Electronic Colloquium on Computational Complexity, Report No. 189 (2020)

# Shadows of Newton polytopes

Pavel Hrubeš* Amir Yehudayoff†

**Abstract**

We define the shadow complexity of a polytope $P$ as the maximum number of vertices in a linear projection of $P$ to the plane. We describe connections to algebraic complexity and to parametrized optimization. We also provide several basic examples and constructions, and develop tools for bounding shadow complexity.

## 1 Introduction

A *polytope* is the convex hull of a finite set of points in Euclidean space. Equivalently, it is a compact set that is defined by finitely many linear inequalities. Polytopes are central in convex geometry and linear optimization algorithms.

Our goal is to understand

*how many vertices can a shadow of a polytope have?*

A shadow of a polytope $P \subseteq \mathbb{R}^n$ is a set of the form $L(P)$, where $L : \mathbb{R}^n \to \mathbb{R}^2$ is a linear map. The shadows of $P$ are two-dimensional polygons, and hence typically much simpler than $P$. The *shadow complexity* of $P$ is

$$\sigma(P) = \max_L |\mathsf{vert}(L(P))|,$$

where $L$ is a linear map and $\mathsf{vert}(Q)$ is the vertex set of the polytope $Q$.

The shadow problem is interesting already in three-dimensional space. Moser's shadow problem asks about the shadow complexity of three-dimensional

*Mathematical Institute of the Czech Academy of Science, pahrubes@gmail.com. Supported by the GACR grant 19-27871X.

†Department of Mathematics, Technion-IIT, amir.yehudayoff@gmail.com



ISSN 1433-8092

polytopes [30]. Specifically, the question is what is the minimum of $\sigma(P)$ over all three dimensional polytopes $P$ with $n$ vertices. The solution is $\Theta(\log n/\log\log n)$; see [10, 28]. In other words, every $n$-vertex polytope in $\mathbb{R}^3$ has a projection to $\mathbb{R}^2$ with at least $\Omega(\log n/\log\log n)$ vertices, and there are polytopes where this is tight. The latter is quite surprising; in such a polytope, most vertices must disappear when projected to the plane.

Our main motivation comes from algebraic complexity theory. This is the study of computations of polynomials over a field. The connection between between polynomials and polytopes is via the notion of *Newton polytope*. Let $\mathbb{F}$ be a field. For a list of variables $x = (x_1, \ldots, x_n)$ and $\alpha \in \mathbb{N}^n$, let $x^\alpha$ be the monomial $\prod_{i=1}^n x_i^{\alpha_i}$. A polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is a formal sum of the form $\sum_{\alpha\in\mathbb{N}^n} a_\alpha x^\alpha$ where $\sup(f) := \{\alpha \in \mathbb{N}^n : a_\alpha \neq 0\}$ is finite. The *Newton* polytope of $f$ is

$$\mathsf{Newt}(f) := \mathsf{conv}(\sup(f))\,,$$

where $\mathsf{conv}(\cdot)$ denotes the convex hull.

Koiran et al. [26] made a bold conjecture relating the complexity of $\mathsf{Newt}(f)$ with the computational complexity of $f$. The *$\tau$-conjecture for Newton polygons* asserts, roughly speaking, that if a *bi-variate* polynomial $f$ is easy to compute then $\mathsf{Newt}(f)$ has a small number of vertices. This conjecture has serious consequences. It implies that the permanent polynomial requires arithmetic circuit of exponential size. This is a central and long-standing open problem in algebraic complexity.

The Newton polytope of the permanent polynomial is the the *Birkhoff* polytope $\mathsf{DS}_n \subseteq \mathbb{R}^{n\times n}$; namely, the set of $n \times n$ doubly stochastic matrices. The vertices of $\mathsf{DS}_n$ are all $n \times n$ permutation matrices. This perspective leads us to the following question.

**Problem 1.** *What is $\sigma(\mathsf{DS}_n)$?*

The Birkhoff polytope has the curious property that it is *both* the Newton polytope of the determinant *and* of the permanent polynomial. This creates friction in the context of the $\tau$-conjecture. Determinant is easy to compute whereas permanent is largely believed to be hard. More specifically, it can be shown that the $\tau$-conjecture implies $\sigma(\mathsf{DS}_n) \leq 2^{O(\sqrt{n}\log^2 n)}$. Proving that $\sigma(\mathsf{DS}_n) = 2^{\Omega(n)}$ refutes this $\tau$-conjecture.[1]

[1]This observation came from Michael Forbes in a private conversation.

Any non-trivial connection between the arithmetic complexity of $f$ and some geometric complexity measure of $\mathsf{Newt}(f)$, such as shadow complexity, will be an exciting development.

We exhibit such a connection in the case of *monotone* computations. A *monotone* arithmetic circuit uses the operations $+, \times$ and only non-negative numbers so that no cancellations can occur in the course of a computation (for definitions see Section 5). They have been considered in the seminal papers of Valiant [42] and of Jerrum and Snir [21], and many others. We show that shadow complexity allows to prove hardness results for monotone computation.

**Theorem 1.1.** *Every monotone formula computing $f$ contains at least $\sigma(\mathsf{Newt}(f))$ leaves.*

What we are really interested in is understanding algebraic circuits, not formulas. We show that in some cases shadow complexity allows to lower bound monotone *circuit* complexity. A polynomial $f$ is *transparent* if $|\mathsf{supp}(f)| = \sigma(\mathsf{Newt}(f))$. In other words, there is a linear map $L : \mathbb{R}^n \to \mathbb{R}^2$ which maps $\mathsf{supp}(f)$ to distinct convexly independent points in $\mathbb{R}^2$.

**Theorem 1.2.** *If $f$ is transparent then every monotone circuit computing $f$ has size at least $\Omega(\sigma(\mathsf{Newt}(f)))$.*

Theorem 1.2 can be used to explicitly find a monotone multilinear polynomial in $n$ variables which requires an arithmetic circuit of size $\Omega(2^{n/3})$; see Corollary 5.11. This is stronger than the $2^{\Omega(n)}$ lower bound from [36], as well as the classical bounds from [42, 21] which are of the form $2^{\Omega(n^{1/2})}$.

**Remark 1.3.** *The transparency assumption is unavoidable. There exists a bivariate polynomial $f$ with a monotone circuit of size $O(n)$ such that $\mathsf{Newt}(f)$ has $2^n$ vertices (see Theorem 5.1).*

Shadow complexity has an algorithmic perspective as well. A polytope naturally defines a linear optimization problem $\Phi(w) = \max_{x \in P} \langle x, w \rangle$, where $\langle x, w \rangle$ is the standard inner product. The maximizers of this optimization problem are vertices of $P$. The Birkhoff polytope, e.g., corresponds to the maximum weight bipartite perfect matching problem. Some additional examples of linear optimization problems include the shortest path problem or the maximum cut problem.

In parametrized complexity, one considers weights that come from a one dimensional space $w(t) = w_0 + tw_1$ parametrized by $t \in \mathbb{R}$. The map $t \mapsto \Phi(w(t))$ is a convex and piecewise linear function. A natural complexity measure for such a map is the number $\beta(P, w(t))$ of the breakpoints in $\Phi(w(t))$. The *parametrized complexity* of $P$ now becomes

$$\beta(P) = \max_{w_0, w_1} \beta(P, w(t)).$$

The quantity $\beta(P)$ has been studied by Carstensen [8, 9], Mulmuley and Shah [31, 32], and many others. Carstensen [9] and later [32] showed that the shortest path problem in an $n$-vertex graph can have $2^{\Omega(\log^2 n)}$ breakpoints, and that the maximum cut problem can have $2^{\Omega(n)}$ breakpoints. In Section 3.4, we give an example of a polytope that corresponds to a linear optimization problem on $n$ variables with $2^{\Omega(n)}$ breakpoints; the previous constructions gave only $2^{\Omega(\sqrt{n})}$ breakpoints.

We observe a fundamental connection between shadow complexity and parameterized complexity.

**Theorem 1.4.** *If* $|\mathsf{vert}(P)| > 1$ *then* $\frac{\sigma(P)}{2} \leq \beta(P) \leq \sigma(P) - 1$.

This means that results from parametrized complexity translate to the language of shadows, and vice versa. Carstensen's lower bound for example implies that

$$\sigma(\mathsf{DS}_n) \geq 2^{\Omega(\log^2 n)}.$$

This is the best lower bound on $\sigma(\mathsf{DS}_n)$ we are aware of. The best upper bound we know is $\sigma(\mathsf{DS}_n) \leq 2^{O(n)}$. This is not entirely obvious and we shall explain this later on (see Proposition 3.11).

The connection between shadow and parametrized complexities leads to interesting conclusions. The idea, in a nutshell, is that if optimization over $P$ is easy then $\beta(P)$ is low. For example, if we can optimize over $P$ by a greedy algorithm then $\beta(P)$ is at most quadratic. We do not want to want to dive into the theory of greedy algorithms, or a formal definition for that matter. Edmonds and Rado [12, 15] proved that if $R \subseteq \{0, 1\}^n$ is a matroid then the optimization problem over $R$ can be solved by a greedy algorithm. Many generalizations of this theorem have been considered (see [43] and references within).

For our purposes, the following simple definition is sufficient. Let $P \subseteq \mathbb{R}^n$ be a polytope and $w \in \mathbb{R}^n$. We denote by $\mathrm{Opt}_P(w)$ the set of vertices $v$ of

$P$ such that $\langle v, w\rangle = \max_{x\in P} \langle x, w\rangle$. Given $w, w' \in \mathbb{R}^n$, we say that they are *order-equivalent* if for every $i, j \in [n]$, we have $w_i \le w_j$ iff $w'_i \le w'_j$. The polytope $P$ is *greedy-like*, if for every order-equivalent $w$ and $w'$, we have $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. In other words, $P$ is greedy-like if for every weight function $w$, where the maximum for $w$ is achieved on $P$ depends only on the order induced by $w$.

**Lemma 1.5.** *If $P \subseteq \mathbb{R}^n$ is a greedy-like polytope then $\beta(P) \le \binom{n}{2}$ and $\sigma(P) \le n(n-1)$.*

A more general link was established by Mulmuley [31]. He considers a model of computation called *PRAM model without bit operations* intended to solve decision problems or optimization problems. This model allows to use basic arithmetic operations such as $+, \times$ as well as $=, \le$, but does not allow access to the individual bits of the inputs. Mulmuley showed[2] that a fast parallel algorithm for optimizing over $P$ gives a small $\beta(P)$. This leads to several interesting lower bounds in this model.

The above can be further linked to our discussion concerning monotone arithmetic circuits. A monotone arithmetic formula can be interpreted as a computation over the semiring $(\mathbb{R}, \min, +, \infty, 0)$ which solves the optimization problem over $\mathsf{Newt}(f)$; see Section 5.1 for more details. This a particular instance of the PRAM model.

Are there general non trivial bounds on shadow complexity? Let $M_\sigma(n)$ be the maximum $\sigma(P)$ over all polytopes $P \subseteq \mathbb{R}^n$ with vertices in $\{0,1\}^n$.

