Fundamenta Informaticae XX (2020) 1–1029
DOI 10.3233/FI-2016-0000
IOS Press

On Proving Parameterized Size Lower Bounds for Multilinear Algebraic Models

Purnata Ghosal
Indian Institute of Technology Madras, Chennai
India 600036
purnatag@gmail.com

B. V. Raghavendra Rao
Indian Institute of Technology Madras, Chennai
India 600036
bvrr@iitm.ac.in

Abstract. We consider the problem of obtaining parameterized lower bounds for the size of arithmetic circuits computing polynomials with the degree of the polynomial as the parameter. We consider the following special classes of multilinear algebraic branching programs:

1) Read Once Oblivious Branching Programs (ROABPs),

2) Strict interval branching programs,

3) Sum of read once formulas with restricted ordering.

We obtain parameterized lower bounds (i.e., $n^{\Omega(t(k))}$ lower bound for some function t of k) on the size of the above models computing a multilinear polynomial that can be computed by a depth four circuit of size $g(k)n^{O(1)}$ for some computable function g.

Further, we obtain a parameterized separation between ROABPs and read-2 ABPs. This is obtained by constructing a degree k polynomial that can be computed by a read-2 ABP of small size such that the rank of the partial derivative matrix under any partition of the variables is large.

1. Introduction

Algebraic Complexity Theory is concerned with complexity of computing polynomials using elementary arithmetic operations such as addition and multiplication over an underlying ring or field. Valiant [1] formalized the notions of algebraic complexity theory and posed proving lower bound on the size of arithmetic circuits computing explicit polynomials as the primary challenge for the area. Following Valiant's work, there has been intense research efforts in the past four decades to prove lower bounds on the size of special classes of arithmetic circuits such as constant depth circuits, multilinear formula and non-commutative models. (See [2, 3] for a survey.) Despite several attempts, the best known size lower bound for the size of an arithmetic circuit computing an explicit polynomial is only super linear [4].

Since any meaningful classification of complexity of polynomials is far from settled, one possibility is to look for relaxed notions of tractability such as parameterized complexity or approximation of polynomials. The notion of approximation of polynomials has been studied in a more geometric setting through various forms of degenerations [5]. In this article, we focus on parameterized complexity of polynomials.

Parameterized Complexity Theory is a multi-dimensional study of computational problems where the complexity of a problem is analyzed in terms of the input size and an additional parameter which can be independent of the input size. This multidimensional view of computation was introduced by Downey and Fellows in their seminal work [6], where they developed foundations of parameterized complexity theory. A decision problem with input size n and parameter k is said to be *fixed-parameter tractable* (FPT) if it has a deterministic $f(k)\text{poly}(n)$ time algorithm. The notion of intractability in parameterized complexity is captured by the complexity class XP and hierarchies of classes such as the W-hierarchy and A hierarchy [6].

Engels [7] initiated the development of a parameterized theory for algebraic complexity classes to obtain algebraic analogue of the theory developed by Downey and Fellows [6]. He suggested suitable notions of tractability and reductions and obtained complete problems for some of the classes introduced, for families of polynomials with a generic parameter.

While a theory with generic parameter is essential to the parameterized study of algebraic complexity, defining specific parameterizations for polynomials and study of their complexity is a first step in this direction. Further, specific parameterizations of polynomial might lead to more insights on the complexity of polynomials.

Müller [8] was the first to introduce parameterizations on polynomials in the context of designing parameterized algorithms for problems on polynomials, such as testing for identity of polynomials given as arithmetic circuits (ACIT). ACIT is one of the fundamental problems in algebraic complexity theory and has close connections to the circuit lower bound problem [9]. Müller studied parameters such as the number of variables in the polynomial, multiplication depth of the circuit computing the polynomial etc., and obtained efficient randomized parameterized algorithms for ACIT.

Polynomials with the degree bounded by a parameter are widely used in the development of efficient parameterized algorithms [10, 11, 12] and in expressing properties of graphs [13].

For example, in [12], the polynomial representing homomorphisms between two graphs has indeed degree equal to the parameter, i.e., the number of vertices in the pattern graph. An efficient computation of the polynomial defined in [12] by arithmetic formulas leads to space-efficient algorithms for detecting homomorphisms from a graph of bounded treewidth. Thus the study of parameterized complexity of polynomials with the degree of a polynomial as the parameter is important for understanding the limitations of these parameterized algorithms.

