Electronic Colloquium on Computational Complexity, Report No. 178 (2020)

Reciprocal Inputs in Arithmetic and Tropical Circuits

Stasys Jukna* Hannes Seiwert† Igor Sergeev‡

Abstract

It is known that the size of monotone arithmetic $(+, \cdot)$ circuits can be exponentially decreased by allowing just one division "at the very end," at the output gate. A natural question is: can the size of $(+, \cdot)$ circuits be substantially reduced if we allow divisions "at the very beginning," that is, if besides nonnegative real constants and variables $x_1, \ldots, x_n$, the circuits can also use their reciprocals $1/x_1, \ldots, 1/x_n$ as inputs. We answer this question in the negative: the gain in circuit size is then always at most quadratic.

Over tropical $(\min, +)$ and $(\max, +)$ semirings, division turns into subtraction; so, reciprocal inputs are then $-x_1, \ldots, -x_n$. We give the same negative answer also for tropical circuits. The question of whether reciprocal inputs can substantially speed up tropical $(\min, +, \max)$ circuits, using both min and max gates, remains open.

Keywords: Arithmetic circuit, tropical circuit, reciprocal input, dynamic programming

1 Introduction

A fundamental question in circuit complexity is: if we allow circuits to use some additional resources, can then the same function be computed using fewer gates? In this paper, we consider the situation when besides input variables $x_1, \ldots, x_n$ the circuits are allowed to use their "reciprocals." Can reciprocal inputs substantially decrease the circuit size?

In the case of Boolean circuits, reciprocal inputs are negations $\overline{x}_1, \ldots, \overline{x}_n$. In every Boolean $(\vee, \wedge, \neg)$ circuit one can move all negations towards the inputs by only doubling the size. As shown by Razborov [15], reciprocal inputs *can* super-polynomially reduce the size of monotone Boolean $(\vee, \wedge)$ circuits; this gap was later increased to exponential by Tardos [17].

In this paper, we consider the role of reciprocal inputs in arithmetic $(+, \cdot)$ as well as in tropical $(\min, +)$ and $(\max, +)$ circuits. In these circuits, inputs are variables $x_1, \ldots, x_n$ and arbitrary nonnegative constants. In the case of arithmetic circuits, there are two types of reciprocal inputs: additive reciprocals $-x_1, \ldots, -x_n$ and multiplicative reciprocals $1/x_1, \ldots, 1/x_n$. We denote the corresponding circuits as $(+, \cdot, -x_i)$ and $(+, \cdot, 1/x_i)$. In $(+, \cdot, -)$ circuits, the subtraction operation $(-)$ can be used. Subtraction-free $(+, \cdot)$ circuits are usually referred to as *monotone* arithmetic circuits.

The $(+, \cdot, -)$ and $(+, \cdot, -x_i)$ circuit complexities are almost the same: subtraction gates can always be moved to the inputs via $-(x + y) = (-x) + (-y)$ and $-x \cdot y = (-x) \cdot y$.

*Faculty of Mathematics and Computer Science, Vilnius University, Lithuania. Research supported by the DFG grant JU 3105/1-2 (German Research Foundation). Email: stjukna@gmail.com

†Faculty of Computer Science and Mathematics, Goethe University Frankfurt, Germany. Email: seiwert@thi.cs.uni-frankfurt.de

‡Research Institute Kvant, Moscow, Russia. Research supported by RFBR grant, project no. 19-01-00294a. Email: isserg@gmail.com

ISSN 1433-8092

So, Valiant's result [18] implies that the $(+,\cdot)/(+,\cdot,-x_i)$ gap *can* be exponential. That is, additive reciprocals $-x_i$ *can* even exponentially decrease the size of monotone arithmetic circuits.

Strassen [16] has shown that if the subtraction $(-)$ operation is allowed, then division $(/)$ gates *cannot* significantly decrease the size of arithmetic circuits: if a polynomial P of degree d can be computed by a $(+,\cdot,/,-)$ circuit of size s (divisions are not restricted to inputs), then P can be also computed by a $(+,\cdot,-)$ circuit of size $\mathcal{O}(sd\log d)$. So, multiplicative reciprocal inputs $1/x_1,\ldots,1/x_n$ in *non-monotone* arithmetic circuits are of little use.

But what about *monotone* $(+,\cdot)$ circuits (without subtraction): can then the division operation $(/)$ help? The question is not trivial because the subtraction operation is crucially used in Strassen's argument to express $1/f$ as power series $1/(1-(1-f))=\sum_{i\geq 0}(1-f)^i$. And indeed, it turned out that Strassen's result does *not* hold for monotone arithmetic circuits: here division *can* exponentially decrease the circuit size.

The exponential $(+,\cdot)/(+,\cdot,/)$ gap for subtraction-free circuits is exhibited by the *spanning tree polynomial* $\kappa_n(x)=\sum_T\prod_{e\in T}x_e$, where the sum is over all spanning trees of a complete undirected graph K_n on n vertices. Jukna and Seiwert [11] have shown that every monotone arithmetic circuit computing κ_n must use $2^{\Omega(\sqrt{n})}$ gates. For the directed version of κ_n (when K_n is a complete directed graph and the sum is taken over all arborescences), the lower bound $2^{\Omega(n)}$ was earlier proved by Jerrum and Snir [9]. On the other hand, Fomin, Grigoriev and Koshevoy [6] have recently shown that both κ_n and its directed version can be computed by a $(+,\cdot,/)$ circuit of size $\mathcal{O}(n^3)$.

The upper bound $\mathcal{O}(n^3)$ also holds when $(+,\cdot,/)$ circuits are allowed to use only one division gate at the end: one can easily move all division gates to the output gate via $(x/y)+z=(x+yz)/y$, $(x/y)z=(xz)/y$ and $(x/y)/z=x/(yz)$. Thus, the $(+,\cdot)/(+,\cdot,/)$ gap *can* be exponential even when only a single division operation "at the very end" (at the output gate) is allowed.

In this paper we ask the "opposite" question: what happens if we only allow divisions "at the very beginning"? Can $(+,\cdot,1/x_i)$ circuits, that is, monotone arithmetic circuits with reciprocal inputs $1/x_1,\ldots,1/x_n$ be super-polynomially smaller than monotone circuits without these additional inputs? That multiplicative reciprocal inputs *can* save gates can be seen on the polynomial x^{31}. It can be computed as $x^{32}\cdot(1/x)$ using 6 gates, but 7 gates are necessary to compute x^{31} by a circuit without reciprocals: this follows from known[1] lower bounds on the minimum length of so-called *addition chains*. So, our question is: can $(+,\cdot,1/x_i)$ circuits be *considerably* smaller than $(+,\cdot)$ circuits?

We answer this question in the negative: the $(+,\cdot)/(+,\cdot,1/x_i)$ gap *cannot* be larger than quadratic (Theorem 1): if an n-variate polynomial P can be computed by a $(+,\cdot,1/x_i)$ circuit of size s, then P can also be computed by a $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2)$.

We also investigate the role of reciprocal inputs in tropical $(\min,+)$ and $(\max,+)$ circuits. Our motivation is that these circuits can simulate basic dynamic programming algorithms. Inputs in tropical circuits are variables $x_1,\ldots,x_n$ and nonnegative real constants; gates perform additions $(+)$ and min or max operations. The tropical $(\min,+)$ and $(\max,+)$ version of the arithmetic division x/y is the (arithmetic) subtraction $x-y$. So, reciprocal inputs $1/x_1,\ldots,1/x_n$ turn into $-x_1,\ldots,-x_n$ in tropical circuits. That is, tropical versions of arithmetic $(+,\cdot,1/x_i)$ circuits are $(\min,+,-x_i)$ and $(\max,+,-x_i)$ circuits.

Over the tropical $(\min,+)$ semiring, the spanning tree polynomial κ_n turns into the min-

[1]See, for example, the On-line Encyclopedia of Integer Sequences at https://oeis.org/A003313.

imum weight spanning tree problem MST_n: given nonnegative weights x_e to the edges e of K_n, compute the minimum weight $\sum_{e\in T} x_e$ of a spanning tree T of K_n. As shown in [11], this problem requires $(\min,+)$ circuits of size $2^{\Omega(\sqrt{n})}$. On the other hand, since tropical circuits are not weaker than monotone arithmetic circuits, the upper bound $\mathcal{O}(n^3)$ of [6] for κ_n carries over to tropical circuits solving MST_n; the same result also holds for $(\max,+)$ circuits computing the $(\max,+)$ version of MST_n. Thus, both gaps $(\min,+)/(\min,+,-)$ and $(\max,+)/(\max,+,-)$ *can* be exponential.

So, again, the question is: what if we restrict tropical circuits to use the subtraction $(-)$ operation only on inputs? We show (Theorem 2) that the $(\min,+)/(\min,+,-x_i)$ gap is never larger than quadratic. The case of tropical $(\max,+,-x_i)$ circuits turns out to be more difficult, but we still are able to show (Theorem 3) that the $(\max,+)/(\max,+,-x_i)$ gap cannot be larger than quadratic for circuits solving *homogeneous* maximization problems. The extension of Theorem 1 to tropical circuits is not trivial because tropical circuits can be much stronger than monotone arithmetic circuits. For example, the shortest path problem can be solved by a polynomial-size $(\min,+)$ circuit resulting from the Bellman–Ford–Moore dynamic programming algorithm, but the corresponding (arithmetic) polynomial requires monotone arithmetic circuits of exponential size [9, Sect. 4.4].

