined by two sets $X \subseteq \mathbb{N}^n$ and $Y \subseteq \mathbb{N}^n$ of vectors is the Minkowski sum $X + Y = \{x + y : x \in X, y \in Y\}$ of these two sets.

Sumsets naturally emerge in every Minkowski circuit Φ. At each Minkowski sum gate following a gate v (if there is any), the set X_v of vectors produced at v is "enlarged" by adding at least one vector to *all* vectors in X_v. So, when we arrive at the output gate w, the entire translates $X_v + y = \{x + y : x \in X_v\}$ of X_v by some vectors $y \in \mathbb{N}^n$ must lie in the set $X_w = B$ produced by the entire circuit. This observation motivates to associate with every gate v its *residue*

$$Y_v = \{y \in \mathbb{N}^n : X_v + y \subseteq B\}$$

which collects all vectors $y \in \mathbb{N}^n$, the translates of X_v by which lie in the set B produced by the entire circuit. For example, if v is the output gate, then $X_v = B$ and $Y_v = \{\vec{0}\}$. If v is an input node, then either $X_v = \{\vec{0}\}$ and $Y_v = B$, or $X_v = \{\vec{e}_i\}$ and $Y_v = \{b - \vec{e}_i : b \in B, b_i \geqslant 1\}$.

Note that neither X_v nor Y_v needs lie in B, but $X_v + Y_v \subseteq B$ already holds for every gate v. Thus, if the circuit Φ has t gates, then we obtain a covering of the set B by t sumsets of the form $X_v + Y_v$

A *norm-measure* is any assignment $\mu : \mathbb{N}^n \to \mathbb{R}_+$ of nonnegative real numbers to vectors in $\mathbb{N}^n$ such that every 0-1 vector with at most one 1 gets norm at most 1, and the norm is monotone and subadditive: $\mu(x) \leqslant \mu(x + y) \leqslant \mu(x) + \mu(y)$ holds for all vectors $x, y \in \mathbb{N}^n$.

Lemma 5.6 (Decomposition lemma [16])**.** *If a set $B \subset \mathbb{N}^n$ can be produced by a Minkowski* $(\cup, +)$ *circuit of size t, then B is a union of t sumsets $X + Y \subseteq B$ with the following property.*

($*$) *For every norm-measure $\mu : \mathbb{N}^n \to \mathbb{R}_+$, for every vector $b \in B$ of norm $\mu(b) > 1$, and every $1/\mu(b) \leqslant \theta < 1$ at least one of these sumsets $X + Y$ contains vectors $x \in X$ and $y \in Y$ such that $x + y = b$ and*

$$\frac{\theta}{2} \cdot \mu(b) < \mu(x) \leqslant \theta \cdot \mu(b)\,.$$

The lemma was originally proved in [16, Theorem D]. Here we give a simpler proof.

Proof. Let Φ be a Minkowski $(\cup, +)$ circuit of size t producing the set B. Since we have only t gates in the circuit, it is enough to show that the collection of sumsets $X_v + Y_v$ associated with the gates v of Φ has the desired property ($*$). So, fix some norm-measure $\mu : \mathbb{N}^n \to \mathbb{R}_+$, some vector $b \in B$ of norm $p := \mu(b) > 1$, and a real number $1/p \leqslant \theta < 1$.

By a *decomposition* of the vector b (or just a *decomposition*, because the vector b is fixed) at a gate v we will mean a pair $(x, y) \in X_v \times Y_v$ of vectors (if there is one) such that $x + y = b$. The *norm* of such a decomposition is the norm $\mu(x)$ of the first vector (that in the set X_v). Note that at the output gate, we have the unique decomposition $(x, y) = (b, \vec{0})$ of b of norm $\mu(x) = \mu(b) = p$.

Claim. *Let v be a gate entered from gates u and w. If there is a decomposition (x, y) of vector b at gate v, then there is a decomposition (x', y') of b at u or w such that $\frac{1}{2} \cdot \mu(x) \leqslant \mu(x') \leqslant \mu(x)$.*

Proof. If $v = u \cup w$ is a union gate, then $X_v = X_u \cup X_w$ and, hence, $Y_v = Y_u \cap Y_w$. So, the same pair (x, y) is a decomposition at the gate u (if $x \in X_u$) or at the gate w (if $x \in X_w$), and the claim is trivial in this case.

Assume now that $v = u + w$ is a Minkowski sum gate. Then $x = x_u + x_w$ for some vectors $x_u \in X_u$ and $x_w \in X_w$. Since vector y belongs to the residue Y_v of gate v, we know that $X_u + X_w + y \subseteq B$ holds. In particular, both inclusions $X_u + (x_w + y) \subseteq B$ and $X_w + (x_u + y) \subseteq B$ must hold. So, vector $x_w + y$ belongs to the residue Y_u of gate u, and vector $x_u + y$ belongs to the residue Y_w of gate w. This implies that the pair $(x_u, x_w + y)$ is a decomposition of b at the gate u, and the pair $(x_w, x_u + y)$ is a decomposition of b at the gate w. Since $x = x_u + x_w$, the monotonicity of the norm implies that both $\mu(x_u)$ and $\mu(x_w)$ are at most $\mu(x)$, while the subadditivity of the norm implies that one of the norms $\mu(x_u)$ and $\mu(x_w)$ of these decompositions must be at least $\frac{1}{2} \cdot \mu(x_u + x_w) = \frac{1}{2} \cdot \mu(x)$, and we can take that input u or w at which the decomposition has larger norm. $\square$

We now start at the output gate with the unique decomposition $(x, y) = (b, \vec{0})$ of vector b, and traverse an input-output path P in the circuit *backwards* by using the following rule: if v is a currently reached gate, and (x, y) is a decomposition at this gate, then go to that of the two inputs of v which has a decomposition (x', y') of norm $\mu(x') \geqslant \frac{1}{2} \cdot \mu(x)$ (if both input gates have this property, then go to any of them). The claim above ensures that we will eventually reach some input node.

If this input node holds the set $\{\vec{0}\}$, then the only decomposition $(x, y) = (\vec{0}, b)$ of vector b at this gate has norm $\mu(x) = \mu(\vec{0}) \leqslant 1$, and if this gate holds $\{\vec{e}_i\}$, then the only decomposition $(x, y) = (\vec{e}_i, b - \vec{e}_i)$ of b at this gate has also norm $\mu(x) = \mu(\vec{e}_i) \leqslant 1$. In both cases, we have that $\mu(x) \leqslant 1$, which is at most θp, because $\theta \geqslant 1/p$.

On the other hand, the (also unique) decomposition $(x, y) = (b, \vec{0})$ of the vector b at the output gate has norm $\mu(x) = \mu(b) = p$, which is strictly larger than θp, because $\theta < 1$. So, there must be an edge (u, v) in the path P at which the jump from $\leqslant \theta p$ to $> \theta p$ happens. That is, there must be a decomposition (x, y) at the gate v and a decomposition (x', y') at the gate u such that $\mu(x) > \theta p$ but $\mu(x') \leqslant \theta p$. By the above claim, we have $\mu(x') \geqslant \frac{1}{2} \cdot \mu(x)$. We have thus found a sumset $X_u + Y_u$ and vectors $x' \in X_u$ and $y' \in Y_u$ such that $x' + y' = b$ and $\frac{1}{2}\theta p < \mu(x') \leqslant \theta p$, as desired. $\square$

5.5. **Decomposition lemma for approximating** $(\max, +)$ **circuits.** In the following lemma, $0 < \gamma < 1$ is an arbitrary fixed "balance" parameter. For a 0-1 vector a, let $|a| := \langle a, a \rangle$ denote the number of 1s in a.

Lemma 5.7. *Let Φ be a* $(\max, +)$ *circuit of size t approximating the maximization problem on a set $A \subseteq \{0, 1\}^n$ within factor $r \geqslant 1$, and let $B \subset \mathbb{N}^n$ be the set of vectors produced by Φ. Then there exist t or fewer sumsets $X + Y \subseteq B$ such that:*

1. *if $x \in X$ and $y \in Y$, then $x + y \leqslant a$ for some $a \in A$;*
2. *$\langle x, y \rangle = 0$ holds for all $x \in X$ and $y \in Y$;*
3. *for every vector $a \in A$ with $|a| \geqslant r/\gamma$ ones, at least one of these sumsets $X + Y$ contains vectors $x \in X$ and $y \in Y$ such that:*

$$\langle a, x + y \rangle \geqslant \tfrac{1}{r} \cdot |a| \,, \; \langle a, x \rangle > \tfrac{\gamma}{2r} \cdot |a| \;\text{ and }\; \langle a, y \rangle \geqslant \tfrac{1-\gamma}{r} \cdot |a| \,. \tag{5}$$

Proof. By Lemma 5.5, we know that the set B has the following two properties:

(i) if $b \in B$, then $b \leqslant a$ for some $a \in A$;
(ii) if $a \in A$, then $\langle a, b \rangle \geqslant \frac{1}{r} \cdot |a|$ for some $b \in B$.

The Minkowski $(\cup, +)$ version Φ' of the circuit Φ has the same size t and produces the same set B. When applied to the Minkowski circuit Φ', Lemma 5.6 gives us a collection of t sumsets $X + Y \subseteq B$ with the following property holding for every norm-measure $\mu : \mathbb{N}^n \to \mathbb{R}_+$, for every vector $b \in B$ of norm $\mu(b) > 1$, and every real number θ satisfying $1/\mu(b) \leqslant \theta < 1$:

(∗) at least one of the sumsets $X + Y$ contains vectors $x \in X$ and $y \in Y$ such that $x + y = b$ and $\frac{\theta}{2} \cdot \mu(b) < \mu(x) \leqslant \theta \cdot \mu(b)$.

Since the set A consists of only 0-1 vectors, property (i) implies that the set B also consists of 0-1 vectors. So, $X + Y \subseteq B$ implies that each of our sumsets $X + Y$ has the first two properties (1) and (2) claimed in Lemma 5.7. It remains to show the third "balancedness" property (3).

Fix an arbitrary vector $a \in A$ with $|a| \geqslant r/\gamma$ ones. Property (ii) of the set B suggests to associate with a the norm-measure $\mu(x) = \mu_a(x) := \langle a, x\rangle$. Then, by (ii), there is a vector $b \in B$ of norm $\mu(b) = \langle a, b\rangle \geqslant p := |a|/r$, which is > 1 since $|a| \geqslant r/\gamma$, $r \geqslant 1$ and $\gamma < 1$. We also have $\gamma \geqslant r/|a| \geqslant 1/\mu(b)$. Hence, by the property $(*)$, at least one of our sumsets $X + Y$ contains vectors $x \in X$ and $y \in Y$ such that $x + y = b$ and the following inequalities hold:

$$\tfrac{1}{2}\gamma \cdot p < \mu(x) = \langle a, x\rangle \leqslant \gamma \cdot p\,. \tag{6}$$

Now, the first inequality in Eq. (5) follows since $\langle a, x + y\rangle = \langle a, b\rangle \geqslant p$, the second inequality in Eq. (5) is the first inequality in Eq. (6) and, since $\langle x, y\rangle = 0$, the third inequality in Eq. (5) follows from $\langle a, x + y\rangle \geqslant p$ and the second inequality in Eq. (6). □

5.6. **The rectangle bound.** A *rectangle* is a family of sets specified by giving a pair $\mathcal{A}, \mathcal{B}$ of families of sets which is *cross-disjoint* in that $A \cap B = \emptyset$ holds for all sets $A \in \mathcal{A}$ and $B \in \mathcal{B}$. The rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ itself consists of all sets $A \cup B$ with $A \in \mathcal{A}$ and $B \in \mathcal{B}$. A rectangle $\mathcal{R}$ *lies below* a family $\mathcal{F}$ if every set of $\mathcal{R}$ is contained in at least one set of $\mathcal{F}$.