Let n be the number of variables and k be a parameter (e.g., degree of the polynomial). Throughout the article, $t(k)$ denotes a computable function that depends only on the parameter, e.g., $t(k) = 2^k, t(k) = 2^{2^k}, t(k) = \sqrt{k}$ etc. Any circuit is said to be of fpt size if the size of the circuit is bounded by $t(k)n^{O(1)}$ for some computable t. By a parameterized lower bound, we mean a lower bound of the form $n^{\Omega(t(k))}$ for a computable function t. It may be noted that the task of proving parameterized lower bounds is more challenging than classical lower bounds. In the case of degree as a parameter, most of the existing lower bounds of the form $n^{\Omega(\sqrt{k})}$(e.g., [14, 15, 16]) for constant depth circuits are already parameterized lower bounds. In contrast, the lower bounds for other special classes such as multilinear formula [17, 18] do not translate easily, especially since the hard polynomials involved are of high degree.

To understand the primary challenge in translating the results in [18, 17] to the parameterized setting we need to delve a bit on the techniques used for proving lower bounds. Raz [17] used the notion of partial derivative matrix under a partition of variables with equal parts (See Section 2 for a detailed definition) as a measure of complexity for polynomials. The idea is to show existence of such partitions where polynomials computed by a multilinear formula of small size will have small rank for the partial derivative matrix. Then, for any polynomial that has large rank under every partition, a natural lower bound on the size follows. However, the analysis done in [17] or subsequent works [19, 18, 20] do not carry forward when the polynomials are parameterized by the degree. Similarly the construction of the polynomial family with high rank partial derivative matrix in [17] and subsequent works do not generalize to the parameterized setting. The main challenge here is construction of low degree polynomials that have maximum value of the measure defined by Raz [17]. Further, obtaining useful upper bounds on the measure for polynomials computed by special classes of circuits also remains a challenging task in the parameterized setting.

In this article we address the challenge of translating lower bounds for the size of multilinear restrictions of arithmetic circuits to parameterized lower bounds.

Results We prove parameterized lower bounds on the size of a ROABP (Theorem 4.1), a strict-interval ABP (Corollary 4.8) and sum of ROPs with restricted ordering of variables (Theorem 4.11) against two families of explicit polynomials. We also obtain a parameterized size separation between ROABPs and read-2 ABPs (Corollary 4.4). For the first three lower bounds (Theorem 4.1, Corollary 4.8 and Theorem 4.11), we construct a parameterized family of polynomials (Theorem 3.4) such that under every equal sized bi-partition of the variables, the rank of the partial derivative matrix is the maximum possible up to a factor that depends

only on the parameter. The construction is built on the polynomial defined by Raz and Yehudayoff [19] and then does a careful analysis of weighted matchings in a complete graph. For the parameterized separation (Corollary 4.4) we construct a parameterized family of polynomials that can be computed by quadratic size read-2 oblivious ABPs (Theorem 4.3), such that under any partition of the variables into two equal parts, the rank of the partial derivative matrix is high (Theorem 3.6.)

Using the second parameterized family of polynomials we construct, we are able to obtain a parameterized version of the separation between read-2 ABPs and ROABPs given in [21] (Theorems 4.2,4.3). This is because this parameterized family of polynomials, where each polynomial can be written as sum of three read-once polynomials, is a parameterized variant of the hard polynomial given in [21] (Theorem 3.6).

Related Works In [22], Chen and Fu studied the parameterized complexity of testing monomials in multivariate polynomials paraemterized by the degree. These results were further improved in [23].

In [24], Chauhan and Rao studied ACIT with degree as the parameter and obtained a randomness-efficient parameterized algorithm for ACIT parameterized by degree. In [25] the authors along with Prakash studied polynomials parameterized by the degree and showed limitations of an existing approach in obtaining deterministic parameterized algorithms for ACIT.

2. Preliminaries

In this section we give necessary definitions related to arithmetic circuits. For more details the reader is referred to an excellent survey by Shpilka and Yehudayoff [3].