**Proposition 1.6.** *There exist constants $0 < c_1 < c_2 < 1$ such that for every $n$ sufficiently large*

$$2^{c_1 n} \le M_\sigma(n) \le 2^{c_2 n}.$$

The fact that $c_2 < 1$ relies on a combinatorial result of Paturi and Zane [34].

## 1.1 Why the plane?

Why do we study projections of polytopes to two dimensions?

First, our results rely on the fact that in two dimensions Minkowski sum (defined in Section 2.3 ) is well-behaved with respect to the number of vertices. In $\mathbb{R}^2$, we have $|\mathsf{vert}(P+Q)| \le |\mathsf{vert}(P)| + |\mathsf{vert}(Q)|$. Already in $\mathbb{R}^3$, only the trivial upper bound $|\mathsf{vert}(P+Q)| \le |\mathsf{vert}(P)| \cdot |\mathsf{vert}(Q)|$ holds.

[2]There is a technical issue of bit-lengths which we avoid.

Second, there exists a polytope in $\mathbb{R}^3$ with $k$ vertices such that every projection to $\mathbb{R}^2$ has only $O(\log k / \log\log k)$ vertices. Hence it may happen that a polytope in $\mathbb{R}^n$ has exponentially many vertices when projected to $\mathbb{R}^3$ but only polynomially many when projected to $\mathbb{R}^2$.

That said, there are non-trivial upper bounds on the number of vertices of $P_1 + \cdots + P_r$ in $\mathbb{R}^d$ if $r$ is large. For the sake of simplicity, we discuss the case of $d = 3$. It follows from a result of Gritzman and Sturmfels [17] that, given polytopes $P_1, \ldots, P_r$ with $n_1, \ldots, n_r$ vertices in $\mathbb{R}^3$,

$$|\mathsf{vert}(P_1 + \cdots + P_r)| \leq O((n_1 + \cdots + n_r)^2)\,.$$

This beats the trivial bound $n_1 n_2 n_3$ already for $r = 3$. The improved bound could be used to derive non-trivial bounds on monotone computations of a bounded depth (see Remark 7.4).

### 1.2 Extension complexity

As a final remark, we briefly discuss a different possible connection between polytopes and algebraic complexity. The *extension complexity of* $P$, denoted $\mathrm{xc}(P)$, as the smallest $r$ such that $P$ is a linear projection of a polytope $Q \subseteq \mathbb{R}^m$ where $Q$ can be defined using $r$ inequalities and an arbitrary number of equalities; see [44, 37, 13] and references within. It is related to communication complexity and algorithms (see, e.g., [35]).

We observe that, like shadow complexity, extension complexity also allows to prove lower bounds on monotone computation. Namely, if $f$ has monotone formula of size $s$ then $\mathrm{xc}(\mathsf{Newt}(f)) \leq O(s)$. This uses simple properties of extension complexity together with a result of Balas [2].

Extension complexity, however, can not yield general lower bounds in the non-monotone setting. There exists a polynomial with a polynomial size arithmetic circuit, but whose Newton polytope has an exponential extension complexity. See Section 5.4 for more details.

## 2 Tools

We start by presenting several tools for bounding shadow complexity, including some elementary facts about Newton polytopes.

## 2.1 Parametrized complexity

Some of the bounds on shadow complexity we describe come from the algorithmic viewpoint. So, we first prove the connection between shadow complexity and parametrized complexity.

*Proof of Theorem 1.4.* It is convenient to argue about

$$B^*(P, w(t)) := \beta(P, w(t)) + 1,$$

which counts to the number of *pieces* of $\Phi(w(t))$. Given $w(t) = w_0 + tw_1$, define $L : \mathbb{R}^n \to \mathbb{R}^2$ by

$$L(x) = (\langle w_0, x\rangle, \langle w_1, x\rangle).$$

Because $\langle w(t), x\rangle = \langle (1, t), L(x)\rangle$, we see that

$$\max_{x\in P} \langle x, w(t)\rangle = \max_{y\in L(P)} \langle y, (1, t)\rangle \ .$$

Since the maximum is always achieved at a vertex of $L(P)$, we obtain $B^*(P, w(t)) \leq \sigma(P)$.

To prove the other inequality, we first show that $B^*(Q) \geq k/2 + 1$ for every polytope $Q$ in $\mathbb{R}^2$ with $k \geq 2$ vertices. Take non-parallel $w_0, w_1 \in \mathbb{R}^2$ so that $\langle v, w_1\rangle$ are distinct for distinct vertices $v$ of $Q$. Let $w(t) = w_0 + tw_1$ and $\bar{w}(t) = -w_0 + tw_1$. Each vertex $v$ of $Q$ can be separated from the other vertices by a hyperplane, and a small perturbation of the hyperplane is still separating. Hence, there exists a non-empty open interval $I$ such that either $\max_{x\in Q} \langle x, w(t)\rangle$ or $\max_{x\in Q} \langle x, \bar{w}(t)\rangle$ is achieved at $x = v$ on $t \in I$. (And $v$ is the only such vertex.) Let $v_1$ be the vertex for which $\langle x, w_1\rangle$ is the largest, and $v_2$ the one where it is smallest. When $t \to \infty$, both $\max_{x\in Q} \langle x, w(t)\rangle$ and $\max_{x\in Q} \langle x, \bar{w}(t)\rangle$ is achieved at $v_1$; similarly for $v_2$ and $t \to -\infty$. It follows that $B^*(Q, w(t)) + B^*(Q, \bar{w}(t)) \geq k + 2$ and so $B^*(Q) \geq k/2 + 1$.

Now, given $P \subseteq \mathbb{R}^n$, let $L : \mathbb{R}^n \to \mathbb{R}^2$ be a linear map so that $L(P)$ has $\sigma(P)$ vertices. By the above, there exists $w(t)$ in $\mathbb{R}^2$ so that $\max_{x\in L(P)} \langle x, w(t)\rangle$ has at least $\sigma(P)/2$ breakpoints. Maximizing $\langle x, w(t)\rangle$ on $L(P)$ is equivalent to maximizing some $w'(t)$ on $P$ and so $\beta(P) \geq \sigma(P)/2$. □

## 2.2 Greedy polytopes

Our goal here is to prove that $\sigma(P)$ is small whenever $P$ is greedy-like (Lemma 1.5).

*Proof of Lemma 1.5.* Let $w(t)$ be a line in $\mathbb{R}^n$. For a given $t$, the weight vector $w(t)$ defines a preorder on $[n]$ by $i \leq_t j$ iff $w(t)_i \leq w(t)_j$. Since $P$ is greedy-like, every breakpoint of $\Phi(w(t)) = \max_{x \in P} \langle x, w(t) \rangle$ occurs at a time where the order $\leq_t$ changes. Hence there exist $i \neq j$ such that the linear function $w(t)_i - w(t)_j$ changes sign. There are $\binom{n}{2}$ pairs, and a linear function can change sign at most once. So, $\Phi(w(t))$ has at most $\binom{n}{2}$ breakpoints. This means $\beta(P) \leq \binom{n}{2}$ and $\sigma(P) \leq n(n-1)$. □

We further show that the definition of greedy-like can be relaxed to weights for which the maximum is achieved at a unique vertex. This weaker notion can be easier to verify, as in the case of Kruskal's algorithm mentioned in Proposition 3.5.

**Lemma 2.1.** *Let $P \subseteq \mathbb{R}^n$ be a polytope. Assume that for every order-equivalent $w, w' \in \mathbb{R}^n$, the equality $Opt_P(w) = Opt_P(w')$ holds whenever $|Opt_P(w)| = 1$. Then $P$ is greedy-like.*

*Proof.* Let $P$ be as in the assumption. Assume that $w, w' \in \mathbb{R}^n$ are order-equivalent with $|\mathrm{Opt}_P(w)| \geq 1$. We want to show that $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. Given $v \in \mathrm{Opt}_P(w)$, we can find $z \in \mathbb{R}^n$ such that $\mathrm{Opt}_P(z) = \{v\}$. Hence for every $\epsilon > 0$ we have $\mathrm{Opt}_P(w + \epsilon z) = \{v\}$. For $\epsilon > 0$ small enough, we also have that $w + \epsilon z$ and $w' + \epsilon z$ are order-equivalent. It follows that $v \in \mathrm{Opt}_P(w' + \epsilon z)$. Letting $\epsilon$ tend to zero, we can conclude $v \in \mathrm{Opt}_P(w')$.

We have shown $\mathrm{Opt}_P(w) \subseteq \mathrm{Opt}_P(w')$. By symmetry, we also have $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. □

## 2.3 Operations on polytopes

Given $A, B \subseteq \mathbb{R}^n$, their *Minkowski sum* is defined as

$$A + B := \{a + b : a \in A, b \in B\}.$$

If $P$ and $Q$ are polytopes then $P + Q$ is also a polytope. In two-dimensions, Minkowski sum has nice properties. Let $P$ be a polytope in $\mathbb{R}^2$ with vertices $v_1, \ldots, v_k$ where $k > 1$. We can assume they are ordered so that $P$ lies in the left closed half plane determined by the line going from $v_i$ to $v_{i+1}$ for $i < k$, and similarly for $v_k$ and $v_1$. Let $E(P)$ be the collection of unit vectors in the direction of these $k$ edges. That is, vectors of the form $(v_{i+1} - v_i)/\|v_{i+1} - v_i\|$ for $i < k$, and $(v_1 - v_k)/\|v_1 - v_k\|$. If $|\mathsf{vert}(P)| \leq 1$ then $E(P) := \emptyset$.

**Lemma 2.2.** *Let $P_1, \ldots, P_r$ be non-empty polytopes in $\mathbb{R}^2$. Then $E(P_1 + \cdots + P_r) = \bigcup_{i=1}^{r} E(P_i)$. Consequently, $|\mathsf{vert}(P_1 + \cdots + P_r)| \leq \sum_{i=1}^{r} |\mathsf{vert}(P_i)|$. The latter holds for empty $P_i$'s as well.*

The lemma is folklore. It can be inferred from Chapter 13.3 in [3], and we give only an outline of proof.

*Proof sketch of Lemma 2.2.* Given a non-empty polytope $P$ and $w \in \mathbb{R}^2$, let $P^w := \{x \in P : \langle x, w\rangle = \max_{z \in P} \langle z, w\rangle\}$ be the set of extreme of points of $P$ in the direction $w$. It is either a vertex or an edge of $P$. For a pair of polytopes we have $(P_1 + P_2)^w = P_1^w + P_2^w$. Every edge of $P_1$ yields an edge of $P_1 + P_2$ with the same direction. Conversely, every edge of $P_1 + P_2$ comes from one of $P_1$ or $P_2$. □

The second operation we use is

$$A \sqcup B := \mathsf{conv}(A \cup B).$$

If $P$ and $Q$ are polytopes then $P \sqcup Q$ is also a polytope.