For real numbers $r \geqslant 1$ and $0 < \gamma < 1$, we say that a set F *appears* (r, γ)*-balanced* in a rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ if there are sets $A \in \mathcal{A}$ and $B \in \mathcal{B}$ such that

$$|F \cap (A \cup B)| \geqslant \tfrac{1}{r} \cdot |F|,\ \ |F \cap A| > \tfrac{\gamma}{2r} \cdot |F| \ \text{ and } \ |F \cap B| \geqslant \tfrac{1-\gamma}{r} \cdot |F|\,. \tag{7}$$

Thus, a set F appears balanced in a rectangle $\mathcal{R}$, if for some pair of disjoint sets whose union belongs to $\mathcal{R}$, the union as well as the sets themselves contain "many enough" elements of F. In our applications, we will only use the last two inequalities in Eq. (7), but the first inequality may also be important in some applications.

The following lemma is just a translation of Lemma 5.7 from the language of vectors to the language of sets. Recall that $\mathsf{Max}_r(\mathcal{F})$ denotes the minimum size of a $(\max, +)$ circuit approximating the maximization problem on $\mathcal{F}$ within the factor r.

Lemma 5.8. *Let $r \geqslant 1$ and $0 < \gamma < 1$. If $\mathsf{Max}_r(\mathcal{F}) \leqslant t$, then there are t or fewer rectangles lying below $\mathcal{F}$ such that every set of $\mathcal{F}$ with at least r/γ elements appears (r, γ)-balanced in at least one of these rectangles.*

Proof. Take a $(\max, +)$ circuit Φ of size $t = \mathsf{Max}_r(\mathcal{F})$ approximating the maximization problem on $\mathcal{F}$ within factor r. Let $A \subseteq \{0, 1\}^n$ be the set of characteristic 0-1 vectors of the sets in $\mathcal{F}$. Hence, the circuit r-approximates the maximization problem on A. Let $B \subset \mathbb{N}^n$ be the set of vectors produced by Φ.

Lemma 5.7 gives us t or fewer sumsets $X + Y \subseteq B$ with properties (1)–(3) listed in this lemma. By property (1), the set B and, hence, each of our t sumsets $X + Y$ consists of only 0-1 vectors. So, each sumset $X + Y$ translates into the rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$, where $\mathcal{A}$ is the family of supports $S_x = \{i : x_i = 1\}$ of vectors $x \in X$, and $\mathcal{B}$ is the family of supports $S_y = \{i : y_i = 1\}$ of vectors $y \in Y$. After this translation, property (1) of Lemma 5.7 implies that each of these rectangles $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ lies below our family $\mathcal{F}$, property (2) yields the cross-disjointness condition ($A \cap B = \emptyset$ for all sets $A \in \mathcal{A}$ and $B \in \mathcal{B}$), and property (3) implies that every set $F \in \mathcal{F}$ with $|F| \geqslant r/\gamma$ elements appears (r, γ)-balanced in at least one of these rectangles. □

In applications, we will use a direct consequence of Lemma 5.8 formulated as a *lower bound* on $\mathsf{Max}_r(\mathcal{F})$. Let $r \geqslant 1$ be a given approximation factor, and $0 < \gamma < 1$ any fixed real parameter (we are free to choose this parameter).

Theorem 5.9 (Rectangle bound)**.** *Let $\mathcal{F}$ be a family of feasible solutions, and $\mathcal{H} \subseteq \mathcal{F}$ some subfamily of feasible solutions, each of cardinality at least r/γ. If for every rectangle $\mathcal{R}$ lying below $\mathcal{F}$, at most h of the sets of $\mathcal{H}$ appear (r, γ)-balanced, then $\mathsf{Max}_r(\mathcal{F}) \geqslant |\mathcal{H}|/h$.*

In particular, Result 4 stated in Section 2 corresponds to the balance parameter $\gamma = 2/3$. In the following applications, we will always take $\mathcal{F} = \mathcal{H}$, but the possibility to choose special subfamilies $\mathcal{H} \subseteq \mathcal{F}$ of feasible solutions may be useful in other applications. Note that, besides of the cross-disjointness and balancedness of rectangles, the fact that the rectangles must lie below $\mathcal{F}$ is also important. If, say, $\mathcal{F}$ is the family of all perfect matchings in some graph, and if a rectangle $\mathcal{R}$ lies below $\mathcal{F}$, then we immediately know that every set of $\mathcal{R}$ must be a union of two vertex-disjoint matchings.

5.7. **Maximization on designs.** We already know that there *exist* many maximization problems for which slight decrease of the allowed approximation factor from $r = 1 + o(1)$ to $r = 1$ can exponentially increase the size of $(\max, +)$ circuits (Proposition 5.2). We also know *explicit* maximization problems for which such a jump in circuit size occurs when decreasing the approximation factor from $r = 2$ to $r = 1$ (Theorem 5.4).

Our goal in this section is to show that such jumps can happen for arbitrarily *large* approximation factors r: a slight decrease of the allowed approximation factor r can make tractable problems intractable. We demonstrate these jumps on maximization problems whose families $\mathcal{F}$ of feasible solutions are "combinatorial designs" (Theorem 5.12 below).

An (m, d)-*design* (know also as a *weak design*) is a family $\mathcal{F}$ which is:

- m-*uniform*: every set has exactly m elements;
- d-*disjoint*: no two distinct sets share d or more elements in common.

We will see soon (the upper bound in Theorem 5.12) that, for some (m, d)-designs $\mathcal{F} \subseteq 2^{[n]}$ with $n = m^2$, $\mathsf{Max}_r(\mathcal{F}) = O(n)$ holds when the approximation factor $r = m/d$ is allowed. On the other hand, we have the following general lower bound on $\mathsf{Max}_r(\mathcal{F})$ for *every* (m, d) design $\mathcal{F}$, when the allowed approximation factor r is only slightly smaller than m/d.

To state this bound, we need an auxiliary notation. For a family $\mathcal{F}$ of sets and a real number $l \geqslant 0$, let $\#_l(\mathcal{F})$ denote the maximal possible number of sets in $\mathcal{F}$ containing a fixed set with l (or more) elements. In other words, $\#_l(\mathcal{F})$ is the maximal possible number of sets in $\mathcal{F}$ whose intersection has l (or more) elements. In particular, if m is the maximum cardinality of a set of $\mathcal{F}$, then $|\mathcal{F}| = \#_0(\mathcal{F}) \geqslant \#_1(\mathcal{F}) \geqslant \ldots \geqslant \#_m(\mathcal{F}) = 1$, and $\#_l(\mathcal{F}) = 0$ for all $l > m$. Note that a nonempty m-uniform family $\mathcal{F}$ is an (m, d)-design if and only if $\#_d(\mathcal{F}) = 1$. Also, $\#_1(\mathcal{F}) = 1$ means that all sets of $\mathcal{F}$ are disjoint.

Lemma 5.10. *Let $\mathcal{F}$ be an (m, d)-design for $1 \leqslant d < m$, $1/(d + 1) \leqslant \gamma < 1$ and $l = \gamma d/2$. For the factor $r = (1 - \gamma)m/d$, we have*

$$\mathsf{Max}_r(\mathcal{F}) \geqslant \frac{|\mathcal{F}|}{\#_l(\mathcal{F})}\,.$$

Proof. We are going to apply the rectangle bound (Theorem 5.9) with the balance parameter γ. First, observe that $|F| \geqslant r/\gamma$ holds for *every* set $F \in \mathcal{F}$: since $|F| = m$, this is equivalent to the inequality $1 \geqslant (1 - \gamma)/\gamma d$, which holds because $\gamma \geqslant 1/(d + 1)$.

Take an arbitrary rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ lying below $\mathcal{F}$. Let $\mathcal{F}_{\mathcal{R}} \subseteq \mathcal{F}$ be the family of all sets $F \in \mathcal{F}$ such that

$$|F \cap A| \geqslant \tfrac{\gamma}{2r} \cdot m = \tfrac{\gamma}{2(1-\gamma)} \cdot d \geqslant \tfrac{\gamma}{2} \cdot d = l \quad \text{and} \quad |F \cap B| \geqslant \tfrac{1-\gamma}{r} \cdot m = d$$

hold for some $A \in \mathcal{A}$ and $B \in \mathcal{B}$. By Theorem 5.9, it is enough to show that $|\mathcal{F}_{\mathcal{R}}| \leqslant \#_l(\mathcal{F})$. We can assume that all sets $B \in \mathcal{B}$ have $|B| \geqslant d$ elements: sets $B \in \mathcal{B}$ of size $|B| < d$ cannot fulfill

$|F \cap B| \geqslant d$ and, hence, can be removed from $\mathcal{B}$ without changing $\mathcal{F}_{\mathcal{R}}$. Similarly, we can assume that all sets $A \in \mathcal{A}$ have $|A| \geqslant l$ elements. Let X be the union of all sets in $\mathcal{A}$; hence, $|X| \geqslant l$.

Claim. *All sets of $\mathcal{F}_{\mathcal{R}}$ contain the set X.*

Proof. Take a set $F \in \mathcal{F}_{\mathcal{R}}$. Then $|F \cap B| \geqslant d$ holds for some set $B \in \mathcal{B}$. On the other hand, since the rectangle $\mathcal{R}$ lies below $\mathcal{F}$, every set $A \cup B$ with $A \in \mathcal{A}$ must lie in some set of $\mathcal{F}$. Since all these sets contain the set B with $|B| \geqslant d$ elements, and since the family $\mathcal{F}$ is d-disjoint, this implies that *all* sets of $\mathcal{A} \vee \{B\}$ and, hence, also the set $X \cup B$ must be contained in *one* set F_B of $\mathcal{F}$. Since both sets F and F_B of $\mathcal{F}$ contain the same set $F \cap B$ of size $|F \cap B| \geqslant d$, and since the family $\mathcal{F}$ is d-disjoint, the equality $F = F_B$ and, hence, the desired inclusion $X \subseteq F$ follows. □

Since the set X has $|X| \geqslant l$ elements, the claim yields $|\mathcal{F}_{\mathcal{R}}| \leqslant \#_{|X|}(\mathcal{F}) \leqslant \#_l(\mathcal{F})$ and, by Theorem 5.9 (applied with $\mathcal{H} = \mathcal{F}$), the desired lower bound $\mathsf{Max}_r(\mathcal{F}) \geqslant |\mathcal{F}|/|\mathcal{F}_{\mathcal{R}}| \geqslant |\mathcal{F}|/\#_l(\mathcal{F})$ follows. □

5.8. **Factor hierarchy theorem.** We will now apply the general lower bound of Lemma 5.10 to explicit designs. Let m be a prime power, $1 \leqslant d \leqslant m$ an integer, and consider the grid $\mathrm{GF}(m) \times \mathrm{GF}(m)$. The *polynomial (m, d)-design* $\mathcal{F}$ consists of all $|\mathcal{F}| = m^d$ subsets S of points in this grid of the form $S = \{(a, p(a)) :\ a \in \mathrm{GF}(m)\}$ for a univariate polynomial $p = p(x)$ of degree at most $d - 1$ over $\mathrm{GF}(m)$. Note that no two points of any of these sets S lie in the same row of the grid.