Let $\mathbb{F}$ denote a field. Unless otherwise stated, $\mathbb{F}$ is considered as an arbitrary field. Let $X = \{x_1, \ldots, x_n\}$ denote the set of variables. For a polynomial $p \in \mathbb{F}[X]$, let $\mathsf{var}(p)$ denote the set of variables that p is dependent on and $\deg(p)$ denote its degree.

Arithmetic Circuits An arithmetic circuit is a model for computing polynomials using the basic operations $+$ and $\times$. An arithmetic circuit C computing a polynomial p is a directed acyclic graph, where every node (called a gate) has in-degree two or zero. The gates of in-degree zero are called input gates and are labeled by constants from the field $\mathbb{F}$ on which the polynomial is defined, or variables from the set X of input variables of the polynomial. Internal gates of C compute either $+$ (sum) or $\times$ (product) of their inputs. Gates of out-degree zero are called output gates. Typically an arithmetic circuit will have a single output gate. Every gate in the circuit C is associated with a unique polynomial in $\mathbb{F}[X]$. The polynomial computed by the circuit, p, is the polynomial associated with its output gate.

The complexity of arithmetic circuits is measured in terms of *size* and *depth*. Size is defined as the number of gates in the circuit. The depth of the circuit represents the length of the longest path from the output node (root) to an input node (leaf) of the circuit. Since a

constant depth arithmetic circuit where fan-in of every gate is bounded by 2 (or even a constant) cannot even read all of the inputs, we assume unbounded fan-in in the case of constant depth circuits. Arithmetic circuits of constant depth have received wide attention [2].

Recall that a monomial $\prod_{i=1}^{n} x_i^{\alpha_i}$ is multilinear (also known as square free) if $\alpha_i \leq 1$ for $1 \leq i \leq n$. A polynomial $p \in \mathbb{F}[X]$ is said to be *multilinear* if p has only multilinear monomials. *Multilinear circuits* where every gate in the circuit computes a multilinear polynomial are natural models of computation for multilinear polynomials. However, it is not known if every efficiently computable multilinear polynomial is also efficiently computable by multilinear circuits [3]. While multilinear circuits are based on a semantic restriction on the circuit, a syntactic version of multilinear circuits received a lot of attention in the literature. An arithmetic circuit C is said to be *syntactic multilinear* if for every product gate with inputs g, h and associated polynomial $f = g \times h$ in C, the set of variables that appear under the sub-circuit rooted at g is disjoint from that of h.

Arithmetic Formulas An arithmetic circuit where the underlying graph is a tree is known as an *arithmetic formula*.

Definition 2.1. (Read-once Polynomials) An arithmetic formula is said to be a read-once formula (ROF for short) if every variable appears as a label in at most one leaf. Polynomials computed by ROFs are known as read-once polynomials (ROPs for short).

It may also be noted that ROFs are a proper subclass of *syntactic multilinear formulas*.

Algebraic Branching Programs An algebraic branching program (ABP) P is a directed acyclic graph with a source vertex s of in-degree 0 and a terminal vertex t of out-degree 0. The rest of the vertices can be divided into layers $L_1, L_2, \ldots, L_{r-1}$ between s and t, s being the only vertex in L_0, the first layer, and t being the only vertex in the last layer L_r. Edges in P are between vertices of consecutive layers. Every edge e is labeled by either a constant from $\mathbb{F}$ or a variable from X. For a directed path ρ in P, let $w(\rho)$ denote the product of edge labels in ρ. For any pair of nodes u, v in P let $[u, v]_P$ denote the polynomial $\sum_{\rho \text{ is a } u \to v \text{ path}} w(\rho)$. The polynomial computed by P is $[s, t]_P$. The *size* of an ABP is the total number of nodes and edges in it and the *depth* of an ABP is the total number of layers in it excluding the layers containing s and t.

An ABP P is said to be *syntactically multilinear* if every input variable is read at most once along any path from s to t in P. An ABP is said to be *oblivious* if the edges between any two layers L_i and L_{i+1} are all labeled by the same variable x_j.

Definition 2.2. (Read-once Oblivious ABPs) A *Read-once Oblivious ABP* (ROABP) P is a syntactically multilinear ABP where the input variables are read at most once, in a fixed order, along any path from s to t, and any variable occurs as a label for edges between at most one pair of layers of the ABP P.