**Lemma 2.3.** *Let $L : \mathbb{R}^n \to \mathbb{R}^m$ be a linear map. Given polytopes $P, Q \subseteq \mathbb{R}^n$, $L(P + Q) = L(P) + L(Q)$ and $L(P \sqcup Q) = L(P) \sqcup L(Q)$.*

*Proof.* The first equality holds by linearity. The second one can be proved by

$$\begin{aligned} L(P \sqcup Q) &= L(\mathsf{conv}(P \cup Q)) = \mathsf{conv}(L(P \cup Q)) \\ &= \mathsf{conv}(L(P) \cup L(Q)) = L(P) \sqcup L(Q)\,. \end{aligned}$$

□

We next relate the shadow complexity of $P$ with the shadow complexity of its faces. A *face* of a polytope $P$ is the intersection of $P$ with a hyperplane $H$ such that $P$ is completely contained in one of the two closed halfspaces determined by $H$. We stipulate that both $\emptyset$ and $P$ are faces of $P$.

**Lemma 2.4.** *Let $F$ be a face of a polytope $P$. Then $\beta(F) \leq \beta(P)$ and $\sigma(F) \leq 2\sigma(P)$.*

For example, this implies $\beta(\mathsf{DS}_{n_1}) \leq \beta(\mathsf{DS}_{n_2})$ whenever $n_1 \leq n_2$. This reflects the fact that finding a maximum perfect matching is harder for larger graphs.

*Proof.* Without loss of generality, assume that $P \subset \mathbb{R}^n$ is contained in the halfspace $\{x \in \mathbb{R}^n : x_1 \geq 0\}$ and that $F \notin \{\emptyset, P\}$ is the intersection with the hyperplane $x_1 = 0$.

Let $w(t)$ be a line in $\mathbb{R}^n$ so that $\beta(F, w(t)) = \beta(F) = k$ with $w(t)_1 = 0$. Let $t_1 < t_2$ be such that the breakpoints of $\max_{x \in F} \langle x, w(t) \rangle$ are contained in the open interval $(t_1, t_2)$. Let $V := \mathsf{vert}(P) \setminus \mathsf{vert}(F)$. Define

$$\mu_F := \min_{x \in F, t \in [t_1, t_2]} \langle x, w(t) \rangle$$

and

$$\mu_P := \max_{v \in V, t \in [t_1, t_2]} \langle v, w(t) \rangle \ .$$

Take $\lambda \in \mathbb{R}$ sufficiently small so that for every $v \in V$, we have $\lambda v_1 + \mu_P < \mu_F$. Define $\bar{w}(t)$ by changing the first coordinate of $w(t)$ to $\lambda + 0 \cdot t$. This means that

$$\max_{x \in F} \langle x, w(t) \rangle = \max_{x \in P} \langle x, \bar{w}(t) \rangle$$

holds on $[t_1, t_2]$. So, $\beta(P, \bar{w}(t)) = k$ and $\beta(P) \geq \beta(F)$.

If $|\mathsf{vert}(F)| \leq 2$, then $\sigma(F) \leq 2\sigma(P)$ holds trivially. Otherwise, $\sigma(F) \leq 2\sigma(P)$ follows from Theorem 1.4. □

## 2.4 Laurent polynomials

It is convenient to work with *Laurent polynomials* instead of polynomials. In a Laurent polynomial, variables are allowed to have negative integer exponents. The notions of $\mathsf{supp}(f)$ and Newton polytope of $f$ are defined in the obvious manner. A Laurent polynomial over $\mathbb{R}$ is *monotone*, if all of its coefficients are non-negative.

**Lemma 2.5.** *Let $f, g$ be Laurent polynomials over $\mathbb{F}$.*

*(i). Then* $\mathsf{Newt}(fg) = \mathsf{Newt}(f) + \mathsf{Newt}(g)$.

*(ii).* $\mathsf{Newt}(f + g) = \mathsf{Newt}(f) \sqcup \mathsf{Newt}(g)$, *provided* $\mathbb{F} = \mathbb{R}$ *and both $f$ and $g$ are monotone.*

*Proof.* Part $(i)$ can be found in [16] for polynomials; it extends to Laurent polynomials. Part $(ii)$ is straightforward to verify. □

An application is that the shadow complexity of $\mathsf{Newt}(g)$ is at least the shadow complexity of any of its factors.

**Lemma 2.6.** *Let $g$ be a non-zero polynomial (over an arbitrary field). If $f$ divides $g$ then $\sigma(\mathsf{Newt}(f)) \leq \sigma(\mathsf{Newt}(g))$.*

*Proof.* Let $L$ be such that $L(\mathsf{Newt}(f)) \subseteq \mathbb{R}^2$ has $\sigma(\mathsf{Newt}(f))$ vertices. By the assumption, we have $g = fh$ for some non-zero polynomial $h$ and so $\mathsf{Newt}(g) = \mathsf{Newt}(f) + \mathsf{Newt}(h)$ by Lemma 2.5. By Lemma 2.3, we have $L(\mathsf{Newt}(g)) = L(\mathsf{Newt}(f)) + L(\mathsf{Newt}(h))$ and so $|\mathsf{vert}(L(\mathsf{Newt}(g)))| \geq |\mathsf{vert}(L(\mathsf{Newt}(f)))|$ by Lemma 2.2. □

# 3 Examples

We now describe some examples, and analyze the shadow complexity of several natural polytopes. We start with polytopes with small $\sigma$, continue with polytopes with large $\sigma$, and then discuss our favorites, the ones where we do not yet know.

## 3.1 The hypercube

Optimizing over the discrete cube $\{0,1\}^n \subset \mathbb{R}^n$ leads to the polytope $\mathsf{Q}_n = [0,1]^n$. The solid cube $\mathsf{Q}_n$ has $2^n$ vertices, but its shadow complexity is small.

**Proposition 3.1.** *$\sigma(\mathsf{Q}_n) = 2n$ and $\beta(\mathsf{Q}_n) = n$.*

The proposition shows that the factor of two in Theorem 1.4 is necessary.

*Proof.* Let $\ell_i \subseteq \mathbb{R}^n$ be the line segment joining the origin and the $i$-th unit vector for $i \in [n]$. The solid cube $\mathsf{Q}_n$ is the Minkowski sum of $\ell_1, \ldots, \ell_n$. Given $L : \mathbb{R}^n \to \mathbb{R}^2$, the image $L(\mathsf{Q}_n)$ is the Minkowski sum of $L(\ell_1), \ldots, L(\ell_n)$ by Lemma 2.3. Since $|\mathsf{vert}(L(\ell_i))| \leq 2$, Lemma 2.2 gives that $|\mathsf{vert}(L(\mathsf{Q}_n))| \leq 2n$. The bound $\sigma(\mathsf{Q}_n) \geq 2n$ is achieved by the same lemma. It is enough to take $L$ so that $L(\ell_i)$ are not parallel to get $|\mathsf{vert}(L(\mathsf{Q}_n))| = 2n$.

The above and Theorem 1.4 imply that $\beta(\mathsf{Q}_n) \geq n$. It remains to prove $\beta(\mathsf{Q}_n) \leq n$. For every $w \in \mathbb{R}^n$, the maximum $\max_{x \in \mathsf{Q}_n} \langle x, w \rangle$ equals the sum of the positive entries in $w$, or zero if all entries are non-positive. A breakpoint of $\max_{x \in \mathsf{Q}_n} \langle x, w(t) \rangle$ can therefore occur only when some coordinate of $w(t)$ changes sign. A linear function can change sign at most once and there are $n$ linear functions. □

**Remark 3.2.** *The solid cube $\mathsf{Q}_n$ is not greedy-like as defined above. This is because in the optimization algorithm, we must distinguish which entries are non-negative. Shifting all coordinates of $w$ by $\lambda$ does not change their order but may change where the maximum is achieved.*

## 3.2 Permutahedra

Given $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$, let

$$P(z) := \mathsf{conv}\{(z_{\pi(1)}, \ldots, z_{\pi(n)}) : \pi \in S_n\},$$

where $S_n$ is the family of permutations of $[n]$. The *permutahedron* is usually defined using the point $z = (0, 1, \ldots, n-1)$. However, we do not insist $z$ to have distinct coordinates. Setting $z$ to be a zero-one vector with $k$ ones, $P(z)$ becomes the convex hull of Boolean vectors of Hamming weight $k$. For every $z$, the polytope $P(z)$ is a linear projection of $\mathsf{DS}_n$. The polytope $P(z)$ typically has $n!$ vertices, but its shadow complexity is always small.

**Proposition 3.3.** *For every $z \in \mathbb{R}^n$, $\sigma(P(z)) \leq n(n-1)$. The bound is attained for $z = (0, 1, \ldots, n-1)$.*

*Proof.* Let $z := (0, 1, \ldots, n-1)$. Let $e_i \in \mathbb{R}^n$ be the $i$-th unit vector. Let $\ell_{i,j}$ be the line segment joining $e_i$ and $e_j$ for $i \neq j$. We claim that the polytope $P(z)$ can be written as the following Mikowski sum

$$P(z) = \bigoplus_{i<j} \ell_{i,j}. \tag{1}$$

Indeed, let $X$ be the $n \times n$ matrix such that $X_{i,j} = x_i^{z_j}$. Observe that

$$P(z) = \mathsf{Newt}(\mathsf{det}(X))\,.$$

The matrix $X$ is a Vandermonde matrix whose determinant, over any field, factorizes as $\mathsf{det}(X) = \prod_{i<j}(x_j - x_i)$. Lemma 2.5 implies (1).

Now, given $L : \mathbb{R}^n \to \mathbb{R}^2$, we thus have $L(P(z)) = \bigoplus_{i<j} L(\ell_{i,j})$. By Lemma 2.2, if we choose $L$ so that the lines $L(\ell_{i,j})$ are non-parallel, the number of vertices of $L(P)$ is $2 \cdot \binom{n}{2} = n(n-1)$.

The general upper bound is an application of Lemma 1.5. We claim that $P(z)$ is greedy-like. Permuting the entries of $z$ does not changes $\sigma$. So, we can assume that $z_1 \leq z_2 \leq \ldots z_n$. Given $w \in \mathbb{R}^n$,

$$\max_{x \in P(z)} \langle x, w \rangle = \max_{\pi \in S_n} \langle z, w_\pi \rangle\,,$$

where $w_\pi := (w_{\pi(1)}, \ldots, w_{\pi(n)})$. The maximum is achieved iff $w_{\pi(1)} \leq w_{\pi(2)} \cdots \leq w_{\pi(n)}$. This means that $\mathrm{Opt}_w(P(z)) = \mathrm{Opt}_{w'}(P(z))$ whenever $w$ and $w'$ are order-equivalent. □

**Remark 3.4.** *Here we provide an additional algebraic proof. Consider* $z = (z_1, \ldots, z_n)$ *with* $z_i = 2^{i-1}$. *The matrix* $X$ *defined by* $X_{i,j} = x_i^{z_j}$ *is a* Moore matrix *[29]. Over* $\mathbb{F} = GF(2)$, *the polynomial* $\mathsf{det}(X(z))$ *factorizes as*

$$\mathsf{det}(X(z)) = \prod_{A \subseteq [n]} \sum_{i \in A} x_i \,.$$

*The number of factors is exponential but we can still get a quadratic upper bound. We have* $P(z) = \bigoplus_{A \subseteq [n]} R_A$ *where* $R_A = \mathsf{conv}\{e_i : i \in A\}$. *Given a projection* $L : \mathbb{R}^n \to \mathbb{R}^2$, *we have* $L(P(z)) = \bigoplus_{A \subseteq [n]} L(R_A)$. *The polytopes* $L(R_A)$ *contain at most* $\binom{n}{2}$ *non-parallel edges and hence* $L(P(z))$ *has again at most* $n(n-1)$ *vertices.*

## 3.3 Spanning trees

Every $\alpha \in \{0,1\}^{\binom{n}{2}}$ can be interpreted as an undirected graph on $n$ vertices. The polytope $\mathsf{TREE}_n$ is defined as the convex hull of spanning trees of the complete $n$-vertex graph.

