Electronic Colloquium on Computational Complexity, Report No. 127 (2018)

APPROXIMATION LIMITATIONS OF TROPICAL CIRCUITS *

STASYS JUKNA† and HANNES SEIWERT‡

Abstract. We develop general lower bound arguments for approximating tropical (min,+) and (max,+) circuits, and use them to prove the first non-trivial, even super-polynomial, lower bounds on the size of such circuits approximating some explicit optimization problems. In particular, these bounds show that the approximation powers of pure dynamic programming algorithms and greedy algorithms are incomparable.

Key words. Tropical circuits, dynamic programming, hardness of approximation, lower bounds

AMS subject classifications. 68Q17

1. Introduction.

A discrete 0-1 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 later being the sum of weights of its elements. Such problems are called 0-1 optimization problems, because feasible solutions are sets, not multi-sets.

For example, the *assignment problem* is, given nonnegative real weights on the edges of the complete bipartite $n\times n$ graph $K_{n,n}$, to compute the minimum weight of a perfect matching. Feasible solutions in this case are all perfect matchings in $K_{n,n}$, viewed as sets of their edges.

Every discrete 0-1 optimization problem problem can be solved by a *tropical* $(\min,+)$ or $(\max,+)$ circuit. Such a circuit is a directed acyclic graph, each whose indegree-zero node holds either one of the variables $x_1,\ldots,x_n$, or a constant $c\in\mathbb{R}_+$. Every other node (called a *gate*) has indegree two, and computes either the sum or minimum/maximum of the values computed at its two predecessors. The *size* of a circuit is the total number of its gates.

Besides being interesting in their own right, the importance of tropical circuits stems form their intimate connection to dynamic programming (DP) algorithms. Many of these algorithms are *pure* in that their recursion equations only use min and addition or max and addition operations, and the structure of the recursion equations do not depend on input weights.

Notable examples of pure DP algorithms are the well-known Bellman–Ford–Moore DP algorithm for the shortest s-t path problem [5, 12, 26], the Floyd–Warshall DP algorithm for the all-pairs shortest paths problem [11, 33] (see Figure 1.1), the Held–Karp DP algorithm for the traveling salesman problem [14], the Dreyfus–Levin–Wagner DP algorithm for the weighted Steiner tree problem [8, 23]. Since every pure DP algorithm is just a special (recursively constructed) tropical circuit, lower bounds on the size of tropical circuits show limits of pure dynamic programming.

An equally notable example of a *non-pure* DP algorithm is the classical Bellman–Dantzig DP algorithm for the knapsack problem [4, 7]. Albeit the algorithm only uses max and addition operations (in the case of maximization), the choice of subproblems

*Research supported by the DFG grant JU 3105/1-1 (German Research Foundation).

†Institute of Computer Science, Goethe University Frankfurt, Frankfurt am Main, Germany (stjukna@gmail.com). Affiliated with Institute of Data Science and Digital Technologies, Faculty of Mathematics and Informatics of Vilnius University, Lithuania.

‡Institute of Computer Science, Goethe University Frankfurt, Frankfurt am Main, Germany (seiwert@thi.cs.uni-frankfurt.de).

ISSN 1433-8092

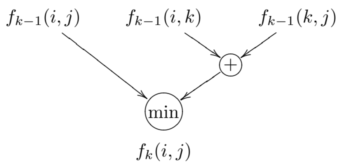

Fig. 1.1: A fragment of a tropical $(\min, +)$ circuit of size $O(n^3)$ implementing the Floyd–Warshall DP algorithm for the all-pairs shortest paths problem. At the gate $f_k(i,j)$, the minimum length of a path from i to j, which only uses nodes $1, \ldots, k$ as inner nodes, is computed.

in its recursion equation *depends* on the input weightings: one of the two subproblems uses the knapsack capacity decreased by the *actual* value of the last item. The dependence on input weightings was crucial in designing fast approximating DP algorithms for the knapsack problem [15].

A general question we approach in this paper is: how many gates a tropical circuit must have in order to *approximate* a given 0-1 optimization problem within a given factor? In particular, we want to compare the approximation power of pure DP algorithms with that of greedy algorithms; we consider greedy algorithms using standard best-in and worst-out heuristics (see subsection 3.6 for details).

Recall that an algorithm approximates a given optimization problem $f : \mathbb{R}_+^n \to \mathbb{R}_+$ within a *factor* $r \geqslant 1$ if for every input $x \in \mathbb{R}_+^n$, 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.

In this case we will also say that the algorithm *r-approximates* f. The factor r may depend on the length n of the inputs x, but not on the inputs themselves. In both cases, the *smaller* the factor r is, the better is the approximation. Factor $r = 1$ means that the problem is solved exactly.

Remark 1.1. Note that, in both cases (minimization and maximization), it is required that the value computed by an algorithm cannot be "better" than the optimal value $f(x)$. Note also that we do not require the approximated values to be achieved on *feasible* solutions: we are only interested in *values* computed by the algorithms.

In their seminal paper [16], Jerrum and Snir considered tropical circuits solving optimization problems *exactly* (within factor $r = 1$). In particular, they proved that the assignment problem (minimum weight perfect matching in $K_{n,n}$) requires $(\min, +)$ circuits of size at least 2^n. On the other hand, a linear programming algorithm, known as the *Hungarian algorithm*, solves the assignment problem using only $O(n^3)$ operations (see, for example, [28]).

They also proved that every $(\min, +)$ circuit solving (exactly) the lightest *directed* spanning tree problem in K_n (known also as the *arborescence* problem) requires $2^{\Omega(n)}$ gates. On the other hand, the family of feasible solutions of the arborescence problem is an intersection of two matroids and, hence, can be approximated by a greedy algorithm within a factor of 2. In [21], we extended this result by showing that also the lightest *undirected* spanning tree problem requires $(\min, +)$ circuits of size $2^{\Omega(\sqrt{n})}$. The family of feasible solutions for this latter problem forms a (graphic) matroid, so that the greedy algorithm can solve this problem *exactly*.

But what if tropical circuits are only required to *approximate* a given optimization problem: can they also then be weaker than greedy and/or linear programming? We will answer this question *affirmatively*. We do this by proving the first non-trivial, even super-polynomial, lower bounds for *approximating* tropical circuits and, hence, also for approximating pure DP algorithms.

2. Results. If a family $\mathcal{F}$ of feasible solutions is *uniform* (all sets of $\mathcal{F}$ have the same cardinality), then there is no difference between the tropical circuit complexities of minimization and maximization problems on $\mathcal{F}$: the minimum size of a $(\min,+)$ circuit solving the minimization problem on $\mathcal{F}$ *coincides* with the minimum size of a $(\max,+)$ circuit solving the maximization problem on $\mathcal{F}$ (see, for example, [20, Lemma 2]).

Somewhat surprisingly, the *approximation* behaviors of tropical $(\min,+)$ and $(\max,+)$ circuits turn out to be entirely different. While the *maximization* problem on *any* family $\mathcal{F} \subseteq 2^{[n]}$ can be approximated by a trivial $(\max,+)$ circuit $\max\{x_1,\ldots,x_n\}$ within a large (but finite) factor $r \leqslant n$, there are many *minimization* problems that cannot be approximated by polynomial-size $(\min,+)$ circuits within any finite factor $r = r(n)$; such is, for example, the assignment problem. We will therefore treat minimization and maximization problems separately.

Below is a summary of our main results (in the order of their later presentation).

(a) A boolean bound for $(\min,+)$ circuits. This bound (Theorem 5.1 in section 5) states that, if the decision version $g(x) = \bigvee_{S\in\mathcal{F}} \bigwedge_{i\in S} x_i$ of the *minimization* problem $f(x) = \min_{S\in\mathcal{F}} \sum_{i\in S} x_i$ requires monotone boolean circuits of size t, then no $(\min,+)$ circuit of size smaller than t can approximate the problem within any finite factor.

Thus, lower bounds for pure DP algorithms approximating *minimization* problems follow from the lower bounds on the size of monotone boolean circuits. In particular, Razborov's lower bound [31] on the size of monotone boolean circuits for the logical permanent implies that no $(\min,+)$ circuit with $n^{o(\log n)}$ gates can approximate the assignment problem on $K_{n,n}$ within any finite factor. This extends the aforementioned result of Jerrum and Snir [16] to *approximating* $(\min,+)$ circuits: linear programming can "outclass" even approximating pure DP algorithms.

In Appendix C, we also show a *converse* of the boolean bound: if the decision version of a minimization problem can be computed by a monotone boolean circuit of size t and of "semantic degree" r (under its particular definition), then the problem can be approximated within the factor r by a $(\min,+)$ circuit of size t. So, the approximation power of $(\min,+)$ circuits is *captured* (not only lower-bounded) by the computational power of monotone boolean circuits.

(b) Greedy can be better in minimization. Using the boolean bound, we show (Theorem 7.7 in subsection 7.2) that there are doubly-exponentially many (in the number of ground-elements) 0-1 minimization problems that can be solved exactly by the greedy algorithm but cannot be approximated within any finite factor by $(\min,+)$ circuits of polynomial size.

(c) A rectangle bound for approximating $(\max,+)$ circuits. In subsection 8.1, we observe that proving nontrivial lower bounds on approximating $(\max,+)$ circuits is a far more difficult task: even Shannon type *counting* arguments seem to fail for such circuits. In particular, with high probability, *one* simple $(\max,+)$ circuit of size $O(n^2)$ approximates the maximization problem on a random family $\boldsymbol{\mathcal{F}} \subseteq \binom{[n]}{n/2}$ within a factor $r = 1 + o(1)$ (Remark 8.2). Being warned by this phenomenon, we develop (in section 6) a general lower bound—the rectangle bound—for approximating $(\max,+)$ circuits, taking into account *specific* structural properties of feasible solutions of the problems to be approximated (Theorems 6.1 and 8.4).

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 $B \in \mathcal{B}$. A rectangle $\mathcal{R}$ *lies below* a family $\mathcal{F}$ of sets if every set of $\mathcal{R}$ is contained in at least one set of $\mathcal{F}$. A set F *appears r-balanced* in the rectangle $\mathcal{A} \vee \mathcal{B}$ if there are $A \in \mathcal{A}$ and $B \in \mathcal{B}$ such that both $|F \cap A|$ and $|F \cap B|$ are at least about $|F|/r$.

Under this proviso, Theorem 8.4 in subsection 8.2 (a set-theoretic version of Theorem 6.1) states that if h is the maximum possible number of feasible solutions $F \in \mathcal{F}$ that can appear r-balanced in a rectangle $\mathcal{R}$ lying below $\mathcal{F}$, then any $(\max, +)$ circuit approximating the maximization problem on $\mathcal{F}$ within the factor r must have at least $|\mathcal{F}|/h$ gates.

(d) A "factor-hierarchy theorem" for $(\max, +)$ circuits. Theorem 8.6 in subsection 8.3 states that, for any prime power m and any integer r dividing m, there is an explicit family $\mathcal{F}$ of feasible solution on $n = m^2$ ground elements such that the maximization problem on $\mathcal{F}$ can be approximated within the factor r by a $(\max, +)$ circuit of size $O(n)$, but for any constant $\epsilon > 0$, at least $n^{\Omega(m/r)}$ gates are necessary to approximate the problem within the factor $(1-\epsilon)r$. That is, already slight improvements of the approximation factor can make tractable problems intractable. This is proved using the rectangle bound. The corresponding families of feasible solutions here are so-called "polynomial designs."

(e) Greedy can be better also in maximization. Theorem 8.7 in subsection 8.4 states that, for every integer $k \geqslant 6$, there is an explicit maximization problem on $n = m^k$ ground elements which can be approximated by the greedy algorithm within factor k, but requires $(\max, +)$ circuits of size exponential in $n^{\Omega(1)}$ to approximate the problem even within the exponentially larger factor $2^k/9$. The theorem is also proved using the rectangle bound. The corresponding families of feasible solutions here are families of perfect matchings in k-uniform k-partite hypergraphs on mk vertices.

That greedy algorithms can have much worse approximation behavior than pure DP algorithms is long known. Namely, there are a lot of optimization problems which are easily solvable by pure DP algorithms even exactly, but the greedy algorithm cannot achieve any finite approximation factor: maximum weight independent set in a path, or in a tree, the maximum weight simple s-t path in a transitive tournament problem, etc.

So, results (b) and (e) imply that the approximation powers of greedy and pure DP algorithms are, in fact, *incomparable*: on some optimization problems, pure DP algorithm can also have much *worse* approximation behavior than greedy.

Remark 2.1 (Regarding input weights) . Our *upper* bounds hold when all non-negative real weights are allowed, whereas *lower* bounds hold even when the input weights are restricted to $\{0, 1\}$ in the case of maximization, and to $\mathbb{N} = \{0, 1, 2, \dots\}$ in the case of minimization. Note that the fewer input weights are allowed, the stronger lower bounds on the circuit size are.

Organization. The paper is organized as follows. In section 3, we recall the concept of Minkowski circuits and sets of vectors "produced" by circuits over any semiring, as well as the two heuristics of greedy algorithms. Section 4 contains two structural lemmas showing how families of approximating feasible solutions are related to the given family of feasible solutions of the optimization problem to be approximated. In the next two sections (sections 5 and 6) we use these lemmas to develop our main technical tools: the "boolean bound" for $(\min, +)$ circuits, and the "rectangle bound" for $(\max, +)$ circuits. The last two sections (sections 7 and 8) are devoted to applications of these tools for specific optimization problems.

For reader's convenience (not to interrupt the main text), some technical results are moved to appendices. In Appendix A, we give a new, simple and self-contained proof of a result from [20] which we use to prove the rectangle bound. In Appendix B, we use convexity arguments to give a *tight* structural characterization of the families of approximative solutions. In Appendix C, we use this characterization to show that our boolean bound for $(\min, +)$ circuits (proved in section 5) actually captures (not only lower-bounds) the power of approximating $(\min, +)$ circuits. Finally, in Appendix D, we prove Proposition 8.3 which states that an explicit Sidon-type maximization problem, requiring $(\max, +)$ circuits of *truly* exponential size to solve it exactly, can be approximated by linear-size $(\max, +)$ circuits within the factor of 2.

3. Preliminaries. In this preparatory section, we introduce our "language:" recall the notion of circuits (over semirings), show how these circuits "produce" (purely syntactically) sets of vectors in $\mathbb{N}^n$, show that these sets are precisely the sets of exponent vectors of polynomials computed by circuits (Lemma 3.1), show that the produced sets of vectors are almost independent on the constant inputs of circuits (Lemma 3.2), recall Minkowski circuits as a bridge between circuits over *arbitrary* semirings and circuits over the arithmetic semiring, sketch the idea of our two-step approach to analyze approximating tropical circuits, and briefly recall greedy algorithms.

3.1. Circuits over semirings. 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.

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 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 node. The *size* of a circuit is the total number of gates in it. Since multiplication distributes over addition, each such circuit computes some polynomial

$$f_A(x_1, \ldots, x_n) = \sum_{a \in A} c_a X^a \quad \text{with} \quad X^a := \prod_{i\colon a_i \neq 0} x_i^{a_i} \tag{3.1}$$

over R in a natural way, where $A \subset \mathbb{N}^n$ is some set of exponent vectors, $\mathbb{N} = \{0, 1, 2, \ldots\}$ is the set of nonnegative integers, coefficients c_a are some semiring elements, and x_i^n stands for $x_i \otimes x_i \otimes \cdots \otimes x_i$ n-times. If the semiring contains the additive unity $\mathsf{0}$ (satisfying $\mathsf{0} \oplus x = x$ and $\mathsf{0} \otimes x = \mathsf{0}$), then we can assume that $c_a \neq \mathsf{0}$ holds for all coefficients in the polynomial (3.1). A circuit is *constant-free* if it has no semiring elements as inputs.

Remark 3.1. Note that in the polynomial (3.1) computed by a *constant-free* circuit, every coefficient c_a is not a semiring element but rather a positive *integer* indicating the number of times the corresponding monomial X^a appears in the polynomial; so, if the underlying semiring is idempotent ($x \oplus x = x$ holds), then we have no such coefficient at all.

In this paper, we will consider circuits over three (idempotent) semirings. In the *boolean* semiring, we have $R = \{0, 1\}$, $x \oplus y := x \vee y$ and $x \otimes y := x \wedge y$. Circuits over this semiring are just monotone boolean circuits. In the tropical $(\min, +)$ (resp.,

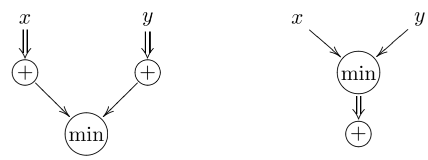

Fig. 3.1: Two constant-free $(\min, +)$ circuits solving the minimization problem $f(x, y) = \min\{2x, 2y\}$ whose set of feasible solutions is $A = \{(2, 0), (0, 2)\}$. The first circuit produces the set A itself, whereas the second saves one gate by producing a *different* set $B = \{(2, 0), (1, 1), (0, 2)\}$. Here $\Downarrow$ stands for two parallel edges.