The main combinatorial property of polynomial designs is the following.

Proposition 5.11. *Let $\mathcal{F}$ be a polynomial (m, d)-design, and $1 \leqslant d \leqslant m$. For every set of $l \leqslant d$ points of the grid $\mathrm{GF}(m) \times \mathrm{GF}(m)$, with no two in the same row, exactly m^{d-l} sets of $\mathcal{F}$ contain this set. In particular, $\#_l(\mathcal{F}) \leqslant m^{d-l}$ holds for every $0 \leqslant l \leqslant d$.*

Proof. This is a direct consequence of a standard result in polynomial interpolation. For any $l \leqslant d$ distinct points $(a_1, b_1), \ldots, (a_l, b_l)$ in $\mathrm{GF}(m)\times\mathrm{GF}(m)$, the number of polynomials $p(x)$ of degree at most $d-1$ satisfying $p(a_1) = b_1, \ldots, p(a_l) = b_l$ is either 0 (if $a_i = a_j$ holds for some $i \neq j$) or is exactly m^{d-l}: this latter number is exactly the number of solutions of the corresponding system of linear equations, with coefficients of p viewed as variables. □

We already know that $(\min, +)$ circuits approximating *minimization* problems on polynomial designs (within any finite factor $r = r(m)$) must be large (Corollary 4.3). Now we show that also the *maximization* problem on the polynomial (m, d)-design is hard to approximate, as long as the allowed approximation factor is smaller than m/d.

Theorem 5.12 (Factor hierarchy theorem)**.** *Let $\mathcal{F}$ be a polynomial (m, d)-design for $1 \leqslant d < m$, and $1/(d + 1) \leqslant \gamma < 1$. Then:*

(i) $\mathsf{Max}_r(\mathcal{F}) \leqslant 3m^2$ *for the factor $r = m/d$, but*
(ii) $\mathsf{Max}_r(\mathcal{F}) \geqslant m^{\gamma d/2}$ *for the factor $r = (1 - \gamma)m/d$.*

That is, the maximization problem on $\mathcal{F}$ can be approximated by a small $(\max, +)$ circuit within the factor $r = m/d$, but the size of approximating $(\max, +)$ circuits drastically increases when this factor is only slightly decreased.

Proof. To show the lower bound (ii), we just apply Lemma 5.10, which yields the lower bound $\mathsf{Max}_r(\mathcal{F}) \geqslant |\mathcal{F}|/\#_l(\mathcal{F})$ for $l = \gamma d/2$. Since $|\mathcal{F}| = m^d$ and, by Proposition 5.11, $\#_l(\mathcal{F}) \leqslant m^{d-l}$, the desired lower bound $\mathsf{Max}_r(\mathcal{F}) \geqslant m^l$ follows.

So, it remains to show the upper bound (i). Given an input weighting x of the points of the grid $\mathrm{GF}(m) \times \mathrm{GF}(m)$, we can first use $m(m - 1)$ max operations to compute m numbers $y_1, \ldots, y_m$, where y_i is the maximum weight of a point in the ith row of the grid. We then apply the $(\max, +)$ circuit for

the top d-of-m selection problem (see Proposition 5.1) to compute the sum W of the *largest* d of the numbers
of d *heaviest* points in the grid with no two in the same row: each y_i picks only one point in the ith row. The main combinatorial property of polynomial designs (Proposition 5.11) implies that these d points are contained in a (unique) set of $\mathcal{F}$. Hence, the found value W cannot exceed the optimal value (the weights are nonnegative). On the other hand, the weight of d heaviest points of an optimal solution $F \in \mathcal{F}$ cannot exceed W. Since $|F| = m$, the weight of this solution cannot exceed $(m/d)W$, as desired. □

Remark 6. The maximizing greedy algorithm also achieves the same approximation factor m/d on the polynomial (m, d)-design: it will also first take the heaviest d points of the grid $\mathrm{GF}(m) \times \mathrm{GF}(m)$, with no two lying in the same row. But this is already the best the greedy algorithm can do.

To show this, take $\epsilon > 0$ arbitrarily small, and set $c := 1/(1 - \epsilon/2) > 1$. Take arbitrary two sets $A \neq B \in \mathcal{F}$, and a subset $S \subset A$ of $|S| = d$ elements. Since $\mathcal{F}$ is an (m, d)-design, S cannot be contained in B. So, give weight $c > 1$ to all elements of S, weight 1 to all elements of $B \setminus S$, and zero weight to the rest. Then the maximizing (best-in) greedy algorithm picks elements of weight c first, gets all $|S| = d$ of them, but then is stuck because no element of weight 1 fits; hence, the greedy algorithm achieves the total weight $c|S| = cd$. But the optimum is at least $|B| = m$. Hence, the approximation factor is at least $m/cd = (1 - \epsilon/2)m/d > (1 - \epsilon)m/d$.

5.9. **Greedy can beat approximating** $(\max, +)$ **circuits.** As Remark 6 shows, Theorem 5.12 does *not* imply that the maximizing greedy algorithm can beat approximating $(\max, +)$ circuits: small $(\max, +)$ circuits can also achieve the greedy approximation factor on designs.

To show that the greedy algorithm can *still* outperform approximating $(\max, +)$ circuits, we consider another maximization problem: maximum weight matchings in k-partite k-uniform hypergraphs. We have a set $V = V_1 \cup \cdots \cup V_k$ of $|V| = mk$ vertices decomposed into k disjoint blocks $V_1, \ldots, V_k$, each of size m. Edges (called also *hyperedges*) are k-tuples $e \in V_1 \times \cdots \times V_k$. The ground set E consists of all $|E| = m^k$ edges. Two edges are *disjoint* if they differ in all k positions. A *matching* is a set of disjoint edges, and is a *perfect matching* if it has the maximum possible number m of edges.

The family $\mathcal{F}_{m,k}$ of feasible solutions of our problem consists of all $|\mathcal{F}_{m,k}| = (m!)^{k-1}$ perfect matchings. So, the maximization problem on $\mathcal{F}_{m,k}$ is, given an assignment of nonnegative weights x_e to the edges $e \in E$, to compute the maximum total weight

$$f(x) = \max\left\{ x_{e_1} + \cdots + x_{e_m} : \ e_i \in E\,, \text{ and } e_i \text{ and } e_j \text{ are disjoint for all } i \neq j \right\}$$

of a perfect matching. Note that in the case $k = 2$, $\mathcal{F}_{m,k}$ consists of perfect matchings in $K_{m,m}$, and the problem is to compute the maximum weight of such a perfect matching.

The greedy algorithm can approximate the maximization problem on $\mathcal{F}_{m,k}$ within the factor k by just always picking the heaviest of the remaining edges, untouched by the partial matching picked so far. On the other hand, we have the following lower bound for $(\max, +)$ circuits approximating this problem.

Theorem 5.13. *Let m be a sufficiently large integer, and $k = k(m)$ be an integer such that $6 \leqslant k \leqslant \log\sqrt{m}$. If $r \leqslant 2^k/9$, then*

$$\mathsf{Max}_r(\mathcal{F}_{m,k}) = 2^{\Omega(\sqrt{m})}\,.$$

Proof. We are going to apply the rectangle bound (Theorem 5.9) with $\gamma := 2/3$; note that then both $\frac{\gamma}{2}$ and $1 - \gamma$ in Eq. (7) are equal to $1/3$. So, take an arbitrary rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ lying below $\mathcal{F} = \mathcal{F}_{m,k}$. Hence, sets in $\mathcal{A}$ and in $\mathcal{B}$ are subsets of (hyper-)edges $e \in V_1 \times \cdots \times V_k$. Since $\mathcal{R}$ lies below our family

$\mathcal{F}$, and $\mathcal{F}$ consist of (perfect) matchings, all sets $A \cup B$ with $A \in \mathcal{A}$ and $B \in \mathcal{B}$ must also be matchings. Take the integer $d := \lceil m/3r \rceil$, and consider the family

$$\mathcal{F}_{\mathcal{R}} = \{ F \in \mathcal{F} : |F \cap A| \geqslant d \text{ and } |F \cap B| \geqslant d \text{ for some } A \in \mathcal{A} \text{ and } B \in \mathcal{B} \}.$$

Our goal is to show a possibly small upper bound $|\mathcal{F}_{\mathcal{R}}| \leqslant h$ on the number of sets in any such family. Then Theorem 5.9 (for the balance parameter $\gamma = 2/3$) gives the lower bound $\mathsf{Max}_r(\mathcal{F}) \geqslant |\mathcal{F}|/h$.

Since the rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ is cross-disjoint, we know that the matchings $A \in \mathcal{A}$ and $B \in \mathcal{B}$ must be edge-disjoint, that is, $A \cap B = \emptyset$ must hold. However, since the sets $A \cup B$ are also matchings ($\mathcal{R}$ lies below $\mathcal{F}$), we actually know that matchings A and B are even *vertex-disjoint*: if $S \subseteq V$ is the set of vertices belonging to at least one edge of a matching in $\mathcal{A}$, and $T \subseteq V$ is the set of vertices belonging to at least one edge of a matching in $\mathcal{B}$, then $S \cap T = \emptyset$ (this is a crucial property). Note that in the proof of the lower bound for designs (Lemma 5.10), the cross-disjointness property of rectangles was *not* used.

So, call a matching $A \subseteq V_1 \times \cdots \times V_k$ an *S-matching* if $A \subseteq (V_1 \cap S) \times \cdots \times (V_k \cap S)$ holds, that is, if edges of A only match vertices of S; *T-matchings* are defined similarly. By the definition of $\mathcal{F}_{\mathcal{R}}$, every perfect matching $F \in \mathcal{F}_{\mathcal{R}}$ has at least d edges lying in $(V_1 \cap S) \times \cdots \times (V_k \cap S)$, and at least d edges lying in $(V_1 \cap T) \times \cdots \times (V_k \cap T)$. In particular, every perfect matching $F \in \mathcal{F}_{\mathcal{R}}$ must contain at least one matching $A \cup B$, where A is an S-matching with $|A| = d$ edges and B is a T-matching with $|B| = d$ edges. It therefore suffices to upper-bound the number of perfect matchings F with this property.

We can pick any such pair (A, B) as follows. Let $S_i = S \cap V_i$ and $T_i = T \cap V_i$ for $i = 1, \ldots, k$. We can assume that each of these $2k$ sets has at least d vertices, for otherwise none of the S-matchings or of the T-matchings could have $\geqslant d$ edges, implying that $\mathcal{F}_{\mathcal{R}} = \emptyset$.

- Pick in each S_i a subset $S_i' \subseteq S_i$ of $|S_i'| = d$ vertices, and in each T_i a subset $T_i' \subseteq T_i$ of $|T_i'| = d$ vertices. There are at most
$$\prod_{i=1}^{k} \binom{m_i}{d} \binom{m - m_i}{d} \leqslant \binom{m}{2d}^k$$
possibilities to do this, where $m_i = |S_i|$.
- Pick a perfect matching A in $S_1' \times \cdots \times S_k'$ and a perfect matching B in $T_1' \times \cdots \times T_k'$. There are only $\left[(d!)^{k-1}\right]^2 = (d!)^{2(k-1)}$ possibilities to do this.