**Proposition 3.5.** $\sigma(\mathsf{TREE}_n) \leq n^4$.

*Proof.* By Lemma 1.5, it is enough to show that $P = \mathsf{TREE}_n$ is greedy-like. Indeed, Kruskal's algorithm for finding a minimum weight spanning tree takes into account only the ordering of weights on the edges. That is, if $w, w'$ are order-equivalent and $\mathrm{Opt}_P(w)$ is a singleton then $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. Hence $\mathsf{TREE}_n$ is greedy-like by Lemma 2.1. . □

**Remark 3.6.** *This is interesting when contrasted with algebraic complexity. Consider the unique polynomial* $\mathsf{Tree}_n$ *with zero-one coefficients so that* $\mathsf{Newt}(\mathsf{Tree}_n) = \mathsf{TREE}_n$. *It is a homogeneous multilinear polynomial of degree* $n-1$. *Proposition 3.5 shows that the shadow complexity of its Newton polytope is polynomial. On the other hand, Jerrum and Snir showed that* $\mathsf{Tree}_n$ *requires exponential monotone arithmetic circuit [21]. They also pointed out that it has a non-monotone circuit of polynomial size. More surprisingly,* $\mathsf{Tree}_n$ *has a monotone circuit with division of polynomial size [14].*

## 3.4 Cliques

The *correlation polytope* $\mathsf{COR}_n \subseteq \mathbb{R}^{n\times n}$ is the convex hull of all symmetric rank-one Boolean matrices:

$$\mathsf{COR}_n = \mathsf{conv}\{bb^t : b \in \{0,1\}^n\}\,.$$

**Proposition 3.7.** $\sigma(\mathsf{COR}_n) = 2^n$.

*Proof.* Let $e_{i,j}$ be the $n \times n$ matrix whose $(i,j)$ entry is one and every other entry is zero. The vertices of $\mathsf{COR}_n$ are of the form $v_A = \sum_{i,j\in A} e_{i,j}$ with $A \subseteq [n]$. Define

$$L(e_{i,j}) := \begin{cases} (2^i, 2^{2i}) & i = j, \\ (0, 2^{i+j}) & i \neq j, \end{cases}$$

and extend it linearly to $\mathbb{R}^{n\times n}$. Setting $n_A := \sum_{i\in A} 2^i$, this guarantees

$$L(v_A) = (\sum_{i\in A} 2^i, \sum_{i,j\in[n]} 2^{i+j}) = (n_A, n_A^2).$$

These $2^n$ points are convexly independent. □

**Remark 3.8.** *The polytope* $\mathsf{COR}_n$ *lives in dimension* $N = n^2$*, and so* $\sigma(\mathsf{COR}) = 2^{\sqrt{N}}$*. The polytope* $\mathsf{ART}_n \subseteq \mathbb{R}^{3n}$*, which we define next, has truly exponential shadow complexity. It is defined as the convex hull of*

$$\Big\{(a_0,\ldots,a_{n-1},b_0,\ldots,b_{2n-1}) \in \{0,1\}^{3n} : \sum_{i=0}^{2n-1} b_i 2^i = \Big(\sum_{i=0}^{n-1} a_i 2^i\Big)^2\Big\}.$$

*In words, b is the binary representation of the square of the number represented by a. It follows that* $\sigma(\mathsf{ART}_n) = 2^n$.

**Remark 3.9.** *The polynomial that corresponds to* $\mathsf{COR}_n$ *is*

$$\mathsf{Clique}_n = \sum_{A\subseteq[n]} \prod_{i,j\in[n]} x_{i,j}\,.$$

*It has* $n^2$ *variables and* $\mathsf{Newt}(\mathsf{Clique}_n) = \mathsf{COR}_n$*. We can interpret the polynomial as counting cliques of all sizes in a directed graph with loops, hence the name.*

## 3.5 More graph-based polytopes

Consider a layered directed graph $G_n$ as follows. The vertex-set of $G_n$ is partitioned into layers $V_0, \dots V_n$. The first and the last layer consist of a single vertex $s$ and $t$. Every other layer has $n$ vertices. The edges are all pairs from $V_i \times V_{i+1}$ directed from layer $i$ to $i+1$. Overall, $G_n$ has $n(n-1)+2$ vertices and $N := (n-2)n^2 + 2n$ edges. Let $\mathsf{CONN}_n \subseteq \mathbb{R}^N$ be the convex hull of incidence vectors of directed paths from $s$ to $t$ in $G_n$. The following proposition can be found in [9, 32].

**Proposition 3.10.** $\sigma(\mathsf{CONN}_n) = 2^{\Theta(\log^2 n)}$.

We now deduce the best bound we are aware of for the Birkhoff polytope.

**Proposition 3.11.** $2^{\Omega(\log^2 n)} \le \sigma(\mathsf{DS}_n) \le 2^{O(n)}$.

*Proof.* As pointed by Mulmuley and Shah in [32], the lower bound for $\mathsf{CONN}_n$ translates to $\mathsf{DS}_n$. For the upper bound, we claim that

$$\sigma(\mathsf{DS}_{2n}) \le 2\binom{2n}{n}\sigma(\mathsf{DS}_n)\,. \tag{2}$$

This indeed implies $\sigma(\mathsf{DS}_n) \le 2^{O(n)}$.

Let us prove (2). Given $A \subseteq [2n]$ with $|A| = n$, let $\Pi_A$ be the set of permutation matrices which, when viewed as a permutation on $[2n]$, map $\{1, \dots, n\}$ to $A$. The set of all $2n \times 2n$ permutation matrices is the union of all $\Pi_A$ with $|A| = n$. Hence,

$$\mathsf{DS}_{2n} = \mathsf{conv}\Big(\bigcup_{A:\,|A|=n} \Pi_A\Big).$$

We can view $\mathsf{conv}(\Pi_A)$ as the Minkowski sum of two copies of $\mathsf{DS}_n$ embedded into $\mathbb{R}^{2n\times 2n}$. Given $L : \mathbb{R}^{2n\times 2n} \to \mathbb{R}^2$ this gives, by Lemma 2.2, $|\mathsf{vert}(L(\mathsf{conv}(\Pi(A))))| \le 2|\mathsf{vert}(L(\mathsf{DS}_n))|$. The bound in (2) follows. □

**Remark 3.12.** *The upper bound on* $\mathsf{DS}_n$ *is more exactly of the form* $2^{(2-o(1))n}$. *In the proof, we implicitly construct a monotone arithmetic formula for* $\mathsf{perm}_n$ *of this size. This matches the lower bound from [39]. Curiously,* $\mathsf{perm}_n$ *has a monotone circuit of size* $O(n2^n)$ *and a (non-monotone) formula of size* $O(n^2 2^n)$.

**Remark 3.13.** *Let* $\mathsf{Mat}_n := (X_0 \cdot X_1 \cdots X_n)_{1,1}$*, where* $X_0, \ldots, X_n$ *are* $n \times n$ *matrices whose entries are distinct variables. Then* $\mathsf{Newt}(\mathsf{Mat}_n) = \mathsf{CONN}_n$.

**Remark 3.14.** *The* perfect matching polytope $\mathsf{MATCH}_n$ *is the the convex hull of incidence vectors of perfect matchings in the complete (non-bipartite) graph on* $2n$ *vertices. A similar argument to the proof of Proposition 3.11 gives*

$$\sigma(\mathsf{DS}_n) \leq \sigma(\mathsf{MATCH}_n) \leq \binom{2n}{n}\sigma(\mathsf{DS}_n) \leq 2^{O(n)}\,.$$

### 3.6 General polytopes

*Proof of Proposition 1.6.* The left inequality follows from Remark 3.8. For $A \subseteq [n]$, let $e_A \in \{0,1\}^n$ be the characteristic vector of the set $A$. Let $\ell_A$ be the line segment joining $e_A$ and $e_\emptyset = 0 \in \mathbb{R}^n$. Suppose towards a contradiction that we have $S \subseteq \{0,1\}^n$ and $L : \mathbb{R}^n \to \mathbb{R}^2$ such that $L(\mathsf{conv}(S))$ has at least $2^{c_2 n}$ vertices. Removing elements of $S$ if necessary, we can assume that $L$ maps elements of $S$ to distinct vertices of $L(\mathsf{conv}(S))$. By [34], there is $c_2 < 1$ so that there exist disjoint $A_0, A_1, A_2, A_3$ with $A_1, A_2, A_3$ non-empty so that $S$ contains the set

$$T := \{e_{A_0} + \sum_{i=1}^{3} \alpha_i e_{A_i} : \alpha_1, \alpha_2, \alpha_3 \in \{0,1\}\}\,.$$

Because $\mathsf{conv}(T) = e_{A_0} + \bigoplus_{i=1}^{3} \ell_{A_i}$, we can conclude that $L(\mathsf{conv}(T))$ has at most six vertices. But $|T| = 8$, contradicting the assumption that $L$ maps $S$ to independent points. □

# 4 Projections

We now discuss some connection between algebraic projections of polynomials and linear projections of Newton polytopes. The results here shall also be used later on.

A *high power* projection (h.p.-projection for short) is a homomorphism

$$\pi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y_1, \ldots, y_m, y_1^{-1}, \ldots y_m^{-1}]$$

such that $\pi(x_i) = a_i y^{\alpha_i}$ for every $x_i$, where $a_i \in \mathbb{F}$ and $\alpha_i \in \mathbb{Z}^m$, and for every $f \in \mathbb{F}[x_1, \ldots, x_n]$,

$$\pi(f(x_1, \ldots, x_n)) = f(\pi(x_1), \ldots, \pi(x_n)).$$

The constants $a_i$ are called the *coefficients* of $\pi$ and $\alpha_i$ its *exponents*. If $\mathbb{F} = \mathbb{R}$ and $\pi$ has non-negative coefficients, then $\pi$ is called *monotone*.