$(\max, +)$) semiring, we have $R = \mathbb{R}_+$ (all nonnegative real numbers), $x \oplus y := \min\{x, y\}$ (resp., $x \oplus y := \max\{x, y\}$) and $x \otimes y := x + y$. Note that, over tropical semirings, the polynomial (3.1) turns into a minimization or maximization problem

$$f(x) = \min_{a \in A} \langle a, x\rangle + c_a \quad \text{or} \quad f(x) = \max_{a \in A} \langle a, x\rangle + c_a\,, \tag{3.2}$$

where here and throughout, $\langle a, x\rangle := a_1x_1 + \cdots + a_nx_n$ stands for the scalar product of vectors $a = (a_1, \ldots, a_n)$ and $x = (x_1, \ldots, x_n)$.

Remark 3.2. Note that, over the $(\min, +)$ semiring R, the polynomial (3.2) satisfies $f(x+y) \geqslant f(x)+f(y)$ for any vectors $x, y \in R^n$, while over the $(\max, +)$ semiring, we have $f(x+y) \leqslant f(x)+f(y)$. In particular, functions computed by $(\min, +)$ circuits are concave, while those computed by $(\max, +)$ circuits are convex; these observations explain our use of convexity arguments in Appendix B.

3.2. Vector-sets produced by circuits. A simple (but important in the analysis of circuits) observation is that every circuit of n variables over a semiring $(R, \oplus, \otimes)$ also *produces* (purely syntactically) a finite set of vectors in $\mathbb{N}^n$ in a natural way. Namely, define the set $X_v \subset \mathbb{N}^n$ of vectors *produced* at a gate v inductively as follows:

- If $v \in R$ is a semiring element, then $X_v := \{\vec{0}\}$, where $\vec{0} = (0, \ldots, 0)$.
- If $v = x_i$ is the ith input variable, then $X_v := \{\vec{e}_i\}$, where $\vec{e}_i$ is the unit vector with 1 at the ith coordinate and zeroes elsewhere.
- If $v = u \oplus w$, then $X_v := X_u \cup X_w$.
- If $v = u \otimes w$, then $X_v := X_u + X_w$, where $A + B := \{a + b \colon a \in A, b \in B\}$ is the sumset (known also as the *Minkowski sum*) of two sets of vectors, and $a + b = (a_1 + b_1, \ldots, a_n + b_n)$ is the componentwise sum of vectors a and b.

The set produced by the entire circuit is the set produced at its output gate.

Remark 3.3. An equivalent definition of the set $B \subset \mathbb{N}^n$ produced by a circuit over a semiring $(\oplus, \otimes)$ is to view this circuit as a monotone *arithmetic* $(+, \times)$ circuit. Then B is exactly the set of exponent vectors of monomials in the polynomial computed by the resulting arithmetic circuit (the values of nonzero coefficients are here irrelevant).

Remark 3.4. Note that, up to the presence or absence of the all-0 vector $\vec{0}$, the set B produced by a circuit Φ does not depend on the values of input nodes holding semiring elements (constants). So, by concentrating on sets of vectors *produced* by circuits, we indeed concentrate entirely on the exponent vectors of the monomials in the computed by circuits polynomials, and totally ignore the actual coefficients of these monomials.

Lemma 3.1. *If $A \subset \mathbb{N}^n$ 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 is A.*

Proof. Simple induction on the circuit size. Let $A \subset \mathbb{N}^n$ be the set of vectors produced, and $f(x)$ the polynomial computed by a circuit Φ.

If Φ consists of a single input node holding a semiring element $c \in R$, then $f(x) = c$ 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 $\vec{e}_i$.

For the induction step, let Φ_A and Φ_B be two circuits, producing sets $A \subset \mathbb{N}^n$ and $B \subset \mathbb{N}^n$. Assume that the circuit Φ_A computes some polynomial $\Phi_A(x) = \sum_{a\in A} c_a X^a$, and Φ_B computes some polynomial $\Phi_B(x) = \sum_{b\in B} c_b X^b$. Then the set of exponent vectors of the polynomial

$$\Phi_A \oplus \Phi_B = \sum_{a\in A} c_a X^a \oplus \sum_{b\in B} c_b X^b$$

is $A \cup B$, and that of

$$\Phi_A \otimes \Phi_B = \Big(\sum_{a\in A} c_a X^a\Big) \otimes \Big(\sum_{b\in B} c_b X^b\Big) = \sum_{a\in A}\sum_{b\in B} c_a c_b X^{a+b}$$

is the Minkowski sum $A + B$. ∎

We will mainly use Lemma 3.1 for circuits over tropical semirings. So, if $A \subset \mathbb{N}^n$ is the set produced by a circuit over the semiring $(\mathbb{R}_+, \min, +)$, then Lemma 3.1 implies that the computed by the circuit function has the form $f(x) = \min_{a\in A}\langle a, x\rangle + c_a$ for some constants $c_a \in \mathbb{R}_+$. If the circuit is constant-free, then the computed function has the form $f(x) = \min_{a\in A}\langle a, x\rangle$.

3.3. Elimination of constant inputs. Our goal is to show that, unlike the functions *computed* by the circuits, the sets of vectors *produced* by the circuits are almost independent of constant inputs. So, let Φ be a (not necessarily constant-free) circuit over some semiring $(R, \oplus, \otimes)$. We can clearly assume that Φ has no gates whose *both* inputs are constants: just replace such gates by constant inputs. We construct the *constant-free version* Φ_* of Φ by repeatedly applying the following transformation:

(∗) If $v = u \circ c$ is a gate, where $\circ \in \{\oplus, \otimes\}$ and c is a constant input node, then contract the edge (u, v), that is, replace every edge (v, w) leaving v by the edge (u, w), and remove the gate v.

After no more such transformation is possible, remove all constant input nodes together with edges leaving them.

These transformations may clearly change the *function* computed by the original circuit Φ. But the following lemma ensures that the resulting constant-free circuit will *produce* almost the same set: only the all-0 vector $\vec{0}$ may be missing.

Lemma 3.2. *If $A \subseteq \mathbb{N}^n$ is the set produced by a circuit, then the constant-free version of this circuit produces either A or $A \setminus \{\vec{0}\}$.*

Proof. Let $v = u \circ c$ be a gate in the circuit, where c is a constant input node. After the transformation (∗), the set X_v produced at gate v was replaced by the set X_u produced at gate u. Recall that at *every* input node c holding a semiring element, the same set $X_c = \{\vec{0}\}$ is produced, regardless of what this element actually is. So, if $\circ = \otimes$, then $X_v = X_u + X_c = X_u + \{\vec{0}\} = X_u$, and if $\circ = \oplus$, then

$X_v = X_u \cup X_c = X_u \cup \{\vec{0}\}$. We thus have that either $X_u = X_v$ or $X_u = X_v \setminus \{\vec{0}\}$ holds, implying that the set produced by the constant-free version of the circuit must either be the same set A, or be this set without the all-0 vector $\vec{0}$. ◻

3.4. Minkowski complexity of vector-sets. It is clear that the same circuit Φ may *compute* different functions over different semirings. It is, however, important to note that the set $A \subset \mathbb{N}^n$ of vectors *produced* by Φ is always the same—it only depends on the circuit itself, not on the underlying semiring. This independence of produced sets on actual semirings is captured by the model of "Minkowski circuits."

A *Minkowski circuit* is a directed acyclic graph with $n+1$ source (indegree zero) nodes holding single-element sets $\{\vec{0}\}, \{\vec{e}_1\}, \ldots, \{\vec{e}_n\}$. 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. So, at each gate, some set of vectors in $\mathbb{N}^n$ is produced in a natural way. The set produced by the circuit is the set produced at its output gate. We will denote the minimum size of a Minkowski circuit producing a set $A \subset \mathbb{N}^n$ by $L(A)$.

Motivation: For every set $A \subset \mathbb{N}^n$, its Minkowski complexity $L(A)$ is exactly the minimum number of gates in a circuit over *any* semiring producing the set A.

3.5. Our approach. The reason to consider sets produced by circuits is that it is often much easier to prove that a given set $B \subset \mathbb{N}^n$ requires many gates to *produce* it than to show that many gates are necessary to *compute* a polynomial (as a function) whose set of exponents is B. This happens because, as we mentioned above, the former task essentially boils down to proving a lower bound on the size of monotone *arithmetic* $(+, \times)$ circuits computing a polynomial whose set of exponent vectors coincides with B (see Remark 3.3).

When dealing with *approximating* circuits, the main difficulty is that we do not know the *precise* structure of these (produced by circuits) sets B of "approximative" solutions. So, we are forced to prove lower bounds on the Minkowski complexity of *all* sets B having particular properties, not just of *one* given set B, as in the case of arithmetic circuits. Still, we will be able to extract enough information about "approximating" sets B from the structure of feasible solutions of the original problem which is to be approximated.

Given a finite set $A \subset \mathbb{N}^n$ (of feasible solutions), the optimization problem *on* A is, for every input weighting $x \in \mathbb{R}_+^n$, to compute the maximum or the minimum weight $\langle a, x\rangle$ of a feasible solution $a \in A$.

At a high level, our approach consists of the following two steps. We are given an optimization problem on some set $A \subset \mathbb{N}^n$ of feasible solutions. Suppose that some (unknown to us) tropical circuit Φ approximates this problem within some factor r, and let $B \subset \mathbb{N}^n$ be the (also unknown) set of vectors produced by Φ. Note that B does not need to coincide with A (see Figure 3.1 for a simple example).

1. Knowing the set A and the fact that the circuit Φ must approximate the optimization problem on A, extract some *structural* information about the set B of "approximative solutions," that is, establish structural properties of any such possible set B.
2. Use this information about the set B to show that Φ must have many gates to (syntactically) *produce* such a set, that is, show that any such set B must have large Minkowski circuit complexity $L(B)$.

3.6. Greedy algorithms. Since we will compare the approximation power of tropical circuits (and pure DP algorithms) with that of greedy algorithms, let us specify what do we actually mean under a "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 simple procedures to end up with a member of $\mathcal{F}$.

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.

Remark 3.5. In the literature on greedy algorithms, it is usually assumed that $\mathcal{F}$ is closed under taking subsets. We do not require this property; this is why we say "lies in a feasible solution" instead of "is a feasible solution" in our description of the algorithm.

In the case of the *maximization* problem on $\mathcal{F}$, given an input weighting $x : E \to \mathbb{R}_+$, the *best-in greedy algorithm* starts with the heaviest-first ordering $x(e_1) \geqslant \ldots \geqslant x(e_n)$, and uses the first-in procedure, while the *worst-out greedy algorithm* starts with the lightest-first ordering $x(e_1) \leqslant \ldots \leqslant x(e_n)$, and uses the first-out procedure.

In the case of the *minimization* problem on $\mathcal{F}$, the *best-in greedy algorithm* starts with the lightest-first ordering $x(e_1) \leqslant \ldots \leqslant x(e_n)$, and uses the first-in procedure, while the *worst-out greedy algorithm* starts with the heaviest-first ordering $x(e_1) \geqslant \ldots \geqslant x(e_n)$, and uses the first-out procedure.

A classical result of Rado [30] and Edmonds [9] is that the best-in greedy algorithm can solve the optimization (maximization or minimization) problem on $\mathcal{F}$ for all weightings $x : E \to \mathbb{R}_+$ if and only if $\mathcal{F}$ is the family of bases of a matroid. If, however, the family $\mathcal{F}$ does not have this (matroid) property, then greedy algorithms can only approximate the optimal value. In this case, it is already crucial what type of greedy strategy (best-in or worst-out) is used.

Example 3.6. To see the difference, consider the problem of finding the maximum weight of an independent set (resp., the minimum weight of a maximal independent set) in a path with three nodes: $\overset{0}{\bullet}\!\!-\!\!\!-\!\!\!-\!\!\overset{1}{\bullet}\!\!-\!\!\!-\!\!\!-\!\!\overset{M}{\bullet}$. The worst-out greedy for maximization will output 1, whereas M is the optimal value. The best-in greedy for minimization will pick 0 and output M, whereas 1 is the optimal value. So, in both cases, the approximation factor is *unbounded* (can be as large as M).

If, however, we use the best-in heuristic for maximization, and worst-out heuristic for minimization, then the approximation factor is always *bounded*.

Proposition 3.3. *For every family $\mathcal{F}$, the approximation factor of the best-in greedy for the maximization and of the worst-out greedy for the minimization problem on $\mathcal{F}$ does not exceed* $\mathrm{r}(\mathcal{F}) := \max\{|S| \colon S \in \mathcal{F}\}$.

Proof. Let $m = \mathrm{r}(\mathcal{F})$, and 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 was the maximizing (best-in) greedy, then e_i was 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 was the minimizing (worst-out) greedy then $\{e_{i+1}, \ldots, e_n\}$ cannot contain any feasible solution (for otherwise, e_i would be not 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. ∎

4. Structure of approximating tropical circuits. In this section, we consider the following question: if a tropical circuit approximates an optimization (minimization or maximization) problem on a given set $A \subset \mathbb{N}^n$ of feasible solutions within a given factor, what can then be said about the set $B \subset \mathbb{N}^n$ of vectors *produced* by that circuit?

To answer this question, we have to fix some notation. For vectors $a, b \in \mathbb{R}^n$ and sets $A, B \subseteq \mathbb{R}^n$ of vectors, we say that:

- a *is contained* in b (and b *contains* a) if $a \leqslant b$ holds, that is, if $a_i \leqslant b_i$ holds for all $i = 1, \ldots, n$;
- B *lies above* A if every vector of B contains at least one vector of A;
- B *lies below* A if every vector of B is contained in at least one vector of A;
- $S_a = \{i \colon a_i \neq 0\}$ is the *support* of a.

Note that the fact that B lies above A does not imply that A lies below B, and vice versa: the former means $(\forall b \in B)(\exists a \in A)\ a \leqslant b$, whereas the latter means $(\forall a \in A)(\exists b \in B)\ a \leqslant b$.

Lemma 4.1 (Maximization) . *Let $A \subseteq \{0,1\}^n$ be some set of feasible solutions, Φ a $(\max, +)$ circuit, and $B \subset \mathbb{N}^n$ the set of vectors produced by Φ. If Φ approximates the maximization problem on A within a factor $r \geqslant 1$ on all input weightings $x \in \{0,1\}^n$, then B has the following properties:*

(i) *B lies below A and, hence, also consists of 0-1 vectors;*
(ii) *for every vector $a \in A$ there is a vector $b \in B$ with $\langle a, b\rangle \geqslant \frac{1}{r}\langle a, a\rangle$.*

Proof. By Lemma 3.1, the circuit Φ solves the maximization problem $\Phi(x) = \max_{b \in B}\langle b, x\rangle + c_b$ for some constants $c_b \in \mathbb{R}_+$. The maximization problem on A is of the form $f(x) = \max_{a \in A}\langle a, x\rangle$. Since the circuit approximates the maximization problem on A within factor r for all 0-1 weightings, we know that $\frac{1}{r} \cdot f(x) \leqslant \Phi(x) \leqslant f(x)$ must hold for all $x \in \{0,1\}^n$. Since $\Phi(x) \leqslant f(x)$ must hold also for the all-0 input weighting $x = \vec{0}$, $c_b = 0$ must hold for all $b \in B$. So, the maximization problem solved by the circuit is actually of the form $\Phi(x) = \max_{b \in B}\langle b, x\rangle$.

Let us first show that the set B must also consist of only 0-1 vectors. Assume contrariwise that some vector $b \in B$ has a position $i \in S_b$ with $b_i \geqslant 2$, and consider the weighting $x \in \{0,1\}^n$ with $x_i = 1$ and $x_j = 0$ for all $j \neq i$. On this weighting, we have $\Phi(x) \geqslant \langle b, x\rangle = b_i \geqslant 2$, whereas $f(x) \leqslant 1$, since all vectors in A are 0-1 vectors. We obtain a contradiction with $\Phi(x) \leqslant f(x)$. So, $B \subseteq \{0,1\}^n$ holds.

To show item (i), suppose contrariwise that some vector $b \in B$ is contained in none of the vectors of A. Since b is a 0-1 vector, this means that $S_b \setminus S_a \neq \emptyset$ holds for all vectors $a \in A$. Then, on the weighting $x := b$, we have $\langle a, x\rangle = \langle a, b\rangle \leqslant \langle b, b\rangle - 1 = \langle b, x\rangle - 1$ for all $a \in A$, a contradiction with $\Phi(x) \leqslant f(x)$.