After a pair (A, B) of matchings is picked, there are at most $[(m - 2d)!]^{k-1}$ possibilities to extend $A \cup B$ to a perfect matching. Thus,

$$|\mathcal{F}_{\mathcal{R}}| \leqslant \binom{m}{2d}^k (d!)^{2(k-1)} [(m-2d)!]^{k-1} = \binom{m}{2d} \left[m! \cdot \binom{2d}{d}^{-1} \right]^{k-1},$$

where the equality follows because $\binom{m}{2d} = m!/(2d)!(m-2d)!$ and $(2d)!/(d!)^2 = \binom{2d}{d}$. Since there are $|\mathcal{F}| = (m!)^{k-1}$ perfect matchings, the rectangle bound (Theorem 5.9) yields the following lower bound on $t = \mathsf{Max}_r(\mathcal{F})$:

$$t \geqslant \frac{|\mathcal{F}|}{|\mathcal{F}_{\mathcal{R}}|} \geqslant \frac{\binom{2d}{d}^{k-1}}{\binom{m}{2d}} \geqslant \left(\frac{2^{2d}}{d} \right)^{k-1} \cdot \left(\frac{2d}{\mathrm{e}m} \right)^{2d} = \frac{1}{d^{k-1}} \left(\frac{2^k d}{\mathrm{e}m} \right)^{2d} \geqslant \frac{1}{d^{k-1}} \left(\frac{2^k}{3\mathrm{e}r} \right)^{2d},$$

where the second inequality follows from the inequalities $\binom{m}{2d} \leqslant (\mathrm{e}m/2d)^{2d}$ and $\binom{2d}{d} \geqslant 2^{2d}/\sqrt{4d} \geqslant 2^{2d}/d$, and the last inequality follows because (by our choice) $d = \lceil m/3r \rceil \geqslant m/3r$. Our approximation factor is $r = 2^k/9$. Since clearly $d \leqslant m$, we have a lower bound

$$t \geqslant \left(\frac{3}{\mathrm{e}}\right)^{6m/2^k} \cdot d^{-k} \geqslant 2^{0.8m/2^k - k\log m}.$$

From our assumption $k \leqslant \log\sqrt{m}$, we have $m/2^k \geqslant \sqrt{m} \gg k\log m$, and the desired lower bound $t \geqslant 2^{\Omega(m/2^k)} \geqslant 2^{\Omega(\sqrt{m})}$ follows. □

6. What do approximating tropical circuits produce?

If we know that a tropical circuit *approximates* a given optimization (minimization or maximization) problem within a given factor, what can then be said about the set of vectors *produced* by that circuit? Using *elementary arguments*, we partially answered this question in Lemmas 4.2 and 5.5: we gave properties, which the sets of produced vectors must *necessarily* have (these properties were already sufficient for our purposes). We will now use convexity arguments to give properties of produced sets that are also *sufficient* for circuits to approximate given problems.

6.1. **A version of Farkas' lemma.** Recall that a vector $c \in \mathbb{R}^n$ is a *convex combination* (or a *weighted average*) of vectors[1] $\vec{b}_1, \dots, \vec{b}_m$ in $\mathbb{R}^n$ if there are real scalars $\lambda_1, \dots, \lambda_m \geqslant 0$ such that

$$\lambda_1 + \cdots + \lambda_m = 1 \quad \text{and} \quad c = \lambda_1 \cdot \vec{b}_1 + \cdots + \lambda_m \cdot \vec{b}_m .$$

It is easy to see the following *averaging property*: for every vector $x \in \mathbb{R}^n$ and every convex combination c of vectors in $\boldsymbol{B}$, we have $\min_{b\in B}\langle b, x\rangle \leqslant \langle c, x\rangle \leqslant \max_{b\in B}\langle b, x\rangle$.

We will need the following formulation of Farkas' lemma due to Fan [8, Theorem 4], see also [27, Corollary 7.1h].

Lemma 6.1 (Farkas' lemma [8])**.** *Let $u, \vec{u}_1, \dots, \vec{u}_m \in \mathbb{R}^n$, and $\alpha, \alpha_1, \dots, \alpha_m \in \mathbb{R}$. The following two assertions are equivalent.*

1. *$\forall y \in \mathbb{R}^n$ inequalities $\langle \vec{u}_1, y\rangle \geqslant \alpha_1, \dots, \langle \vec{u}_m, y\rangle \geqslant \alpha_m$ imply $\langle u, y\rangle \geqslant \alpha$.*
2. *$\exists\ \lambda_1, \dots, \lambda_m \in \mathbb{R}_+$ such that $u = \sum_i \lambda_i \vec{u}_i$ and $\alpha \leqslant \sum_i \lambda_i\alpha_i$.*

This relates optimization with convex combinations.

Lemma 6.2. *For any vectors $a, \vec{a}_1, \dots, \vec{a}_m \in \mathbb{R}^n$ the following two assertions are equivalent.*

1. *$\forall x \in \mathbb{R}^n_+ :\ \langle a, x\rangle \geqslant \min_i\langle \vec{a}_i, x\rangle$.*
2. *$\exists\ \lambda_1, \dots, \lambda_m \in \mathbb{R}_+ :\ \ \sum_i \lambda_i = 1$ and $a \geqslant \sum_i \lambda_i \vec{a}_i$.*

Proof. The implication (2) ⇒ (1) follows directly from the aforementioned averaging property of convex combinations. To show the converse implication (1) ⇒ (2), observe that the assertion (1) is equivalent to the assertion that the set of inequalities $\langle \vec{a}_i, x\rangle \geqslant z$ and $\langle \vec{e}_j, x\rangle \geqslant 0$ for $i = 1, \dots, m$ and $j = 1, \dots, n$ implies the inequality $\langle a, x\rangle \geqslant z$. We use the inequalities $\langle \vec{e}_j, x\rangle \geqslant 0$ to ensure that we only consider vectors x in $\mathbb{R}^n_+$ (with no negative entries).

By taking $y = (x, z)$, $u = (a, -1)$, $\vec{u}_i = (\vec{a}_i, -1)$ for $i = 1, \dots, m$, and $\vec{u}_{m+j} = (\vec{e}_j, 0)$ for $j = 1, \dots, n$, the above assertion turns into the assertion that for every vector y in $\mathbb{R}^{n+1}$, the system of inequalities $\langle \vec{u}_i, y\rangle \geqslant 0$ for $i = 1, \dots, m+n$, implies the inequality $\langle u, y\rangle \geqslant 0$. Then, by Lemma 6.1, there exist $\lambda_1, \dots, \lambda_{m+n} \in \mathbb{R}_+$ such that

$$(a, -1) = \sum_{i=1}^{m} \lambda_i(\vec{a}_i, -1) + \sum_{j=1}^{n} \lambda_{m+j}(\vec{e}_j, 0).$$

[1] We use the arrow notation $\vec{b}_i$ for vectors only when they are indexed.

This yields $\lambda_1 + \cdots + \lambda_m = 1$ and $a \geqslant \sum_i \lambda_i \vec{a}_i$, as desired. ☐

The following direct consequence of Lemma 6.2 compares the values of optimization problems. For a set U of real vectors, let $\mathrm{Conv}(U)$ denote the set of all convex combinations of vectors in U, that is, the convex hull of U. Say that a set $U \subseteq \mathbb{R}^n$ *lies above* a set $V \subseteq \mathbb{R}^n$ if $\forall u \in U\ \exists v \in V: \ u \geqslant v$, and that U *lies below* V if $\forall u \in U\ \exists v \in V: \ u \leqslant v$.

Lemma 6.3. *Let $U, V \subset \mathbb{R}^n$ be finite sets of vectors. Then*

(1) $\forall x \in \mathbb{R}^n_+ : \ \min_{u \in U} \langle u, x\rangle \geqslant \min_{v \in V} \langle v, x\rangle$ *if and only if U lies above* $\mathrm{Conv}(V)$*;*
(2) $\forall x \in \mathbb{R}^n_+ : \ \max_{u \in U} \langle u, x\rangle \leqslant \max_{v \in V} \langle v, x\rangle$ *if and only if U lies below* $\mathrm{Conv}(V)$.

Claim (1) follows directly from Lemma 6.2. The second claim (2) also follows from Lemma 6.2 by using the equality $\max(x, y) = -\min(-x, -y)$.

6.2. **Consequences for tropical circuits.** Recall that the maximization (resp., minimization) problem on a given set $A \subset \mathbb{N}^n$ of feasible solutions is, for every input weighting $x \in \mathbb{R}^n_+$, to compute the maximum (resp., minimum) weight $\langle a, x\rangle = a_1x_1 + \cdots + a_nx_n$ of a feasible solution $a \in A$.

The consequence of Farkas' lemma (Lemma 6.3) directly yields the following complete characterization of the properties of sets of vectors produced by approximating $(\max, +)$ circuits.

Lemma 6.4 (Maximization)**.** *Let $A \subset \mathbb{N}^n$ be some finite set of vectors, Φ be a* $(\max, +)$ *circuit, and $B \subset \mathbb{N}^n$ the set of vectors produced by Φ. Then the following two assertions are equivalent.*

(1) Φ *approximates the maximization problem on A within a factor r.*
(2) *B lies below* $\mathrm{Conv}(A)$ *and $\frac{1}{r} \cdot A$ lies below* $\mathrm{Conv}(B)$.

Proof. By Lemma 3.2, we can assume the circuit Φ is constant-free. Hence, by Proposition 3.1, the circuit solves the maximization problem of the form $\Phi(x) = \max_{b \in B} \langle b, x\rangle$. The maximization problem on A is of the form $f(x) = \max_{a \in A}\langle a, x\rangle$. The circuit approximates the maximization problem on A within factor r if and only if $\frac{1}{r} \cdot f(x) \leqslant \Phi(x) \leqslant f(x)$ holds for all weightings $x \in \mathbb{R}^n_+$.

When applied with $U = B$ and $V = A$, Lemma 6.3(2) implies that the inequality $\Phi(x) \leqslant f(x)$ holds if and only if B lies below $\mathrm{Conv}(A)$. When applied with $U = \frac{1}{r} \cdot A$ and $V = B$, this lemma implies that the inequality $\frac{1}{r} \cdot f(x) \leqslant \Phi(x)$ holds if and only if $\frac{1}{r} \cdot A$ lies below $\mathrm{Conv}(B)$. ☐

We say that a set $U \subseteq \mathbb{R}^n$ *lies tightly above* the convex hull $\mathrm{Conv}(V)$ of a set $V \subseteq \mathbb{R}^n$ if for every vector $u \in U$, the inequality $u \geqslant c$ holds for some convex combination c of vectors $v \in V$ with the *same* support as that of u. That is, we now additionally have that *none* of the vectors $v \in V$ in the convex combination c has a zero in a position i where $u_i \neq 0$.

Lemma 6.5 (Minimization)**.** *Let $A \subset \mathbb{N}^n$ be some finite set of vectors, Φ be a* $(\min, +)$ *circuit, and $B \subset \mathbb{N}^n$ the set of vectors produced by Φ.*

Then the following two assertions are equivalent.

(1) Φ *approximates the minimization problem on A within a factor r.*
(2) *B lies above* $\mathrm{Conv}(A)$ *and $r \cdot A$ lies above* $\mathrm{Conv}(B)$.

If $A \subseteq \{0, 1\}^n$ and A is an antichain, then these assertions are equivalent to:

3. *B lies above A and $r \cdot A$ lies tightly above* $\mathrm{Conv}(B)$.