2 Results

In this section, we describe our results and related facts more precisely. A summary of known and new results is depicted in Fig. 1.

Arithmetic circuits Our first result (Theorem 1) shows that the $(+,\cdot)/(+,\cdot,1/x_i)$ gap can never be larger than quadratic. That is, reciprocal inputs cannot significantly decrease the size of monotone arithmetic circuits.

Theorem 1. *If an n-variate polynomial P can be computed by a $(+,\cdot,1/x_i)$ circuit of size s, then P can also be computed by a $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2)$.*

Theorem 1 has the following consequence which may be of independent interest. It concerns the monotone arithmetic $(+,\cdot)$ circuit complexity of "complementary" polynomials. Every subset $S\subseteq[n]=\{1,\dots,n\}$ has its associated multilinear monomial $X_S=\prod_{i\in S} x_i$. The *complement* of a multilinear polynomial $P(x)=\sum_{S\in\mathcal{F}} X_S$ is the multilinear polynomial

$$\text{co-}P(x) := \sum_{S\in\mathcal{F}} \prod_{i\notin S} x_i\,.$$

Can the $(+,\cdot)$ circuit complexity of co-P be much smaller than that of P? The following consequence of Theorem 1 answers this question in the *negative*.

Corollary 1. *If a multilinear n-variate polynomial P can be computed by a $(+,\cdot)$ circuit of size s, then its complement* co-P *can be computed by a $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2+n^3)$.*

Proof. Take a $(+,\cdot)$ circuit computing $P(x_1,\dots,x_n)=\sum_{S\in\mathcal{F}} X_S$, and let s be the size of this circuit. If we replace each input variable x_i by its reciprocal $1/x_i$, the obtained $(+,\cdot,1/x_i)$ circuit has the same size s, and computes the Laurent polynomial[2] $P(1/x_1,\dots,1/x_n)$. We

[2] Laurent polynomials may have negative exponents.

have

$$\text{co-}P(x) = \sum_{S\in\mathcal{F}} \prod_{i\notin S} x_i = \prod_{i=1}^{n} x_i \cdot \sum_{S\in\mathcal{F}} \prod_{i\in S} \frac{1}{x_i} = \prod_{i=1}^{n} x_i \cdot P(1/x_1,\ldots,1/x_n)\,.$$

Hence, the polynomial co-P can be computed by a $(+,\cdot,1/x_i)$ circuit of size $t = s + n$. By Theorem 1, the polynomial co-P can be computed by a $(+,\cdot)$ circuit of size $\mathcal{O}(nt^2) = \mathcal{O}(ns^2 + n^3)$, as claimed. ∎

Tropical circuits Many dynamic programming algorithms (DP algorithms) are "pure" in that they only use the basic operations $(\min,+)$ or $(\max,+)$ in their recursion equations. Notable examples of pure DP algorithms for combinatorial optimization problems are the well-known Bellman–Ford–Moore shortest s-t path algorithm [2, 7, 13], the Floyd–Warshall all-pairs shortest paths algorithm [5, 19], and the Held–Karp travelling salesman algorithm [8], and the Dreyfus–Levin–Wagner Steiner tree algorithm [3, 12].

Tropical $(\min,+)$ or $(\max,+)$ circuits constitute a natural model for pure DP algorithms. Such a circuit uses fanin-2 $(\min,+)$ or $(\max,+)$ operations. Inputs are variables $x_1,\ldots,x_n$ and arbitrary nonnegative constants. That is, tropical circuits use addition $(+)$ gates instead of arithmetic multiplication $(\cdot)$, and use min or max gates (but not both) instead of arithmetic addition gates $(+)$. Tropical $(\min,+,-x_i)$ or $(\max,+,-x_i)$ circuits can additionally use $-x_1,\ldots,-x_n$ as inputs.

Tropical $(\min,+)$ circuits compute tropical polynomials $P(x) = \min_{a\in A}\langle a,x\rangle + c_a$, where $A \subseteq \mathbb{N}^n$ is a finite set of vectors (these are exponent vectors of monomials in the arithmetic case), $c_a \in \mathbb{R}_+$ are nonnegative *constant terms*, and $\langle a,x\rangle = a_1x_1 + \cdots + a_nx_n$ is the scalar product of vectors a and x. In tropical $(\max,+)$ circuits, we have max instead of min. Tropical $(\min,+,-x_i)$ and $(\max,+,-x_i)$ circuits compute tropical *Laurent* polynomials: in this case, we have $A \subseteq \mathbb{Z}^n$, that is, some variables x_i may have negative "exponents" a_i.

We call two tropical n-variate Laurent polynomials P and Q *equivalent* if $P(x) = Q(x)$ holds for all $x \in \mathbb{R}^n_+$. A tropical circuit Φ *computes* a given tropical polynomial P if the Laurent polynomial produced by Φ is equivalent to P. That is, we only require that the circuit solves the corresponding optimization problem on all *nonnegative* weights.

Remark 1 (Why only nonnegative weights?). The reason to restrict ourselves to nonnegative input weights $x_1,\ldots,x_n \in \mathbb{R}_+$ is twofold. First, efficient dynamic programming algorithms usually work well on nonnegative weightings, but fail when negative weights are allowed. Second, if tropical circuits are required to solve a given optimization problem on arbitrary weightings, then their power is almost the same as that of monotone arithmetic circuits [9, Theorem 2.6] (i.e., we have no new model then). On the other hand, if only nonnegative weights are allowed, then tropical circuits can be even exponentially more powerful than monotone arithmetic circuits. Say, the shortest path problem can be solved by a polynomial-size $(\min,+)$ circuit resulting from the Bellman–Ford–Moore dynamic programming algorithm, but the corresponding (arithmetic) polynomial requires monotone arithmetic circuits of exponential size [9, Sect. 4.4].

The next theorem gives a tropical analogue of Theorem 1 in the case of minimization.

Theorem 2 (Minimization)**.** *If a tropical* $(\min,+)$ *polynomial* $P(x_1,\ldots,x_n)$ *can be computed by a* $(\min,+,-x_i)$ *circuit of size* s*, then* P *can also be computed by a* $(\min,+)$ *circuit of size* $\mathcal{O}(ns^2)$*.*

The situation with tropical $(\max, +, -x_i)$ circuits (solving *maximization* problems) turns out to be more delicate. In this case, we can prove the analogue of Theorem 2 only for polynomials $P(x) = \max_{a\in A}\langle a, x\rangle + c_a$ that are *homogeneous*: there is an $m \in \mathbb{N}$ such that $a_1 + \cdots + a_n = m$ holds for all $a \in A$.

Theorem 3 (Maximization)**.** *If a homogeneous* $(\max, +)$ *polynomial* $P(x_1, \ldots, x_n)$ *can be computed by a* $(\max, +, -x_i)$ *circuit of size* s*, then* P *can also be computed by a* $(\max, +)$ *circuit of size* $\mathcal{O}(ns^2)$*.*

By analogy with the arithmetic case, one can define the *complement* of a tropical polynomial $P(x) = \min_{S\in\mathcal{F}} \sum_{i\in S} x_i$ as co-$P(x) = \min_{S\in\mathcal{F}} \sum_{i\notin S} x_i$. But then no tropical analogue of Corollary 1 holds: the gap between the circuit complexities of tropical polynomials P and co-P *can* be exponentially large. For example, let $\mathcal{F}$ be the family of subsets of edges of a complete bipartite $n\times n$ graph $K_{n,n}$ consisting of all $n!$ complements of perfect matchings and all n^2 singletons. Then the $(\min, +)$ polynomial $P(x) = \min_{S\in\mathcal{F}} \sum_{e\in S} x_e$ can be computed by a trivial $(\min, +)$ circuit of size n^2 (which just outputs the minimum weight of an edge). On the other hand, the complement polynomial co-$P(x) = \min_{S\in\mathcal{F}} \sum_{e\notin S} x_e$ is equivalent to the tropical permanent polynomial $\mathrm{Per}_n(x) = \min_M \sum_{e\in M} x_e$, where the minimum is over all perfect matchings M in $K_{n,n}$ (the minimum in co-P is always achieved on a perfect matching $\overline{S}$), and it is known that Per_n requires $(\min, +)$ circuits of size $2^{\Omega(n)}$ [9].

The situation, however, changes if instead of complementary tropical polynomials, we consider their duals, where in the dual polynomial, min turns into max, and vice versa.

Corollary 2. *If a polynomial* $P(x) = \max_{a\in A} \langle a, x\rangle$ *with* $A \subseteq \{0,1\}^n$ *can be computed by a* $(\max, +)$ *circuit of size* s*, then the dual polynomial* $P^*(x) = \min_{a\in A} \langle \vec{1} - a, x\rangle$ *can be computed by a* $(\min, +, -x_i)$ *circuit of size* $n + s$ *and by a* $(\min, +)$ *circuit of size* $\mathcal{O}(ns^2 + n^3)$*.*

Proof. Let Φ be a $(\max, +)$ circuit of size s computing P. Turn the circuit Φ into a $(\min, +, -x_i)$ circuit: replace every max gate by a min gate and every input variable x_i by its reciprocal $-x_i$. The resulting $(\min, +, -x_i)$ circuit Φ' has the same size and computes the Laurent polynomial $Q(x) = \min_{a\in A} \langle a, -x\rangle = \min_{a\in A} \langle -a, x\rangle$. Then the $(\min, +, -x_i)$ circuit $x_1 + \cdots + x_n + \Phi'(x)$ of size $n+s$ computes $\langle \vec{1}, x\rangle + \min_{a\in A}\langle -a, x\rangle = \min_{a\in A}\langle \vec{1} - a, x\rangle = P^*(x)$. By Theorem 2, the polynomial P^* can be computed by a $(\min, +)$ circuit of size $\mathcal{O}(ns^2 + n^3)$, as claimed. ∎

For the duals of $(\min, +)$ polynomials, the same argument using Theorem 3 instead of Theorem 2 gives a weaker fact.