To show property (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 $\Phi(x) < m/r$, whereas $f(x) \geqslant \langle a, a\rangle \geqslant m$, a contradiction with $\frac{1}{r} f(x) \leqslant \Phi(x)$. ∎

Remark 4.1. The observation we made at the beginning of the proof is worth to be restated separately: every *minimal* $(\max, +)$ circuit Φ approximating the maximization problem $f(x) = \max_{a\in A}\langle a, x\rangle$ on any set $A \subset \mathbb{N}^n$ of feasible solutions must be *constant-free*. This holds because on the input weighting $x = \vec{0}$, the circuit must output $\Phi(x) \leqslant f(x) = 0$; so, we can safely set all constant inputs (if any) to zero.

Lemma 4.2 (Minimization) . *Let $A \subseteq \{0,1\}^n$ be an antichain, Φ be a $(\min,+)$ circuit, and $B \subset \mathbb{N}^n$ the set of vectors produced by Φ. If Φ approximates the minimization problem on A within a finite factor r on all input weightings $x \in \mathbb{N}^n$, then B has the following properties:*

(i) *B lies above A;*

(ii) *for every vector $a \in A$, at least one vector $b \in B$ has the same support as a.*

Note that the actual value of the factor r is here irrelevant: it only must be finite.

Proof. By Lemma 3.1, the circuit Φ solves the minimization problem $\Phi(x) = \min_{b\in B}\langle b, x\rangle + c_b$ for some constants $c_b \in \mathbb{R}_+$. The minimization problem on A is of the form $f(x) = \min_{a\in A}\langle a, x\rangle$. We know that $f(x) \leqslant \Phi(x) \leqslant r \cdot f(x)$ must hold for all $x \in \mathbb{N}^n$.

To show property (i), suppose contrariwise that some vector $b \in B$ contains none of the vectors of A. Since vectors in A are 0-1 vectors, this means that every vector $a \in A$ must have a 1 in some position $i \notin S_b$ (where $b_i = 0$). So, take an integer $M \geqslant \max\{c_b \colon b \in B\}$ (which exists, because the set B is finite), and consider the assignment $x \in \{0, M+1\}^n$ of weights such that $x_i = M+1$ if $i \notin S_b$, and $x_i = 0$ if $i \in S_b$. On this weighting, we have $\Phi(x) \leqslant \langle b, x\rangle + c_b = 0 + c_b \leqslant M$. But since every vector $a \in A$ has 1 is some position i where $x_i = M+1$, we have that $\langle a, x\rangle \geqslant M+1$ for all $a \in A$ and, hence, also $f(x) \geqslant M+1$, contradicting the inequality $f(x) \leqslant \Phi(x)$.

To show property (ii), suppose contrariwise that there is a vector $a \in A$ such that $S_b \neq S_a$ holds for all vectors $b \in B$. The case $S_b \subset S_a$ (proper inclusion) is impossible: by item (i), we then would have $S_{a'} \subset S_a$ for some vector $a' \in A$, contradicting the fact that A is an antichain. So, the only possibility is that $S_b \setminus S_a \neq \emptyset$ holds for all vectors $b \in B$. To show that this is also impossible, take $M := rn+1$, and consider the weighting $x \in \{1, M\}^n$ such that $x_i = 1$ for all $i \in S_a$ and $x_i = M$ for all $i \notin S_a$. Then $f(x) \leqslant \langle a, x\rangle = \langle a, a\rangle \leqslant n$. But since every vector $b \in B$ must have a position $i \notin S_a$ such that $b_i \geqslant 1$, we have $\Phi(x) \geqslant M = rn+1 > r \cdot f(x)$, a contradiction. ∎

Remark 4.2. In Lemmas 4.1 and 4.2, we have admittedly stated only the simplest properties of the sets B created by circuits, because their proofs are then simple, and because these properties already suffice to derive our main lower-bounding tools—the “boolean bound” (Theorem 5.1) and the “rectangle bound” (Theorem 6.1). Moreover, these properties hold already when circuits have to approximate the given optimization problems on only boolean or only nonnegative integer weights. If, however, we have *all* weights from $\mathbb{R}_+$ in our disposal, then it is possible to describe the structure of the sets created by approximating circuits even *tightly*. We do this in Appendix B using convexity arguments.

5. Boolean bound for $(\min,+)$ circuits. In this section, we show that the approximation power of *minimizing* pure DP algorithms is lower bounded by the monotone *boolean* circuit complexity of the corresponding *decision* problems.

As we already mentioned, every finite set $A \subset \mathbb{N}^n$ of vectors (of *feasible solutions*) defines two natural discrete optimization problems with a linear objective function: given a vector $x \in \mathbb{R}^n_+$ of weights, compute the minimum or the maximum weight $\langle a, x\rangle = a_1x_1 + \cdots + a_nx_n$ of a feasible solution $a \in A$. The *decision version* of these two problems is a monotone boolean function defined by A:

$$f_A(x) = \bigvee_{a\in A} \bigwedge_{i\in S_a} x_i \,, \tag{5.1}$$

where $S_a = \{i\colon a_i \neq 0\}$ is the support of vector a. Note that, for every input $x \in \{0,1\}^n$, we have

$$f_A(x) = 1 \text{ if and only if } S_x \supseteq S_a \text{ for some } a \in A. \tag{5.2}$$

Let $\mathrm{Min}_r(A)$ denote the minimum size of a $(\min,+)$ circuit approximating the minimization problem on A within the factor r, and let $\mathrm{Bool}(A)$ denote the minimum number of gates in a monotone boolean $(\vee,\wedge)$ circuit computing the corresponding to A boolean function $f_A(x)$.

Theorem 5.1 (Boolean bound for minimization) . *If* $\{\vec{0}\} \neq A \subset \{0,1\}^n$ *is an antichain, then for every approximation factor* $r = r(n) \geqslant 1$*, we have* $\mathrm{Min}_r(A) \geqslant \mathrm{Bool}(A)$.

Thus, no $(\min,+)$ circuit with *fewer* than $t = \mathrm{Bool}(A)$ gates can approximate the minimization problem on A within any finite factor. That is, if fewer than t gates are used, then regardless of how large approximation factor r we will take, there will be an input weighting $x \in \mathbb{R}_+^n$ on which the circuit makes an error: the computed value will be either strictly smaller or more than r times larger than the optimal value on x. Note, however, that Theorem 5.1 is not a "sentence of death" for $(\min,+)$ circuits: it does *not* exclude that, using t or more gates, $(\min,+)$ circuits may achieve small approximation factors.

Proof. Take a $(\min,+)$ circuit Φ of size $t = \mathrm{Min}_r(A)$ 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; hence, $L(B) \leqslant t$. By Lemma 3.1, the circuit Φ solves the minimization problem $\Phi(x) = \min_{b\in B}\langle b, x\rangle + c_b$ for some constants $c_b \in \mathbb{R}_+$. By Lemma 4.2, we know that the set B must have the following two structural properties:

(i) for every $b \in B$ there is an $a \in A$ such that $b \geqslant a$ and, hence, $S_b \supseteq S_a$;

(ii) for every $a \in A$ there is a $b \in B$ such that $S_a = S_b$ and, hence, also $S_a \supseteq S_b$.

Since A is an antichain, and since $A \neq \{\vec{0}\}$ (there would be nothing to prove otherwise), $\vec{0} \notin A$ follows and property (i) implies that $\vec{0} \notin B$ must hold as well. So, Lemma 3.2 implies that the constant-free version Φ^* of our circuit Φ must produce the same set B satisfying the properties (i) and (ii).

Let Ψ be the monotone boolean circuit obtained from Φ^* by replacing all min gates by OR gates, and all sum gates by AND gates. Since the sets of vectors produced by circuit do not depend on the underlying semiring, the circuit Ψ produces the same set B with both properties (i) and (ii). So, it remains to show that $\Psi(x) = f_A(x)$ holds for all $x \in \{0,1\}^n$.

By Lemma 3.1, the circuit Ψ computes the boolean function $\Phi(x) = \bigvee_{b\in B} \bigwedge_{i\in S_b} x_i$. Together with (5.2), property (i) implies $\Psi(x) \leqslant f_A(x)$, while property (ii) implies $f_A(x) \leqslant \Psi(x)$ for all inputs $x \in \{0,1\}^n$. So, the boolean circuit Ψ computes our boolean function f_A, and has size at most t, as desired. ∎

Remark 5.1. Actually, Theorem 5.1 has also an inverse: under an appropriate definition of the "semantic degree" of monotone boolean circuits, we even have an equality $\mathrm{Min}_r(A) = \mathrm{Bool}_r(A)$, where $\mathrm{Bool}_r(A)$ is the minimum size of a monotone boolean circuit of semantic degree at most r computing the boolean function $f_A(x)$

defined by (5.1). We show this in Appendix C. Thus, the approximation power of tropical $(\min, +)$ circuits is *captured* (not only lower bounded) by the computational power of monotone boolean circuits.

6. A rectangle bound for (max,+) circuits. Under a *rectangle* with parts $X \subseteq \mathbb{N}^n$ and $Y \subseteq \mathbb{N}^n$ we will mean the sumset $X + Y = \{x + y \colon x \in X, y \in Y\}$ of these two parts. Say that a vector $a \in \mathbb{N}^n$ *appears* (r, ϵ)*-balanced* in a rectangle $X + Y$ if there are vectors $x \in X$ and $y \in Y$ such that for $p := \frac{1}{r} \cdot \langle a, a \rangle$, we have

$$\langle a, x + y \rangle \geqslant p, \quad \tfrac{1}{2}\epsilon p < \langle a, x \rangle \leqslant \epsilon p \quad \text{and} \quad \langle a, y \rangle \geqslant (1 - \epsilon)p\,. \tag{6.1}$$

For a set $A \subset \mathbb{N}^n$ of feasible solutions, let $\mathrm{Max}_r(A)$ denote the minimum size of a tropical $(\max, +)$ circuit approximating the maximization problem $f(x) = \max_{a \in A}\langle a, x \rangle$ on A within the factor r on all input weightings $x \in \{0, 1\}^n$.

Theorem 6.1 (Rectangle bound) . *Let $A \subseteq \{0, 1\}^n$ be a set of vectors, $1 \leqslant r < m$ be an approximation factor, and $r/m \leqslant \epsilon < 1$. If $\mathrm{Max}_r(A) \leqslant t$, then there exist t or fewer rectangles $X + Y$ lying below A such that every vector $a \in A$ with at least m ones appears (r, ϵ)-balanced in at least one of these rectangles.*

Note that the fact that a rectangle $X + Y$ *lies below* a set of 0-1 vectors implies that the rectangle is "cross disjoint" in that $\langle x, y \rangle = 0$ holds for all $x \in X$ and $y \in Y$.

We will derive Theorem 6.1 from the following general "decomposition lemma" for Minkowski circuits. A *norm-measure* is any assignment 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 sub-additive in that the norm of a sum of two vectors does not exceed the sum of their norms.

Recall that the Minkowski complexity, $L(B)$, of a finite set $B \subset \mathbb{N}^n$ of vectors is the minimum number of set-theoretic union and Minkowski sum operations required to produce B when starting from simplest sets $\{\vec{0}\}, \{\vec{e}_1\}, \ldots, \{\vec{e}_n\}$.

Lemma 6.2 (Decomposition lemma [20]) . *Let $B \subset \mathbb{N}^n$ be a set of vectors, $p \geqslant 2$ and $1/p \leqslant \epsilon < 1$. If $L(B) \leqslant t$, then B is a union of t or fewer rectangles $X + Y \subseteq B$ with the following property:*

- $(*)$ *for every norm measure $\mu : \mathbb{N}^n \to \mathbb{R}_+$ and for every vector $b \in B$ of norm $\mu(b) \geqslant p$, at least one of these rectangles $X + Y$ contains vectors $x \in X$ and $y \in Y$ such that $x + y = b$ and $\frac{1}{2}\epsilon p \leqslant \mu(x) \leqslant \epsilon p$.*

Important here is that we can choose different norms μ for different vectors $b \in B$. This flexibility will be crucial in our proof of Theorem 6.1. Lemma 6.2 itself was originally proved in [20, Theorem D]. In Appendix A, we give a simpler, direct and self-contained proof, not depending on other concepts related to Minkowski circuits used in [20].

Proof of Theorem 6.1. Take a $(\max, +)$ circuit Φ of size t approximating the maximization problem on A within factor r, and let $B \subset \mathbb{N}^n$ be the set of vectors produced by Φ; hence, $L(B) \leqslant t$. By Lemma 4.1, we know that

- (i) the set B must lie below A and, hence, must also consist of 0-1 vectors;
- (ii) for every vector $a \in A$ with at least m ones there must be a vector $b = b_a \in B$ such that

$$\langle a, b \rangle \geqslant p := \tfrac{1}{r}\, \langle a, a \rangle \geqslant \tfrac{1}{r}\, m\,. \tag{6.2}$$

We are going to apply Lemma 6.2. Property (ii) suggests to associate with every vector $a \in A$ the norm-measure $\mu_a(x) := \langle a, x \rangle$. By Lemma 6.2, the set B is a union of at most t rectangles $X + Y \subseteq B$ with the property $(*)$.

Now take a vector $a \in A$ and a vector $b = b_a \in B$ satisfying (6.2). Then $\mu_a(b) = \langle a, b\rangle \geqslant p$. By the property $(*)$, there is a rectangle $X + Y \subseteq B$ and vectors $x \in X$ and $y \in Y$ such that $x + y = b$ and $\epsilon p/2 < \mu_a(x) = \langle a, x\rangle \leqslant \epsilon p$; hence, the middle inequalities of (6.1) hold. Since $\langle a, x + y\rangle = \langle a, b\rangle \geqslant p$, we also have the first inequality of (6.1). Since $\langle a, y\rangle = \langle a, x + y\rangle - \langle a, x\rangle \geqslant p - \epsilon p$ also holds, the vector a appears (r, ϵ)-balanced in this rectangle, as desired. ∎

In the next sections, we will use the just created general tools—the boolean bound (Theorem 5.1) and the rectangle bound (Theorem 6.1)—to prove lower bounds on the size of approximating tropical circuits for some specific 0-1 optimization problems. To state and prove these lower bounds, it will be convenient to switch back from *vectors* to *sets*. For this view an n-dimensional 0-1 vector as incidence vector of the corresponding subset of $[n] = \{1, \ldots, n\}$.

7. Approximation limitations of (min,+) circuits. We already know (see Proposition 3.3) that the standard (worst-out) greedy algorithm can approximate the minimization problem on *any* family $\mathcal{F}$ within the (large but finite) factor $\mathrm{r}(\mathcal{F}) = \max\{|S|\colon S \in \mathcal{F}\}$. In contrast, we will now show that small tropical $(\min, +)$ circuits are unable to approximate some minimization problems within any (arbitrarily large but finite) factor $r \geqslant 1$.

For a family $\mathcal{F}$ of feasible solutions, and a real number $r \geqslant 1$, let $\mathrm{Min}_r(\mathcal{F})$ denote the minimum size of a $(\min, +)$ circuit approximating the minimization problem $f(x) = \min_{S\in\mathcal{F}} \sum_{i\in S} x_i$ on $\mathcal{F}$ within the factor r. Recall that, in our lower bounds, we will only require the circuit to do this on nonnegative *integer* weights. Let also

$$\mathrm{Min}(\mathcal{F}) := \inf_{r\geqslant 1} \mathrm{Min}_r(\mathcal{F})\,.$$

Note that $\mathrm{Min}(\mathcal{F}) \geqslant t$ means that $\mathrm{Min}_r(\mathcal{F}) \geqslant t$ holds for *any* finite approximation factor $r = r(n) \geqslant 1$. By Theorem 5.1, we know that $\mathrm{Min}(\mathcal{F})$ is at least the monotone boolean circuit complexity $\mathrm{Bool}(f_{\mathcal{F}})$ of the decision version $f_{\mathcal{F}}(x) = \bigvee_{S\in\mathcal{F}} \bigwedge_{i\in S} x_i$ of the minimization problem on $\mathcal{F}$. We thus can use known lower bounds on the size of monotone boolean circuits to show limitations on the approximation power of minimizing pure DP algorithms. In fact, we even have the equality $\mathrm{Min}(\mathcal{F}) = \mathrm{Bool}(f_{\mathcal{F}})$ (see Theorem C.2 in Appendix C).