An h.p.-projection $\pi$ induces a linear map $L_\pi : \mathbb{R}^n \to \mathbb{R}^m$ by setting $L_\pi(e_i) = \alpha_i$ and extending it linearly to $\mathbb{R}^n$. It follows that $\mathsf{supp}(\pi(f)) \subseteq L_\pi(\mathsf{supp}(f))$. The inclusion may be strict, as some monomials of $f$ can cancel out in the projection. If no cancellations occur, we indeed have $\mathsf{Newt}(\pi(f)) = L_\pi(\mathsf{Newt}(f))$. This is satisfied, e.g., if $f$ is monotone and the coefficients of $\pi$ are positive, or if the coefficients are algebraically independent.

In the other direction, take $L : \mathbb{R}^n \to \mathbb{R}^m$ a linear map defined by $m \times n$ matrix with integer coefficients. Consider a h.p.-projection $\pi_L$ of the form $\pi(x_i) = a_i x_i^{L(e_i)}$. If we choose the coefficients $a_i$ to be sufficiently independent, we again obtain $L(\mathsf{Newt}(f)) = \mathsf{Newt}(\pi_L(f))$.

The following we do not really need, but it puts things into perspective. A similar fact has been noted by Grochow [18].

**Proposition 4.1.** *Let $f$ be a monotone polynomial. Assume that a Laurent polynomial $g$ is a monotone h.p.-projection of $f$. Then $\mathsf{Newt}(g)$ is a linear projection of some face of $\mathsf{Newt}(f)$. Hence $\sigma(\mathsf{Newt}(g)) \leq 2\sigma(\mathsf{Newt}(f))$.*

*Proof.* Assume $g = \pi(f)$ with $\pi$ an h.p.-projection. Let $A \subseteq [n]$ be the set of $i \in [n]$ with $a_i = 0$. Let $f^*$ be the polynomial obtained by substituting 0 for $x_i$ for every $i \in A$. The polytope $\mathsf{Newt}(f^*)$ is a face of $\mathsf{Newt}(f)$. Indeed, since $f$ has non-negative exponents, $\mathsf{Newt}(f^*) = \mathsf{Newt}(f) \cap H$ where $H$ is the hyperplane defined by $\sum_{i \in A} z_i = 0$, and $\mathsf{Newt}(f)$ lies in the halfspace $\sum_{i \in A} z_i \geq 0$.

We can now write $\pi(f) = \pi^*(f^*)$ where $\pi^*$ has positive coefficients. This means that $\mathsf{supp}(\pi(f)) = L_{\pi^*}(\mathsf{supp}(f^*))$ and hence $\mathsf{Newt}(\pi(f)) = L_{\pi^*}(\mathsf{Newt}(f^*))$. The bound on $\sigma$ follows from Lemma 2.4 □

The following we do need:

**Lemma 4.2.** *Let $f$ be a polynomial over an infinite field $\mathbb{F}$. Assume that $\sigma(\mathsf{Newt}(f)) = k$. Then there exists a bivariate Laurent polynomial $g \in \mathbb{F}(y_1, y_2, y_1^{-1}, y_2^{-1})$ which is an h.p.-projection of $f$ so that $\mathsf{Newt}(g)$ has $k$ vertices. Moreover, if $f$ is a homogeneous polynomial then $g$ is a polynomial. If $\mathbb{F} = \mathbb{R}$, then the coefficients in the projection can be assumed positive.*

*Proof.* Let $L(z) = Az$ with $A \in \mathbb{R}^{2 \times n}$ be a linear map so that

$$|\mathsf{vert}(L(\mathsf{Newt}(f)))| = k.$$

We can assume that the entries of $A$ are rational, because a small perturbation of $A$ cannot decrease $|\mathsf{vert}(L(\mathsf{Newt}(f)))|$. Now, we can assume that the entries of $A$ are integers, because we can multiply $A$ by a suitable integer.

Moreover, when $f$ is homogeneous of degree $d$, increasing all entries of $A$ by $\lambda$ corresponds to shifting $L(\mathsf{Newt}(f))$ by $(\lambda d, \lambda d)$, which does not change the number of vertices. Hence, in this case, $A$ can be taken with non-negative integer entries.

Let us now consider a h.p.-projection $\pi$ with $\pi(x_i) = a_i y^{L(e_i)}$. It follows that $\mathsf{supp}(\pi(f)) \subseteq L(\mathsf{supp}(f))$. Now, we claim that we can choose the coefficients $a_i$ so that equality holds. This can be seen as follows. Given $\alpha \in \mathsf{supp}(f)$, the coefficient of $y^{L(\alpha)}$ in $\pi(f)$ is a non-zero polynomial $h_\alpha$ in the coefficients of $\pi$. Hence, if $\mathbb{F}$ is infinite, there exist non-zero coefficients for which $h_\alpha$ is non zero for every $\alpha \in \mathsf{supp}(f)$. If $\mathbb{F} = \mathbb{R}$, they can be taken positive. □

**Remark 4.3.** *We emphasize the difference between linear projections of polytopes and algebraic projections of polynomials. Since the permanent polynomial is VNP-complete,* $\mathsf{Clique}_n$ *from Remark 3.9 is a projection (in the common sense) of* $\mathsf{perm}_m$ *with* $m$ *polynomial in* $n$*. However,* $\mathsf{Newt}(\mathsf{Clique}_n)$ *is not a linear projection of* $\mathsf{Newt}(\mathsf{perm}_m)$*, neither of any of its faces, unless* $m$ *is exponentially large [18]. The idea is that* $\mathsf{DS}_m$ *has* $O(m^2)$ *facets whereas* $\mathsf{Newt}(\mathsf{Clique}_n)$ *is not a projection of any polytope with few facets. It follows that* $\mathsf{Clique}_n$ *is not a monotone projection of* $\mathsf{perm}_m$*.*

# 5 Monotone computation

As the standard model of computation of polynomials we take the *arithmetic circuit* model. It is a (finite) directed acyclic graph whose every node has in-degree zero or two. Nodes of in-degree zero (input nodes) are labelled with a constant or a variable. Nodes of in-degree two are labelled with operations $+$ or $\times$. Every node of the circuit computes a polynomial in $\mathbb{F}$ in the natural way. As the *size* of the circuit, we take the number of nodes. A circuit is called a *formula* if its underlying graph is a tree. For more background and motivation, see [40] and references within.

Our focus is mainly on monotone computation. A polynomial over $\mathbb{R}$ is *monotone* if all of its coefficients are non-negative. Similarly, a *monotone arithmetic circuit* can use only non-negative constants.

## 5.1 Optimization problems

We start with a somewhat surprising connection between monotone computation and Newton polytopes. A monotone circuit over $\mathbb{R}$ computing $f$ can be interpreted as a computation over the semi-ring $M = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$. That is, replace "+" by "min", replace "×" by "+", replace "0" by "∞", and replace every "$a > 0$" by "0". A circuit with operations from $M$ has also been called a *tropical* circuit [22]. The resulting circuit computes the function $f^* : \mathbb{R}^n \to \mathbb{R}$ which turns out to be precisely

$$f^*(w) = \min_{x \in \mathsf{Newt}(f)} \langle x, w \rangle .$$

For example, any monotone circuit for the permanent polynomial can also be viewed as a tropical circuit for the minimum weight perfect matching in a bipartite graph. Computations over general semi-rings have been considered in [21, 22], where the reader can find more details.

## 5.2 Shadows and monotone computations

We explore some connections between the structure of the Newton polytope of $f$ and monotone arithmetic computations. We prove that shadow complexity allows to prove lower bounds on monotone complexity (Theorems 1.1 and 1.2). We also show that in general Theorem 1.1 does not hold for circuits instead of formulas and so the assumption of transparency in Theorem 1.2 cannot be removed.

**Theorem 5.1.** *For every $n$, there exists a monotone bivariate polynomial $f_n$ such that $f_n$ has a monotone arithmetic circuit of size $O(n)$ and $\mathsf{Newt}(f)$ has $2^n$ vertices.*

Theorem 5.1 is proved in Section 8. The construction is reminiscent of that in [4] of a univariate polynomial with circuit of size $s$ and $2^{\Omega(s)}$ real roots (cf. Chapter 12 of [7]). A weaker bound can also be deduced as follows:

**Remark 5.2.** *Recall the polynomial $\mathsf{Mat}_n$ from Remark 3.13. Then $\mathsf{Mat}_n$ has a monotone circuit of size $O(n^4)$ whereas $\sigma(\mathsf{Newt}(\mathsf{Mat}_n)) = 2^{\Omega(\log^2 n)}$.*

**Remark 5.3.** *When a monotone arithmetic formula is interpreted as a tropical formula (cf. Section 5.1), it becomes an instance of parallel computation in*

*the PRAM model without bit operations of Mulmuley [31]. Hence Theorem 1.1 can be seen as special case*[3] *of Theorem 3.3 from [31].*

## 5.3 Monotone formulas

Here we show that shadow complexity allows to lower bound monotone formula complexity.

A *high powered* circuit (h.p.-circuit for short) is an arithmetic circuit in which every input node is labelled by a term $ax_1^{k_1} \cdots x_n^{k_n}$ with $a \in \mathbb{F}$ and $k_1, \ldots, k_n \in \mathbb{Z}$. The size of the circuit is the number of its nodes.

In other words, we have given the circuit a power to compute every term $ax^\alpha$ at a unit cost. This is especially important in the case of h.p.-formula. An arithmetic formula of size $s$ can compute a polynomial of degree at most $s$, whereas there is no such restriction in an h.p.-formula. Furthermore, we have allowed the variables to have negative exponents and hence an h.p.-circuit computes a Laurent polynomial instead of a polynomial. But this is only a cosmetic detail.

**Theorem 5.4.** *Let $f$ be a monotone bivariate Laurent polynomial such that $\mathsf{Newt}(f)$ has $k$ vertices. Then every monotone h.p.-formula computing $f$ has at least $k$ leaves.*

*Proof.* Straightforward induction using Lemma 2.5 and 2.2. □

We can now prove that every monotone formula computing $f$ contains at least $\sigma(\mathsf{Newt}(f))$ leaves.

*Proof of Theorem 1.1.* By Lemma 4.2 there exists a bivariate $g$ which is a monotone h.p.-projection of $f$ so that $\mathsf{Newt}(g)$ has $k$ vertices. The projection also transforms a monotone formula for $f$ to a monotone h.p.-formula for $g$. □

## 5.4 Lower bounds from extension complexity

As mentioned in Section 1.2, one can obtain monotone formula lower bounds also from extensions complexity of Newton polytopes. The main ingredient is the following lemma.

[3]This is not a "black box" reduction. Mulmuley's result has an additional parameter representing bit size of the input, whereas we have no such thing.

**Lemma 5.5.** *For polytopes $P, Q \subseteq \mathbb{R}^n$ we have*

$$\mathrm{xc}(P+Q) \leq \mathrm{xc}(P) + \mathrm{xc}(Q) \quad \textit{and} \quad \mathrm{xc}(P \sqcup Q) \leq \mathrm{xc}(P) + \mathrm{xc}(Q) + 2.$$

*Proof.* The first inequality is rather obvious. The second follows from a theorem of Balas [2], see also [11]. □

The lower bound is now proved by a straightforward induction.