Let π be a permutation of the variables. An interval in π is a set of the form $\{\pi(i), \pi(i+1), \ldots, \pi(j)\}$ for some $i < j$. Arvind and Raja [26] studied a restriction of multilinear ABPs called as *interval ABPs* where every node in the ABP computes a polynomial whose variable set forms an interval in $\{1, \ldots, n\}$. In this article, we consider a restriction of interval ABPs which we call as *strict interval* ABPs.

Definition 2.3. A syntactically multilinear ABP P is said to be a π-strict interval ABP, if for any pair of nodes (a, b) in P, the index set X_{ab} of the variables occurring on all paths from a to b is contained in some π-interval I_{ab} in $[n]$ and for any other node c in P, the intervals I_{ab} and I_{bc} are non-overlapping.

In the following we define notion of partial derivative matrix whose rank is an important complexity measure for polynomials.

Partial Derivative Matrix of a polynomial

Nisan [27] defined the partial derivative matrix of a polynomial, considered its rank as a complexity measure for non-commutative polynomials and proved exponential lower bounds for the size of non-commutative formulas and ABPs. Raz [17] considered a variant of the partial derivative matrix and proved super polynomial size lower bounds for multilinear formulas. We describe the partial derivative matrix introduced by Raz [17] in more detail.

A partition of X is an injective map $\varphi : X \to Y \cup Z$, where Y and Z are two disjoint sets of variables such that $|X| = |Y \cup Z|$. An equi-partition is a partition $\varphi : X \to Y \cup Z$ such that $|Y| = |Z| = |X|/2$.

Definition 2.4. [17] Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial of degree k, $\varphi : X \to Y \cup Z$ be a partition of the input variables of f. Then the partial derivative matrix of f with respect to φ, denoted by M_{f^φ} is an $A \times B$ matrix, where the rows are indexed by the set of all multilinear monomials μ of degree at most k in the variables Y, and columns indexed by the set of all multilinear monomials ν of degree at most k in the variables in Z, i.e., $A = \sum_{i=0}^{k} \binom{|Y|}{i}$ and $B = \sum_{i=0}^{k} \binom{|Z|}{i}$.

For monomials μ and ν respectively in variables Y and Z, the entry $M_{f^\varphi}(\mu, \nu)$ is the coefficient of the monomial $\mu\nu$ in f.

For a multilinear polynomial $p \in \mathbb{F}[X]$ and an equi-partition φ, let $\mathsf{rank}_\varphi(p)$ be the rank of the matrix M_{p^φ} over $\mathbb{F}$.

Let $X = \{x_1, \ldots, x_n\}$ be the set of variables where n is even. A partition of X is an injective function $\varphi : X \to Y \cup Z$, where Y and Z are two disjoint sets of variables. A partition φ is said to be an equi-partition if $|Y| = |Z| = n/2$. In the remainder of the article, we assume that the number of variables n is an even number. We use the partial derivative matrix defined by Raz for polynomials parameterized by degree, k.

The following fundamental properties of the rank of a partial derivative matrix were given by Raz [17]. We include a proof for completeness.

Lemma 2.5. Let f, g and h be multilinear polynomials of degree at most k in $\mathbb{F}[X]$. Then,

(Sub-additivity): If $f = g + h$, then $\forall \varphi : X \to Y \cup Z$, $\mathsf{rank}_\varphi(f) \leq \mathsf{rank}_\varphi(g) + \mathsf{rank}_\varphi(h)$, with equality when $\mathsf{var}(g) \cap \mathsf{var}(h) = \emptyset$.

(Sub-multiplicativity): If $f = g \times h$ then $\forall \varphi : X \to Y \cup Z$, $\mathsf{rank}_\varphi(f) \leq \mathsf{rank}_\varphi(g) \cdot \mathsf{rank}_\varphi(h)$, with $\mathsf{rank}_\varphi(f) = \mathsf{rank}_\varphi(g) \times \mathsf{rank}_\varphi(h)$ when $\mathsf{var}(g) \cap \mathsf{var}(h) = \emptyset$.