7.1. Explicit lower bounds. 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. Since every feasible solution for this problem (a triangle) has only three edges, the worst-out greedy algorithm achieves approximation factor $r = 3$ on this problem (by Proposition 3.3). It is also clear that a trivial pure DP algorithm can solve this problem even exactly by using $O(n^3)$ operations. This is already almost the best pure DP algorithms can do.

Corollary 7.1. *If $\mathcal{F}$ is the family of all triangles in K_n, then* $\mathrm{Min}(\mathcal{F})$ *is at least* $\Omega(n^3/\log^4 n)$.

Proof. It is known (see, e.g., [1, Lemma 3.14] or [18, Theorem 9.19]) that the decision version of this problem requires monotone boolean circuits with $\Omega(n^3/\log^4 n)$ gates. ∎

Recall that the assignment problem (lightest perfect matching in $K_{n,n}$) can be solved by a linear programming algorithm using only $O(n^3)$ operations. The corresponding family of feasible solutions is here the family of all perfect matchings, each viewed as its set of edges.

Corollary 7.2. *If $\mathcal{F}$ is the family of all perfect matchings in $K_{n,n}$, then* $\mathrm{Min}(\mathcal{F})$ *is at least $n^{\Omega(\log n)}$.*

Proof. The decision version of the assignment problem is the boolean permanent function which, as proved by Razborov [31], requires monotone boolean circuits of size $n^{\Omega(\log n)}$. ∎

Let m be a prime power, and consider the complete bipartite $m \times m$ graph $K_{m,m} = U \times V$ with $U = \mathrm{GF}(m)$ and $V = \mathrm{GF}(m)$. Every polynomial $p(x)$ over $\mathrm{GF}(m)$ determines a subgraph of $K_{m,m}$ consisting of m edges $(u, p(u))$ for $u \in U$. The *polynomial* (m, d)*-design* consists of all m^d subgraphs of $K_{m,m}$ determined by polynomials of degree at most $d - 1$ over $\mathrm{GF}(m)$. Note that $\mathrm{Min}_r(\mathcal{F}) \leqslant m^{d+1}$ is a trivial upper bound, even for $r = 1$.

Corollary 7.3. *If $\mathcal{F}$ is a polynomial (m, d)-design with $d \leqslant (m/4 \ln m)^{1/2}$, then* $\mathrm{Min}(\mathcal{F})$ *is at least $m^{\Omega(d)}$.*

Proof. The monotone boolean function defined by the family $\mathcal{F}$ was introduced by Andreev [2] who proved a lower bound of $m^{\Omega(d)}$ for this function when d is at most about $m^{1/4}$. Alon and Boppana [1] used Razborov's method of approximations to show that this lower bound actually holds for all d at most $(m/4 \ln m)^{1/2}$. Theorem 5.1 yields the desired lower bound for $(\min, +)$ circuits. ∎

A family $\mathcal{F}$ of subgraphs of K_n is (k, l)*-clique-like* $(1 \leqslant l < k \leqslant n)$ if it contains all subgraphs G with clique number $\omega(G) \geqslant k$, and does not contain any subgraph G with chromatic number $\chi(G) \leqslant l$. The corresponding to any such family $\mathcal{F}$ monotone boolean function accepts all graphs G with $\omega(G) \geqslant k$, and rejects all graphs G with $\chi(G) \leqslant l$. On the remaining graphs, the function can take arbitrary values. For $l = k - 1$, this is the well-known k-CLIQUE function.

It is known (see [17, Theorem 3.4] or [18, Theorem 9.26]) that any (k, l)-clique like function requires monotone boolean circuits of size exponential in $\min\{l, n/k\}^{1/4}$. Theorem 5.1 yields the following consequence for approximating $(\min, +)$ circuits.

Corollary 7.4. *If a family $\mathcal{F}$ is (k, l)-clique-like, then* $\mathrm{Min}(\mathcal{F})$ *is exponential in* $\min\{l, n/k\}^{1/4}$.

By Proposition 3.3, the greedy algorithm *can* approximate the minimization problem on any family $\mathcal{F} \subseteq 2^{[n]}$ within the (large but finite) factor n. So, the corollaries above could wake an impression as if they already show that the greedy algorithm can have a better approximation behavior than $(\min, +)$ circuits. This, however, is not the case. The point is that, in order to achieve this (finite) factor, the minimizing greedy algorithm must use the *worst-out* oracle, and this oracle may be forced to solve (at each step) hard (even NP-hard) decision problems: does the remaining set S of ground elements still contain at least one feasible solution $F \in \mathcal{F}$? For example, in the case of lightest k-clique problem for $k = n/2$, the oracle must be able to solve solve an NP-complete problem: does the current graph S still contain a k-clique?

On the other hand, the decision problems to be solved by the *best-in* oracle are usually much simpler (are "local"): does the new partial solution can still be extended to a feasible solution? However, it is easy to see that, like the $(\min, +)$ circuits of too small size, the best-in greedy algorithm *cannot* approximate any of the problems considered above within any finite factor (see also Example 3.6).

Still, in the next section, we will combine the boolean bound (Theorem 5.1) with counting arguments to show that also the best-in greedy algorithm *can* be better than pure DP algorithms: there *exists* a huge number of minimization problems which can be solved even exactly by the best-in greedy algorithm, but which cannot be approximated within any finite factor by $(\min,+)$ circuits of polynomial size.

7.2. (min,+) versus greedy. Recall that a *matroid* is a nonempty family $\mathcal{F} \subseteq 2^E$ of subsets of a finite set E satisfying the *base exchange axiom*:

- for every $A \neq B \in \mathcal{F}$ and every $a \in A \setminus B$ there is a $b \in B \setminus A$ such that $(A \setminus \{a\}) \cup \{b\}$ is a member of $\mathcal{F}$.

Remark 7.1. Our shortcut "matroid" $\mathcal{F}$ actually stands for "family of bases of a matroid:" the matroid itself is usually treated as the downward closure $\mathcal{F}^*$ of $\mathcal{F}$. Note, however, that optimization problems also in this case are defined as minimization or maximization problem on the family $\mathcal{F}$ (of bases), not on the family $\mathcal{F}^*$. In particular, the *minimization* problem on $\mathcal{F}^*$ is to compute the minimum weight of a member of $\mathcal{F}$.

By the Rado theorem [30], if $\mathcal{F}$ is a matroid, then the minimization problem on $\mathcal{F}$ can be solved exactly by the greedy algorithm (even by the best-in greedy algorithm). In contrast, we will now show that most matroids require $(\min,+)$ circuits of exponential size to be even only approximated within any finite factor.

Let $\binom{[n]}{k}$ denote the family of all k-element subsets of $[n] = \{1,\ldots,n\}$. 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 $A \neq B \in \mathcal{H}$.

The following general construction of matroids was implicit in several papers, including those of Piff and Welsh [29], and Knuth [22], and was made explicit by Bansal, Pendavingh and Van der Pol [3, Lemma 8].

Lemma 7.5. *If $\mathcal{H} \subseteq \binom{[n]}{k}$ is separated, then for every subfamily $\mathcal{H}' \subseteq \mathcal{H}$, the family $\mathcal{F} = \binom{[n]}{k} \setminus \mathcal{H}'$ is a matroid.*

Proof. Suppose contrariwise that $\mathcal{F}$ is not a matroid. Then there exist two sets $A \neq B \in \mathcal{F}$ violating the base exchange property: there is an $a \in A \setminus B$ such that $(A \setminus \{a\}) \cup \{b\} \not\in \mathcal{F}$ 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 exchange axiom fails for A and B, neither of S and T can belong to $\mathcal{F}$; hence, both sets S and T belong to $\mathcal{H}'$. But $\mathrm{dist}(S,T) = |\{b,c\}| = 2$, a contradiction with $\mathcal{H}'$ being separated. ∎

Thus, for every separated family $\mathcal{H} \subseteq \binom{[n]}{k}$, Lemma 7.5 gives us $2^{|\mathcal{H}|}$ matroids.

Lemma 7.6 (Knuth [22]) . *For every $1 \leqslant k \leqslant n$, there is a separated family $\mathcal{H} \subseteq \binom{[n]}{k}$ of size $|\mathcal{H}| \geqslant \binom{n}{k}/2n$.*

Proof. Let $l = \lfloor \log_2 n \rfloor + 1$, and let $\vec{v}_1,\ldots,\vec{v}_n$ be vectors in $\{0,1\}^l$ corresponding to the binary representations of numbers from 1 to n. Associate with every vector $b \in \{0,1\}^l$ the family $\mathcal{H}_b$ of all k-element subsets S of $[n]$ such that $\sum_{i\in S} \vec{v}_i = b \bmod 2$. Since vectors $\vec{v}_1,\ldots,\vec{v}_n$ are distinct, the sum modulo 2 of any two of them cannot be the all-0 vector. So, $\mathrm{dist}(S,T) > 2$ must hold for all $S \neq T \in \mathcal{H}_b$. Thus, every family $\mathcal{H}_b$ is separated. Since there are only $2^l \leqslant 2n$ such families, and they exhaust the entire family $\binom{[n]}{k}$, there must be a vector b for which $|\mathcal{H}_b| \geqslant \binom{n}{k}/2n$ holds. ∎

Theorem 7.7. *There exist double-exponentially many in n matroids $\mathcal{F} \subseteq 2^{[n]}$ such that* $\mathrm{Min}(\mathcal{F}) \geqslant \mathrm{Bool}(f_{\mathcal{F}}) = 2^{\Omega(n)}$.

Proof. When applied with $k = \lfloor n/2 \rfloor$, Lemmas 7.5 and 7.6 imply that there are at least $M := 2^{\binom{n}{k}/2n} \geqslant 2^{2^n/4n^{3/2}}$ matroids over $[n]$: at least so many possibilities are to choose a subfamily $\mathcal{H}' \subseteq \mathcal{H}$. On the other hand, the number of monotone boolean circuits of size s does not exceed $L(s) := 2^s(n+2+s)^{2s}$, where n is the number of input variables: for each gate we have two choices for the type of the gate and at most $(n+2+s)^2$ choices for two inputs that feed the gate. This implies that there are at most $L(s)$ families $\mathcal{F}$ of monotone boolean circuit complexity at most s. By taking $s := 2^n/n^3$, we obtain $\log L(s) = O(s \log(n+s)) = O(2^n/n^2)$. So, at least $M - L(s) \geqslant L(s)$ matroids have monotone boolean complexity at least s. Theorem 5.1 yields the same lower bound for approximating $(\min, +)$ circuits. ∎

8. Approximation limitations of (max,+) circuits. As we have seen in the previous section, the approximation power of tropical circuits solving *minimization* problems is determined by the size of monotone boolean circuits solving the decision versions of these problems: if the latter circuits require t gates, then no tropical $(\min, +)$ circuit of size smaller than t can approximate the problem within any finite factor.

The situation with *maximization* problems is entirely different: here the approximation factor is always *bounded*. For example, a trivial $(\max, +)$ circuit $\Phi(x) = \max\{x_1, \ldots, x_n\}$ approximates the maximization problem on *any* family $\mathcal{F} \subseteq 2^{[n]}$ within the (large but finite) factor $\mathrm{r}(\mathcal{F}) := \max\{|S| \colon S \in \mathcal{F}\}$ (assuming that each ground element $i \in [n]$ belongs to at least one set of $\mathcal{F}$). Recall that we only allow nonnegative weights. This trivial observation can be extended as follows.

The *top k-of-n selection* problem $f_{n,k} : \mathbb{R}^n \to \mathbb{R}$ is, given a vector $x = (x_1, \ldots, x_n)$, to compute the sum of the k largest entries of x. Note that the family of feasible solutions of this problem is the family $\binom{[n]}{k}$ of all k-element subsets of $[n]$. For example, $f_{6,3}(1,3,5,2,7,5) = 7 + 5 + 5 = 17$. In particular, $f_{n,1}(x) = \max\{x_1, \ldots, x_n\}$ and $f_{n,n}(x) = x_1 + \cdots + x_n$.

Under the *top k-of-n DP algorithm* we will mean the following simple DP algorithm which solves the top k-of-n selection problem $f_{n,k}$ by starting from $f_{0,0} = 0$ and using the recursion

$$f_{m,l} = \max\{f_{m-1,l}, f_{m-1,l-1} + x_m\} \tag{8.1}$$

for $m = 1, \ldots, n$, $l = 1, \ldots, \min\{k, m\}$, with $f_{m,0} = f_{m-1.m} = 0$. The recursion is related to the Pascal identity $\binom{m}{l} = \binom{m-1}{l} + \binom{m-1}{l-1}$ for binomial coefficients. The number of operations performed by this algorithm and, hence, also the number of gates of the resulting $(\max, +)$ circuit is at most $2kn$.

Say that a family $\mathcal{F} \subseteq 2^{[n]}$ is *k-dense* ($k \geqslant 1$) if every k-element subset of $[n]$ is contained in at least one set of $\mathcal{F}$.

Proposition 8.1. *The top k-of-n DP algorithm uses at most $2kn$ operations to approximate the maximization problem on every k-dense family $\mathcal{F} \subseteq 2^{[n]}$ of feasible solutions within the factor $r = \mathrm{r}(\mathcal{F})/k$.*

Proof. 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 $f_{n,k}(x) \leqslant f(x)$. On the other hand, $f(x)$ is at least the sum of weights of the heaviest k elements in an optimal solution on input x. Since no solution has more than $\mathrm{r}(\mathcal{F})$ elements, the optimal value cannot exceed $(\mathrm{r}(\mathcal{F})/k) \cdot f(x)$, as desired. ∎

Example 8.1. Let $2 \leqslant k \leqslant n$ be even, and let $\mathcal{F}$ be the family of all $\binom{n}{k}$ k-cliques in K_n, each clique viewed as the set of its $\binom{k}{2}$ edges. A trivial approximation factor for the maximization problem on $\mathcal{F}$ is $\mathrm{r}(\mathcal{F}) = \binom{k}{2}$. However, since every set of $l := k/2$ edges lies in at least one k-clique, Proposition 8.1 implies that $O(kn^2)$ operations are enough to approximate the maximization problem on $\mathcal{F}$ by a pure DP algorithm within the quadratically smaller factor $r = \binom{k}{2}/l \leqslant k$.

8.1. Hard to solve exactly, easy to approximate. The following proposition shows that one (fixed) small $(\max, +)$ circuit can approximate a huge number of maximization problems within a small factor $r = 1 + o(1)$, while an exponential number of gates is necessary to solve any of these problems exactly (with $r = 1$). In other words, while *most* problems are hard to solve exactly, they are trivially approximable by just *one* (fixed) $(\max, +)$ circuit. This emphasizes the difficulties to come up with strong *lower* bounds on the size of approximating $(\max, +)$ circuits, even for slightly larger than 1 approximation factors r.

For a family $\mathcal{F}$ of feasible solutions and a real number $r \geqslant 1$, let $\mathrm{Max}_r(\mathcal{F})$ denote the minimum size of a $(\max, +)$ circuit approximating the maximization problem on $\mathcal{F}$ within the factor r.

Proposition 8.2. *There are double-exponentially many in n matroids $\mathcal{F} \subseteq 2^{[n]}$ such that $\mathrm{Max}_1(\mathcal{F}) = 2^{\Omega(n)}$ but the maximization problem on each of these matroids can be approximated within a factor $r = 1 + o(1)$ by one and the same $(\max, +)$ circuit of size n^2.*

Proof. Let n be a sufficiently large even integer, and $k = n/2$. Lemmas 7.5 and 7.6 give us at least $M := 2^{\binom{n}{k}/2n} \geqslant 2^{2^n/4n^{3/2}}$ matroids $\mathcal{F} \subset \binom{[n]}{k}$ with the property that the Hamming distance between any two distinct sets $A \neq B \in \binom{[n]}{k} \setminus \mathcal{F}$ is > 2. We claim that each such family must be $(k-1)$–dense. To see this, take any set $T \in \binom{[n]}{k-1}$, any two distinct elements $x, y \notin T$, and consider the k-element sets $S_1 = T \cup \{x\}$ and $S_2 = T \cup \{y\}$. Since the Hamming distance between S_1 and S_2 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 matroids $\mathcal{F} \subset \binom{[n]}{k}$ which are $(k-1)$-dense. By Proposition 8.1, one $(\max, +)$ circuit of size at most $2kn = n^2$ for the top $(k-1)$-of-n problem approximates the maximization problem on each of these M matroids within the factor $r = k/(k-1) = 1 + 2/(n-2)$.