Proof. By Lemma 3.2, we can assume the circuit Φ is constant-free. Hence, by Proposition 3.1, the circuit solves the minimization problem of the form $\Phi(x) = \min_{b \in B} \langle b, x\rangle$. The minimization problem on A is of the form $f(x) = \min_{a \in A}\langle a, x\rangle$. The circuit approximates the minimization problem on A within factor r if and only if $f(x) \leqslant \Phi(x) \leqslant r \cdot f(x)$ holds for all weightings $x \in \mathbb{R}^n_+$. When applied with $U = B$ and $V = A$, Lemma 6.3(1) implies that the inequality $\Phi(x) \geqslant f(x)$ holds if and only if B lies above $\mathrm{Conv}(A)$. When applied with $U = r \cdot A$ and $V = B$, this lemma implies that the inequality $r \cdot f(x) \geqslant \Phi(x)$ holds if and only if $r \cdot A$ lies above $\mathrm{Conv}(B)$. This shows the equivalence of the assertions (1) and (2).

Suppose now that $A \subseteq \{0,1\}^n$, and that A is an antichain. The implication (3) ⇒ (2) is obvious. So, assume that the set B has property (2), i.e., that B lies above $\mathrm{Conv}(A)$ and $r \cdot A$ lies above $\mathrm{Conv}(B)$.

To show that B lies above the set A (not only above its convex hull), take an arbitrary vector $b \in B$. Since B lies above $\mathrm{Conv}(A)$, there must be a vector $a \in A$ and a scalar $0 < \lambda \leqslant 1$ such that $b \geqslant \lambda \cdot a$. Since a is a 0-1 vector, and b is a nonnegative *integer* vector, $b \geqslant a$ must hold.

To show that $r \cdot A$ lies tightly above $\mathrm{Conv}(B)$, take an arbitrary vector $a \in A$. Since, by (2), the set $r \cdot A$ lies above $\mathrm{Conv}(B)$, the inequality $r \cdot a \geqslant c$ must hold for some convex combination $c = \lambda_1 \cdot \vec{b}_1 + \cdots + \lambda_m \cdot \vec{b}_m$ of vectors in B, where all scalars λ_i are positive. It remains to show that then $S_{\vec{b}_i} = S_a$ holds for all $i = 1, \ldots, m$, i.e., that all the vectors in this convex combination have the same support as our vector a.

Since (as we have just shown) the set B lies above the set A, there must be (not necessarily distinct) vectors $\vec{a}_1, \ldots, \vec{a}_m$ in A such that $\vec{b}_i \geqslant \vec{a}_i$ for all $i = 1, \ldots, m$ and, hence, $c = \sum_{i=1}^m \lambda_i \cdot \vec{b}_i \geqslant \sum_{i=1}^m \lambda_i \cdot \vec{a}_i$. The inequality $r \cdot a \geqslant c$ implies that $S_a \supseteq S_{\vec{b}_i} \supseteq S_{\vec{a}_i}$ must hold for all i. Since A is an antichain and consists of only 0-1 vectors, this implies $\vec{a}_i = a$ for all i. We thus have $S_{\vec{b}_i} = S_a$ for all $i = 1, \ldots, m$, as desired. □

Remark 7 (Approximation using arithmetic circuits). Lemma 6.5 implies that in order to show that the minimization problem on an antichain $\mathcal{F} \subseteq 2^{[n]}$ can be r-approximated by a $(\min, +)$ circuit of size t, it is enough to design a monotone *arithmetic* $(+, \times)$ circuit Φ of size $\leqslant t$ such that the polynomial computed by this circuit has the following two properties:

(1) for every monomial $\prod_{i \in T} x_i^{d_i}$ there is a set $S \in \mathcal{F}$ with $S \subseteq T$;
(2) for every set $S \in \mathcal{F}$ there is a monomial $\prod_{i \in T} x_i^{d_i}$ with $T = S$ and all $d_i \leqslant r$.

Indeed, property (1) ensures that the set B of vectors produced by the arithmetic circuit Φ lies above the set A of characteristic 0-1 vectors of sets in $\mathcal{F}$, while property (2) ensures that the set $r \cdot A$ lies tightly above $\mathrm{Conv}(B)$. By Lemma 6.5, the $(\min, +)$ version of Φ r-approximates the minimization problem on $\mathcal{F}$.

7. A TIGHT BOOLEAN BOUND FOR $(\min, +)$ CIRCUITS

In Section 4.1, we have shown (Theorem 4.3) that the monotone boolean circuit complexity of the decision versions of minimization problems is a *lower bound* on the size of $(\min, +)$ circuits approximating these problems within *any* finite approximation factor $r \geqslant 1$. Lemma 6.5 will allow us to take the factor r into account, that is, to show that approximating $(\min, +)$ circuits and monotone boolean circuits are even more tightly related (as given in Theorem 7.1 below). For this purpose, we introduce the concept of "semantic degree" of monotone boolean circuits.

7.1. **Semantic degree of boolean circuits.** A *minterm* of a monotone boolean function $f(x_1, \ldots, x_n)$ is a vector $a \in \{0,1\}^n$ such that $f(a) = 1$, but $f(a') = 0$ for any vector a' obtained by switching any single 1-entry of a to 0. The boolean function *defined* by a finite set $A \subset \mathbb{N}^n$ of vectors is of the form

$$f_A(x) = \bigvee_{a \in A} \bigwedge_{i \in S_a} x_i \,,$$

where, as before, $S_a = \{i : a_i \neq 0\}$ is the support of vector a. In particular, if $A \subset \{0,1\}^n$ is the set of minterms of a boolean function, then this function is of the form f_A (is defined by the set of its minterms).

A monotone boolean $(\vee, \wedge)$ circuit ϕ for a boolean function f_A (defined by its set A of minterms) not only *computes* the function f_A, but also *produces* (purely syntactically) some finite set $B \subset \mathbb{N}^n$ of vectors, as given in Section 3.2. By Proposition 3.1, the circuit ϕ computes the boolean function f_B defined by the set B. Since the circuit ϕ *computes* the function f_A, we know that $f_B(x) = f_A(x)$ must hold for all $x \in \{0,1\}^n$. The "semantic degree" of the circuit ϕ (motivated by Lemma 6.5) gives an upper bound on the magnitudes of entries of particular (not all) vectors of the set B.

Namely, we define the *semantic degree*, $\deg(\phi)$, of ϕ as the minimum real number r such that the set $r \cdot A$ lies tightly above the convex hull $\mathrm{Conv}(B)$ of the set B produced by the circuit ϕ. Recall that this means that for every minterm $a \in A$ there are vectors $\vec{b}_1, \ldots, \vec{b}_m \in B$ and positive scalars $\lambda_1, \ldots, \lambda_m$ such that $\lambda_1 + \cdots + \lambda_m = 1$, $S_{\vec{b}_1} = \ldots = S_{\vec{b}_m} = S_a$ and

$$a \leqslant \lambda_1 \cdot \vec{b}_1 + \cdots + \lambda_m \cdot \vec{b}_m \leqslant r \cdot a\,. \tag{8}$$

We use the adjective "semantic" because $\deg(\phi)$ depends on the *function* computed by ϕ, that is, on the set A of minterms of this function. Note that the first inequality in Eq. (8) always holds because a is a 0-1 vector, and $S_{\vec{b}_i} = S_a$ holds for all vectors b_i (we included this inequality just for clarity of the concept).

7.2. **The converse of the boolean bound (Theorem 4.3).** For a finite set $A \subset \mathbb{N}^n$, let $\mathsf{Bool}_r(A)$ denote the minimum size of a monotone boolean circuit of semantic degree at most r computing the boolean function f_A defined by A.

Theorem 7.1 (Tight boolean bound)**.** *If $A \subset \{0,1\}^n$ is an antichain, then $\mathsf{Min}_r(A) = \mathsf{Bool}_r(A)$ holds for every $r \geqslant 1$.*

Proof. To show $\mathsf{Min}_r(A) \leqslant \mathsf{Bool}_r(A)$, take a monotone boolean $(\vee, \wedge)$ circuit ϕ of semantic degree r computing the boolean function f_A defined by A. We can assume that the circuit is constant-free: 0 and 1 are the only possible constants, and they can be trivially eliminated from the circuit. Let $B \subset \mathbb{N}^n$ be the set of vectors produced by the circuit ϕ. By Proposition 3.1, the circuit ϕ computes the boolean function f_B defined by this set, that is, $f_B(x) = f_A(x)$ holds for all inputs $x \in \{0,1\}^n$.

Let Φ be the (also constant-free) tropical $(\min, +)$ version of the boolean circuit ϕ obtained from ϕ by replacing each $\vee$-gate by a min-gate, and each $\wedge$-gate by a $+$-gate. The $(\min, +)$ circuit Φ produces the same set B.

Since $f_B(x) \leqslant f_A(x)$ must hold for all inputs $x \in \{0,1\}^n$, for every $b \in B$ there must be a vector $a \in A$ with $S_b \supseteq S_a$. Since vectors in A are 0-1 vectors, this latter inclusion yields $b \geqslant a$. Thus, the set B lies above A and, hence, also above $\mathrm{Conv}(A)$. Since the circuit ϕ has semantic degree r, we additionally have that the set $r \cdot A$ lies above $\mathrm{Conv}(B)$. By Lemma 6.5, the circuit Φ approximates the minimization problem on A within the factor r.

To show $\mathsf{Bool}_r(A) \leqslant \mathsf{Min}_r(A)$, take a tropical $(\min, +)$ circuit Φ approximating the minimization problem on A within the factor r, and let $B \subset \mathbb{N}^n$ be the set of vectors produced by Φ. By Lemma 3.2, we can assume that the circuit Φ is constant-free. Let ϕ be the boolean version of the tropical circuit Φ obtained by replacing each min-gate by an $\vee$-gate, and each $+$-gate by an $\wedge$-gate. (Recall that min and $\vee$ are "additions" and $+$ and $\wedge$ are "multiplications" in the corresponding semirings.) The circuit ϕ produces the same set B. By Lemma 6.5, we know that the set B has the following two properties:

(i) B lies above A ;
(ii) $r \cdot A$ lies tightly above $\mathrm{Conv}(B)$.

By property (ii), the semantic degree of the boolean circuit ϕ is at most r. On the other hand, property (i) implies that the support of every vector $b \in B$ contains the support of at least one vector $a \in A$, and property (ii) implies that the support of every vector $a \in A$ contains the support of at least one vector $b \in B$. In terms of Proposition 4.1, this means that the set B is similar to the set A. Since (by Proposition 3.1) the circuit computes a boolean function f_B defined by the set B, Proposition 4.1 itself implies that ϕ computes the boolean function defined by the set A. □

7.3. **Bounds on semantic degree.** An *r-bounded copy* of a boolean vector $a \in \{0,1\}^n$ is an integer vector $b \in \mathbb{N}^n$ which has the same nonzero positions as a, and every nonzero position of b is at most r. In particular, the unique 1-bounded copy of a is the vector a itself. Recall that $|a| = \langle a, a\rangle$ is the number of ones in a 0-1 vector a.

Proposition 7.2. *Let ϕ be a monotone boolean circuit computing a boolean function f, $A \subset \{0,1\}^n$ the set of minterms of f, and $B \subset \mathbb{N}^n$ the set of vectors produced by the circuit ϕ. Then the following holds.*

(1) $\deg(\phi) = 1$ *if and only if $A \subseteq B$.*
(2) $\deg(\phi) \leqslant r$ *holds if for every $a \in A$ the set B contains at least one r-bounded copy of a.*
(3) *If* $\deg(\phi) \leqslant r$ *holds, then for every $a \in A$ the set B contains at least one s-bounded copy of a for $s \leqslant r|a| - |a| + r$.*

Note that (1) is a special case of (2) and (3) for $r = 1$

Proof. The "if" direction in (1) is obvious. The "only if" direction follows from a simple observation: a convex combination $\lambda_1 b_1 + \cdots + \lambda_m b_m$ of positive *integers* b_i with all $\lambda_i > 0$ can be equal 1 only if $b_1 = \ldots = b_m = 1$.