Corollary 3. *If a polynomial* $P(x) = \min_{a\in A} \langle a, x\rangle$ *with* $A \subseteq \{0,1\}^n$ *can be computed by a* $(\min, +)$ *circuit of size* s*, then the dual polynomial* $P^*(x) = \max_{a\in A} \langle \vec{1} - a, x\rangle$ *can be computed by a* $(\max, +, -x_i)$ *circuit of size* $n + s$*. If* P *is homogeneous, then* P^* *can be computed by a* $(\max, +)$ *circuit of size* $\mathcal{O}(ns^2 + n^3)$*.*

What about circuits that may use *both* min and max operations? It can be easily shown (Proposition 1 below) that the power of $(\min, +, \max, -x_i)$ circuits is almost the same as that of general tropical $(\min, +, -)$ circuits (with subtraction gates). Since the $(\min, +)/(\min, +, -)$ gap can be exponential [6], we know that the size of $(\min, +)$ circuits can be exponentially decreased by allowing max gates *and* reciprocal inputs $-x_i$. It turned out that already the $(\min, +)/(\min, +, \max)$ gap can be exponential: the minimum weight

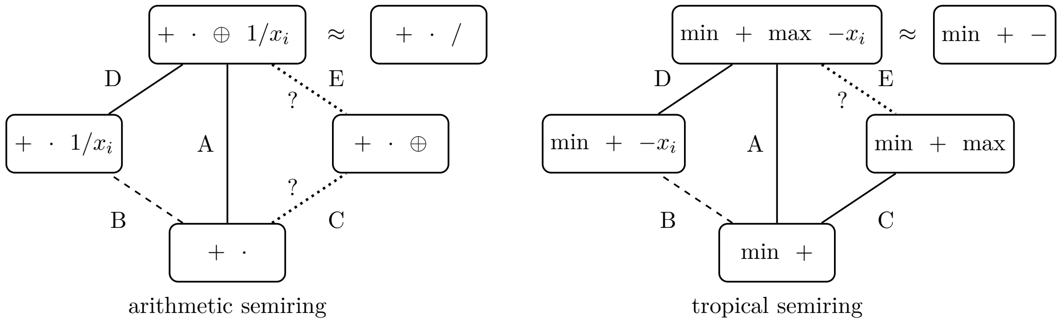

Figure 1: Basic subclasses of arithmetic $(+,\cdot,/)$ and tropical $(\min,+,-)$ circuits. Here, $x \oplus y = (x^{-1}+y^{-1})^{-1}$ is the harmonic sum operation; its tropical $(\min,+)$ version is $\max(x,y)$. The $\approx$ relation means that the sizes of the corresponding circuits coincide up to (small) multiplicative constants (Proposition 1). That the gap (A), in both the arithmetic and the tropical case, can be exponential follows from the lower bounds proved in [9, 11] and the upper bounds for the same polynomials proved in [6]. That already the tropical gap (C) can be exponential was proved in [10]. The main result of this paper is that the gap (B) cannot be larger than quadratic for arithmetic and tropical $(\min,+)$ circuits computing any polynomials (Theorems 1 and 2), and for $(\max,+)$ circuits computing homogeneous polynomials (Theorem 3). Together with the gap (A), this implies that the gap (D) can be exponential in both arithmetic and tropical cases. The status of the arithmetic and tropical gap (E), as well as of the arithmetic gap (C) remains open.

spanning tree problem MST_n can be solved by a $(\min,+,\max)$ circuit of size $\mathcal{O}(n^3)$ [11], but any $(\min,+)$ circuit solving MST_n must have $2^{\Omega(\sqrt{n})}$ gates [10]. Together with Theorem 2, this implies that even the $(\min,+,-x_i)/(\min,+,\max)$ gap can be exponential.

Corollary 4. *The* MST_n *problem can be solved by a* $(\min,+,\max)$ *circuit of size* $\mathcal{O}(n^3)$*, but any* $(\min,+,-x_i)$ *circuit for this problem must have* $2^{\Omega(\sqrt{n})}$ *gates.*

We summarize our results concerning the subclasses of arithmetic $(+,\cdot,/)$ and tropical $(\min,+,-)$ circuits in Fig. 1. In the tropical $(\min,+)$ semiring, addition $x+y$ turns into $\min(x,y)$, multiplication $x\cdot y$ into $x+y$, and division x/y into subtraction $x-y$. The max operation is related with the min operation via the equality $\min(x,y)+\max(x,y)=x+y$. So, in the arithmetic $(+,\cdot)$ semiring, the analogue of the max operation is the *harmonic sum* (the half of the harmonic mean) operation:

$$x \oplus y := \frac{1}{\frac{1}{x}+\frac{1}{y}} = \frac{x\cdot y}{x+y}\,.$$

Note that $(x+y)\cdot(x\oplus y) = x\cdot y$ and $(x+y)^{-1} = x^{-1}\oplus y^{-1}$. These are arithmetic versions of the tropical equations $\min(x,y)+\max(x,y)=x+y$ and $-\min(x,y)=\max(-x,-y)$. In particular, the arithmetic circuits corresponding to tropical $(\min,+,\max)$ circuits are $(+,\cdot,\oplus)$ circuits.

The $\approx$ relation in Fig. 1 is the content of the following simple fact.

Proposition 1. *The* $(+,\cdot,/)$ *and* $(+,\cdot,\oplus,1/x_i)$ *circuit complexities are proportional, that is, differ by at most absolute constant factors. The same holds for the* $(\min,+,-)$ *and* $(\min,+,\max,-x_i)$ *complexities.*

Here, we allow $(\min,+,\max,-x_i)$ circuits to use also negative constants as inputs.

Proof. We treat both arithmetic and tropical circuits in one argument: in the former, we have $x_i^{-1} = 1/x_i$, while in the latter, we have $x_i^{-1} = -x_i$ and $x \oplus y = \max(x, y)$. One direction is trivial: $(+, \cdot, \oplus, {x_i}^{-1})$ circuits can be simulated by at most three times larger $(+, \cdot, /)$ circuits. To show the converse direction, take an arbitrary $(+, \cdot, /)$ circuit Φ of size s. We replace each gate g by two gates g and g^{-1}. In particular, if $g = x_i$ is an input variable, then we have two input gates $g = x_i$ and $g^{-1} = x_i^{-1}$ in the new circuit Φ'. If $g = c$ is an input constant, then we have two input gates $g = c$ and $g^{-1} = c^{-1}$ in the new circuit. If a and b are the gates entering a non-input gate g in Φ, then wire these gates in Φ' according to the following rules.

- if $g = a \cdot b$ in Φ, then $g = a \cdot b$ and $g^{-1} = a^{-1} \cdot b^{-1}$ in Φ';
- if $g = a + b$ in Φ, then $g = a + b$ and $g^{-1} = a^{-1} \oplus b^{-1}$ in Φ';
- if $g = a/b$ in Φ, then $g = a \cdot b^{-1}$ and $g^{-1} = a^{-1} \cdot b$ in Φ'.

Thus, the new circuit Φ' uses only gates from $\{+, \cdot, \oplus\}$, has size at most $2s$, and computes the same function. ∎

So, since the $(+, \cdot)/(+, \cdot, /)$ gap can be exponential, reciprocal inputs $1/x_1, \ldots, 1/x_n$, *together* with the harmonic sum operation $x \oplus y$, *can* exponentially decrease the size monotone arithmetic circuits. Theorem 1 shows that the presence of $\oplus$-gates is crucial to achieve such a speed-up. So, a natural question is whether reciprocal inputs can considerably decrease the size of $(+, \cdot, \oplus)$ circuits, that is, whether the $(+, \cdot, \oplus)/(+, \cdot, \oplus, 1/x_i)$ gap can be large.

The situation with tropical $(\min, +)$ circuits, where the harmonic sum operation $x \oplus y$ turns into $\max(x, y)$, is similar. By Corollary 4, we know that the $(\min, +, -x_i)/(\min, +, \max)$ gap can be exponential. This means that the extension of $(\min, +)$ circuits by including the max operation leads to a stronger model than the inclusion of reciprocal inputs $-x_i$. A natural question is whether reciprocal inputs can be useful in $(\min, +, \max)$ circuits at all. In Section 8, we discuss these open questions in more detail.

3 Preliminaries

Arithmetic $(+, \cdot)$ and tropical $(\min, +)$ or $(\max, +)$ circuits are circuits over the corresponding semirings. To treat these circuits "under one umbrella," let[3] $(\mathbb{R}_+, +, \cdot)$ stand for any of these three semirings. That is, in tropical semirings, "addition" $x + y$ turns into $\min(x, y)$ or $\max(x, y)$, while "multiplication" $x \cdot y$ in both these semirings turns into addition $x + y$. The multiplicative identity element is 1 in the arithmetic semiring, and is 0 in the tropical semirings (since $0 + x = x + 0 = x$). Hence, the multiplicative inverse (or reciprocal) of $x \in \mathbb{R}_+$ is $x^{-1} := 1/x$ in the arithmetic semiring, and is $x^{-1} := -x$ in both tropical semirings. The *size* of a circuit is the number of its non-input gates.