On the other hand, by Remark 4.1, minimal $(\max, +)$ circuits solving the maximization problem on any family $\mathcal{F}$ must be constant-free. So, by counting constant-free $(\max, +)$ circuits (instead of boolean circuits), the the same argument as in the proof of Theorem 7.7 implies that $\mathrm{Max}_1(\mathcal{F}) = 2^{\Omega(n)}$ holds for doubly-exponentially many of these matroids $\mathcal{F}$. ∎

Remark 8.2. It can be easily shown that *random* maximization problems are also easy to approximate. Let $m = n/2$, $k = m - 2$, and consider a random family $\boldsymbol{\mathcal{F}}$ of m-element subsets of $[n]$ with each m-element subset being included in $\mathcal{F}$ independently with probability $1/2$. Since one k-element set is contained in $l = \binom{n-k}{2} = \Omega(n^2)$ m-element sets, the probability that a fixed k-element set will be contained in *none* of the sets of $\boldsymbol{\mathcal{F}}$ is $(1/2)^l = 2^{-\Omega(n^2)}$. So, by the union bound, the family $\boldsymbol{\mathcal{F}}$ is *not* k-dense with probability at most $\binom{n}{k} \cdot 2^{-\Omega(n^2)} = 2^{-\Omega(n^2)}$. That is, the family $\boldsymbol{\mathcal{F}}$ is k-dense with probability at least $1 - 2^{-\Omega(n^2)}$. By Proposition 8.1, with this probability, the top k-of-n DP algorithm will approximate the maximization problem on a random family $\boldsymbol{\mathcal{F}} \subseteq \binom{[n]}{m}$ within the factor $r = m/k = 1 + o(1)$. This, in particular, implies that, unlike for circuits solving maximization problems *exactly* $(r = 1)$, simple counting arguments cannot show even the *existence* of maximization problems requiring a cubic number of gates to approximate them within, say, the factor $r = 1.001$.

Proposition 8.2 shows a mere *existence* of maximization problems which are difficult to solve exactly but are easy to approximate within a slightly larger than 1 factor. There are, however, also *explicit* problems exhibiting a similar gap.

A family $\mathcal{F}$ of sets is a *Sidon family* if for any sets A, B, C, D in $\mathcal{F}$, $A \cup B = C \cup D$ and $A \cap B = C \cap D$ implies $\{A, B\} = \{C, D\}$. In terms of the incidence 0-1 vectors of sets, this means the following: if we know the sum of incidence vectors of two members of $\mathcal{F}$, then we know which vectors were added.

Example 8.3. Let $\mathcal{F}$ be the family of all $|\mathcal{F}| = \binom{n}{k}$ k-cliques in K_n, viewed as sets of their $\binom{k}{2}$ edges. It is easy to verify that no union of two k-cliques can contain some third k-clique. Indeed, the latter clique must then have a node u not in the first clique and a node v not in the second clique. If $u = v$ then the node u is not covered, and if $u \neq v$ then the edge $\{u, v\}$ is not covered by the first two cliques, a contradiction.

Gashkov and Sergeev [13, Theorem 1] have shown that large Sidon families yield almost maximal lower bounds on the monotone *arithmetic* circuit complexity: if $\mathcal{F} \subseteq 2^{[n]}$ is a Sidon family, then any multilinear polynomial with no negative coefficients, whose family of monomials (viewed as subsets of $[n]$) coincides with $\mathcal{F}$, requires monotone arithmetic $(+, \times)$ circuits of size at least $|\mathcal{F}|$ (it is clear that size $n|\mathcal{F}|$ always suffices). For *uniform* Sidon families (all sets have the same cardinality), this yields the same lower bound $\mathrm{Max}_1(\mathcal{F}) \geqslant |\mathcal{F}|$ on the size of $(\max, +)$ circuits solving the maximization problem on $\mathcal{F}$ *exactly*; see, e.g., [16, Corollary 2.10] or [19, Theorem 9]. In view of this, the following proposition is somewhat surprising: maximization problems on some exponentially large (even explicit) Sidon families can be easily approximated already within factor $r = 2$.

Proposition 8.3. *Let m be an odd integer, and $n = 4m$. Then there is an explicit Sidon family $\mathcal{F} \subseteq 2^{[n]}$ such that $\mathrm{Max}_1(\mathcal{F}) \geqslant 2^{n/4}$ but $\mathrm{Max}_2(\mathcal{F}) \leqslant n$.*

Proof. See Appendix D. ∎

8.2. Set version of the rectangle bound. Proposition 8.2 and Remark 8.2 show that already for small approximation factors $r > 1$, it may be difficult to come up with a large lower bound on $\mathrm{Max}_r(\mathcal{F})$, even by using counting arguments: a single small $(\max, +)$ circuit can approximate a doubly-exponential number of maximization problems within factor $1 + o(1)$. Still, our rectangle bound (Theorem 6.1) will allow us to prove such lower bounds.

Since we are now dealing with sets instead of vectors, it is worth to restate Theorem 6.1 in terms of sets. Under a *rectangle* we will now mean a cross-union

$$\mathcal{A} \vee \mathcal{B} = \{A \cup B \colon A \in \mathcal{A} \text{ and } B \in \mathcal{B}\}$$

of two families which are *cross-disjoint* in that $A \cap B = \emptyset$ holds for all $A \in \mathcal{A}$ and $B \in \mathcal{B}$. We say that a set F *appears* (r, ϵ)*-balanced* in a rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ if there are $A \in \mathcal{A}$ and $B \in \mathcal{B}$ such that for $p := |F|/r$, we have

$$|F \cap (A \cup B)| \geqslant p \quad \text{and} \quad \tfrac{1}{2}\epsilon p \leqslant |F \cap A| \leqslant \epsilon p\,. \tag{8.2}$$

Since, $A \cap B = \emptyset$, we also have $|F \cap B| \geqslant p - |F \cap A| \geqslant (1 - \epsilon)p$. Finally, we say that a family $\mathcal{R}$ *lies below* a family $\mathcal{F}$ if every set of $\mathcal{R}$ is contained in at least one set of $\mathcal{F}$. In these terms, Theorem 6.1 turns into the following fact.

Theorem 8.4 (Set version of the rectangle bound) . *Let $\mathcal{F}$ be a family of sets, $1 \leqslant r < m$ be an approximation factor, and $r/m \leqslant \epsilon < 1$. If $\mathrm{Max}_r(\mathcal{F}) \leqslant t$, then* *there exist t or fewer rectangles lying below $\mathcal{F}$ such that every set $F \in \mathcal{F}$ with $|F| \geqslant m$ elements appears (r,ϵ)-balanced in at least one of them.*

Thus, in order to show that $\mathrm{Max}_r(\mathcal{F})$ is large, it is enough to choose some subfamily $\mathcal{H} \subseteq \mathcal{F}$, some parameters ϵ and $m \geqslant r$ satisfying $r/m \leqslant \epsilon < 1$, and to show that, for every rectangle $\mathcal{R}$ lying below $\mathcal{F}$, the family

$$\mathcal{H}_{\mathcal{R}} = \{F \in \mathcal{H} \colon F \text{ appears } (r,\epsilon)\text{-balanced in } \mathcal{R}\}$$

cannot have more than h sets; then the lower bound $\mathrm{Max}_r(\mathcal{F}) \geqslant |\mathcal{H}|/h$ follows. Besides the cross-disjointness and balancedness, the property of "lying below $\mathcal{F}$" is here crucial. If, say, $\mathcal{F}$ is the family of all perfect matchings in $K_{n,n}$, and if a rectangle $\mathcal{R}$ lies below $\mathcal{F}$, then we know that every set of $\mathcal{R}$ must be a matching.

We now apply Theorem 8.4 to explicit maximization problems.

8.3. Maximization on designs. For a nonnegative real number l and a family $\mathcal{F}$ of sets, let $\#_l(\mathcal{F})$ denote the maximal possible number of sets in $\mathcal{F}$ containing a fixed set with l (or more) elements:

$$\#_l(\mathcal{F}) = \max_{|X|\geqslant l} |\{F \in \mathcal{F} \colon F \supseteq X\}|\,.$$

In other words, $\#_l(\mathcal{F})$ is the maximal possible number of sets in $\mathcal{F}$ whose intersection has l (or more) elements. Hence, $|\mathcal{F}| = \#_0(\mathcal{F}) \geqslant \#_1(\mathcal{F}) \geqslant \ldots \geqslant \#_m(\mathcal{F}) = 1$, where m is the maximum number of elements in a set of $\mathcal{F}$. For example, if $\mathcal{F}$ is the set of edges of a graph, then $\#_1(\mathcal{F})$ is the maximum degree of a vertex in this graph.

Say that a family $\mathcal{F}$ is an (m,d)-*design* $(1 \leqslant d \leqslant m)$ if every of its sets has at least m elements, and no two of them share d or more elements in common, that is, $\#_d(\mathcal{F}) \leqslant 1$ holds. We will see soon (Theorem 8.6) that the maximization problem on some (m,d)-designs can be approximated by a small $(\max,+)$ circuit within the factor $r = \frac{m}{d}$. The following theorem shows that this is actually the best $(\max,+)$ circuits can achieve.

Theorem 8.5 (Arbitrary designs) . *Let $\mathcal{F}$ be an (m,d)-design. Then for every $\epsilon \geqslant 1/(d+1)$, for which $r := (1-\epsilon)\frac{m}{d} \geqslant 1$, we have*

$$\mathrm{Max}_r(\mathcal{F}) \geqslant \frac{|\mathcal{F}|}{\#_l(\mathcal{F})} \quad \textit{for} \quad l = \epsilon d/2\,.$$

Proof. We are going to apply the rectangle bound (Theorem 8.4). Let us first show that the parameter ϵ satisfies the conditions $r/m \leqslant \epsilon < 1$ of this theorem. The condition $r/m \leqslant \epsilon$ is equivalent to $1 \leqslant \epsilon(d+1)$, which is fulfilled because $\epsilon \geqslant 1/(d+1)$. The condition $\epsilon < 1$ is fulfilled because $r \geqslant 1$.

Take now an arbitrary rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ lying below $\mathcal{F}$. Set $p := m/r$, and let $\mathcal{F}_{\mathcal{R}} \subseteq \mathcal{F}$ be the family of all sets $F \in \mathcal{F}$ such that

$$|F \cap A| \geqslant \tfrac{1}{2}\epsilon p \geqslant l \quad \text{and} \quad |F \cap B| \geqslant (1-\epsilon)p = d$$

holds for some $A \in \mathcal{A}$ and $B \in \mathcal{B}$. By Theorem 8.4, 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: we can remove all smaller sets without changing $\mathcal{F}_{\mathcal{R}}$.

For each set $B \in \mathcal{B}$, every set of the family $\mathcal{A} \vee \{B\} = \{A \cup B \colon A \in \mathcal{A}\}$ must be contained in at least one set of $\mathcal{F}$ (since $\mathcal{R}$ lies below $\mathcal{F}$). But all these sets of $\mathcal{F}$ must then contain the same set B with $|B| \geqslant d$ elements. Since no two sets of $\mathcal{F}$ can share

d or more elements in common ($\mathcal{F}$ is an (m,d)-design), this implies that *all* sets of $\mathcal{A} \vee \{B\}$ must be contained in one and the same set F_B of $\mathcal{F}$. In particular, if X is the union of all sets in $\mathcal{A}$, then $F_B \supseteq X \cup B$.

Now take an arbitrary set $F \in \mathcal{F}_{\mathcal{R}}$. Then $|F \cap B| \geqslant d$ holds for some $B \in \mathcal{B}$. We already know that the set B is contained in a set F_B of $\mathcal{F}$. So, $|F \cap F_B| \geqslant d$. Since $\mathcal{F}$ is an (n,d)-design, this implies that $F = F_B$. Hence,

$$\mathcal{F}_{\mathcal{R}} \subseteq \{F_B \colon B \in \mathcal{B}\}\,.$$

Every set F_B in this latter family must contain the same set X (union of all sets in $\mathcal{A}$). Since every set in $\mathcal{A}$ has at least l elements, their union X also has $|X| \geqslant l$ elements. So,

$$|\mathcal{F}_{\mathcal{R}}| \leqslant |\{F_B \colon B \in \mathcal{B}\}| \leqslant |\{F \in \mathcal{F} \colon F \supseteq X\}| \leqslant \#_l(\mathcal{F})\,,$$

as desired. ∎

We now apply Theorem 8.5 to polynomial (m,d)-designs $\mathcal{F}$ (defined before Corollary 7.3). That is, m is a prime power, and ground elements are $n = m^2$ edges of the complete bipartite $m \times m$ graph $K_{m,m} = U \times V$ with $U = \mathrm{GF}(m)$ and $V = \mathrm{GF}(m)$. Members of $\mathcal{F}$ are $|\mathcal{F}| = m^d$ subgraphs $\{(a, p(a)) \colon a \in U\}$ of $K_{m,m}$ determined by polynomials $p(x)$ of degree at most $d-1$ over $\mathrm{GF}(m)$.

A standard result in polynomial interpolation is that 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}: so many solutions of the corresponding system of linear equations, with coefficients of p viewed as variables, there are. Thus, in the polynomial (m,d)-design $\mathcal{F}$, we have

$$\#_l(\mathcal{F}) = m^{d-l} \quad \text{for all} \quad l = 0, 1, \ldots, d\,. \tag{8.3}$$

As we already mentioned in the previous section, Andreev [2] showed that the monotone boolean functions corresponding to polynomial designs require monotone boolean circuits of size $m^{\Omega(d)}$. Nisan and Wigderson [27] used polynomial designs to construct quick pseudorandom generators. We also know (Corollary 7.3) that the *minimization* problem on such designs cannot be approximated within *any* finite factor r by using fewer than $n^{\Omega(d)}$ gates. We will now show that polynomial designs are also hard to approximate by $(\max, +)$ circuits.

Theorem 8.6 (Polynomial designs) . *Let $\mathcal{F}$ be the polynomial (m,d)-design with $d \leqslant m-1$. Then $\mathrm{Max}_r(\mathcal{F}) \leqslant 3m^2$ holds for $r := \frac{m}{d}$, but for any $\frac{1}{d+1} \leqslant \epsilon \leqslant 1 - \frac{d}{m}$,*

$$\mathrm{Max}_{(1-\epsilon)r}(\mathcal{F}) \geqslant m^{\epsilon d/2}\,.$$

In particular, the maximization problem of an (m,d)-design with $d = m/2$ can be approximated within factor $r = 2$ using only $3m^2$ gates, but $m^{\Omega(m)}$ gates are necessary to do this within any factor $2 - c$ for an arbitrarily small constant $c > 0$. So, already a slight decrease of the approximation factor leads to an exponential increase in the circuit size.

Proof. The lower bound follows directly from the property (8.3) and Theorem 8.5: for $l = \epsilon d/2$, we have $\#_l(\mathcal{F}) = m^{d-l}$ and, hence, $\mathrm{Max}_{(1-\epsilon)r}(\mathcal{F}) \geqslant |\mathcal{F}|/\#_l(\mathcal{F}) \geqslant m^d/m^{d-l} = m^l$.

To show the upper bound $\mathrm{Max}_r(\mathcal{F}) \leqslant 3m^2$ for $r = m/d$, 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 an edge incident to the vertex $i \in U$. Then it is enough to use additional $2dm$ operations to compute the sum W of the largest d of these numbers, by solving the top d-of-m selection problem (see (8.1)). Then W is a sum of weights of d *heaviest* edges with no common vertex in U. This later property of the edges (no common vertex on the left), together with the property (8.3) of polynomial designs (for $l = d$), implies that these d edges 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 edges of an optimal solution $S \in \mathcal{F}$ cannot exceed W. Since $|S| = m$, the weight of this solution cannot exceed $(m/d)W$, as desired. ∎

Remark 8.4. Note that Theorem 8.6 does *not* show that greedy can achieve *better* approximation factors than pure DP algorithms. The (best-in) greedy algorithm *also* achieves approximation factor $r = \frac{m}{d}$ on polynomial (m, d)-designs because it also computes the sum of the d largest numbers among $y_1, \ldots, y_m$, where y_i is the maximum weight of an edge incident to the vertex $i \in U$. But just as pure DP algorithms the greedy algorithm cannot achieve any better factor $r = (1 - \epsilon)\frac{m}{d}$ for $\epsilon > 0$.

The proof is straight-forward, but let us do this for completeness. Take arbitrary two sets $S \neq T \in \mathcal{F}$, and a subset $A \subset S$ of $|A| = d$ elements. Since $\mathcal{F}$ is an (m, d)-design, we have that $|S \cap T| \leqslant d - 1$, implying that A is not contained in T. Give now weight $1/(1 - \epsilon) > 1$ to all elements of A, weight 1 to all elements of $T \setminus A$, and zero weight to the rest. Then the (best-in) greedy algorithm tries elements of weight $1/(1 - \epsilon)$ first, gets all $|A| = d$ of them, but then is stuck because no element of weight 1 fits; hence, the greedy algorithm achieves total weight $g = d/(1 - \epsilon)$. But the optimum is at least $|T| = m$. Hence, $r \geqslant |T|/g \geqslant (1 - \epsilon)\frac{m}{d}$, as claimed.