Proof:
Let $\varphi : X \to Y \cup Z$ be any partition of X , $A = \sum_{i=0}^{k} \binom{|Y|}{i}$ and $B = \sum_{i=0}^{k} \binom{|Z|}{i}$.

Sub-additivity: Suppose that $\overline{M_{g^\varphi}}$ and $\overline{M_{h^\varphi}}$ be matrices of order $A \times B$ obtained respectively from M_{g^φ} and M_{h^φ} by adding additional zero entries. Then $M_{f^\varphi} = \overline{M_{g^\varphi}} + \overline{M_{h^\varphi}}$ and hence $\mathsf{rank}_\varphi(g + h) \leq \mathsf{rank}_\varphi(g) + \mathsf{rank}_\varphi(h)$, as the rank of a matrix is a sub-additive function.

Additionally, if $\mathsf{var}(g) \cap \mathsf{var}(h) = \emptyset$, then for any two monomials $m_1 \in \mathbb{F}[Y]$ and $m_2 \in \mathbb{Z}$, either $\overline{M_{g^\varphi}}[m_1, m_2] = 0$ or $\overline{M_{h^\varphi}}[m_1, m_2] = 0$. Therefore, $\mathsf{rank}_\varphi(g + h) = \mathsf{rank}_\varphi(g) + \mathsf{rank}_\varphi(h)$.

Sub-multiplicativity: Let g and h be variable disjoint, $\mathsf{var}(g) \cap \mathsf{var}(h) = \emptyset$. Let $\varphi|_g : X_g \to Y_1 \cup Z_1$, and $\varphi|_h : X_h \to Y_2 \cup Z_2$, where $X_g = \mathsf{var}(g)$, $X_h = \mathsf{var}(h)$, $Y = Y_1 \cup Y_2$, $Z = Z_1 \cup Z_2$. Let, $\mathsf{M} = M_{g^{\varphi|_g}} \otimes M_{h^{\varphi|_h}}$ where $\otimes$ denotes the tensor product.

Note that each row index of M can be written as $m_{11}m_{12}$, a product of a multilinear monomial m_{11} in variables in Y_1, and a multilinear monomial m_{12} in variables in Y_2, respectively. Similarly, each column index of M can be written as $m_{21}m_{22}$, a product of a monomial in variables in Z_1, m_{21} and a monomial in variables in Z_2, m_{22} respectively. Now, M_{f^φ} is the sub-matrix of M obtained by removing rows and columns that are indexed by monomials of degree larger than k. Also, rows and columns of M that are indexed by monomials of degree larger than k will have no non-zero entries, for, $f = g \times h$ and f is a degree k multilinear monomial. Hence, we conclude that $\mathsf{rank}_\varphi(f) = \mathsf{rank}_\varphi(g) \cdot \mathsf{rank}_\varphi(h)$.
□

Rank upper bound for degree-k polynomials

In the following, we assume that $k << n/2$.

Lemma 2.6. For any equi-partition $\varphi : X \to Y \cup Z$, and any multilinear polynomial p of degree k, we have $\mathsf{rank}_\varphi(p) \leq (k+2)\binom{n/2}{k/2}$.

Proof:
Let $p \in \mathbb{F}[x_1, \ldots, x_n]$ be a degree-k multilinear polynomial. We fix an arbitrary equi-partition $\varphi : X \to Y \cup Z$ with $|Y| = |Z| = n/2$.

For $d \leq k$ let A_d be a matrix constructed from M_{p^φ} such that rows labeled by degree d monomials in the variables $x \in X$ such that $\varphi(x) \in Y$ are copied from M_{p^φ}, and all other rows correspond to all zero entries. We can now express $M_{p^\varphi} = A_0 + A_1 + \cdots + A_k$.

Then by sub-additivity, $\mathsf{rank}_\varphi(p) \leq \sum_{d=0}^{k} \mathsf{rank}(A_d)$. Since p has degree bounded by k, all but $\binom{n/2}{k-d}$ columns of A_d are zero columns. Thus $\mathsf{rank}(A_d) \leq \min\{\binom{n/2}{d}, \binom{n/2}{k-d}\}$. Substituting these values for $\mathsf{rank}_\varphi(p)$, we have, $\mathsf{rank}_\varphi(p) \leq 2\sum_{d=0}^{k/2}\binom{n/2}{d} \leq (k+2)\binom{n/2}{k/2}$.