Claim (2) is also obvious, because for every 0-1 vector a, the inequality $b \leqslant r \cdot a$ holds for *every* r-bounded copy b of a.

To show claim (3), assume that $\deg(\phi) \leqslant r$. Take any minterm $a \in A$ and let $m = |a|$ be the number of ones in a. By the definition of the semantic degree, we know that there must be a convex combination $c = \sum_{i=1}^{l} \lambda_i \cdot \vec{b}_i$ of vectors $b_i \in B$ with all supports $S_{\vec{b}_i} = S_a$ such that $c \leqslant r \cdot a$ holds. By Carathéodory's theorem [5], if a vector is in the convex hull of some set $P \subseteq \mathbb{R}^m$ of vectors, then this vector can be written as a convex combination of $m+1$ or fewer vectors in P. So, by taking $P = \{\vec{b}_1, \ldots, \vec{b}_l\}$, we can assume that $l \leqslant |a| + 1 = m + 1$.

Consider the vectors $\vec{b}'_i := \vec{b}_i - a \geqslant \vec{0}$ (the vectors $\vec{b}'_i$ are nonnegative, because vectors $\vec{b}_i$ have the same support as a). Then $c = a + c'$ with $c' := \sum_{i=1}^{l} \lambda_i \vec{b}'_i = c - a \leqslant r \cdot a - a = (r-1) \cdot a$. Since $\lambda_1 + \cdots + \lambda_l = 1$, there must be an i such that $\lambda_i \geqslant 1/l \geqslant 1/(m+1)$. From $\lambda_i \cdot \vec{b}'_i \leqslant c' \leqslant (r-1) \cdot a$, and since a is a 0-1 vector, we have that all entries of vector $\vec{b}'_i$ must be at most $(r-1)/\lambda_i \leqslant (r-1)(m+1)$. Hence, all entries of the vector $\vec{b}_i$ are at most $(r-1)(m+1) + 1 = rm - m + r$, as desired. □

Remark 8. An apparent advantage of Proposition 7.2 is that it avoids the somewhat involved definition of the semantic degree via convex hulls. Items (i) and (ii) may be useful when proving *upper* bounds, while items (i) and (iii) may be useful when proving *lower* bounds on the size of monotone boolean circuits of bounded semantic degree.

Remark 9. Note that the upper bound $s \leqslant r|a| - |a| + r$ in item (iii) of Proposition 7.2 cannot be substantially improved. Take $m = |a|$ vectors $\vec{b}_i := a + m(r-1)\vec{e}_i$, and let all $\lambda_i := 1/m$. Then the convex combination $c = \sum_{i=1}^{m} \lambda_i \cdot \vec{b}_i = a + (r-1) \cdot a = r \cdot a$ satisfies $c \leqslant r \cdot a$, but every vector $\vec{b}_i$ in this combination has $s = 1 + m(r-1) = r|a| - |a| + 1$ as one of it entries.

The following example shows that the semantic degree of monotone boolean circuits can be small even when some vectors produced by the circuit have very large entries.

Example 1 (Shortest paths). Let A be the set of characteristic 0-1 vectors of all simple paths in K_n between two fixed vertices s and t. Then the boolean function f_A defined by A is the s-t connectivity function STCONN on n-vertex graphs. The Bellman–Ford pure DP algorithm for the shortest s-t path problem gives us a monotone boolean $(\vee, \wedge)$ circuit ϕ of size $O(n^3)$ computing the boolean function f_A. The circuit has gates u_j^l at which the existence of a path from vertex s to vertex j with at most l edges is detected. Then $u_j^1 = x_{s,j}$ for all $j \neq s$, and the recursion of Bellman–Ford is to compute u_j^{l+1} as the OR of u_j^l and all $u_i^l \wedge x_{i,j}$ for $i \notin \{s, j\}$. The output gate is u_t^{n-1}.

The vectors of the set $B \subset \mathbb{N}^n$ produced by the Bellman–Ford circuit ϕ correspond not to (simple) paths but rather to *walks* of length at most $n - 1$ from s to t. Since a walk can traverse the same edge many times, some vectors in B have entries much larger than 1. Still, by Proposition 7.2(1), $\deg(\phi) = 1$ holds: every (simple) s-t path is also a walk of length at most $n - 1$, implying that $A \subseteq B$.

7.4. **Semantic versus syntactic degree.** The standard, "syntactic" definition of the degree is the following. Each input node holding a variable has degree 1. The degree of an OR gate is the *maximum* of the degrees of its input gates, and the degree of an AND gate is the *sum* of the degrees of its input gates. The following proposition shows that the semantic degree never exceeds the syntactic degree.

Proposition 7.3. *Let ϕ_1 and ϕ_2 be any two monotone boolean circuits. Then*

$$\deg(\phi_1 \vee \phi_2) \leqslant \max\left\{\deg(\phi_1), \deg(\phi_2)\right\} \text{ and } \deg(\phi_1 \wedge \phi_2) \leqslant \deg(\phi_1) + \deg(\phi_2).$$

Proof. For $i \in \{1, 2\}$, let $A_i \subseteq \{0, 1\}^n$ be the set of minterms of the boolean function computed by ϕ_i, and let $B_i \subset \mathbb{N}^n$ be the set of vectors produced by ϕ_i. Let $r_i = \deg(\phi_i)$ be the semantic degree of ϕ_i.

Take an arbitrary minterm a of ϕ. If $\phi = \phi_1 \vee \phi_2$, then $B = B_1 \cup B_2$ is the union of the set B_1 and B_2, and $a \in A_i$ for some $i \in \{1, 2\}$. We know that $c \leqslant r_i \cdot a$ must hold for some vector c in $\mathrm{Conv}(B_i) \subseteq \mathrm{Conv}(B)$. So, $\deg(\phi_1 \vee \phi_2) \leqslant \max\{r_1, r_2\}$ in this case.

If $\phi = \phi_1 \wedge \phi_2$, then $B = B_1 + B_2$ is the Minkowski sum of the sets B_1 and B_2, and $a = \vec{a}_1 \vee \vec{a}_2$ is a componentwise OR of some minterms $\vec{a}_1 \in A_1$ and $\vec{a}_2 \in A_2$. We know that $\vec{c}_1 \leqslant r_1 \cdot \vec{a}_1$ and $\vec{c}_2 \leqslant r_i \cdot \vec{a}_2$ must hold for some vectors $\vec{c}_1 \in \mathrm{Conv}(B_1)$ and $\vec{c}_2 \in \mathrm{Conv}(B_2)$. A well-known property of Minkowski sums is that $\mathrm{Conv}(B_1) + \mathrm{Conv}(B_2) = \mathrm{Conv}(B_1 + B_2)$ always holds. Hence, the vector $c = \vec{c}_1 + \vec{c}_2$ belongs to $\mathrm{Conv}(B)$ and satisfies $c = \vec{c}_1 + \vec{c}_2 \leqslant r_1 \cdot \vec{a}_1 + r_2 \cdot \vec{a}_2 \leqslant r_1 \cdot a + r_2 \cdot a = (r_1 + r_2) \cdot a$. So, $\deg(\phi_1 \wedge \phi_2) \leqslant r_1 + r_2$ holds in this case. □

The following example illustrates that, together with Proposition 7.3, the upper bound $\mathsf{Min}_r(A) \leqslant \mathsf{Bool}_r(A)$ given by Theorem 7.1 allows one to show that some minimization problems *can* be approximated by small $(\min, +)$ circuits within (large) but finite factors by proving upper bounds for monotone *boolean* circuits of bounded semantic degree. Recall that *some* minimization problems cannot be approximated by $(\min, +)$ circuits of polynomial size within *any* finite factor $r = r(n)$ at all (Section 4.2).

Example 2 (Spanning trees). In the *minimum weight spanning tree* problem $\mathcal{T}_n$, we are given an assignment of nonnegative real weights to the edges of K_n, and the goal is to compute the minimum weight of a spanning tree in K_n; the weight of a subgraph is the sum of weights of its edges. We have shown in [17] that $\mathsf{Min}_1(\mathcal{T}_n) = 2^{\Omega(\sqrt{n})}$.

On the other hand, the decision version of this problem is the graph connectivity problem. Using the (pure) DP algorithm of Bellman and Ford, for every pair (s, t) of vertices, the s-t connectivity problem can be solved by a monotone boolean circuit $\phi_{s,t}$ of size $O(n^3)$ and semantic degree $\deg(\phi_{s,t}) = 1$

(see Example 1). So, the connectivity problem can be solved by the circuit $\phi_{1,2} \wedge \phi_{1,3} \wedge \cdots \wedge \phi_{1,n}$ of size $O(n^4)$. By Proposition 7.3, the circuit has semantic degree $r \leqslant n-1$. Theorem 7.1 implies that $\mathsf{Min}_r(\mathcal{T}_n) = O(n^4)$ holds for some *finite* factor $r \leqslant n-1$.

8. Conclusion and open problems

Developing a workable taxonomy of existing algorithmic paradigms in *rigorous* mathematical terms is an important long-term goal. When pursuing this goal, the main difficulty is to prove unconditional *lower bounds* on the complexity of algorithms from particular classes, that is, to prove lower bounds not relying on unproven complexity assumptions like $\mathbf{P} \neq \mathbf{NP}$.

In this paper, we consider the class of all pure DP algorithms, take tropical circuits as their natural mathematical model, and prove the first non-trivial (even super-polynomial) unconditional lower bounds for *approximating* pure DP algorithms in this model. The results imply that the approximation powers of greedy and pure DP algorithms are incomparable. Some interesting questions still remain open.

Given a family $\mathcal{F} \subseteq 2^{[n]}$ of feasible solutions, and an approximation factor $r \geqslant 1$ let, as before, $\mathsf{Max}_r(\mathcal{F})$ denote the minimum number of gates in a $(\max, +)$ circuit approximating the maximization problem $f(x) = \max_{S\in\mathcal{F}} \sum_{i\in S} x_i$ on $\mathcal{F}$ within the factor r. In the case of minimization problems and $(\min, +)$ circuits, the corresponding complexity measure is $\mathsf{Min}_r(\mathcal{F})$.

8.1. Minimization.

We have shown in Theorem 4.6 that there *exist* a lot of monotone boolean functions f such that minterms of f are bases of a matroid, and f requires monotone boolean circuits of exponential size. But we do not know of any *explicit* matroid for which the corresponding boolean function requires large monotone boolean circuits.

Problem 1. *Prove a super-polynomial lower bound on the monotone boolean circuit complexity of an* explicit *boolean function whose minterms are bases of a matroid.*

Let $\mathcal{T}_n$ be the family of all spanning trees in a complete n-vertex graph K_n. Since $\mathcal{T}_n$ is a matroid, both minimization and maximization problems can be solved exactly (within factor $r = 1$) by the greedy algorithm. On the other hand, we know that $\mathsf{Min}_1(\mathcal{T}_n) = 2^{\Omega(\sqrt{n})}$ [17]. We also know that $\mathsf{Min}_r(\mathcal{T}_n) = O(n^4)$ holds if factor $r = n-1$ is allowed (Example 2).