8.4. Maximum weight matchings in hypergraphs. As we mentioned in Remark 8.4, Theorem 8.6 does *not* imply that greedy can achieve better approximation factors than pure DP algorithms. To show that greedy still can be better, we consider another maximization problem: maximum weight matchings in hypergraphs.

Consider a k-uniform k-partite hypergraph on mk vertices. That is, 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 $n := |E| = m^k$ edges. Two edges e and e' are disjoint ($e \cap e' = \emptyset$) if they differ in all k positions. A *matching* is a set of disjoint edges, and is *perfect* if it has the maximum possible number m of edges.

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

$$f(x) = \max\{x_{e_1} + \cdots + x_{e_m} : e_i \in E \text{ and } e_i \cap e_j = \emptyset \text{ for all } i \neq j\}$$

of a perfect matching. Since no perfect matching has more than m edges, $\mathrm{Max}_r(\mathcal{F}_{n,k}) \leqslant m^k = n$ is a trivial upper bound for the approximation factor $r = m$, by using a circuit which simply picks the heaviest edge.

The greedy algorithm can approximate the maximization problem on $\mathcal{F} = \mathcal{F}_{n,k}$ 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.

Theorem 8.7 (Matchings) . *Let* $4 \leqslant k \leqslant \log m - 3 \log\log m$ *be an integer, and*

$n = m^k$ be sufficiently large. If $r \leqslant 2^k/9$, then

$$\mathrm{Max}_r(\mathcal{F}_{n,k}) = 2^{\Omega(n^{1/k}/2^k)}\,.$$

Thus, the greedy algorithm can achieve the approximation factor k, but no polynomial size $(\max,+)$ circuit can achieve an even exponentially worse factor. Actually, the gap occurs already for small (constant) values of k. Say, for $k=6$, the greedy algorithm can approximate the maximization problem on $\mathcal{F}_{n,k}$ within factor $r=6$, but any pure DP algorithm approximating this problem even within factor $r = 2^6/9 > 7$ must use $2^{n^{\Omega(1)}}$ operations.

Proof. We are going to apply the rectangle bound (Theorem 8.4) with $\epsilon := 2/3$. So, take an arbitrary rectangle $\mathcal{R} = \mathcal{A} \vee \mathcal{B}$ lying below $\mathcal{F} = \mathcal{F}_{n,k}$. Hence, all sets $A \cup B$ with $A \in \mathcal{A}$ and $B \in \mathcal{B}$ must be matchings. Take $d := \lceil m/3r \rceil$, and consider the family

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

Our goal is to upper bound the maximum possible number $|\mathcal{F}_{\mathcal{R}}|$ of sets in such a family.

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 $A \cup B$ must also be matchings ($\mathcal{R}$ lies below $\mathcal{F}$), we actually know that matchings A and B are also *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 we have that $S \cap T = \emptyset$ (this is a crucial property).

So, call a matching $A \subset V_1 \times \cdots \times V_k$ *S-matching* if $A \subset S^k$ 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 S, and at least d edges lying in 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 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 $\mathcal{F}_{\mathcal{R}} = \emptyset$.

1. Pick in each S_i a subset $U_i \subseteq S_i$ of $|U_i| = d$ vertices, and in each T_i a subset $W_i \subseteq T_i$ of $|W_i| = d$ 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 $m_i = |S_i|$. Here we used the (clear from combinatorial interpretation) inequality $\binom{x}{a}\binom{y}{b} \leqslant \binom{x+y}{a+b}$.

2. Pick a perfect matching A in $U_1 \times \cdots \times U_k$ and a perfect matching B in $W_1 \times \cdots \times W_k$. 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,

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

Now let $t = \mathrm{Max}_r(\mathcal{F})$ be the minimum number of gates in a $(\max, +)$ circuit. Since there are $|\mathcal{F}| = (m!)^{k-1}$ perfect matchings, the rectangle bound (Theorem 8.4) implies that

$$t \geqslant \frac{|\mathcal{F}|}{|\mathcal{F}_{\mathcal{R}}|} \geqslant \frac{\Delta^{k-1}}{\binom{m}{2d}},$$

where

$$\Delta := \frac{m!}{\binom{m}{2d}} \cdot \frac{1}{(d!)^2(m-2d)!} = \frac{(2d)!}{(d!)^2} = \binom{2d}{d}.$$

Since $\binom{2d}{d} \geqslant 2^{2d}/\sqrt{4d} \geqslant 2^{2d}/d$ and $\binom{m}{2d} \leqslant (\mathrm{e}m/2d)^{2d}$, we have

$$t \geqslant \binom{2d}{d}^{k-1} \cdot \binom{m}{2d}^{-1} \geqslant \frac{(2^{k-1})^{2d}}{d^{k-1}} \cdot \frac{(2d)^{2d}}{(\mathrm{e}m)^{2d}} = \left(\frac{2^k d}{\mathrm{e}m}\right)^{2d} \cdot d^{1-k}.$$

Since, by our assumption, the approximation factor is $r = 2^k/9$, and since we have set $d := \lceil m/3r \rceil$, we have $3m/2^k \leqslant d \leqslant 3m/2^k + 1$. So,

$$t \geqslant \left(\frac{3}{\mathrm{e}}\right)^{6m/2^k} \cdot d^{-k} \geqslant 2^{0.8m/2^k - k\log 4m}.$$

Since $k \leqslant \log m - 3\log\log m$, we have $m/2^k \geqslant \log^3 m \gg k\log 4m$, and the desired lower bound $t = 2^{\Omega(m/2^k)} = 2^{\Omega(n^{1/k}/2^k)}$ follows. ∎

9. Conclusion and open problems. We proved the first non-trivial lower bounds for *approximating* tropical circuits. Since pure DP algorithms are just special (recursively constructed) tropical circuits, these bounds hold also for approximating pure DP algorithms. The results imply that the approximation powers of greedy and pure DP algorithms are incomparable. Some interesting questions still remain open.

Minimization. We have shown in Theorem 7.7 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.

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.*

A related (more general) problem is to develop lower bound arguments for monotone boolean circuits of *bounded* semantic degree (see Appendix C), that are easier to apply than Razborov's general method of approximations, and its symmetric versions; see [18, Chapter 9].

Let $\mathcal{T}_n$ be the family of all spanning trees in a complete n-vertex graph K_n. Since this family is (the family of bases of) the graphic matroid, both minimization and maximization problems can be solved by greedy algorithms. On the other hand, it is known that $2^{\Omega(\sqrt{n})}$ gates are necessary to solve the minimization problem on $\mathcal{T}_n$ by a $(\min, +)$ circuit *exactly*, that is, $\mathrm{Min}_1(\mathcal{T}_n) = 2^{\Omega(\sqrt{n})}$ [21]. But what if we only want to *approximate* the minimum weight spanning tree problem, is $\mathrm{Min}_r(\mathcal{T}_n)$ super-polynomial also for $r > 1$? It can be shown (see Example C.3 in Appendix C) that $\mathrm{Min}_r(\mathcal{T}_n) = O(n^4)$ holds for some *finite* factor $r \leqslant n - 1$.

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

Maximization. The next problem asks whether Theorem 8.7 holds for matchings in bipartite graphs. In the *heaviest matching* problem $\mathcal{M}_n$, we are given an assignment of nonnegative real weights to the edges of a complete bipartite $n \times n$ graph $K_{n,n}$, and the goal is to compute the minimum weight of a matching in $K_{n,n}$. The greedy algorithm can approximate this problem within the factor 2.

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

For $k = 2$, the calculations made in the proof of Theorem 8.7 result in a trivial bound. So, new arguments are necessary in the bipartite case.

Our next question concerns the *maximization* problem on the graphic matroid $\mathcal{T}_n$. We know that, for factor $r = 1$, we have $\mathrm{Max}_1(\mathcal{T}_n) = 2^{\Omega(\sqrt{n})}$ [21].

Problem 4. *Is* $\mathrm{Max}_2(\mathcal{T}_n)$ *also exponential in* n*?*

We have shown in Theorem 7.7 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?

Problem 5. *Do there* exist *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 8.2, the answer is "yes" for $r = 1$. But this proposition and Remark 8.2 indicate that direct counting arguments may fail to answer this question for slightly larger approximation factors r.

Tradeoffs between minimization and maximization. Given a family $\mathcal{F} \subseteq 2^{[n]}$ of feasible solutions, what is the difference in the hardness of approximation of the minimization and of the minimization problems on the *same* family $\mathcal{F}$?

A family $\mathcal{F} \subseteq 2^{[n]}$ is *uniform* if all its sets have the same cardinality. For non-uniform families $\mathcal{F}$, both gaps $\mathrm{Max}_1(\mathcal{F})/\mathrm{Min}_1(\mathcal{F})$ and $\mathrm{Min}_1(\mathcal{F})/\mathrm{Max}_1(\mathcal{F})$ can be exponential. For example, $\mathrm{Max}_1(\mathcal{F})/\mathrm{Min}_1(\mathcal{F}) = 2^{\Omega(n)}$ holds for the family $\mathcal{F}$ of all simple paths in K_n from vertex 1 to vertex n (see, e.g., [19, Section 7]). But it is known that no gap is possible for uniform families $\mathcal{F}$: then $\mathrm{Min}_1(\mathcal{F}) = \mathrm{Max}_1(\mathcal{F})$ holds (see, for example, [19, Theorem 9]).

The following proposition shows that the situation is entirely different if we consider *approximating* circuits: even for uniform families $\mathcal{F}$, the gap $\mathrm{Min}_r(\mathcal{F})/\mathrm{Max}_s(\mathcal{F})$ can be exponential for the approximation factors $s = 1 + o(1)$ and $r \geqslant 1$ arbitrarily large.

Proposition 9.1. *There are doubly exponentially many in* n *uniform families* $\mathcal{F} \subseteq 2^{[n]}$ *such that* $\mathrm{Max}_r(\mathcal{F}) \leqslant n^2$ *already for* $r = 1 + o(1)$ *but* $\mathrm{Min}_r(\mathcal{F}) \geqslant \mathrm{Bool}(\mathcal{F}) = 2^{\Omega(n)}$ *holds for all* $r \geqslant 1$.

Proof. Matroids $\mathcal{F}$ given by Lemmas 7.5 and 7.6 are uniform families, and Proposition 8.2 shows that $\mathrm{Max}_{1+o(1)}(\mathcal{F}) \leqslant n^2$ holds for every such matroid $\mathcal{F}$. On the other hand, Theorem 7.7 shows that for doubly exponentially many in n of such matroids, and for any finite factor $r \geqslant 1$, we have $\mathrm{Min}_r(\mathcal{F}) \geqslant \mathrm{Bool}(\mathcal{F}) = 2^{\Omega(n)}$. ∎

Large $\mathrm{Min}_r(\mathcal{F})/\mathrm{Max}_s(\mathcal{F})$ gaps for $r \geqslant s \geqslant 1$ are also achievable on *explicit* uniform families $\mathcal{F}$. For example, if $\mathcal{F}$ is the polynomial (m,d)-design with $d = m^{1/3}$, then $\mathrm{Min}_r(\mathcal{F}) = m^{\Omega(d)}$ holds for any finite factor $r \geqslant 1$ (Corollary 7.3), but $\mathrm{Max}_s(\mathcal{F}) = O(m^2)$ already for factor $s = m/d$ (Theorem 8.6). If $\mathcal{F}$ is the family of all k-cliques in K_n for $k = \sqrt{n}$, then $\mathrm{Min}_r(\mathcal{F}) = 2^{\Omega(n^{1/8})}$ holds for all $r \geqslant 1$ (Corollary 7.4), but $\mathrm{Max}_s(\mathcal{F}) = O(n^2)$ for $s = k$ (Example 8.1 and Proposition 8.1).

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

Note that the separating family $\mathcal{F}$ is here required to be uniform (or at least form an antichain): without this requirement, the gap can be *artificially* made large. To see this, take an arbitrary uniform family $\mathcal{F} \subseteq 2^{[n]}$ with large $\mathrm{Max}_r(\mathcal{F})$ (as in Theorems 8.6 and 8.7), and extend it to a nonuniform family $\mathcal{F}'$ by adding all single element sets. Then $\mathrm{Min}_1(\mathcal{F}') \leqslant n$ (just compute the minimum weight of a single element), but $\mathrm{Max}_r(\mathcal{F}')$ still remains large.

Appendix A. Proof of Decomposition Lemma. The goal of this section is to give a short and direct proof of Lemma 6.2. Recall that a *norm-measure* is any assignment 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 sub-additive in that the norm of a sum of two vectors does not exceed the sum of their norms.

Let $B \subset \mathbb{N}^n$ be a set of vectors, $p \geqslant 2$ and $1/p \leqslant \epsilon < 1$. Suppose that B can be produced by a Minkowski circuit Φ of size t. Our goal is to show that then B can be written as a union of t rectangles $X + Y \subseteq B$ with the following property:

($*$) for every norm-measure $\mu : \mathbb{N}^n \to \mathbb{R}_+$, and for every vector $b \in B$ of norm $\mu(b) \geqslant p$, at least one of these rectangles $X + Y$ contains vectors $x \in X$ and $y \in Y$ such that $x + y = b$ and $\frac{1}{2}\epsilon p \leqslant \mu(x) \leqslant \epsilon p$.

For a gate v of Φ, let $X_v \subset \mathbb{N}^n$ be the set of vectors produced at this gate. That is, X_v is some of the sets $\{\vec{0}\}, \{\vec{e}_1\}, \ldots, \{\vec{e}_n\}$ if v is an input node, $X_v = X_u \cup X_w$ 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.

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 at w. 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 $Y_v = B$ (if v holds a constant), or $X_v = \{\vec{e}_i\}$ and $Y_v = \{b - \vec{e}_i : b \in B, b_i = 1\}$ (if v holds the variable x_i).

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. So, since we have only t gates v in the circuit, it is enough to show that the collection of rectangles $X_v + Y_v$ over all gates v has the desired property.

To show this, fix some norm-measure $\mu : \mathbb{N}^n \to \mathbb{R}_+$, and some vector $b \in B$ of norm $\mu(b) \geqslant p$. By a *b-decomposition* at a gate v we will mean a pair (x, y) of vectors (if there is one) with $x \in X_v$ and $y \in Y_v$ such that $x + y = b$. Our goal is to show that the b-decomposition $x + y = b$ at some gate v must satisfy $\frac{1}{2}\epsilon p \leqslant \mu(x) \leqslant \epsilon p$.

Define the *weight* of a pair (x, y) of vectors as the norm $\mu(x)$ of the first vector of the pair. Note that at the output gate, we have the (unique) b-decomposition $(x, y) = (b, \vec{0})$ of weight $\mu(x) = \mu(b) \geqslant p$.

Claim A.1. *If (x, y) is a b-decomposition at a gate v, and if u and w are the gates entering v, then there is a b-decomposition at gate u or at gate w whose weight is at least half of the weight of the pair (x, y).*

Proof. If v is a Max 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 b-decomposition at gate u (if $x \in X_u$) or at gate w (if $x \in X_w$), and the claim is trivial in this case.

Assume now that v is a 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 $X_u + X_w + Y_v \subseteq B$ 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 b-decomposition at gate u, and the pair $(x_w, x_u + y)$ is a b-decomposition at gate w. Since $x = x_u + x_w$, and since the norm is subadditive, one of the norms $\mu(x_u)$ and $\mu(x_w)$ must be at least $\frac{1}{2} \cdot \mu(x)$. It therefore suffices to take the input with the larger norm. ∎

We now start at the output gate with the b-decomposition $(x, y) = (b, \vec{0})$, and traverse an input-output path P *backwards* using the following rule: if v is a currently reached gate, then go to that of the two inputs whose b-decomposition has larger weight (in the case of equality, go to any of the inputs). Claim A.1 ensures that we will eventually reach some input node. If this node holds $\{\vec{0}\}$, then the only b-decomposition $(x, y) = (\vec{0}, b)$ has weight $\mu(x) = \mu(\vec{0}) \leqslant 1$, and if this gate holds $\{\vec{e}_i\}$, then the only b-decomposition $(x, y) = (\vec{e}_i, b - \vec{e}_i)$ also has weight $\mu(x) = \mu(\vec{e}_i) \leqslant 1$, which is at most ϵp, because $\epsilon \geqslant 1/p$.