**Theorem 5.6.** *Assume that $f$ has a monotone formula of size $s$. Then* $\mathrm{xc}(\mathsf{Newt}(f)) \leq O(s)$.

**Remark 5.7.** *The Pfaffian $\mathsf{Pf}_n$ is the polynomial so that $\mathsf{Pf}_n^2 = \mathsf{det}(X)$, where $X$ is the $2n \times 2n$ antisymmetric matrix with $X_{i,i} = 0$ and $X_{i,j} = -X_{j,i} = x_{i,j}$ if $i < j$. The Pfaffian has an arithmetic circuit of size polynomial in $n$, and a formula of size $2^{O(\log^2 n)}$; see [42]. The Newton polytope $\mathsf{Newt}(\mathsf{Pf}_n)$ is the perfect matching polytope $\mathsf{MATCH}_n$, as described in Remark 3.14. By a result of Rothvoss [38], $\mathsf{MATCH}_n$ has extension complexity $2^{\Omega(n)}$.*

## 5.5 Monotone circuits

We move to proving the circuit lower bound stated in Theorem 1.2. We first observe that Minkowski sum typically can not avoid convex independence.

**Lemma 5.8.** *Let $A, B \subseteq \mathbb{R}^2$ be non-empty sets such that $A+B$ is a convexly independent set with $|A| \geq |B|$. Then either $|A| \leq 2$ or $|B| \leq 1$.*

*Proof.* For the sake of contradiction, assume that $A+B$ is convexly independent, $|A| \geq 3$ and $|B| \geq 2$. By Lemma 2.2, the convex hull of $A+B$ has at most $|A|+|B|$ vertices. By the size assumption, there exist $a_1 \neq a_2 \in A$ and $b_1 \neq b_2 \in B$ with $a_1 + b_1 = a_2 + b_2$. The point $a_1 + b_1$ is the average of $a_1 + b_2$ and $a_2 + b_1$ and it is distinct from them, a contradiction. □

**Theorem 5.9.** *Let $f$ be a monotone bivariate Laurent polynomial such that $\mathsf{supp}(f)$ is convexly independent and $|\mathsf{supp}(f)| = k$. Then $f$ requires monotone h.p.-circuit with $k/4$ gates.*

Theorem 5.9 implies Theorem 1.2 via Lemma 4.2.

*Proof.* The lower bound is proved using the following "progress" measure. Given $A \subseteq \mathbb{R}^2$ and $\epsilon \in \{0,1\}$, let $A^\epsilon := A$ if $\epsilon = 1$ and $A^\epsilon := \emptyset$ if $\epsilon = 0$. Given $v \in \mathbb{R}^2$, let $v + A := \{v\} + A$. Let $\mathcal{A}$ be a finite set of finite subsets of $\mathbb{R}^2$. For functions $\epsilon : \mathcal{A} \to \{0,1\}$ and $v : \mathcal{A} \to \mathbb{R}^2$, let

$$\mathcal{A}_{\epsilon,v} = \bigcup_{A \in \mathcal{A}} (v(A) + A)^{\epsilon(A)}.$$

Let

$$\mu(\mathcal{A}) = \max_{\epsilon,v} \{|\mathcal{A}_{\epsilon,v}| : \ \mathcal{A}_{\epsilon,v} \text{ is convexly independent}\}.$$

**Claim.** *Let $\mathcal{A}' = \mathcal{A} \cup \{B\}$ and $A_1, A_2 \in \mathcal{A}$. Then*

$$\mu(\mathcal{A}') \leq \mu(\mathcal{A}) + |B|\,, \tag{3}$$
$$\mu(\mathcal{A}') \leq \mu(\mathcal{A}) + 2\,, \ \textit{if } B = u + A_1 \textit{ for some } u \in \mathbb{R}^2\,, \tag{4}$$
$$\mu(\mathcal{A}') \leq \mu(\mathcal{A}) + 4\,, \ \textit{if } B = A_1 \cup A_2\,, \tag{5}$$
$$\mu(\mathcal{A}') \leq \mu(A) + 4\,, \ \textit{if } B = A_1 + A_2\,. \tag{6}$$

*Proof of Claim.* Inequality (3) is straightforward.

To prove (4), suppose that $\epsilon, v$ are such that $\mathcal{A}'_{\epsilon,v}$ is convexly independent. Suppose $\epsilon(A_1) = \epsilon(B) = 1$ and $v(A_1) + A_1 \neq v(B) + B$; otherwise we have $|\mathcal{A}'_{\epsilon,v}| \leq \mu(\mathcal{A})$. Then $(v(A_1) + A_1) \cup (v(B) + B) = \{v(A_1), v(B) + u\} + A_1$ is convexly independent. Since $|\{v(A_1), v(B) + u\}| = 2$, by Lemma 5.8, $A_1$ has size at most 2. This means $\mu(\mathcal{A}') \leq \mu(\mathcal{A}) + 2$ by (3).

For (5), observe that $\mu(\mathcal{A}') \leq \mu(\mathcal{A} \cup \{u_1 + A_1, u_2 + A_2\})$ whenever $u_1, u_2 \neq 0$ are distinct and apply (4) twice.

Finally, we prove (6). If $B = A_1 + A_2$ is not convexly dependent, it contributes nothing to $\mu$. Assume that $B$ is convexly independent and $|A_1| \geq |A_2| > 0$. By Lemma 5.8, either $|A_1 + A_2| \leq 4$ or $|A_2| = 1$. In the former case, $\mu(\mathcal{A}') \leq \mu(\mathcal{A}) + 4$ by (3). In the latter, $A_2 = \{u\}$ for some $u$ and $\mathcal{A}' = \mathcal{A} \cup \{u + A_1\}$ and we can apply (4). □

Let us call a h.p.-circuit *transparent*, if every gate in the circuit computes a polynomial with convexly independent support. Given a circuit $\Psi$ and a node $u$, let $\mathsf{supp}(u)$ be the support of the Laurent polynomial computed by $u$. Let $\mathcal{A}_\Psi$ be the set $\{\mathsf{supp}(u) : u \in \Psi\}$.

Using the Claim, we can show that whenever a transparent and monotone $\Psi$ has $s$ gates then $\mu(\mathcal{A}_\Psi) \leq 4s$. The proof is by induction. The induction base $s = 1$ trivially holds. It remains to verify the induction step. Let $u$ be an

output gate of $\Psi$. If $u$ is also an input gate, apply (3). If $u = u_1 + u_2$ then $\mathsf{supp}(u) = \mathsf{supp}(u_1) \cup \sup(u_2)$ and (5) completes the proof. If $u = u_1 \times u_2$ then $\mathsf{supp}(u) = \mathsf{supp}(u_1) + \sup(u_2)$ and (6) completes the proof.

Finally, consider a monotone circuit $\Psi$ for $f$ of minimal size $s$. No gate in the circuit computes the zero polynomial (unless $f$ itself the zero polynomial). The circuit is transparent because a monotone computation does not cancel monomials unless multiplying by zero, and because $+, \times$ can not "undo" convex independence. This means that $\mu(\mathcal{A}_\Psi) \leq 4s$. On the other hand, since $\mathsf{supp}(f)$ consists of $k$ convexly independent points, we have $\mu(\mathcal{A}_\Psi) \geq |\mathsf{supp}(f)| = k$. □

Other illustrative consequences are the following:

**Corollary 5.10.** $\sum_{k=0}^{n} x^k y^{k^2}$ *requires monotone h.p.-arithmetic circuit of size* $\Omega(n)$.

Recall the $\mathsf{Clique}_n$ polynomial from Remark 3.9 and the polytope $\mathsf{ART}_n$ from Remark 3.8. Let $\mathsf{Art}_n$ be the unique polynomial with zero-one coefficients so that $\mathsf{Newt}(\mathsf{Art}_n) = \mathsf{ART}_n$.

**Corollary 5.11.** *Both* $\mathsf{Clique}_n$ *and* $\mathsf{Art}_n$ *require monotone arithmetic circuits of size* $\Omega(2^n)$.

*Proof.* Proposition 3.7 and Remark 3.8 show that $\mathsf{Clique}_n$ and $\mathsf{Art}_n$ are transparent with shadow complexity $2^n$. □

## 5.6 Generalizations

The results of this section can be strengthened in several ways. First, one could extend the notion of monotone computation to any field. A monotone circuit would be such that for every sum gate $f_1 + f_2$, no monomial can vanish[4]: $\mathsf{supp}(f_1 + f_2) = \mathsf{supp}(f_1) \cup \mathsf{supp}(f_2)$. Then Theorem 1.1 goes through.

Second, one may consider circuits with high-power gates. This would be an arithmetic circuit which, apart from the $+, \times$ gates, can use also unary gates of the form $(\ )^k$ which raises its input to a power of $k \in \mathbb{N}$. A similar notion has appeared in the context of additive complexity of a polynomial and counting real roots of univariate polynomials (see Section 12.3 of [7] and

[4]Monomials can however vanish on a product as in $(x + y)(x - y) = x^2 - y^2$.

references within). Our lower bounds hold also in this setting. This is because $\mathsf{Newt}(f^k)$ with $k > 0$ is merely a scaling of $\mathsf{Newt}(f)$.

Finally, our results extend to other semi-rings as well. For definitions of polynomials over semi-rings and their computations see, e.g., [21, 22]. Let $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ be the Boolean semi-ring.

**Proposition 5.12.** *Theorems 1.1 and 1.2 hold also over* $\mathbb{B}$.

*Proof.* Given a circuit over $\mathbb{B}$ computing $f$, we can interpret it as a computation over $\mathbb{R}$ by replacing $\wedge$ by $\times$ and replacing $\vee$ by $+$. The circuit then computes a polynomial $f^*$ over $\mathbb{R}$ with $\mathsf{supp}(f^*) = \mathsf{supp}(f)$. Since the two theorems take into account only $\mathsf{supp}(f^*)$, they hold over $\mathbb{B}$ as well. □

# 6 Divisions

The model of monotone circuits can be extended to include *division gates*. We may allow the circuit to use an extra gate computing $f/g$. A monotone circuit with divisions can compute a non-monotone polynomial; e.g., $x^2 - x + 1 = \frac{x^3+1}{x+1}$.

Monotone circuits with divisions were extensively studied by Fomin et al. [14]. They proved, among other nice things, a separation between monotone circuits and monotone circuits with division. The *Spanning Tree* polynomial (see Section 3.3) has a polynomial size monotone circuit with divisions but requires an exponential size monotone circuit by [21]. This is in sharp contrast with the result of Strassen that division gates cannot help in the general arithmetic setting[5].

Super-polynomial lower bounds on monotone circuits with division computing a *monotone* polynomial $f$ are not known. In [14], strong lower bounds were given for a non-monotone $f$. The non-monotonicity, however, is more than a subtlety. Their proof hinges on the fact that $(x-1)^2 + 2^{-2^{n+1}}$ can be written as $f/g$ with $f, g$ monotone, whereas they require degrees $2^{2^n}$.