Polynomials As customary, an n-variate *polynomial* over a semiring $(\mathbb{R}_+, +, \cdot)$ is an expression of the form $P(x) = \sum_{a \in A} c_a X^a$, where $A \subseteq \mathbb{N}^n$ is some finite set of *exponent vectors*, $X^a = \prod_{i=1}^n x_i^{a_i}$ are *monomials*, $c_a X^a$ are *terms*, and $c_a > 0$ are *coefficients* of the polynomial P. The *degree* of a monomial X^a is the sum $a_1 + \cdots + a_n$ of its exponents, and the *degree* of a polynomial is the maximum degree of its monomials. *Laurent polynomials* may have negative exponents: in this case, we have $A \subseteq \mathbb{Z}^n$. To stress that a given Laurent polynomial P is

[3] As customary, $\mathbb{R}_+$ stands for the set of all nonnegative real numbers, $\mathbb{Z}$ for the set of all integers, and $\mathbb{N}$ for the set of all nonnegative integers.

actually a polynomial (has no negative exponents), we will sometimes call P a *non-Laurent* polynomial; thus, the terms "polynomial" and "non-Laurent polynomial" are equivalent.

In tropical semirings, monomials X^a turn into linear forms $\langle a, x\rangle = a_1x_1 + \cdots + a_nx_n$, and terms turn into affine forms $\langle a, x\rangle + c_a = a_1x_1 + \cdots + a_nx_n + c_a$. Hence, "exponents" a_i are the coefficients of these forms. So, in the tropical $(\min, +)$ semiring, Laurent polynomials are expressions of the form $P(x) = \min_{a\in A} \langle a, x\rangle + c_a$; in the $(\max, +)$ semiring, we have max instead of min.

Polynomials produced by circuits By a $(+, \cdot, x_i{}^{-1})$ *circuit* we will mean a conventional circuit Φ using the semiring operations $+$ and $\cdot$ as gates. Inputs are variables $x_1, \ldots, x_n$, their reciprocals $x_1{}^{-1}, \ldots, x_n{}^{-1}$ and arbitrary *nonnegative* constants $c \in \mathbb{R}_+$.

Every $(+, \cdot, x_i{}^{-1})$ circuit *produces* (purely syntactically) a unique Laurent polynomial in a natural way: at an addition $(+)$ gate, add the two produced polynomials, and at a multiplication $(\cdot)$ gate, multiply every term of one produced polynomial with every term of the other produced polynomial, and take the sum of the resulting terms. That is, at each input gate holding x_i, x_i^{-1} or a constant c, the corresponding Laurent polynomials x_i, x_i^{-1} or c are produced. If $P(x) = \sum_{a\in A} c_aX^a$ and $Q(x) = \sum_{b\in B} c_bX^b$ are Laurent polynomials produced at the predecessors of some gate, then the polynomial produced at that gate is

$$P + Q = \sum_{a\in A} c_aX^a + \sum_{b\in B} c_bX^b$$

if it is an "addition" gate, and is

$$P \cdot Q = \sum_{a\in A}\sum_{b\in B} c_ac_bX^{a+b}$$

if it is a "multiplication" gate. In particular, if we deal with tropical $(\min, +, -x_i)$ circuits, then the (Laurent) polynomials produced at predecessors of a gate are of the form $P(x) = \min_{a\in A}\langle a, x\rangle + c_a$ and $Q(x) = \min_{b\in B}\langle b, x\rangle + c_b$ with $A, B \subseteq \mathbb{Z}^n$ and all $c_a, c_b \in \mathbb{R}_+$. The polynomial produced at this gate is either $\min\{P, Q\}$ or $P + Q = \min_{a\in A} \min_{b\in B} \langle a + b, x\rangle + c_a + c_b$.

Computing versus producing We say that two n-variate Laurent polynomials P and Q are *equivalent* if $P(x) = Q(x)$ holds for all $x \in \mathbb{R}_+^n$. An arithmetic or tropical circuit Φ (with or without reciprocal inputs) *computes* a given Laurent polynomial P if the Laurent polynomial produced by Φ is equivalent to P.

It is well known and easy to show (using, for example, the multivariate version of the "fundamental theorem of algebra") that if two arithmetic Laurent polynomials are equivalent, then they coincide as formal expressions, that is, have the same Laurent monomials with the same nonzero coefficients (see Section 5). Thus, if an arithmetic $(+, \cdot, 1/x_i)$ circuit computes a polynomial, then it also produces this polynomial.

This does not hold for tropical circuits: a lot of distinct tropical Laurent polynomials may be equivalent to one fixed tropical polynomial. For example, for any $(\min, +)$ polynomial P, all polynomials $\min\{x, x + P\}$ are equivalent to the polynomial x. The following tight structural characterization of equivalent tropical Laurent polynomials was proved by Jerrum and Snir [9, Corollary A3] using a version of Farkas' lemma due to Fan [4, Theorem 4] (where a vector u *lies above* a vector v if $u \geq v$, and u *lies below* v if $u \leq v$ holds): two

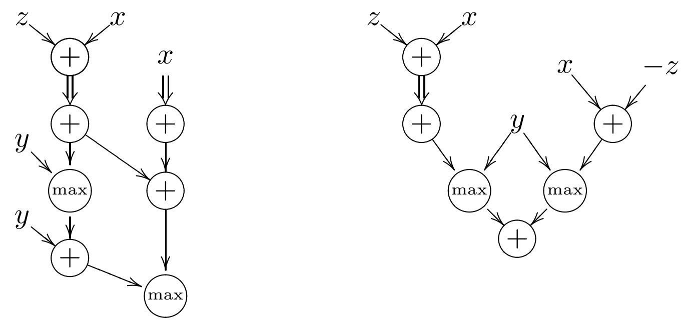

Figure 2: On the left: a $(\max,+)$ circuit (without reciprocal inputs) computing the tropical polynomial $P(x,y,z) = \max\{2x+y+2z, 2y, 3x+z\}$ using 7 gates; here $\Downarrow$ stands for two parallel edges. However, the $(\max,+,-x_i)$ circuit on the right computes the same polynomial using only 6 gates. It can be shown that this circuit is optimal: at least 6 gates are necessary to compute P. Moreover, every $(\max,+,-x_i)$ circuit of size 6 for this polynomial P must produce a term with negative "exponents" (we omit a tedious proof via case considerations). The latter circuit *produces* the Laurent polynomial $Q = \max\{2x+y+2z, 2y, 3x+z, x+y-z\}$ which is different from the computed polynomial P: it has a "redundant" term $x+y-z$. Note that the variable z has a negative "exponent" -1 in this term.

tropical Laurent polynomials $P(x) = \min_{a\in A}\langle a,x\rangle + c_a$ and $Q(x) = \min_{b\in B}\langle b,x\rangle + c_b$ are equivalent if and only if every vector (a,c_a) lies above some convex combination of vectors (b,c_b), and every vector (b,c_b) lies above some convex combination of vectors (a,c_a). Due to the equality $\max(x,y) = -\min(-x,-y)$, the same holds for $(\max,+)$ polynomials with "lies above" replaced by "lies below."

For example, the Laurent polynomials $P = \max\{2x, 2y\}$ and $Q = \max\{2x, x-y, 2y\}$ are equivalent because $(1,-1) \leq \frac{1}{2}(2,0) + \frac{1}{2}(0,2)$. The variable y has a negative "exponent" in Q, so that the Laurent polynomial Q cannot be produced by a $(\max,+)$ circuit at all. The example given in Fig. 2 shows that even *optimal* $(\max,+,-x_i)$ circuits computing (non-Laurent) $(\max,+)$ polynomials may produce negative "exponents."

Factors of polynomials Let $P = \sum_{a\in A} c_a X^a$ be a (non-Laurent) polynomial, and $X^b = \prod_{i=1}^n x_i^{b_i}$ a monomial; hence, $A \subseteq \mathbb{N}^n$ and $b \in \mathbb{N}^n$. The monomial X^b is a *factor* of P if it divides all monomial of P, that is, if all vectors $a-b$ with $a \in A$ are nonnegative. The monomial X^b whose ith exponent is $b_i = \min\{a_i : a \in A\}$ is the *greatest factor* of P. That is, the greatest factor of P is the monomial of largest possible degree $b_1 + \cdots + b_n$ dividing all monomials of P. The *contraction* of the polynomial P is the polynomial

$$[P] := P/M = \sum_{a\in A} c_a X^{a-b}\,.$$

where $M = X^b$ is the greatest factor of P. For example, if $P = x_i$ (a single variable), then $M = x_i$ and $[P] = 1$ (the "multiplicative" identity), while if $P = c \in \mathbb{R}_+$ (a constant), then $M = 1$ and $[P] = c$. In tropical semirings, the "multiplicative" identity is 0. So, the contraction of a tropical $(\max,+)$ polynomial $P = \max_{a\in A}\ \langle a,x\rangle + c_a$ is $[P] := P - M = \max_{a\in A}\ \langle a-b,x\rangle + c_a$.