Problem 2. *Is* $\mathsf{Min}_2(\mathcal{T}_n)$ *polynomial in n?*

8.2. Maximization.

The next question concerns the *maximization* problem on the matroid $\mathcal{T}_n$ of spanning trees in K_n. We know that, for factor $r = 1$, we have $\mathsf{Max}_1(\mathcal{T}_n) = 2^{\Omega(\sqrt{n})}$ [17].

Problem 3. *Is* $\mathsf{Max}_2(\mathcal{T}_n)$ *polynomial in n?*

In Theorem 5.13, we considered the maximum weight problem on k-partite *hypergraphs*. For $k = 2$, the calculations made in the proof of Theorem 5.13 result in a trivial bound. This rises a natural question: does a similar lower bound hold also for matchings in *bipartite* graphs? Let $\mathcal{M}_n$ be the family of all perfect matchings in a complete bipartite $n \times n$ graph. The greedy algorithm can approximate the maximization problem on $\mathcal{M}_n$ within the factor 2.

Problem 4. *Is* $\mathsf{Max}_2(\mathcal{M}_n)$ *polynomial in n?*

We have shown in Theorem 4.6 that the *minimization* problem on many matroids cannot be efficiently approximated by pure DP algorithms within any finite factor r. But what happens with *maximization* problems? By Proposition 5.2, we know that there are a lot of matroids $\mathcal{F} \subseteq 2^{[n]}$ such that $\mathsf{Max}_1(\mathcal{F}) = 2^{\Omega(n)}$ but $\mathsf{Max}_r(\mathcal{F}) \leqslant n^2$ holds already for $r = 1 + o(1)$.

Problem 5. *Are there matroids, on which the maximization problem cannot be efficiently approximated by pure DP algorithms within some factor $r \geqslant 1+\epsilon$ for a constant $\epsilon > 0$?*

Note that here we only ask for the mere *existence*. By Proposition 5.2, the answer is "yes" for $r = 1$. But this proposition and Proposition 5.3 indicate that direct counting arguments may fail to answer this question for slightly larger approximation factors r.

8.3. **Tradeoffs between minimization and maximization.** If a family $\mathcal{F}$ of feasible solutions is *uniform* (all sets of $\mathcal{F}$ have the same cardinality), then $\mathsf{Min}_1(\mathcal{F}) = \mathsf{Max}_1(\mathcal{F})$ (see, for example, [16, Lemma 2]). That is, if we consider *exactly* solving tropical circuits (factor $r = 1$), then there is no difference between the tropical circuit complexity of the minimization and the maximization problem on the *same* (uniform) set $\mathcal{F}$ of feasible solutions.

But the situation is entirely different if we consider *approximating* circuits: Theorem 4.6 and Proposition 5.2 give us doubly-exponentially many in n matroids $\mathcal{F} \subseteq 2^{[n]}$ such that $\mathsf{Max}_{1+o(1)}(\mathcal{F}) \leqslant n^2$, but $\mathsf{Min}_r(\mathcal{F}) = 2^{\Omega(n)}$ for any finite factor $r = r(n) \geqslant 1$.

Problem 6. *Are there uniform families $\mathcal{F}$ for which the gap $\mathsf{Max}_r(\mathcal{F})/\mathsf{Min}_s(\mathcal{F})$ is exponential for $r \geqslant s > 1$?*

Note that the separating family $\mathcal{F}$ is here required to be uniform (or at least to form an antichain): without this requirement, the gap can be *artificially* made large. To see this, take an arbitrary uniform family $\mathcal{H} \subseteq 2^{[n]}$ with large $\mathsf{Max}_r(\mathcal{H})$ (as in Theorems 5.12 and 5.13), and extend it to a nonuniform family $\mathcal{F}$ by adding all single element sets. Then $\mathsf{Min}_1(\mathcal{F}) \leqslant n$ (just compute the minimum weight of a single element), but $\mathsf{Max}_r(\mathcal{F})$ still remains large.

8.4. **Pure DP algorithms with subtraction.** Can the size of tropical approximating circuits be substantially reduced by allowing (besides min / max and +) also *subtraction* (−) gates? In the case of the approximation factor $r = 1$ (exact solution), we already know the answer: subtraction gates *can* then even exponentially decrease the circuit size. Namely, we already know that both directed and undirected versions of the MST problem (minimum weight spanning tree problem) on n-vertex graphs require tropical $(\min, +)$ circuits of size $2^{\Omega(\sqrt{n})}$ [15, 17] but, as shown by Fomin, Grigoriev and Koshevoy [10], both these problems are solvable by tropical $(\min, +, -)$ circuits of size only $O(n^3)$. Unfortunately, no non-trivial lower bounds for $(\min, +, -)$ circuits are known so far. So, at least two natural questions arise.

- Prove lower bounds for $(\min, +, -)$ circuits, at least when $r = 1$.
- What about larger approximation factors $r > 1$?

Note that, when restricted to the *boolean* domain $\{0, 1\}$, $(\min, +, -)$ circuits have the entire power of *unrestricted* boolean $(\vee, \wedge, \neg)$ circuits: $x \wedge y = \min(x, y)$, $x \vee y = \min(1, x + y)$ and $\neg x = 1 - x$. The point, however, is that $(\min, +, -)$ circuits must correctly work over the entire *real* domain $\mathbb{R}_+$.

Appendix A. Greedy algorithms

Since we compared the approximation power of tropical circuits (and pure DP algorithms) with that of the greedy algorithm, here we specify what we actually mean by "the" greedy algorithm.

Let $\mathcal{F} \subseteq 2^E$ be some family of feasible solutions forming an antichain (no two members of $\mathcal{F}$ are comparable under set inclusion). Given an ordering $e_1, \ldots, e_n$ of the elements of E, there are two trivial heuristics to end up with a member of $\mathcal{F}$ by treating the elements one-by-one in this fixed order.

First-in: Start with the empty partial solution, treat the elements one-by-one and, at each step, *add* the next element to the current partial solution if and only if the extended partial solution still lies in at least one feasible solution.

First-out: Start with the entire set E as a partial solution, treat the elements one-by-one and, at each step, *remove* the next element from the current partial solution if and only if the reduced partial solution still contains at least one feasible solution.

Recall that an optimization (maximization or minimization) problem on $\mathcal{F}$ is, given an assignment of nonnegative real weights to the ground elements, to compute the maximum or the minimum weight of a feasible solution, the latter being the sum of weights of its elements.

In this paper, by *the greedy algorithm* we always mean the algorithm which, on every input weighting $x : E \to \mathbb{R}_+$, starts with the heaviest-first ordering $x(e_1) \geqslant x(e_2) \geqslant \ldots \geqslant x(e_n)$ of the elements of E, and uses:

- the first-in heuristic ("best-in" strategy) in the case of *maximization*;
- the first-out heuristic ("worst-out" strategy) in the case of *minimization*.

That is, at each step, the "oracle" of the maximizing greedy algorithm decides whether the current set is still contained in at least one feasible solution, while that of the minimizing greedy algorithm decides whether the current set still contains at least one feasible solution.

We denote the approximation factor achieved by the greedy algorithm on a corresponding optimization (minimization or maximization) problem on $\mathcal{F}$ by $r_{\text{greed}}(\mathcal{F})$. It is well known (see, for example, [23, Theorem 1.8.4] that $r_{\text{greed}}(\mathcal{F}) = 1$ if and only if $\mathcal{F}$ is (the family of bases of) a matroid. If $\mathcal{F}$ is not a matroid, then greedy algorithms can only approximate the corresponding optimization problems. In this case, it is already crucial what greedy strategy is used.

Example 3. The choice of these special heuristics (first-in for maximization and first-out for minimization) is not an accident. Namely, a greedy algorithm starting with the lightest-first ordering $x(e_1) \leqslant x(e_2) \leqslant \ldots \leqslant x(e_n)$, and using the first-out heuristic ("worst-out" strategy) for maximization or first-in heuristic ("best-in" strategy) for minimization would be unable to approximate some optimization problems within any finite factor. To give a simple example, consider the path with three nodes $\overset{a}{\bullet}\!-\!\!-\!\overset{b}{\bullet}\!-\!\!-\!\overset{c}{\bullet}$, and let $\mathcal{F}$ be the family consisting of just two sets $\{a, c\}$ and $\{b\}$ (the maximal independent sets in this path). If we take an arbitrarily large number $M > 1$, and give weights $x(a) = 0$, $x(b) = 1$ and $x(c) = M$, then both these greedy algorithms will treat the vertices in the order a, b, c. The worst-out maximizing greedy on $\mathcal{F}$ will output $x(b) = 1$ while the optimum is M, and the best-in greedy for minimization will output $x(a) + x(c) = 0 + M$, while the optimum is 1. In both cases, the achieved approximation factor is $r \geqslant M$ (unbounded).

If, however, the greedy algorithm uses the "right" strategies for maximization and for minimization, then the approximation factor is always *bounded* (albeit possibly growing with the size of feasible solutions). Say that a family $\mathcal{F}$ of sets is *m-bounded* if $|S| \leqslant m$ holds for all $S \in \mathcal{F}$.

Proposition A.1. *For every m-bounded family $\mathcal{F}$, we have $r_{\text{greed}}(\mathcal{F}) \leqslant m$, and there exist m-bounded antichains $\mathcal{F}$ for which $r_{\text{greed}}(\mathcal{F}) = m$.*

Proof. To show the upper bound, take an arbitrary weighting $x : E \to \mathbb{R}_+$. Consider the heaviest-first ordering $x(e_1) \geqslant \ldots \geqslant x(e_i) \geqslant \ldots \geqslant x(e_n)$. Let e_i be the *first* element accepted by the greedy algorithm. Let $S \in \mathcal{F}$ be an optimal solution for the input x, and $A \in \mathcal{F}$ be the solution found by the algorithm. Let also $x(S) = \sum_{i \in S} x(e_i)$ and $x(A) = \sum_{i \in A} x(e_i)$ be their weights.

If this is the maximizing (best-in) greedy, then e_i is the first element belonging to at least one feasible set. So, $S \cap \{e_1, \ldots, e_{i-1}\} = \emptyset$, implying that $x(S) \leqslant |S| \cdot x(e_i) \leqslant m \cdot x(e_i) \leqslant m \cdot x(A)$, as desired.

If this is the minimizing (worst-out) greedy, then $\{e_{i+1}, \dots, e_n\}$ cannot contain any feasible solution (for otherwise, e_i would not be accepted). So, some element e_j with $j \leqslant i$ must belong to the optimal solution S. But then $x(S) \geqslant x(e_j) \geqslant x(e_i)$, whereas $x(A) \leqslant |A| \cdot x(e_i) \leqslant m \cdot x(e_i)$, implying that $x(A) \leqslant m \cdot x(S)$, as desired.

To show that $r_{\text{greed}}(\mathcal{F}) \geqslant m$ holds for some m-bounded antichains $\mathcal{F}$, take an arbitrarily small number $\epsilon > 0$, and consider the star $K_{1,m}$ centered in a and with leaves $b_1, \dots, b_m$. Let $\mathcal{F}$ consist of the only two maximal independent sets $\{a\}$ and $\{b_1, \dots, b_m\}$ in this graph. Give the weight $c := 1/(1-\epsilon/2) > 1$ to the center 1, and weights 1 to the leaves. The maximizing (best-in) greedy will output c while the optimum is m, and the minimizing (worst-out) greedy algorithms will output m while the optimum is c. In both cases, the achieved approximation factor is $r \geqslant m/c > (1-\epsilon)m$. □

Appendix B. Sidon sets: proof of Theorem 5.4

A set $A \subset \mathbb{N}^n$ of vectors is a *Sidon set* if for all vectors $a, b, c, d \in A$: if $a + b = c + d$, then $\{a, b\} = \{c, d\}$. That is, knowing the sum of two vectors in A, we know which vectors were added. Let (as before) $\mathsf{Max}_r(A)$ denote the minimum size of a tropical $(\max, +)$ circuit r-approximating the problem $f(x) = \max_{a \in A} \langle a, x \rangle$ on A.