This question can be phrased more generally. If $f$ can be computed by a monotone circuit with divisions of size $s$ then we can find non-zero $h$ and $g$ with monotone circuit size $O(s)$ such that $fh = g$. In other words, $f$ divides $g$.

[5] This holds for polynomials of low degree; the spanning tree polynomial indeed has this property.

**Problem 2.** *Find an explicit monotone $f_n$ (with polynomially many variables and of a polynomial degree) such that $g$ requires superpolynomial monotone circuit whenever $g \neq 0$ and $f_n$ divides $g$.*

A seminal result of Kaltofen [23], see also [6], states the following: if $f$ of degree $d$ can be computed by a circuit of size $s$, we can compute each factor of $f$ by a (non-monotone) circuit of size polynomial in $s$ and $d$. We believe that in fact $d$ can be replaced by the degree of the factor. This means that in the non-monotone setting, Problem 2 is equivalent to proving lower bound on $f_n$.

Shadow complexity gives a partial solution to Problem 2.

**Theorem 6.1.** *Let $f$ be a (not necessarily monotone) real polynomial such that $\sigma(\mathsf{Newt}(f)) = k$. Assume that $g \neq 0$ is a monotone polynomial such that $f$ divides $g$. Then every monotone formula computing $g$ contains at least $k$ leaves.*

*Proof.* Lemma 2.6 gives $\sigma(\mathsf{Newt}(g)) \geq \sigma(\mathsf{Newt}(f))$, and we can apply Theorem 1.1. □

Shadow complexity also provides lower bounds on monotone *circuit* complexity provided the degree is not too large. This is another partial solution to Problem 2.

**Proposition 6.2.** *Let $f$ be either $\mathsf{Clique}_n$ or $\mathsf{Art}_n$. Let $g \neq 0$ be a monotone polynomial such that $f$ divides $g$.*

*(i). $g$ requires monotone formula with $2^n$ leaves.*

*(ii). If $g$ has degree $d \leq 2^{o(n^{\frac{1}{2}})}$, then $g$ requires monotone circuit of size $2^{\Omega(n^{\frac{1}{2}})}$.*

*(iii). If $g = \alpha f$ with $\alpha$ a monomial of an arbitrary degree, then $g$ requires monotone arithmetic circuit of size $\Omega(2^n)$.*

*Proof.* $(i)$ follows from Theorem 6.1 and the fact that $f$ is transparent (see Proposition 3.7 and Remark 3.8). Similarly, $\mathsf{Newt}(\alpha f)$ is merely a shift of $\mathsf{Newt}(f)$ and hence it remains transparent, which gives $(iii)$.

For $(ii)$ we use a result of Hyafil [20]: If $g$ has a monotone circuit of size $s$, then it has a monotone formula of size $2^{O(\log s \log d + \log^2 d)}$. Part $(i)$ completes the proof. □

The degree assumption in $(ii)$ is rather artificial. A monotone circuit with divisions can result in $g$ with an exponential degree, as is the case in the circuit from [14] computing the spanning tree polynomial. Nevertheless, this yields lower bounds at least for monotone formulas with division.

**Theorem 6.3.** *The polynomials* $\mathsf{Clique}_n$ *and* $\mathsf{Art}_n$ *require monotone formula with division of size* $2^{\Omega(n)}$.

*Proof.* Brent's [5] argument that formulas with division can be balanced implies that if $f$ has monotone formula with divisions of size $s$, then $f = g/h$ where both $g$ and $h$ have monotone formulas of size polynomial in $s$. Proposition 6.2 part $(i)$ completes the proof □

**Remark 6.4.** *Transparency is fragile. If* $f$ *is transparent then* $f^2$ *is not necessarily so. In fact, if* $f$ *is monotone then* $f^2$ *is* never *transparent unless* $|\mathsf{supp}(f)| \leq 1$. *Hence, the techniques from Proposition 6.2 do not give anything when* $g = f^m$ *and* $m$ *is exponentially large.*

**Remark 6.5.** *A different partial solution to Problem 2 can be inferred from monotone* Boolean *lower bounds. Let* $\mathsf{Clique}_{k,n}$ *be the polynomial* $\sum_A \prod_{i,j \in A} x_{i,j}$, *where* $A$ *ranges over* $k$*-element subsets of* $[n]$. *For* $k := \lfloor (n/\log n)^{2/3}/4 \rfloor$, *and for every* $m$, *the polynomial* $(\mathsf{Clique}_{k,n})^m$ *requires a monotone arithmetic circuit of size* $2^{n^{\Omega(1)}}$.

*Indeed, a monotone arithmetic can be interpreted as a monotone Boolean circuit (cf. Section 5.1). Hence, a monotone arithmetic circuit for* $(\mathsf{Clique}_{k,n})^m$ *translates to a monotone Boolean circuit deciding whether a graph has a* $k$*-clique. This requires an exponential circuit by a result of Alon and Boppana [1].*

# 7 $\tau$-Conjecture for Newton polygons

Koiran et al. made the following conjecture [26].

**Conjecture 1** ([26])**.** *Let* $\mathbb{F}$ *be a field. Let* $f \in \mathbb{F}[x_1, x_2]$ *be a bivariate polynomial which can be written as*

$$f = \sum_{i=1}^{p} \prod_{j=1}^{q} f_{i,j}\,, \quad \textit{where } |\mathsf{supp}(f_{i,j})| \leq r\,, \tag{7}$$

*then* $\mathsf{Newt}(f)$ *has at most* $O((pqr)^c)$ *vertices (for some absolute constant* $c$*).*

The authors of [26] have shown that Conjecture 1 implies VP≠VNP over the field in question. The conjecture is related to a similar conjecture by Koiran from [24] about the number of real roots of *univariate* polynomials. In [19], it was shown that the conjecture from [24] in fact implies Conjecture 1. Theorem 5.4 validates the conjecture in the monotone setting:

**Remark 7.1.** *Let $f$ be as in (7) with $f_{ij}$ monotone. Then* $\mathsf{Newt}(f)$ *has at most $pqr$ vertices.*

The conjecture can be used to upper-bound the shadow complexity.

**Proposition 7.2.** *Let $\mathbb{F}$ be an infinite field. Assume Conjecture 1 holds over $\mathbb{F}$. Assume that a polynomial $f$ of degree $d$ has an arithmetic circuit of size $s$. Then $\sigma(\mathsf{Newt}(f)) \leq s^{O(\sqrt{d}\log d)}$.*

*Proof.* First, observe that if Conjecture 1 is true, it is also true when $f$ and $f_{ij}$ in (7) are allowed to be Laurent polynomials.

Now, if $f$ has a circuit of size $s$, then $f$ has a depth-four circuit of size $s^{O(\sqrt{d}\log d)}$; see [25] and references within. This means that we can write

$$f = \sum_{i=1}^{p} \prod_{j=1}^{q} f_{i,j}\,, \quad \text{where } |\mathsf{supp}(f_{i,j})| \leq r\,,$$

with $pqr \leq s^{O(\sqrt{d}\log d)}$.

Suppose that $\sigma(\mathsf{Newt}(f)) = k$. By Lemma 4.2, there is a h.p.-projection $\pi$ so that the Newton polytope of the bivariate Laurent polynomial $\pi(f)$ has $k$ vertices. Hence $\pi(f) = \sum_{i=1}^{p} \prod_{j=1}^{q} \pi(f_{i,j})$. Since $|\mathsf{supp}(\pi(f_{i,j}))| \leq r$, Conjecture 1 implies $k \leq O((pqr)^c)$ and hence $k \leq s^{O(d\log d)}$. □

This gives quantitative bounds for some specific polytopes, mainly the Birkhoff polytope and the Matching polytope from Remark 3.14:

**Corollary 7.3.** *Assume that Conjecture 1 holds over* some *infinite field. Then both $\sigma(\mathsf{DS}_n)$ and $\sigma(\mathsf{MATCH}_n)$ are at most $2^{O(\sqrt{n}\log^2 n)}$.*

*Proof.* $\mathsf{DS}_n$ is the Newton polytope of the determinant polynomial which has an arithmetic circuit of size $s = n^{O(1)}$. For $\mathsf{MATCH}_n$, the same holds by Remark 5.7. □

We do not know whether these conclusions hold or not. Another implication of Conjecture 1 is that $\sigma(\mathsf{Q}_{k,n}) \leq n^{O(1)}$, where $\mathsf{Q}_{k,n}$ is the convex hull of vectors in $\{0,1\}^n$ of Hamming weight $k$. It follows from Proposition 3.3 that this is actually true: $\sigma(\mathsf{Q}_{k,n}) \leq n^2$.

**Remark 7.4.** *Results of Gritzman and Sturmfels [17] (cf. Section 1.1) imply the following monotone three-dimensional version. Let $f$ be as in (7), where $f_{ij} \in \mathbb{R}[x_1, x_2, x_3]$ are monotone. Then $\mathsf{Newt}(f) \subseteq \mathbb{R}^3$ has at most $O(p(qr)^2)$ vertices.*

# 8 An easy polynomial with many vertices

Here we construct a bivariate polynomial with a monotone arithmetic circuit of linear size, but whose Newton polytope has exponentially many vertices. This proves Theorem 5.1.

We use the following notation. Given $(a,b) \in \mathbb{R}^2$,

$$(a,b) \cdot P := \{(ax, by) : (x,y) \in P\}.$$

Given $a \in \mathbb{R}$,

$$aP := (a,a) \cdot P.$$

**Observation 8.1.** *For a bivariate polynomial $f(x,y)$,*

$$\mathsf{Newt}(f(x^a, y^b)) = (a,b)\mathsf{Newt}(f(x,y)) \text{ and } \mathsf{Newt}(f^a) = a\mathsf{Newt}(f).$$

The building block of the polynomial are the following two polytopes. Let $P_n$ be the polytope with vertices $\{(k,k^2) : 0 \leq k \leq n-1\}$. Let $Q_n$ be the polytope with vertices $\{(k,k^2+k) : 0 \leq k \leq n-1\}$. These polytopes can be constructed inductively as follows.

**Lemma 8.2.** *For every $n \geq 1$,*

$$P_{2n} = (2,4) \cdot P_n \sqcup ((1,1) + (2,4) \cdot Q_n)) \quad (8)$$

$$Q_{2n} = (1,2) \cdot (P_n + Q_n) \sqcup \{(2n-1, 2n(2n-1))\}. \quad (9)$$

*Proof.*

Part (8). Let $0 \leq k \leq 2n-1$. If $k = 2r$ is even then $r \leq n-1$ and

$$(k,k^2) = (2,4)(r,r^2)$$

with $(r, r^2)$ a vertex of $P_n$. If $k = 2r + 1$ is odd then $r \leq n - 1$ and

$$(k, k^2) = (2r + 1, 4r^2 + 4r + 1) = (1, 1) + (2, 4) \cdot (r, r^2 + r)\,,$$

where $(r, r^2 + r)$ is a vertex of $Q_n$. This shows the containment $\subseteq$ in (8). The other direction holds since $P_n \sqcup Q_n$ can have at most $2n$ vertices.