4 Syntactic elimination of reciprocal inputs

The goal of this section is to prove that we can efficiently eliminate reciprocal inputs ${x_1}^{-1}, \ldots, {x_n}^{-1}$ from a $(+, \cdot, {x_i}^{-1})$ circuit, as long as the Laurent polynomial *produced* by this circuit has no negative exponents (Lemma 3). Recall that a $(+, \cdot, {x_i}^{-1})$ circuit stands either for an arithmetic $(+, \cdot, 1/x_i)$ or for a tropical $(\min, +, -x_i)$ or $(\max, +, -x_i)$ circuit. That is, in tropical circuits, "addition" is either $x + y := \min\{x, y\}$ or $x + y := \max\{x, y\}$, "multiplication" is $x \cdot y := x + y$, and reciprocal inputs are $x_i^{-1} := -x_i$.

The following lemma shows that the Laurent polynomials produced by $(+, \cdot, {x_i}^{-1})$ circuits are of a very special form "polynomial P divided by a monomial M," where both P and M have not much larger $(+, \cdot)$ circuits.

Lemma 1. *Let Φ be a $(+, \cdot, {x_i}^{-1})$ circuit of size s. Then the Laurent polynomial Q produced by Φ is of the form $Q = P/M$, where P is a polynomial and M a monomial such that both P and M can be simultaneously produced by a $(+, \cdot)$ circuit of size at most $4s$.*

Here, as customary, P/M stands for $P \cdot M^{-1}$; for example, in the case of tropical circuits, P/M is the tropical (Laurent) polynomial of the form $P - M$.

Proof. Our goal is to transform the $(+, \cdot, {x_i}^{-1})$ circuit Φ circuit into a $(+, \cdot)$ circuit Φ' of size at most $4s$ producing the pair (P, M) with $Q = P/M$. We are going to build Φ' by traversing the circuit Φ from inputs towards outputs. Let 1 be the multiplicative identity element of the underlying semiring; this is constant 1 in the arithmetic case, and is constant 0 in the tropical case.

At the inputs x_i, x_i^{-1} and $c \in \mathbb{R}_+$, the corresponding pairs (P, M) are $(x_i, 1)$, $(1, x_i)$ and $(c, 1)$. Now assume that the Laurent polynomials Q_1 and Q_2 produced at the predecessors of some gate already have the desired representations $Q_1 = P_1/M_1$ and $Q_2 = P_2/M_2$. We have to show that the Laurent polynomial produced at this gate is also of the form $Q = P/M$, where P is a polynomial and M is a monomial. If this is a "multiplication" gate, then[4] $Q = Q_1 \cdot Q_2 = (P_1/M_1) \cdot (P_2/M_2) = P_1P_2/M_1M_2$, and we can take $P = P_1P_1$ and $M = M_1M_2$; we replaced one multiplication gate by two multiplication gates in this case. If this is an "addition" gate, then $Q = Q_1 + Q_2 = (P_1/M_1) + (P_2/M_2) = (P_1M_2 + P_2M_1)/M_1M_2$, and we can take $P = P_1M_2 + P_2M_1$ and $M = M_1M_2$. In this case, we replaced one addition gate by one addition gate and three multiplication gates in this case. The resulting $(+, \cdot)$ circuit Φ' has at most four times more gates than the original $(+, \cdot, {x_i}^{-1})$ circuit Φ, as claimed. ∎

If Q is a Laurent polynomial produced by a $(+, \cdot, x_i^{-1})$ circuit, then Lemma 1 gives us a not much larger $(+, \cdot)$ circuit (without reciprocal inputs) that simultaneously produces a polynomial P and a monomial M such that $Q = P/M$. That is, we have removed reciprocal inputs $x_1^{-1}, \ldots, x_n^{-1}$ at the cost of introducing one "division by a monomial" gate. Our goal is to eliminate also this division gate. If Q is a non-Laurent polynomial (has no negative exponents), then the monomial M in the representation $Q = P/M$ must divide all monomials of the polynomial P. The next Lemma 2 shows that, at the cost of a quadratic increase in size, it is possible to eliminate the last division gate in the case when M is the "largest possible" monomial, that is, when M is the greatest factor of P: then $Q = [P] \cdot M$, where $[P]$ is the contraction of the polynomial P.

The following lemma holds for circuits over any semiring $(+, \cdot)$ with the following property:

[4] As customary, we will sometimes omit the "multiplication" symbol $\cdot$ and write xy for $x \cdot y$.

($*$) if P_1 and P_2 are polynomials produced at some gates of a $(+,\cdot)$ circuit, then every factor of the polynomial $P_1 + P_2$ is a factor of both polynomials P_1 and P_2.

Non-monotone arithmetic $(+,\cdot,-)$ circuits do not have this property (due to possible cancellations). For example, x is a factor of $P = P_1 + P_2$ with $P_1 = x + y$ and $P_2 = x - y$ but is neither a factor of P_1 nor of P_2. But monotone arithmetic circuits as well as tropical circuits already have this property.

Lemma 2. *If a polynomial P of n variables can be produced by a $(+,\cdot)$ circuit of size s, then its contraction $[P]$ can be produced by a $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2)$.*

Proof. Let Φ be a $(+,\cdot)$ circuit of size s producing P. Call a monomial $X^a = \prod_{i=1}^n x_i^{a_i}$ *small* if $a_i \leq 2^s$ for all $i = 1,\ldots,n$. Since the multiplication gates have fanin two, every factor of the polynomial produced at any gate of Φ is small. Under an *extended* $(+,\cdot)$ circuit we understand a $(+,\cdot)$ circuit in which arbitrary small monomials can be used as inputs for free.

The following claim reduces the problem of producing the contraction $[P]$ of our polynomial P to producing these small monomials.

Claim. *The contraction $[P]$ of P can be produced by an extended $(+,\cdot)$ circuit of size $3s$.*

Proof. Our goal is to transform the $(+,\cdot)$ circuit Φ producing the polynomial P into an extended $(+,\cdot)$ circuit Φ' of size at most $3s$ producing $[P]$. Again, we are going to build Φ' by traversing the circuit Φ from inputs towards outputs. At the inputs x_i and $c \in \mathbb{R}_+$ of the circuit Φ, we have $[x_i] = 1$ and $[c] = c$.

Now assume that we are dealing with some gate in Φ producing a polynomial P from the polynomials P_1 and P_2 produced by its predecessors. By construction, we have already built a part of Φ' producing the contractions $[P_1] = P_1/M_1$ and $[P_2] = P_2/M_2$, where M_1 and M_2 are the greatest factors of the polynomials P_1 and P_2. If $P = P_1 \cdot P_2$, then $M = M_1 \cdot M_2$ is the greatest factor of P, and we can produce the contraction $[P]$ of P as $[P] = P/M = (P_1/M_1) \cdot (P_2/M_2) = [P_1] \cdot [P_2]$; in this case, we need no new gates. If $P = P_1 + P_2$ and M is the greatest factor of P, then we can produce $[P]$ as

$$[P] = [P_1 + P_2] = \frac{P_1 + P_2}{M} = \frac{M_1 \cdot [P_1] + M_2 \cdot [P_2]}{M} = M_1' \cdot [P_1] + M_2' \cdot [P_2]$$

where $M_1' = M_1/M$ and $M_2' = M_2/M$. Property ($*$) ensures that M divides both monomials M_1 and M_2. So, M_1' and M_2' are small (non-Laurent) monomials, and can be used as inputs for free. The resulting extended $(+,\cdot)$ circuit Φ' has at most $3s$ gates and produces the polynomial $[P]$. ∎

The extended $(+,\cdot)$ circuit guaranteed by the claim may have up to $t = 2s$ small monomials as inputs. Each variable in each of these monomials may have degree up to 2^s. These additional input monomials can be removed at the cost of a quadratic increase in circuit size. Namely, by repeated squaring, for each variable x_i, all the univariate monomials $x_i^2, x_i^{2^2}, \ldots, x_i^{2^s}$ can be simultaneously produced using only s multiplication gates. Then each x_i^d with $d \leq 2^s$ can be produced using at most s additional multiplications (by looking at the binary expansion of the exponent d). Thus, every small input monomial of the circuit given by Claim 1 can be produced by using at most $2ns$ multiplication gates, and we obtain a (non-extended) $(+,\cdot)$ circuit of size at most $2ns \cdot t + 3s = 4ns^2 + 3s = \mathcal{O}(ns^2)$ producing the contraction $[P]$ of P, as desired. ∎

In the proofs of Theorems 1 to 3, we will use the following simple consequence of Lemmas 1 and 2 for circuits producing non-Laurent polynomials (without negative exponents).

Lemma 3. *If an n-variate non-Laurent polynomial can be produced by a $(+,\cdot,{x_i}^{-1})$ circuit of size s, then it can be produced by a $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2)$.*

Proof. Let Q be an n-variate non-Laurent polynomial (hence, there are no negative exponents), and suppose that Q can be produced by a $(+,\cdot,{x_i}^{-1})$ circuit Φ of size s. Lemma 1 gives us a $(+,\cdot)$ circuit Φ' of size at most $4s$ simultaneously producing a polynomial P and a monomial M such that $P = Q\cdot M$. Let X^a be the greatest factor of the polynomial Q. Since $[P] = [Q\cdot M] = [Q]$, we obtain $Q = [Q]\cdot X^a = [P]\cdot X^a$. By Lemma 2, the polynomial $[P]$ can be produced by a $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2)$. On the other hand, since the polynomial P can be produced by a $(+,\cdot)$ circuit of size at most $4s$ (by the circuit Φ'), no variable can have degree larger than $d = 2^{4s}$ in P. Since the monomial X^a divides all monomials of P, no entry of the vector a can be larger than 2^{4s} as well. So, by repeated squaring, the monomial X^a can be produced by a $(+,\cdot)$ circuit of size $\mathcal{O}(n\log d) = \mathcal{O}(ns)$, and we have the desired $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2)$ producing our polynomial Q. ∎

Remark 2. Even if the produced polynomial Q is a Laurent polynomial with negative exponents, we still have $Q = [P]\cdot X^a$ for the Laurent monomial X^a with each a_i being the minimum of the (possible negative) "exponents" of x_i in Q. But then the Laurent monomial X^a might have negative exponents, and could not be produced by a $(+,\cdot)$ circuit at all. This is why we require the produced polynomial Q to be a non-Laurent polynomial (without negative exponents).