On the other hand, the (also unique) b-decomposition $(x, y) = (b, \vec{0})$ at the output gate has weight $\mu(x) = \mu(b) \geqslant p$, which is strictly larger than ϵp, because $\epsilon < 1$. So, there must be an edge (u, v) in the path P at which the jump from $\leqslant \epsilon p$ to $> \epsilon p$ happens. That is, there must be a b-decomposition $x + y = b$ at gate u and a b-decomposition $x' + y' = b$ at gate v such that $\mu(x) \leqslant \epsilon p$ but $\mu(x') > \epsilon p$. Together with Claim A.1, the latter inequality gives the inequality $\mu(x) \geqslant \frac{1}{2} \cdot \mu(x') > \epsilon p/2$.

We have thus found a gate $X_u + Y_u$ and vectors $x \in X_u$ and $y \in Y_u$ such that $x + y = b$ and $\epsilon p/2 < \mu(x) \leqslant \epsilon p$, as desired. ∎

Appendix B. Tight structure of approximating 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 $a \in A$. Note that now (unlike in section 4) we have *all* input weightings $x \in \mathbb{R}^n_+$ in our disposal.

Our goal is to answer the following question: if we know that a given tropical circuit approximates a given optimization (minimization or maximization) problem within a given factor, what can then be said about the set B of vectors *produced* by that circuit? Using elementary arguments, we partially answered this question in section 4: we gave properties, which the set B must *necessarily* have. Now we will use convexity arguments to give a complete characterization of the properties of B which also are *sufficient* for the circuit to approximate a given problem.

B.1. A consequence of Farkas' lemma. We will use the following fact relating convexity with optimization. For a real vector $a = (a_1, \ldots, a_n)$ and a scalar $\lambda \in \mathbb{R}$, $\lambda \cdot a$ stands for the vector $(\lambda a_1, \ldots, \lambda a_n)$. If $A \subseteq \mathbb{R}^n$ is a set of vectors, then $\lambda \cdot A$ stands for the set of vectors $\{\lambda \cdot a \colon a \in A\}$.

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

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

[1] We will only use arrows $\vec{a}_i$ for *indexed* vectors to indicate that these are vectors, not their entries.

By Carathéodory's theorem [6], we can always assume that $m \leqslant n+1$. 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 B,

$$\min_{b \in B} \langle b, x\rangle \leqslant \langle c, x\rangle \leqslant \max_{b \in B} \langle b, x\rangle . \tag{B.1}$$

Let $\mathrm{Conv}(B)$ denote the set of all convex combinations of vectors in B, that is, the convex hull of B. As before, for two vectors $c, v \in \mathbb{R}^n$, we write $c \leqslant v$ if $c_i \leqslant v_i$ holds for all $i = 1, \ldots, n$.

The following consequence of Farkas' lemma was observed already by Jerrum and Snir [16, Corollary A3].

Lemma B.1. *Let B be a finite set of vectors and v a vector in $\mathbb{R}^n$. Then*

$$\forall x \in \mathbb{R}^n_+ \colon \max_{b \in B} \langle b, x\rangle \geqslant \langle v, x\rangle \quad \textit{if and only if} \quad \exists c \in \mathrm{Conv}(B) \colon c \geqslant v . \tag{B.2}$$

$$\forall x \in \mathbb{R}^n_+ \colon \min_{b \in B} \langle b, x\rangle \leqslant \langle v, x\rangle \quad \textit{if and only if} \quad \exists c \in \mathrm{Conv}(B) \colon c \leqslant v . \tag{B.3}$$

Proof. Note that claim (B.3) is equivalent to (B.2): just take the set $-B$ and the vector $-v$. Also, the "if" direction in (B.2) follows directly from the averaging property (B.1). To prove the "only if" direction in (B.2), we use the following form of Farkas' lemma [10]; see, for example, Schrijver [32, Corollary 7.1d]:

- If, for all $x \in \mathbb{R}^n$, the linear inequalities $\langle \vec{a}_1, x\rangle \leqslant 0, \ldots, \langle \vec{a}_m, x\rangle \leqslant 0$ imply the linear inequality $\langle w, x\rangle \leqslant 0$, then w is a nonnegative linear combination of $\vec{a}_1, \ldots, \vec{a}_m$.

Now let $B = \{\vec{b}_1, \ldots, \vec{b}_m\}$. Our assumption is that $\max_i \langle \vec{b}_i, y\rangle \geqslant \langle v, y\rangle$ holds for all $y \in \mathbb{R}^n_+$. This is equivalent to: for every $y \in \mathbb{R}^n_+$ and $z \in \mathbb{R}$, the linear inequalities $\langle \vec{b}_1, y\rangle \leqslant z, \ldots, \langle \vec{b}_m, y\rangle \leqslant z$ imply the linear inequality $\langle v, y\rangle \leqslant z$.

Set $w := (v, -1)$, $\vec{a}_i := (\vec{b}_i, -1)$ for $i = 1, \ldots, m$ and $\vec{a}_{m+i} := (-\vec{e}_i, 0)$ for $i = 1, \ldots, n$, where $\vec{e}_i \in \{0,1\}^n$ is the ith unit vector. By taking $x := (y, z)$ for $y \in \mathbb{R}^n$ and $z \in \mathbb{R}$, our assumption turns into: linear inequalities $\langle \vec{a}_1, x\rangle \leqslant 0, \ldots, \langle \vec{a}_{m+n}, x\rangle \leqslant 0$ imply the linear inequality $\langle w, x\rangle \leqslant 0$; the last inequalities $\langle \vec{a}_{m+i}, x\rangle \leqslant 0$ ensure that $y \in \mathbb{R}^n_+$. So, Farkas' lemma implies that there exist nonnegative scalars $\lambda_1, \ldots, \lambda_{m+n}$ such that

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

This yields $\lambda_1 + \cdots + \lambda_m = 1$ and $v \geqslant c := \sum_{i=1}^m \lambda_i \vec{b}_i$, as desired. ∎

The following direct consequence of Lemma B.1 compares the values of optimization problems. Interesting in our context here are the "only if" directions.

Lemma B.2. *Let $U, V \subset \mathbb{R}^n$ be finite sets of vectors. Then*

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

B.2. Consequences for approximating tropical circuits. The following lemma states that a $(\max, +)$ circuit approximates the maximization problem on a set $A \subset \mathbb{N}^n$ of feasible solutions within a factor r if and only if the set $B \subset \mathbb{N}^n$ produced by the circuit is "sandwiched" between $\frac{1}{r} \cdot A$ and A.

Lemma B.3 (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. The maximization problem on A is of the form $f(x) = \max_{a \in A}\langle a, x\rangle$. By Lemma 3.1, the circuit Φ solves the maximization problem of the form $\Phi(x) = \max_{b \in B}\langle b, x\rangle + c_b$ for some constants $c_b \in \mathbb{R}_+$. So, the circuit approximates the maximization problem on A within factor r if and only if $f(x)/r \leqslant \Phi(x) \leqslant f(x)$ holds for all weightings $x \in \mathbb{R}^n_+$. Since $\Phi(x) \leqslant f(x)$ must hold for the all-0 input weighting $x = \vec{0}$, $c_b = 0$ must hold for all $b \in B$. So, the problem solved by the circuit is actually of the form $\Phi(x) = \max_{b \in B}\langle b, x\rangle$.

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

We have a similar characterization of sets produced by *constant-free* approximating $(\min, +)$ circuits, that is, for circuits, where constants are not allowed as inputs.

Recall that a vector $a \in \mathbb{R}^n$ *lies above* the convex hull $\mathrm{Conv}(B)$ of a set $B \subseteq \mathbb{R}^n$ of vectors if $a \geqslant c$ holds for some convex combination $c = \lambda_1 \cdot \vec{b}_1 + \cdots + \lambda_m \cdot \vec{b}_m$ of vectors in B. If $S_a = \{i\colon a_i \neq 0\}$ is the support of vector a, then we only know that $S_{\vec{b}_i} \subseteq S_a$ must then hold for all i. If we have a stronger property that $S_{\vec{b}_i} = S_a$ holds for all vectors $\vec{b}_i$ in such a combination (for which $\lambda_i \neq 0$, of course), then we say that vector a *lies tightly above* $\mathrm{Conv}(B)$.

Lemma B.4 (Minimization) . *Let* Φ *be a constant-free* $(\min, +)$ *circuit, and* $B \subset \mathbb{N}^n$ *the set of vectors produced by* Φ. *If* $A \subset \mathbb{N}^n$, *then the first two of the following three assertions are equivalent. If* $A \subseteq \{0,1\}^n$ *and* A *is an antichain, then all three 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* $\mathrm{Conv}(B)$.
(3) B *lies above* A, *and* $r \cdot A$ *lies tightly above* $\mathrm{Conv}(B)$.

Proof. By Lemma 3.1, the circuit Φ solves the minimization problem $\Phi(x) = \min_{b \in B}\langle b, x\rangle$ (there are no free coefficients since the circuit is constant-free). The minimization problem on A is also 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 $x \in \mathbb{R}^n_+$. By Lemma B.1, this happens if and only if the set B lies above $\mathrm{Conv}(A)$ (to ensure $f(x) \leqslant \Phi(x)$) and the set $r \cdot A$ lies above $\mathrm{Conv}(B)$ (to ensure $\Phi(x) \leqslant r \cdot f(x)$). This proves the equivalence of assertions (1) and (2).

Now suppose that $A \subseteq \{0,1\}^n$, and A is an antichain. The implication (3) $\Rightarrow$ (2) is obvious. For the converse implication, suppose that assertion (2) holds. To show that then B must lie above A (not only above $\mathrm{Conv}(A)$), take an arbitrary vector $b \in B$. Since, by item (2), the set B lies above $\mathrm{Conv}(A)$, there must be a vector $a \in A$ and a scalar $0 < \lambda \leqslant 1$ such that $b \geqslant \lambda \cdot a$. Vector a is a 0-1 vector, and b is a nonnegative integer vector. So, $b \geqslant a$ must hold.

To show that $r \cdot A$ lies *tightly* above $\mathrm{Conv}(B)$, take an arbitrary vector $a \in A$. Since, by item (2), the set $r \cdot A$ lies above $\mathrm{Conv}(B)$, the inequality $c \leqslant r \cdot a$ must hold for some convex combination $c = \lambda_1 \cdot \vec{b}_1 + \cdots + \lambda_m \cdot \vec{b}_m$ of vectors in B. We can assume w.l.o.g. that all scalars λ_i are positive. It remains to show that then $S_{\vec{b}_i} = S_a$ must hold for all $i = 1, \ldots, m$.

Since 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 $c \leqslant r \cdot a$ implies that $S_{\vec{a}_i} \subseteq S_{\vec{b}_i} \subseteq S_a$ 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 $c \geqslant a$ and $S_{\vec{b}_i} = S_a$ for all $i = 1, \ldots, m$, as desired. ∎

B.3. Eliminating constant inputs. In general, optimization problems solved by tropical circuits Φ need not be *monic*: they can be of the form $\Phi(x) = \min_{b \in B}\langle b, x\rangle + c_b$ or $\Phi(x) = \max_{b \in B}\langle b, x\rangle + c_b$, where the free coefficients $c_b \in \mathbb{R}_+$ may be nonzero constants. Now, an optimization problem on a given set $A \subset \mathbb{N}^n$ of feasible solutions is always monic: it is of the form $f(x) = \min_{a \in A}\langle a, x\rangle$ or $f(x) = \max_{a \in A}\langle a, x\rangle$.

Intuitively, constant inputs should not help to solve or approximate monic optimization problems: these inputs can only contribute to the free coefficients. In the case of *maximization* problems, this intuition is easy to confirm: on the all-0 input weighting $x = \vec{0}$, we then have $\max\{c_b \colon b \in B\} = \Phi(\vec{0}) \leqslant f(\vec{0}) = 0$. So, $c_b = 0$ must hold for all $b \in B$, and we can safely replace all constant inputs by zeros.

In the case of *minimization* problems, such a simple argument does not work. Still, it is possible to eliminate constant inputs also from $(\min, +)$ circuits by combining Lemmas 3.2 and B.2.

Lemma B.5. *Let $A \subset \mathbb{N}^n$ and $\vec{0} \notin A$. If the minimization problem on A can be r-approximated by a $(\min, +)$ circuit of size s, then this problem can also be r-approximated by a constant-free $(\min, +)$ circuit of size at most s.*

Proof. Let Φ be a $(\min, +)$ circuit r-approximating the minimization problem $f(x) = \min_{a \in A}\langle a, x\rangle$ on A, and $B \subset \mathbb{N}^n$ be the set of vectors produced by Φ. By Lemma 3.1, the circuit Φ solves the minimization problem $\Phi(x) = \min_{b \in B}\langle b, x\rangle + c_b$ for some free coefficients $c_b \in \mathbb{R}_+$. Let us first show that also then (when some free coefficients c_b may be nonzero) the set B must have the two properties given in item (ii) of Lemma B.4.

Claim B.6. *The set B lies above* $\mathrm{Conv}(A)$ *and the set $r \cdot A$ lies above* $\mathrm{Conv}(B)$.

Proof. Since the original circuit Φ r-approximates f, we know that for all inputs $x \in \mathbb{R}_+^n$, the inequalities $f(x) \leqslant \Phi(x) \leqslant r \cdot f(x)$ must hold.

To show that B must lie above $\mathrm{Conv}(A)$, apply Lemma B.2(ii) with $U := \{(a, 0) \colon a \in A\}$ and $V := \{(b, c_b) \colon b \in B\}$. The first inequality $f(x) \leqslant \Phi(x)$ then implies that V must lie above $\mathrm{Conv}(U)$. So, the projection B of V onto the first n positions must also lie above the projection $\mathrm{Conv}(A)$ of $\mathrm{Conv}(U)$ onto these positions.

To show that $r \cdot A$ must lie above $\mathrm{Conv}(B)$, apply Lemma B.2(ii) with $U := \{(b, c_b) \colon b \in B\}$ and $V := \{(r \cdot a, 0) \colon a \in A\}$. The second inequality $\Phi(x) \leqslant r \cdot f(x)$ then implies that V must lie above $\mathrm{Conv}(U)$. So, the projection $r \cdot A$ of V onto the first n positions must also lie above the projection $\mathrm{Conv}(B)$ of U onto these positions. (Actually, we have a bit more: since the $(n+1)$-th position of all vectors in V is zero, every vector of $r \cdot A$ must contain a convex combination of vectors $b \in B$ with $c_b = 0$. But we will not use this additional property.) ∎

Now let Φ^* be the constant-free version of Φ (see subsection 3.3 for its construction). By Lemma 3.2, the circuit Φ^* produces either the set B or the set $B \setminus \{\vec{0}\}$. Since $\vec{0} \notin A$, the function f is unbounded: for every constant $c \in \mathbb{R}_+$, there must be an input weighting $x \in \mathbb{R}_+^n$ such that $f(x) > c$ holds. This implies that $\vec{0} \notin B$: otherwise, we would have that $f(x) \leqslant \Phi(x) \leqslant \langle \vec{0}, x\rangle + c_{\vec{0}} = c_{\vec{0}}$ must hold for all inputs $x \in \mathbb{R}_+^n$.

So, the circuit Φ^* produces the same set B as the original circuit Φ. Since the circuit Φ^* is constant-free, we can apply Lemma B.4. Together with Claim B.6, this lemma implies that the circuit Φ^* also r-approximates our minimization problem f of A, as desired. ∎

Appendix C. A converse of the boolean bound. In section 5, we have shown (Theorem 5.1) 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. The goal of this appendix is to show that approximating $(\min, +)$ circuits and monotone boolean circuits are even more tightly related.

Recall that the *decision version* of the minimization problem $f(x) = \min_{a\in A} \langle a, x\rangle$ is the monotone boolean function

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

where $S_a = \{i\colon a_i \neq 0\}$ is the support of vector a. Note that, for every input $x \in \{0,1\}^n$, we have

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

Lemma C.1. *Let $A \subset \{0,1\}^n$ be an antichain, and Φ be a monotone boolean circuit. Then Φ computes f_A if and only if the set $B \subset \mathbb{N}^n$ produced by Φ has the following two properties:*

(i) *for every vector $b \in B$ there is a vector $a \in A$ such that $S_b \supseteq S_a$;*
(ii) *for every vector $a \in A$ there is a vector $b \in B$ such that $S_a = S_b$.*

Proof. By Lemma 3.1, the boolean function computed by the circuit Φ is of the form $\Phi(x) = \bigvee_{b\in B} \bigwedge_{i\in S_b} x_i$. So, the "if" direction follows directly from the property (C.1).