Part (9). We first describe the vertices of $(1, 2)(P_n + Q_n)$. We claim that

$$\begin{aligned} \mathsf{vert}((1,2)(P_n+Q_n)) = \{v_0, v_1, \ldots, v_{2n-2}, u\}, \\ \text{where } v_k := (k, k^2 + k),\ u := (n - 1, 2n(n - 1)). \end{aligned} \tag{10}$$

Given $0 \leq k \leq 2n - 2$, let us show that $v_k$ is a vertex of $(1, 2)(P_n + Q_n)$. If $k = 2r$ is even, we have $r \leq n - 1$ and

$$(k, k^2 + k) = (2r, 4r^2 + 2r) = (1, 2)(r, r^2) + (1, 2)(r, r^2 + r).$$

If $k = 2r + 1$ is odd, we have $r \leq n - 2$ and

$$(k, k^2 + k) = (2r + 1, 4r^2 + 6r + 2) = (1, 2)(r + 1, (r + 1)^2) + (1, 2)(r, r^2 + r).$$

This means that $v_k \in (1, 2)(P_n + Q_n)$. Now, every $(z_1, z_2) \in (1, 2)(P_n + Q_n)$ satisfies $z_2 \geq z_1^2 + z_1$, because

$$2r_1^2 + 2(r_2^2 + r_2) - (r_1 + r_2)^2 - (r_1 + r_2) = (r_1 - r_2)^2 - (r_1 - r_2) \geq 0.$$

Since $v_k$ lies on the curve $z_2 = z_1^2 + z_1$, and the curve is strictly convex, $v_k$ cannot be convex combination of other points in $(1, 2)(P_n + Q_n)$. So, $v_k$ is indeed a vertex. To show that $u$ is a vertex, note that both $(1, 2)P_n$ and $(1, 2)Q_n$ are contained in the halfplane $\{(z_1, z_2) \in \mathbb{R}^2 : z_2 \leq 2nz_1\}$. On the boundary $z_2 = 2nz_1$, $(1, 2)Q_n$ has vertices $(0, 0)$ and $u$, and $(1, 2)P_n$ only the vertex $(0, 0)$. This implies $u$ is a vertex of $(1, 2)(P_n + Q_n)$. This proves the containment $\subseteq$ in (10). Equality holds since $P_n + Q_n$ can have at most $2n$ vertices.

To infer (9) from (10), note that $u$ lies on the line connecting the origin and $v_{2n-1} = (2n - 1, 2n(2n - 1))$. □

*Proof of Theorem 5.1.* Inductively define a sequence of bivariate polynomials. The base case is

$$p_0 = 1 \text{ and } q_0 = 1.$$

The inductive step is

$$p_{n+1} = p_n(x^2, y^4)^2 + x^N y^N q_n(x^2, y^4)^2$$

and

$$q_{n+1} = p_n(x^2, y^4) q_n(x^2, y^4) + x^{N(N-1)} y^{N^2(N-1)}$$

where $N = 2^{n+1}$.

We claim that for every $n \geq 0$,

$$\mathsf{Newt}(p_n) = 2^n P_{2^n} \text{ and } \mathsf{Newt}(q_n) = 2^n Q_{2^n} . \tag{11}$$

For $n = 0$, this follows from $\mathsf{Newt}(p_0) = \mathsf{Newt}(q_0) = \{(0,0)\} = P_1 = Q_1$. The induction step uses Lemma 2.5 and Observation 8.1. Assume that (11) holds for a given $n \geq 0$. Then

$$\mathsf{Newt}(p_n(x^2, y^4)) = 2^n(2,4)P_{2^n} \text{ and } \mathsf{Newt}(q_n(x^2, y^4)) = 2^n(2,4)Q_{2^n}.$$

Using (8),

$$\begin{aligned}
\mathsf{Newt}(p_{n+1}) =& 2 \cdot 2^n(2,4)P_{2^n} \sqcup ((N,N) + 2 \cdot 2^n(2,4)Q_{2^n})) \\
=& 2^{n+1}((2,4)P_{2^n} \sqcup ((1,1) + (2,4)Q_{2^n}))) \\
=& 2^{n+1} P_{2^{n+1}} .
\end{aligned}$$

Similarly, part (9) gives

$$\begin{aligned}
\mathsf{Newt}(q_{n+1}) =& 2^n(2,4)(P_{2^n} + Q_{2^n}) \sqcup \{(N(N-1), N^2(N-1))\} \\
=& 2^{n+1} ((1,2)(P_{2^n} + Q_{2^n}) \sqcup \{(N-1, N(N-1))\}) \\
=& 2^{n+1} Q_{2^{n+1}} .
\end{aligned}$$

This proves (11).

To compute $p_n, q_n$, first construct a circuit of size $O(n)$ that simultaneously computes $x^M, x^{M(M-1)}$, $y^M, y^{M^2(M-1)}$ for every $M = 2^m$ with $m \leq n$. Now, construct a circuit for $p_n$ and $q_n$ inductively. Given a circuit for $p_n$ and $q_n$, we can construct a new one computing $p_{n+1}, q_{n+1}$ by introducing a constant number of extra gates. □

# 9 Open problems

We conclude with the main open problems of this paper.

**Open Problem 1.** *Is $\sigma(\mathsf{DS}_n)$ or $\sigma(\mathsf{MATCH}_n)$ exponential in $n$?*

**Open Problem 2.** *Is Conjecture 1 true? If not, is it true when $f$ in (7) is required to have convexly independent support?*

**Open Problem 3.** *Find an explicit monotone $f_n$ (with polynomially many variables and of a polynomial degree) such that $g$ requires superpolynomial monotone arithmetic circuit whenever $g \neq 0$ and $f_n$ divides $g$.*

**Acknowledgement.** We thank Michael Forbes for pointing out the connection between shadow complexity and Conjecture 1.

# References

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

[2] E. Balas. Disjunctive programming: properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44, 1998.

[3] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 2 edition, 2000.

[4] A. Borodin and S. Cook. On the number of additions needed to cumpute specific polynomial. *SIAM J. Comput.*, 5:146–157, 1976.

[5] R. P. Brent. The parallel evaluation of general arithmetic expressions. *J. ACM*, 21:201–206, 1974.

[6] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and Computation in Mathematics*. Springer, 2000.

[7] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *A series of comprehensive studies in mathematics*. Springer, 1997.

[8] P. Carstensen. Complexity of some parametric integer and network programming problems. *Math. Programming*, 26:64–75, 1983.

[9] P. Carstensen. *The complexity of some problems in parametric linear and combinatorial programming*. PhD thesis, Univ. of Michigan, 1983.

[10] B. Chazelle, H. Edelsbrunner, and L. J. Guibas. The complexity of cutting complexes. *Discrete Comput Geom*, 4:139–181, 1989.

[11] M. Confronti, M. D. Summa, and Y. Faenza. Balas formulation for the union of polytopes is optimal. *Math. Programming*, 180:311–326, 2020.

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

[13] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: Exponential separation and strong lower bounds. *CoRR*, abs/1111.0837, 2011.

[14] S. Fomin, D. Grigoriev, and G. Koshevoy. Subtraction-free complexity, cluster transformations, and spanning trees. *Found Comput Math*, 16:1–31, 2016.

[15] D. Gale. Optimal assignments in an ordered set: an application of matroid theory. *J. Combin. Theory* 4, pages 1073–1082, 1968.

[16] S. Gao. Absolute irreducibility of polynomials via newton polytopes. *Journal of Algebra*, 237(2):501–520, 2001.

[17] P. Gritzmann and B. Sturmfels. Minkowski addition of polytopes: Computational complexity and applications to Gröbner bases. *SIAM J. Disc. Math.*, 6(2), 1993.

[18] J. A. Grochow. Monotone projection lower bounds from extended formulation lower bounds. *Theory of Computing*, 13:1–15, 2017.

[19] P. Hrubeš. On the distribution of runners on a circle. *European Journal of Combinatorics*, 89, 2020.

[20] L. Hyafil. On the parallel evaluation of multivariate polynomials. *SIAM J. Comput.*, 8(2):120–123, 1979.

[21] M. Jerrum and M. Snir. Some exact complexity results for straight-line computations over semirings. *Journal of the ACM*, 1982.

[22] S. Jukna. Lower bounds for tropical circuits and dynamic programs. *Theory of Computing Systems*, 57:160–194, 2015.

[23] E. Kaltofen. Uniform closure properties of p-computable functions. In *STOC*, pages 330–337, 1987.

[24] P. Koiran. Shallow circuits with high-powered inputs. In *Symposium on Innovations in Computer Science*. Tsingua University Press, Beijing, 2011.

[25] P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012.

[26] P. Koiran, N. Portier, S. Tavenas, and S. Thomassé. A $\tau$-conjecture for Newton polygons. *Foundations of computational mathematics*, 15(1):187–197, 2015.

[27] S. Berkowitz L. Valiant, S. Skyum and C. Rackoff. Fast parallel computation of polynomials using few processors. *Siam J. Comp.*, 12:641–644, 1983.

[28] J. G. Lagarias, Y. Luo, and A. Padrol. Moser's shadow problem. *ArXiv*, 2013.

[29] E. H. Moore. A two-fold generalization of fermat's theorem. *Bull. Amer. Math. Soc.*, 2(7):189–199, 1896.

[30] L. Moser. Poorly formulated unsolved problems in combinatorial geometry. In *mimeographed notes*. (East Lansing conference), 1966.

[31] K. Mulmuley. Lower bounds in a parallel model without bit operations. *SIAM J. Comput.*, 28(4):1460–1509, 1999.

[32] K. Mulmuley and P. Shah. A lower bound for the shortest path problem. *Journal of Computer and System Sciences*, 62(2):253–267, 2001.

[33] N. Nisan. Lower bounds for non-commutative computation. In *Proceeding of the 23th STOC*, pages 410–418, 1991.

[34] R. Paturi and F. Zane. Dimension of projections in Boolean functions. *SIAM J. Disc. Math.*, 11(4):624–632, 1998.

[35] A. Rao and A. Yehudayoff. Communication Complexity: And Applications. Cambridge University Press. doi:10.1017/9781108671644

[36] R. Raz and A. Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. J. Comput. Syst. Sci. 77(1), pages 167–190, 2011.

[37] T. Rothvoß. Some 0/1 polytopes need exponential size extended formulations. *CoRR*, abs/1105.0036, 2011.

[38] T. Rothvoß. The matching polytope has exponential extension complexity the matching polytope has exponential extension complexity. *J. ACM*, 2017.

[39] E. Shamir and M. Snir. On the depth complexity of formulas. *Journal Theory of Computing Systems*, 13(1):301–322, 1979.

[40] A. Shpilka and A. Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science,* 5(3-4), 2010.

[41] H. R. Tiwari. On computing the shadows and slices of polytopes. *arXiv*, 2008.

[42] L. G. Valiant. Negation can be exponentially powerful. *Theoretical Computer Science*, 12:303–314, 1980.

[43] A. Vince. A framework for the greedy algorithm. *Discrete Applied Mathematics* 121, pages 247–260, 2002.

[44] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.



ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il