5 Proof of Theorem 1: arithmetic circuits

Suppose that an n-variate polynomial P can be computed by a monotone arithmetic $(+,\cdot,1/x_i)$ circuit Φ of size s. Our goal is to show that P can also be computed by a monotone $(+,\cdot)$ circuit of size $\mathcal{O}(ns^2)$. To show this, let Q be the Laurent polynomial produced by the circuit Φ. Since Φ computes P, we know that $P(x) = Q(x)$ holds for all $x \in \mathbb{R}_+^n$. In view of Lemma 3, it is enough to show that $Q = P$, i.e., that the circuit Φ produces the polynomial P itself.

A basic fact about arithmetic polynomials is that if a univariate polynomial F of degree d vanishes on any set $S \subseteq \mathbb{R}$ of $|S| \geq d+1$ points, then F is a zero polynomial. Easy induction on the number of variables extends this fact to multivariate polynomials (see, for example, [1, Lemma 2.1]): if $F \in \mathbb{R}[x_1,\ldots,x_n]$ is a nonzero polynomial of degree d, then for every subset $S \subseteq \mathbb{R}$ of size $|S| \geq d+1$, the polynomial F takes nonzero value on at least one point $x \in S^n$.

Now take a monomial M of sufficiently large degree such that the polynomial MQ has no negative exponents, and consider the polynomial $F = MP - MQ$. Assume for the sake of contradiction that $Q \neq P$. Then F is a nonzero polynomial of finite degree d and, when applied with any set $S \subseteq \mathbb{R}_+$ of size $|S| \geq d+1$, the aforementioned fact yields $F(x) \neq 0$ and, hence, also $P(x) \neq Q(x)$ for some $x \in \mathbb{R}_+^n$, a contradiction. ∎

6 Proof of Theorem 2: minimization

Let $P(x_1,\ldots,x_n) = \min_{a\in A}\langle a,x\rangle + c_a$ be a tropical $(\min,+)$ polynomial; hence $A \subseteq \mathbb{N}^n$ and $c_a \in \mathbb{R}_+$. Suppose that P can be computed by a $(\min,+,-x_i)$ circuit Φ of size s. Our goal is to show that P can also be computed by a $(\min,+)$ circuit of size $\mathcal{O}(ns^2)$.

The circuit Φ produces some tropical Laurent polynomial $Q(x) = \min_{b\in B} \langle b, x\rangle + c_b$ with $B \subseteq \mathbb{Z}^n$ and $c_b \in \mathbb{R}_+$ for all $b \in B$, and with the property that $Q(x) = P(x)$ holds for all $x \in \mathbb{R}^n_+$. In order to apply Lemma 3, it is enough to show that Q has no negative "exponents," i.e., that $B \subseteq \mathbb{N}^n$ holds. This follows from the general characterization of equivalent tropical Laurent polynomials given by Jerrum and Snir [9] (see Section 3): for every vector $b \in B$ there must be a convex combination c of vectors in A such that $b \geq c$ holds; since the vector c is nonnegative, the vector b must also be nonnegative. But in the case of minimization, the inclusion $B \subseteq \mathbb{N}^n$ can also be shown more directly.

Assume for the sake of contradiction that $b_i < 0$, that is, $b_i \leq -1$ holds for some vector $b \in B$ and some position i. Let $K = 1 + c_b$, where c_b is the "coefficient" of the tropical term $\langle b, x\rangle + c_b$ of the polynomial Q, and consider the weighting $x \in \{0, K\}^n$ which sets $x_i := K$ and $x_j := 0$ for all $j \neq i$. On this weighting, we have $P(x) \geq 0$ but $Q(x) \leq \langle b, x\rangle + c_b = b_i \cdot K + c_b \leq -K + c_b < 0$, a contradiction with $Q(x) = P(x)$. □

Remark 3. If we required $(\max, +, -x_i)$ circuits to correctly compute a given polynomial on all input weightings from $\mathbb{R}^n$ (including negative weights), then the same argument would yield the $(\max, +)$ version of Theorem 2: just give a sufficiently small negative weight to a "bad" position i with $b_i < 0$ to enforce $Q(x) > P(x)$.

But since we only require $(\max, +, -x_i)$ circuits to correctly compute a given $(\max, +)$ polynomial on *nonnegative* weightings (see Remark 1 for why this relaxation is reasonable), the above argument does not work: the set $B \subseteq \mathbb{Z}^n$ of "exponent" vectors of the Laurent $(\max, +)$ polynomial Q can contain any vector $b \in \mathbb{Z}^n$ such that $b \leq a$ holds for some $a \in A$. The example in Fig. 2 shows that, in general, negative "exponents" cannot be excluded.

7 Proof of Theorem 3: maximization

Let $Q(x) = \max_{b\in B} \langle b, x\rangle + c_b$ be a tropical Laurent $(\max, +)$ polynomial; hence, $B \subseteq \mathbb{Z}^n$ and $c_b \in \mathbb{R}_+$ for all $b \in B$. Recall that the *degree* of a (tropical) term $\langle b, x\rangle + c_b$ is the sum $\langle b, \vec{1}\rangle = b_1 + \cdots + b_n$ of its "exponents." A Laurent polynomial is *homogeneous* if all its terms have the same degree. The *higher envelope* of Q is the Laurent polynomial $\lceil Q\rceil = \max_{b\in\lceil B\rceil} \langle b, x\rangle + c_b$, where $\lceil B\rceil \subseteq B$ is the set of all "exponent" vectors $b \in B$ of Q whose degree is maximal. Note that the Laurent polynomial $\lceil Q\rceil$ is always homogeneous.

Lemma 4. *Let Q be a Laurent $(\max, +)$ polynomial, and P a homogeneous non-Laurent $(\max, +)$ polynomial of degree m. If Q is equivalent to P, then the higher envelope $\lceil Q\rceil$ of Q is a non-Laurent polynomial of degree m, and is also equivalent to P.*

Proof. Let $P(x) = \max_{a\in A} \langle a, x\rangle + c_a$; hence, $A \subseteq \mathbb{N}^n$, $c_a \in \mathbb{R}_+$ and $\langle a, \vec{1}\rangle = a_1 + \cdots + a_n = m$ holds for all $a \in A$. Let $Q(x) = \max_{b\in B} \langle b, x\rangle + c_b$; hence, $B \subseteq \mathbb{Z}^n$ and $c_b \in \mathbb{R}_+$ for all $b \in B$. Suppose that Q is equivalent to P; hence, $Q(x) = P(x)$ holds for all $x \in \mathbb{R}^n_+$. The (possibly Laurent, at this moment) polynomial $\lceil Q\rceil$ is homogeneous by its definition. Let us first show that $\lceil Q\rceil$ has the same degree m as P.

Claim 1. $\langle b, \vec{1}\rangle = m$ *for all* $b \in \lceil B\rceil$.

Proof. Let $c_A := \max_{a\in A} c_a$ and $c_B = \max_{b\in B} c_b$ be the largest "coefficients" of polynomials P and Q. If $\langle b, \vec{1}\rangle \leq m - 1$ held for some vector $b \in \lceil B\rceil$, then $\langle b, \vec{1}\rangle \leq m - 1$ would hold for all vectors $b \in B$. So, if we take a sufficiently large number r, say $r = 1 + c_B$, then for the vector $\vec{r} = (r, \ldots, r)$, we obtain $Q(\vec{r}) = \max_{b\in B} \{r \cdot \langle b, \vec{1}\rangle + c_b\} \leq rm - r + c_b \leq rm - 1$, while

$P(\vec{r}) = \max_{a\in A} \{r \cdot \langle a, \vec{1}\rangle + c_a\} \geq rm$. On the other hand, if $\langle b, \vec{1}\rangle \geq m+1$ held for some vector $b \in \lceil B \rceil$, then on the vector $\vec{r}$ with $r = 1 + c_A$, we would have $Q(\vec{r}) \geq \langle b, \vec{r}\rangle \geq rm + r$, while $P(\vec{r}) \leq rm + c_A < Q(\vec{r})$. Thus, $\langle b, \vec{1}\rangle = m$ holds for all $b \in \lceil B \rceil$. ∎

Let us now show that $\lceil Q \rceil$ is a non-Laurent polynomial, that is, has no negative "exponents."

Claim 2. $\lceil B \rceil \subseteq \mathbb{N}^n$.