Let m be an odd integer, and $n = 4m$. Our goal is to show that then there is an explicit Sidon set $A \subseteq \{0,1\}^n$ such that $\mathsf{Max}_1(A) \geqslant 2^{n/4}$ but $\mathsf{Max}_2(A) \leqslant n$. For this, consider the cubic parabola $C = \{(z, z^3) : z \in \{0,1\}^m\} \subseteq \mathrm{GF}(2^{2m})$. As customary, we view vectors in $z \in \{0,1\}^m$ as coefficient-vectors of polynomials of degree at most $m-1$ over GF(2) when rising them to a power. Note, however, that in the definition of Sidon sets, the sum of vectors is taken over the semigroup $(\mathbb{N}, +)$, not over GF(2); in particular, $a + a = 0$ holds only for $a = 0$.

For a finite set $A \subset \mathbb{N}^n$ of vectors, let $L(A)$ denote the minimum size of a Minkowski $(\cup, +)$ circuit producing A. We will use the following three facts. Recall that a set $A \subseteq \{0,1\}^n$ is *uniform* if all its vectors have the same number of ones.

(1) The cubic parabola $C \subseteq \{0,1\}^{2m}$ is a Sidon set [21, Theorem 2].
(2) $L(A) \geqslant |A|$ holds for every Sidon set $A \subset \mathbb{N}^n$ [12, Theorem 1].
(3) If $A \subseteq \{0,1\}^n$ is uniform, then $\mathsf{Max}_1(A) \geqslant L(A)$ [15, Theorem 2.9].

The cubic parabola C is not uniform, and we cannot apply (3) to it. But, using a simple trick (suggested by Igor Sergeev, personal communication), we can extend this set to a uniform Sidon set. For a 0-1 vector a, let $\underline{a}$ denote the componentwise negation of a. For example, if $a = (0,0,1)$ then $\underline{a} = (1,1,0)$. Consider the following set of vectors:

$$A = \{(c, \underline{c}) : c \in C\} = \left\{(a, a^3, \underline{a}, \underline{a^3}) : a \in \{0,1\}^m\right\} \subseteq \{0,1\}^n .$$

This set is already uniform: every vector of A has exactly $2m$ ones. The set A is also a Sidon set because, by (1), the set C was such. So, (2) and (3) imply that $\mathsf{Max}_1(A) \geqslant |A| = 2^m = 2^{n/4}$.

It remains therefore to prove the upper bound $\mathsf{Max}_2(A) \leqslant n$. We have $n = 4m$ variables $x_1, \dots, x_{4m}$. Our approximating circuit will solve the maximization problem on the set $B = B' \cup B''$, where

$$B' = \{(a, 0, \underline{a}, 0) : a \in \{0,1\}^m\} \quad \text{and} \quad B'' = \{(0, a, 0, \underline{a}) : a \in \{0,1\}^m\} .$$

The maximization problem on B is to compute $f(x) = \max\{g(x), h(x)\}$, where

$$g(x) = \max \sum_{i=1}^{m} a_i x_i + \sum_{i=2m+1}^{3m} (1-a_i)x_i \,;$$

$$h(x) = \max \sum_{i=m+1}^{2m} a_i x_i + \sum_{i=3m+1}^{4m} (1-a_i)x_i$$

with both maximums taken over all vectors $a \in \{0,1\}^{4m}$. Since $g(x)$ is just the sum $\max\{x_1, x_{2m+1}\} + \max\{x_2, x_{2m+2}\} + \cdots + \max\{x_m, x_{3m}\}$, and similarly for $h(x)$, the maximization problem f can be solved using only $4m = n$ gates.

It remains to show that f indeed approximates the maximization problem on A within factor $r = 2$. As we have shown in Section 6.2 (see Lemma 6.4), this happens precisely when the set B lies below A, and the set $\frac{1}{2} \cdot A$ lies below the convex hull Conv(B) of B. It is clear that the first subset B' of B lies below A. We have to show that this holds also for the second subset B''. For this, it is enough to show that B'' coincides with the set of all vectors $(0, a^3, 0, \underline{a^3})$ for $a \in \{0,1\}^m$.

It is known that a polynomial x^k permutes GF(q) if and only if $q-1$ and k are relatively prime; see, for example, Lidl and Niederreiter [20, Theorem 7.8]. In our case, we have $q = 2^m$ and $k = 3$. Since m is odd, we have $m = 2t+1$ for some $t \in \mathbb{N}$. Easy induction on t shows that $p(t) := 2^{2t+1} + 1$ is divisible by 3: the basis $t = 0$ is obvious, because $p(0) = 3$, and the induction step $p(t+1) = 2^{2(t+1)+1} + 1 = 4(2^{2t+1} + 1) - 3 = 4 \cdot p(t) - 3$ follows from the induction hypothesis. So, $q - 1 = p(t) - 2$ cannot be divisible by 3, that is, $q-1$ and 3 are relatively prime and, hence, the mapping $a \mapsto a^3$ is a *bijection*. This gives us a crucial fact:

$$\left\{(0, a^3, 0, \underline{a^3}) :\ a \in \{0,1\}^m\right\} = \left\{(0, a, 0, \underline{a}) :\ a \in \{0,1\}^m\right\} = B''\,.$$

Hence, the entire set $B = B' \cup B''$ lies below A, that is, every vector of B is covered by at least one vector of A. By Lemma 6.4, it remains to show that the set $\frac{1}{2} \cdot A$ lies below the convex hull Conv(B). So, take an arbitrary vector $u = \frac{1}{2} \cdot (a, a^3, \underline{a}, \underline{a^3})$ in $\frac{1}{2} \cdot A$. This vector is a convex combination $\frac{1}{2} \cdot v + \frac{1}{2} \cdot w$ of vectors $v = (a, 0, \underline{a}, 0)$ and $w = (0, a^3, 0, \underline{a^3})$ of B, as desired. □

ACKNOWLEDGMENTS

We thank Georg Schnitger and Igor Sergeev for inspiring discussions.

REFERENCES

[1] N. Alon and R. Boppana, *The monotone circuit complexity of boolean functions*, Combinatorica, 7 (1987), pp. 1–22, https://doi.org/10.1007/BF02579196.

[2] A. Andreev, *On a method for obtaining lower bounds for the complexity of individual monotone functions*, Sov. Math. Dokl., 31 (1985), pp. 530–534.

[3] N. Bansal, R. Pendavingh, and J. van der Pol, *On the number of matroids*, Combinatorica, 35 (2015), pp. 253–277, https://doi.org/10.1007/s00493-014-3029-z.

[4] R. Bellman, *On a routing problem*, Quarterly of Appl. Math., 16 (1958), pp. 87–90, https://doi.org/10.1090/qam/102435.

[5] C. Carathéodory, *Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen*, Mathematische Annalen, 64 (1907), pp. 95–115.

[6] S. Dreyfus and R. Wagner, *The Steiner problem in graphs*, Networks, 1 (1971), pp. 195–207, https://doi.org/10.1002/net.3230010302.

[7] J. Edmonds, *Matroids and the greedy algorithm*, Math. Programming, 1 (1971), pp. 127–136, https://doi.org/10.1007/BF01584082.

[8] K. FAN, *Linear Inequalities and Related Systems*, Princeton University Press, 1956, ch. On systems of linear inequalities, pp. 99–156.
[9] R. FLOYD, *Algorithm 97, shortest path*, Comm. ACM, 5 (1962), p. 345, https://doi.org/10.1145/367766.368168.
[10] S. FOMIN, D. GRIGORIEV, AND G. KOSHEVOY, *Subtraction-free complexity, cluster transformations, and spanning trees*, Found. Comput. Math., 15 (2016), pp. 1–31, https://doi.org/10.1007/s10208-014-9231-y.
[11] L. FORD, *Network flow theory*, Tech. Report P-923, The Rand Corp., 1956.
[12] S. GASHKOV AND I. SERGEEV, *A method for deriving lower bounds for the complexity of monotone arithmetic circuits computing real polynomials*, Sbornik: Mathematics, 203 (2012), pp. 1411–1147.
[13] R. GRAHAM AND N. A. SLOANE, *Lower bounds for constant weight codes*, IEEE Trans. Inform. Theory, 26 (1980), pp. 37–43, https://doi.org/TIT.1980.1056141.
[14] M. HELD AND R. KARP, *A dynamic programming approach to sequencing problems*, SIAM J. on Appl. Math., 10 (1962), pp. 196–210, https://doi.org/10.1137/0110015.
[15] M. JERRUM AND M. SNIR, *Some exact complexity results for straight-line computations over semirings*, J. ACM, 29 (1982), pp. 874–897, https://doi.org/10.1145/322326.322341.
[16] S. JUKNA, *Tropical complexity, Sidon sets and dynamic programming*, SIAM J. Discrete Math., 30 (2016), pp. 2064–2085, https://doi.org/10.1137/16M1064738.
[17] S. JUKNA AND H. SEIWERT, *Greedy can beat pure dynamic programming*, Inf. Process. Letters, 142 (2019), pp. 90–95, https://doi.org/10.1016/j.ipl.2018.10.018.
[18] D. KNUTH, *The asymptotic number of geometries*, J. Combin. Theory, Ser. A, 16 (1974), pp. 398–400, https://doi.org/10.1016/0097-3165(74)90063-6.
[19] A. LEVIN, *Algorithm for the shortest connection of a group of graph vertices*, Sov. Math. Dokl., 12 (1971), pp. 1477–1481.
[20] R. LIDL AND H. NIEDERREITER, *Introduction to Finite Fields and their Applications.*, Cambridge University Press, 1986.
[21] B. LINDSTRÖM, *Determination of two vectors from the sum*, J. Combin. Theory, Series B, 6 (1969), pp. 402–407, https://doi.org/10.1016/S0021-9800(69)80038-4.
[22] E. MOORE, *The shortest path through a maze*, in Proc. Internat. Sympos. Switching Theory, vol. II, 1957, pp. 285–292.
[23] J. OXLEY, *Matroid theory*, Oxford University Press, 1992.
[24] M. PIFF AND D. WELSH, *The number of combinatorial geometries*, Bull. Lond. Math. Soc., 3 (1971), pp. 55–56, https://doi.org/10.1112/blms/3.1.55.
[25] R. RADO, *A theorem on independence relations*, Quart. J. Math., 13 (1942), pp. 83–89, https://doi.org/10.1093/qmath/os-13.1.83.
[26] A. RAZBOROV, *Lower bounds on monotone complexity of the logical permanent*, Math. Notes of the Acad. of Sci. of the USSR, 37 (1985), pp. 485–493, https://doi.org/10.1007/BF01157687.
[27] A. SCHRIJVER, *Theory of Integer and Linear Programming*, Wiley, 1986.
[28] A. VITERBI, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Trans. on Information Theory, 13 (1967), pp. 260–269, https://doi.org/10.1109/TIT.1967.1054010.
[29] S. WARSHALL, *A theorem on boolean matrices*, J. ACM, 9 (1962), pp. 11–12.