To show that "only if" direction, assume that $\Phi(x) = f_A(x)$ holds for all $x \in \{0,1\}^n$. If the property (i) is violated, then there is a vector $b \in B$ such that $S_a \setminus S_b \neq \emptyset$ for all vectors $a \in A$. But then on the vector $x \in \{0,1\}^n$ with $x_i = 1$ for $i \in S_b$ and $x_i = 0$ for $i \notin S_b$, we have $\Phi(x) = 1$ but $f_A(x) = 0$, a contradiction. The same argument also shows that for every $a \in A$ there must be a vector $b \in B$ with $S_a \supseteq S_b$. By property (i), there must also be a vector $a' \in A$ such that $S_b \supseteq S_{a'}$. Since A is an antichain, this yields $a = a'$. So, $S_b = S_a$, showing property (ii). ∎

A *minterm* of a monotone boolean function $f : \{0,1\}^n \to \{0,1\}$ is a vector $x \in \{0,1\}^n$ such that $f(x) = 1$ but $f(x') = 0$ for every vector $x' < x$ obtained by switching some 1-entry of x to 0. Note that, if $A \subset \{0,1\}^n$ is an antichain, then A is exactly the set of all minterms of the boolean function f_A defined by A. If Φ is a monotone boolean circuit computing f_A, then Lemma C.1(ii) gives the following property: for every minterm $a \in A$ the circuit must produce a vector b with the same nonzero positions as a. This, however, does not restrict the magnitude of nonzero positions of vector b. The notion of the "semantic degree" of Φ (which itself is motivated by Lemma B.4(3)) takes this magnitude into account.

Namely, let Φ be a monotone boolean circuit computing f_A, and $B \subset \mathbb{N}^n$ the set produced by Φ. The *semantic degree*, $\deg(\Phi)$, of Φ is the minimum number r for which the set $r \cdot A$ lies tightly above $\mathrm{Conv}(B)$. So, $\deg(\Phi)$ is the minimal number r for which the following holds:

- for every minterm $a \in A$ there is a convex combination $c = \lambda_1\vec{b}_1 + \cdots + \lambda_m\vec{b}_m$ of vectors $\vec{b}_i \in B$ such that $a \leqslant c \leqslant r \cdot a$ and $S_{\vec{b}_1} = \ldots = S_{\vec{b}_m} = S_a$.

That is, for every minterm $a \in A$, a convex combination of some produced vectors with the same support as a must be "sandwiched" between a and $r \cdot a$.

In particular, $\deg(\Phi) \leqslant d$ holds if for every minterm $a \in A$ the circuit produces a vector b such that $S_b = S_a$ and $b_i \leqslant d$ holds for all positions i. Note, however, that the semantic degree of a circuit can be small even if some vectors in the produced set B have very large entries. In particular, $\deg(\Phi) = 1$ holds if and only if all minterms are also produced, that is, if $A \subseteq B$ holds.

Example C.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. The boolean function f_A is then 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 $(\vee, \wedge)$ circuit Φ of size $O(n^3)$ computing the boolean function f_A. The circuit has gates $f_{l,j}$ at which the existence of a path from vertex s to vertex j with at most l edges is detected. Then $f_{1,j} = x_{s,j}$ for all $j \neq s$, and the recursion of Bellman–Ford is given by letting $f_{l+1,j}$ to be the OR of $f_{l,j}$ and all $f_{l,i} \wedge x_{i,j}$ for $i \notin \{s, j\}$. The output gate is $f_{n-1,t}$. The vectors of the set $B \subset \mathbb{N}^n$ of vectors 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 may have entries much larger than 1. Still, $\deg(\Phi) = 1$ holds: every (simple) s-t path is also a walk of length at most $n-1$, implying that $A \subseteq B$.

Remark C.2 (Magnitude of entries) . If Φ has semantic degree r, then we know that for every vector $a \in A$ there must be a convex combination $c = \sum_{i=1}^{l} \lambda_i \cdot \vec{b}_i \leqslant r \cdot a$ with all $S_{\vec{b}_i} = S_a$. By Carathéodory's theorem [6], if c is a point in the convex hull of some set $P \subseteq \mathbb{R}^m$, then c can be written as a convex combination of $m+1$ or fewer points in P. So, we can assume that $l \leqslant m+1$ where $m = |S_a|$ is the number of nonzero entries of vector 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 $\lambda_i \cdot \vec{b}_i \leqslant r \cdot a$, we have that all entries of vector $\vec{b}_i$ must be at most $r(m+1)a_i$. So, $\deg(\Phi) = r$ implies that for every vector $a \in A$ there must be a vector $b \in B$ such that $a \leqslant b \leqslant r(|S_a|+1) \cdot a$.

The following theorem shows that the *approximation* power of tropical $(\min, +)$ circuits is actually captured (not only lower bounded) by the *computational* power of monotone boolean circuits. For a set $A \subset \mathbb{N}^n$, let $\mathrm{Bool}_r(A)$ denote the minimum size of a monotone boolean circuit of semantic degree at most r computing the boolean function f_A. Hence, $\mathrm{Bool}_r(A) \geqslant \mathrm{Bool}(A)$ holds for every $r \geqslant 1$.

Theorem C.2. *If $A \subset \{0,1\}^n$ is an antichain, then $\mathrm{Min}_r(A) = \mathrm{Bool}_r(A)$ holds for every $r \geqslant 1$.*

Proof. To show $\mathrm{Min}_r(A) \leqslant \mathrm{Bool}_r(A)$, take a monotone boolean circuit Φ of semantic degree r computing the boolean function f_A defined by A. We can assume that the circuit is constant-free: we can repeatedly replace $v \wedge 0$ by 0, $v \wedge 1$ by v, $v \vee 0$ by v and $v \vee 1$ by 1. Let $B \subset \mathbb{N}^n$ be the set of vectors produced by the circuit Φ. 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 implies that the set B must lie above A. Since the circuit Φ has semantic degree r, we additionally have that the set $r \cdot A$ lies above $\mathrm{Conv}(B)$ (even tightly). By Lemma B.4, the $(\min, +)$ version of Φ (an also constant-free circuit) must approximate the minimization problem on A within factor r.

To show $\mathrm{Bool}_r(A) \leqslant \mathrm{Min}_r(A)$, take a $(\min, +)$ 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 B.5, we can assume that the circuit is constant-free. By Lemma B.4, we have that: (1) the set B must lie above A, and (2) the set $r \cdot A$ must lie tightly above $\mathrm{Conv}(B)$. The former property (1) yields property (ii) of Lemma C.1, while the latter property (2) implies property (i) of this lemma. So, the semantic degree of the boolean version of Φ is at most r and, since properties (i) and (ii) of Lemma C.1 are fulfilled, this version computes the boolean function f_A. ∎

We use the adjective "semantic" because $\deg(\Phi)$ depends on the function computed by Φ, that is, on the set of minterms A. The standard, "syntactic" definition of the degree is the following: each input variable has degree 1, take the maximum of degrees at each OR gate, and the sum of degrees at each AND gate.

The following simple proposition shows that the semantic degree never exceeds the syntactic degree.

Proposition C.3. *For any two monotone boolean circuits Φ_1 and Φ_2, we have*

$$\deg(\Phi_1 \vee \Phi_2) \leqslant \max\{\deg(\Phi_1), \deg(\Phi_2)\} \text{ and } \deg(\Phi_1 \wedge \Phi_2) \leqslant \deg(\Phi_1) + \deg(\Phi_2).$$

Proof. For $i \in \{1, 2\}$, let $A_i \subseteq \{0,1\}^n$ be the set of minterms of Φ_i, and let $B_i \subset \mathbb{N}^n$ be the set of vectors produced by Φ_i. Let r_i be the semantic degree of Φ_i.

Let us first consider the case of an OR gate. Then the circuit $\Phi = \Phi_1 \vee \Phi_2$ produces the set $B = B_1 \cup B_2$. Take an arbitrary minterm a of Φ. Then $a \in A_i$ for some $i \in \{1, 2\}$. Let r_i be the semantic degree of Φ_i. Then $c \leqslant r_i \cdot a$ must hold for some vector c in $\mathrm{Conv}(B_i) \subseteq \mathrm{Conv}(B)$. So, $\deg(\Phi_1 \vee \Phi_2) \leqslant \max\{r_1, r_2\}$, as desired.

Let us now consider the case of an AND gate. Then the circuit $\Phi = \Phi_1 \wedge \Phi_2$ produces the set $B = B_1 + B_2$. Let r_i be the semantic degree of Φ_i. Take an arbitrary minterm a of Φ. Then $a = \vec{a}_1 \vee \vec{a}_2$ is a componentwise Or of some vectors $\vec{a}_i \in A_i$. Let r_i be the semantic degree of Φ_i. Then $\vec{c}_i \leqslant r_i \cdot \vec{a}_i$ must hold for some vector $\vec{c}_i$ in $\mathrm{Conv}(B_i)$. An important 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 $c = \vec{c}_1 + \vec{c}_2 \leqslant r_1 \cdot \vec{a}_1 + r_2 \cdot \vec{a}_2 \leqslant (r_1 + r_2)[\vec{a}_1 \vee \vec{a}_2] = (r_1 + r_2) \cdot a$. This shows $\deg(\Phi_1 \wedge \Phi_2) \leqslant r_1 + r_2$, as desired. ∎

The following example illustrates that, together with Proposition C.3, the inequality $\mathrm{Min}_r(A) \leqslant \mathrm{Bool}_r(A)$ of Theorem C.2 can be used to show that some minimization problems *can* be approximated by small $(\min, +)$ circuits (even constant-free ones) with large, but finite factors. Recall that, as we have shown in section 7, some minimization problems cannot be approximated by $(\min, +)$ circuits of polynomial size within *any* finite factor at all.

Example C.3 (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 [21] that $\mathrm{Min}_1(\mathcal{T}_n) = 2^{\Omega(\sqrt{n})}$. 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 1 (see Example C.1). So, the connectivity problem can be solved by a circuit $\Phi_{1,2} \wedge \Phi_{1,3} \wedge \cdots \wedge \Phi_{1,n}$ of size $O(n^4)$. By Proposition C.3, the circuit has semantic degree $r \leqslant n - 1$. Theorem C.2 implies that $\mathrm{Min}_r(\mathcal{T}_n) = O(n^4)$ holds for some *finite* factor $r \leqslant n - 1$.

Appendix D. Sidon sets: proof of Proposition 8.3. Let m be an odd integer, and $n = 4m$. Our goal is to show that there is an explicit Sidon set $A \subseteq \{0,1\}^n$

of vectors such that $\mathrm{Max}_1(A) \geqslant 2^{n/4}$ but $\mathrm{Max}_2(A) \leqslant n$. A set $A \subset \mathbb{N}^n$ of vectors is a *Sidon set* if knowing the sum of two vectors in A, we know which vectors were added: for every a, b, c, d in A, if $a + b = c + d$ then $\{c, d\} = \{a, b\}$.

We first recall some known results. Consider the cubic parabola

$$C = \{(x, x^3) \colon x \in \{0,1\}^m\}$$

over $\mathrm{GF}(2^m)$; we view vectors in $x \in \{0,1\}^m$ as coefficient-vectors of polynomials of degree at most $m-1$ over $\mathrm{GF}(2)$ when rising them to a power. Note, however, that in the definition of Sidon sets, the sum of vectors is taken over the semigroup $(\mathbb{N}^{2m}, +)$, not over $\mathrm{GF}(2^{2m})$; in particular, $a + a = 0$ holds only for $a = 0$.

As before, $L(A)$ denotes the Minkowski circuit complexity of a set $A \subset \mathbb{N}^n$. We will use the following three facts.

(1) The cubic parabola $C \subseteq \{0,1\}^{2m}$ is a Sidon set; Lindström [25, Theorem 2].
(2) $L(A) \geqslant |A|$ holds for every Sidon set $A \subset \mathbb{N}^n$; Gashkov and Sergeev [13, Theorem 1].
(3) If $A \subset \{0,1\}^n$ is uniform (all vectors of A have the same number of 1s), then $\mathrm{Max}_1(A) \geqslant L(A)$; Jerrum and Snir [16, Theorem 2.9].

The cubic parabola C is not uniform, and we cannot apply (3) to it. But 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:

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

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 $\mathrm{Max}_1(A) \geqslant |A| = 2^m = 2^{n/4}$.

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

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

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

$$g(x) = \max_{a \in \{0,1\}^{4m}} \sum_{i=1}^{m} a_i x_i + \sum_{i=2m+1}^{3m} (1 - a_i) x_i ;$$

$$h(x) = \max_{a \in \{0,1\}^{4m}} \sum_{i=m+1}^{2m} a_i x_i + \sum_{i=3m+1}^{4m} (1 - a_i) x_i .$$

Since $g(x) = \max\{x_1, x_{2m+1}\} + \max\{x_2, x_{2m+2}\} + \cdots + \max\{x_m, x_{3m}\}$, 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 $r = 2$. For this to hold, we have (by Lemma B.3) to show that the set B lies below A, and that the set $\frac{1}{2} \cdot A$ lies below $\mathrm{Conv}(B)$. 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 $\mathrm{GF}(q)$ if and only if $q - 1$ and k are relatively prime; see, for example, Lidl and Niederreiter [24, 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:

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

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 B.3, it remains to show that the set $\frac{1}{2} \cdot A$ lies below the convex hull $\mathrm{Conv}(B)$. 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 interesting discussions.

REFERENCES

[1] N. Alon and R. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

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

[3] N. Bansal, R. Pendavingh, and J.G. van der Pol. On the number of matroids. *Combinatorica*, 35(3):253–277, 2015.

[4] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[5] R. Bellman. On a routing problem. *Quarterly of Appl. Math.*, 16:87–90, 1958.

[6] C. Carathéodory. Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. *Mathematische Annalen*, 64(1):95–115, 1907.

[7] G.B. Dantzig. Discrete variable extremum problems. *Operations research*, 5:266–277, 1957.

[8] S.E. Dreyfus and R.A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971.

[9] J. Edmonds. Matroids and the greedy algorithm. *Math. Programming*, 1:127–136, 1971.

[10] J. Farkas. Über die Theorie der einfachen Ungleichungen. *J. Reine Angew. Math.*, 124:1–24, 1902.

[11] R.W. Floyd. Algorithm 97, shortest path. *Comm. ACM*, 5:345, 1962.

[12] L.R. Ford. Network flow theory. Technical Report P-923, The Rand Corp., 1956.

[13] S.B. Gashkov and I.S. Sergeev. A method for deriving lower bounds for the complexity of monotone arithmetic circuits computing real polynomials. *Sbornik: Mathematics*, 203(10):1411–1147, 2012.

[14] M. Held and R.M. Karp. A dynamic programming approach to sequencing problems. *SIAM J. on Appl. Math.*, 10:196–210, 1962.

[15] O.H. Ibarra and J.C. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22:463–468, 1975.

[16] M. Jerrum and M. Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982.

[17] S. Jukna. Combinatorics of monotone computations. *Combinatorica*, 19(1):65–85, 1999.

[18] S. Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer-Verlag, 2012.

[19] S. Jukna. Lower bounds for tropical circuits and dynamic programs. *Theory of Comput. Syst.*, 57(1):160–194, 2015.

[20] S. Jukna. Tropical complexity, Sidon sets and dynamic programming. *SIAM J. Discrete Math.*, 30(4):2064–2085, 2016.

[21] S. Jukna and H. Seiwert. Greedy can also beat pure dynamic programming. Technical Report Nr. 49, Electronic Colloq. Comput. Complexity (ECCC), 2018.

[22] D.E. Knuth. The asymptotic number of geometries. *J. Combin. Theory, Ser. A*, 16(3):398–400, 1974.

[23] A.Y. Levin. Algorithm for the shortest connection of a group of graph vertices. *Sov. Math. Dokl.*, 12:1477–1481, 1971.

[24] R. Lidl and H. Niederreiter. *Introduction to Finite Fields and their Applications.* Cambridge University Press, 1986.

[25] B. Lindström. Determination of two vectors from the sum. *J. Combin. Theory, Series B*, 6(4):402–407, 1969.

[26] E.F. Moore. The shortest path through a maze. In *Proc. Internat. Sympos. Switching Theory*, volume II, pages 285–292, 1957.

[27] N. Nisan and A. Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[28] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization.* Prentice-Hall, 1982.

[29] M.J. Piff and D.J.A. Welsh. The number of combinatorial geometries. *Bull. Lond. Math. Soc.*, 3:55–56, 1971.

[30] R. Rado. A theorem on independence relations. *Quart. J. Math.*, 13(1):83–89, 1942.

[31] A.A. Razborov. Lower bounds on monotone complexity of the logical permanent. *Math. Notes of the Acad. of Sci. of the USSR*, 37(6):485–493, 1985.

[32] A. Schrijver. *Theory of Integer and Linear Programming.* Wiley, 1986.

[33] S. Warshall. A theorem on boolean matrices. *J. ACM*, 9:11–12, 1962.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il