Proof. Assume for the sake of contradiction that some vector $b \in \lceil B \rceil$ has a negative entry, and let $I = \{i\colon b_i > 0\}$. Since, by Claim 1, $\langle b, \vec{1}\rangle = m$ holds, we have $\sum_{i\in I} b_i \geq m+1$. Take $r := 1 + c_A$, and consider the weighting x with $x_i = r$ for all $i \in I$, and $x_i = 0$ for all $i \notin I$. Then $Q(x) \geq \langle b, x\rangle = r \cdot \sum_{i\in I} b_i \geq rm + r$, but $P(x) \leq P(\vec{r}) \leq rm + c_A < Q(x)$, a contradiction. ∎

Claim 3. *The polynomial $\lceil Q \rceil$ is equivalent to Q and, hence, also to P.*

Proof. Fix an arbitrary input weighting $x \in \mathbb{R}^n_+$. We have to show that $\lceil Q \rceil(x) = Q(x)$ holds. By Claims 1 and 2, $\lceil Q \rceil$ is a homogeneous polynomial of degree m. Hence, $\lceil Q \rceil(x + \vec{1}) = \lceil Q \rceil(x) + m$. Since Q is equivalent to P, and since the polynomial P is homogeneous of degree m, we also have $Q(x+\vec{1}) = P(x+\vec{1}) = P(x) + m = Q(x) + m$. So, it remains to show that $Q(x+\vec{1}) = \lceil Q \rceil(x+\vec{1})$ holds. Let $b \in B$ be a vector on which the maximum in $Q(x+\vec{1})$ is achieved; hence, $\langle b, x\rangle + c_b + \langle b, \vec{1}\rangle = Q(x) + m$. Since $\langle b, x\rangle + c_b \leq Q(x)$ holds, $\langle b, \vec{1}\rangle \geq m$ and, hence, $b \in \lceil B \rceil$ follows. That is, the maximum in $Q(x+\vec{1})$ is achieved on a vector in $\lceil B \rceil$, as desired.

This completes the proof of Claim 3 and, thus, the proof of Lemma 4. ∎

Remark 4. Note that the homogeneity of the polynomial P was only used in the proof of Claim 3. If P is a not necessarily homogeneous polynomial of degree m, then for every $x \in \mathbb{R}^n_+$ we still have $Q(x+\vec{r}) = \lceil Q \rceil(x+\vec{r})$ for all $r \geq 1 + Q(x)$: if $b \in B$ is a vector on which the maximum in $Q(x+\vec{r})$ is achieved, then $\langle b, x\rangle + c_b + r\cdot\langle b, \vec{1}\rangle = Q(x+\vec{r}) = P(x+\vec{r}) \geq rm$. Since $\langle b, x\rangle + c_b \leq Q(x) \leq r-1$, we obtain $r - 1 + r\cdot\langle b, \vec{1}\rangle \geq rm$ and, hence, $\langle b, \vec{1}\rangle \geq m - 1 + 1/r$. Since $\langle b, \vec{1}\rangle$ is an integer, we obtain $\langle b, \vec{1}\rangle \geq m$, that is, $\langle b, \vec{1}\rangle = m$. But for $0 \leq r \leq Q(x)$, the maximum in $Q(x+\vec{r})$ may be achieved on a vector $b \in B$ with $\langle b, \vec{1}\rangle < m$.

Proof of Theorem 3. Let $P(x) = \max_{a\in A} \langle a, x\rangle + c_a$ be a homogeneous $(\max, +)$ polynomial of some degree m; hence, $A \subseteq \mathbb{N}^n$, $c_a \in \mathbb{R}_+$ and $\langle a, \vec{1}\rangle = a_1 + \cdots + a_n = m$ holds for all $a \in A$. Suppose that the polynomial P can be computed by a $(\max, +, -x_i)$ circuit Φ of size s. Our goal is to show that then P can be computed by a $(\max, +)$ circuit of size $\mathcal{O}(ns^2)$.

Let $Q(x) = \max_{b\in B} \langle b, x\rangle + c_b$ be the tropical Laurent polynomial produced by the circuit Φ; hence, $B \subseteq \mathbb{Z}^n$ and $c_b \in \mathbb{R}_+$ for all $b \in B$. Since the circuit Φ computes the polynomial P, we know that Q is equivalent to P. Lemma 4 implies that the higher envelope $\lceil Q \rceil$ of Q is a non-Laurent polynomial (has no negative "exponents") and is also equivalent to P. So, by Lemma 3, it is enough to show that the polynomial $\lceil Q \rceil$ can be produced by a $(\max, +, -x_i)$ circuit of size $\leq s$. As observed already by Jerrum and Snir [9, Theorem 2.4], the desired circuit Φ' can be obtained from Φ by appropriately discarding ingoing edges of some of its max gates.

Namely, Laurent polynomials x_i, $-x_i$, $c \in \mathbb{R}_+$ produced at input gates coincide with their higher envelopes. If the higher envelopes $\lceil Q_1 \rceil$ and $\lceil Q_2 \rceil$ of the Laurent polynomials Q_1 and Q_2

produced at the predecessors of a gate in Φ are already produced in the new circuit, then do the following. If this is an addition $(+)$ gate, then do nothing: since $\langle b_1 + b_2, \vec{1}\rangle = \langle b_1, \vec{1}\rangle + \langle b_2, \vec{1}\rangle$, the higher envelope $\lceil Q_1 + Q_2 \rceil = \lceil Q_1 \rceil + \lceil Q_2 \rceil$ is already produced at this gate. If this is a max gate, and if one of $\lceil Q_1 \rceil$ and $\lceil Q_2 \rceil$ has smaller degree than the other, then delete the ingoing edge from the corresponding ("smaller") predecessor gate. If the degrees of $\lceil Q_1 \rceil$ and $\lceil Q_2 \rceil$ are equal, then do nothing. ∎

8 Final remarks and open problems

The main message of this paper is that reciprocal inputs cannot substantially decrease the size of monotone arithmetic $(+, \cdot)$ as well as of tropical $(\min, +)$ circuits (Theorems 1 and 2). The same holds for $(\max, +)$ circuits (Theorem 3) as long as the computed $(\max, +)$ polynomial $P(x) = \max_{a\in A}\langle a, x\rangle + c_a$ is homogeneous, that is, $a_1 + \cdots + a_n = m$ holds for some $m \in \mathbb{N}$ and all $a \in A$.

Problem 1. *Can the* $(\max, +)/(\max, +, -x_i)$ *gap be super-polynomial for non-homogeneous* $(\max, +)$ *polynomials?*

As a possible candidate for a polynomial showing a large $(\max, +)/(\max, +, -x_i)$ gap we suggest the non-homogeneous *heaviest co-path* polynomial: $\text{co-Path}_n(x) = \max_p \sum_{e\notin p} x_e$, where the maximum is over all simple paths p in K_n from the vertex $s = 1$ to the vertex $t = n$; we view paths as sets of their edges. Note that this polynomial is not homogeneous: the degrees of its monomials vary between $\binom{n}{2} - n + 1$ and $\binom{n}{2} - 1$. The polynomial co-Path_n is the dual of the s-t path polynomial $\text{Path}_n(x) = \min_p \sum_{e\in p} x_e$, which can be computed by a $(\min, +)$ circuit of size $\mathcal{O}(n^3)$ resulting from the Bellman–Ford–Moore dynamic programming algorithm for the shortest s-t path problem. So, by Corollary 3, the polynomial co-Path_n can be computed by a $(\max, +, -x_i)$ circuit using $\mathcal{O}(n^3)$ gates.

Problem 2. *Prove or disprove that the* $(\max, +)$ *polynomial* co-Path_n *requires* $(\max, +)$ *circuits of super-polynomial size.*

By Corollary 4, we know that the $(\min, +, -x_i)/(\min, +, \max)$ gap can be exponential. So, a natural question is whether reciprocal inputs can substantially speed up $(\min, +, \max)$ circuits? Recall that the $(\min, +, \max, -x_i)$ and $(\min, +, -)$ circuit complexities are proportional (Proposition 1). So, the problem actually is the following.

Problem 3. *Can the* $(\min, +, \max)/(\min, +, -)$ *gap be super-polynomial?*

To answer this question in the affirmative, it is enough to find a minimization problem f that can be solved by a "small" $(\min, +, -)$ circuit but requires "large" $(\min, +, \max)$ circuits. The latter task (lower bound) can be settled by proving a large lower bound on the monotone Boolean circuit complexity of the Boolean version of the minimization problem f. Namely, define the *Boolean version* of a minimization problem $f(x) = \min_{S\in\mathcal{F}} \sum_{i\in S} x_i$ to be the monotone Boolean function $g(x) = \bigvee_{S\in\mathcal{F}} \bigwedge_{i\in S} x_i$.

Proposition 2. *If a minimization problem can be solved by a* $(\min, +, \max)$ *circuit of size* s*, then the Boolean version of this problem can be computed by a monotone Boolean* $(\wedge, \vee)$ *circuit of size* s*.*

The intuition behind is simple. For a nonnegative number x, let $[x] = 0$ if $x = 0$, and $[x] = 1$ if $x > 0$. Then for all $x, y \in \mathbb{R}_+$, we have $[\min(x, y)] = [x] \wedge [y]$, and $[\max(x, y)] = [x + y] = [x] \vee [y]$ (see Appendix A for details).

So, Problem 3 reduces to showing that some optimization problem, whose Boolean version requires "large" monotone Boolean circuits, can be solved by "small" tropical circuits using subtraction gates. A possible candidate could be the well-known *assignment problem*. The corresponding $(\min, +)$ polynomial is the tropical permanent polynomial $\mathrm{Per}_n(x) = \min_M \sum_{e \in M} x_e$, where the minimum is over all perfect matchings M in the complete bipartite $n \times n$ graph. As shown by Jerrum and Snir [9], Per_n requires $(\min, +)$ circuits of size $2^{\Omega(n)}$. In fact, Per_n requires even $(\min, +, \max)$ circuits of super-polynomial size.

Corollary 5. *Any* $(\min, +, \max)$ *circuit computing* Per_n *must have* $n^{\Omega(\log n)}$ *gates.*

Proof. The *logical permanent function* is the monotone Boolean function $\mathrm{per}_n(x)$ of n^2 variables that, given an input vector $x \in \{0, 1\}^{n^2}$, outputs 1 if and only if the subgraph of $K_{n,n}$ specified by x has a perfect matching. A celebrated result of Razborov [15] is that every monotone Boolean circuit computing per_n must have $n^{\Omega(\log n)}$ gates. Since per_n is the Boolean version of Per_n, Proposition 2 yields the same lower bound on the size of $(\min, +, \max)$ circuits computing Per_n. ∎

Problem 4. *Can the tropical permanent* Per_n *be computed by a* $(\min, +, -)$ *circuit of polynomial size?*

Actually, even the following weaker version of Problem 3 remains open.

Problem 5. *Can the* $(\min, \max)/(\min, +, -)$ *gap be super-polynomial? That is, can subtractions help for bottleneck problems?*

A next challenge is to prove nontrivial *lower* bounds on the $(\min, +, -)$ circuit complexity. By "nontrivial" we mean a lower bound for a tropical polynomial of not too large degree. Say, a lower bound of 2^n on the $(\min, +, -)$ circuit complexity of the polynomial $P(x, y) = \max\{2^{2^n} x, y\}$ of doubly exponential degree is trivial: the size of a circuit computing a given polynomial is always at least the logarithm of the "degree" of this polynomial.

Problem 6. *Prove any nontrivial lower bound on the size of* $(\min, +, -)$ *circuits.*

The arithmetic analogue of $(\min, +, -)$ circuits is that of arithmetic $(+, \cdot, /)$ circuits. To prove an exponential lower bound for $(+, \cdot, /)$ circuits, Fomin, Grigoriev and Koshevoy [6] used an indirect argument: the hard to compute polynomials are constructed using *subtraction*. Even if $(+, \cdot, /)$ circuits cannot subtract, some polynomials with negative coefficients are still computable by $(+, \cdot, /)$ circuits. For example, the polynomial $f(x, y) = x^2 - xy + y^2$ can be computed by a $(+, \cdot, /)$ circuit $\Phi = (x^3 + y^3)/(x + y)$. A general result of Pólya [14] states that if F is a homogeneous polynomial such that $F(x) > 0$ for all $x \in \mathbb{R}^n_+$ such that $x_1 + \cdots + x_n = 1$, then $F(x) = P(x)/(x_1 + \cdots + x_n)^r$ for some $r \geq 1$ and some positive polynomial P (with only positive coefficients). Hence, every such polynomial F *can* be computed by a $(+, \cdot, /)$ circuit.

In [6], the authors present an explicit homogeneous n-variate polynomial F of degree only four (but with negative coefficients) which fulfills Pólya's condition and, hence, *can* be represented as a quotient P/Q of two positive polynomials. But the authors show that in any such representation $F = P/Q$, the polynomial P must have degree $d \geq 2^{2^{n-2}}$. Since the degree of the polynomial computed by a $(+,\cdot,/)$ circuit of size s cannot exceed 2^s, this implies that any $(+,\cdot,/)$ circuit computing F must have $s \geq \log_2 d \geq 2^{n-2}$ gates.

The case of tropical $(\min,+,-)$ circuits is less clear. First, in the tropical world, there exist no "additive" inverses at all (here "addition" is either min or max) and, hence, there is no tropical analogue of the arithmetic polynomial constructed in [6]. A next problem is whether any tropical analogue of Pólya's theorem holds. In particular, the mere fact that the function computed by a $(\min,+,-)$ circuit is "nonnegative" (takes nonnegative values on nonnegative inputs) does not imply that this function can be computed by a $(\min,+,\max)$ circuit at all: consider, for example, the circuit $\Phi(x,y) = \max\{x,y\} - \min\{x,y\}$ computing the nonnegative Laurent polynomial $P(x,y) = \max\{x-y, y-x\}$. So, Problem 6 may be an even harder challenge than Problem 3.

When dealing with $(\min,+,-)$ circuits, the following simple observation may be useful: if a tropical polynomial P can be computed by a $(\min,+,-)$ circuit of size s, then P can be computed by a $(\min,+,-)$ circuit of size $2s$ with at most one subtraction gate as the output gate. Namely, we can move subtraction $(-)$ gates towards the output gate using the equations $\min\{a-b,c\} = \min\{a,b+c\} - b$, $(a-b)+c = (a+c)-b$ and $(a-b)-c = a-(b+c)$. At each min gate two new gates are added, while at $+$ and $-$ gates no new gates are added. Thus, $(\min,+,-)$ circuits are essentially of the form $\Phi(x) = \Phi_1(x) - \Phi_2(x)$, where Φ_1 and Φ_2 are $(\min,+)$ circuits.

A Proof of Proposition 2

Recall that the *Boolean version* of a minimization problem $f(x) = \min_{S\in\mathcal{F}} \sum_{i\in S} x_i$ is the monotone Boolean function $g(x) = \bigvee_{S\in\mathcal{F}} \bigwedge_{i\in S} x_i$. Suppose that the problem f can be solved by a $(\min,+,\max)$ circuit $\Phi(x_1,\ldots,x_n)$ of size s. Our goal is to show that then its Boolean version g can be computed by a monotone Boolean $(\wedge,\vee)$ circuit of size s.

For a nonnegative real number $x \in \mathbb{R}_+$, let $[x] = 0$ if $x = 0$, and $[x] = 1$ if $x > 0$. For a vector $x \in \mathbb{R}_+^n$, let $[x] = ([x_1],\ldots,[x_n]) \in \{0,1\}^n$ denote the corresponding Boolean vector. Consider the Boolean function $\hat{f}(x) = \bigwedge_{S\in\mathcal{F}} \bigvee_{i\in S} x_i$. Note that, for all inputs $x \in \{0,1\}^n$, we have

$$[f(x)] = \left[\min_{S\in\mathcal{F}} \sum_{i\in S} x_i\right] = \bigwedge_{S\in\mathcal{F}} \left[\sum_{i\in S} x_i\right] = \bigwedge_{S\in\mathcal{F}} \bigvee_{i\in S} [x_i] = \hat{f}([x]) = \hat{f}(x)\,. \tag{1}$$

Let $\hat{\Phi}(x_1,\ldots,x_n)$ be a monotone Boolean $(\wedge,\vee)$ circuit obtained from Φ as follows: replace each constant input $c \in \mathbb{R}_+$ by $[c]$, replace each min gate $\min(u,v)$ by an AND gate $u \wedge v$, each max gate $\max(u,v)$ and each addition gate $u+v$ by an OR gate $u \vee v$. We know that $\Phi(x) = f(x)$ and, hence, also $[\Phi(x)] = [f(x)]$ holds for all $x \in \{0,1\}^n$. So, by Eq. (1), we have only to show that $[\Phi(x)] = \hat{\Phi}([x])$ holds for all inputs $x \in \{0,1\}^n$, because then $\hat{\Phi}(x) = \hat{\Phi}([x]) = [\Phi(x)] = [f(x)] = \hat{f}(x)$, that is, the Boolean circuit $\hat{\Phi}$ computes $\hat{f}$. The function $\hat{f}$ if the dual of the Boolean version $g(x) = \bigvee_{S\in\mathcal{F}} \bigwedge_{i\in S} x_i$ of our minimization problem f. Thus, the dual of the circuit $\hat{\Phi}$, obtained from $\hat{\Phi}$ by interchanging AND and OR gates, computes g.

If $\Phi = x_i$, then $[\Phi(x)] = x_i = \hat{\Phi}(x)$ (the input x is Boolean), and if $\Phi = c$ is a constant $c \in \mathbb{R}_+$, then $[\Phi(x)] = [c] = \hat{\Phi}(x)$. The rest follows by induction on the size of the circuit Φ using the equalities holding for all numbers $u,v \in \mathbb{R}_+$: $[\min(u,v)] = [u] \wedge [v]$, and $[\max(u,v)] = [u+v] = [u] \vee [v]$. ∎

Acknowledgment

The first author would like to thank Vladimir Lysikov for interesting discussions at the initial stage.

References

[1] N. Alon and M. Tarsi. Colorings and orientations of graphs. *Combinatorica*, 12:125–134, 1992.

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

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

[4] K. Fan. *Linear Inequalities and Related Systems*, chapter On systems of linear inequalities, pages 99–156. Princeton University Press, 1956.

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

[6] S. Fomin, D. Grigoriev, and G. Koshevoy. Subtraction-free complexity, cluster transformations, and spanning trees. *Found. Comput. Math.*, 15:1–31, 2016.

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

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

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

[10] S. Jukna and H. Seiwert. Greedy can beat pure dynamic programming. *Inf. Process. Letters*, 142:90–95, 2019.

[11] S. Jukna and H. Seiwert. Sorting can exponentially speed up pure dynamic programming. *Inf. Process. Letters*, Article Nr. 105962:159–160, 2020.

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

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

[14] G. Pólya. Über positive Darstellung von Polynomen. In *Vierteljschr. Naturforsch. Ges. Zürich*, volume 73. 1928.

[15] 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.

[16] V. Strassen. Vermeidung von Divisionen. *J. Reine Angew. Math.*, 264:184–202, 1973.

[17] É. Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

[18] L. G. Valiant. Negation can be exponentially powerful. *Theor. Comput. Sci.*, 12:303–314, 1980.

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

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il