This proof holds for $k < n/2$. Since in the parameterized domain, k is typically much smaller than n, the above calculations are sufficient for us. □

3. Construction of high rank polynomials

For any complexity measure $\mu : \mathbb{F}[X] \to \mathbb{R}_{\geq 0}$ for polynomials, to be useful, we need a class of polynomials where the measure is "small" and an explicit family of polynomials where the measure is "large". In this section, we consider the latter task and show construction of two parameterized polynomial families $f = (f_{n,2k})_{k\geq 0}$ and $h = (h_{2n,k})$ such that the rank of the partial derivative matrix is large for almost all partitions.

The first family is computable by a depth four circuit of fpt (i.e., $t(k)n^{O(1)}$ where $t(k) = k^{O(k)}$) size. For any partition φ, $\mathsf{rank}_\varphi(f)$ matches the maximum possible value defined by the upper bound described in Lemma 2.6, upto a factor that depends only on the parameter.

The second family h is a sum of three ROPs, also computable by a circuit of fpt size. In the case of h, $\mathsf{rank}_\varphi(h)$ attains the maximum possible value upto a constant factor in the exponent. Hence, $\mathsf{rank}_\varphi(h) \geq t_2(k)n^{ck}$ where t_2 is a computable function on k and $c < 1/2$.

A full rank polynomial

We know, by Lemma 2.6, that for a multilinear polynomial g of degree k in n variables, the maximum possible value of $\mathsf{rank}_\varphi(g)$ over all partitions φ is at most $(k+2)\binom{n/2}{k/2}$. Though it is possible to construct polynomials that achieve this bound under a fixed partition φ, it is not immediately clear if there is a polynomial g computed by small circuits that is full rank under every equi-partition. In the following, we give the description of a multilinear polynomial of degree k that has rank $n^{k/2}/t(k)$ where t is a function that depends only on k. We assume that $2k$ divides n.

We consider K_{2k}, the complete graph on $2k$ vertices. Suppose $V_1 \cup \cdots \cup V_{2k} = X$ be a partition of the variable set $X = \{x_1, \ldots, x_n\}$ such that $|V_i| = |V_j|$ for $1 \leq i < j \leq 2k$. For convenience let $V_i = \{x_{(i-1)n/2k+1}, \ldots, x_{in/2k}\}$, where we assume a natural ordering among the variables, i.e., $x_j \succeq x_i,\ \forall j \geq i$. We consider the variable set V_i as the label of vertex i of the graph K_{2k} for $1 \leq i \leq 2k$. For each edge (i,j) of K_{2k}, we define a polynomial p_{ij} on the vertex set $V_i \cup V_j$. These *edge polynomials* p_{ij} will be used in the subsequent construction of the polynomial f.

Let $\mathcal{M}$ be the set of all possible perfect matchings on $G = K_{2k}$. We define a parameterized family of polynomial $f = (f_{n,2k})_{n>1,2k|n}$, $f_{n,2k} \in \mathbb{G}[x_1, x_2, \ldots, x_n]$ as follows:

$$f(x_1, x_2, \ldots, x_n) = \sum_{M\in\mathcal{M}} \zeta_M \prod_{(i,j)\sim M} (1 + p_{ij}(V_i \cup V_j)),$$

where ζ_M for $M \in \mathcal{M}$ are formal variables. We define the edge polynomial p_{ij} as an n/k-variate quadratic multilinear polynomial, such that,

$$p_{ij}(x_i, \ldots, x_{j+n/2k-1}) = \sum_{k<\ell} \omega_{k,\ell} x_k x_\ell.$$

Here $\omega_{i,j}$, for $1 \leq i < j \leq n/k$, are also formal variables. So the polynomial f is defined over $\mathbb{G}[X]$ where $\mathbb{G}$ is an extension of the field $\mathbb{F}$ containing $\{\omega_{i,j}\} \cup \{\zeta_M \mid M \in \mathcal{M}\}$. The addition of 1 to p_{ij} ensures an increase in the $\mathsf{rank}_\varphi(p_{ij})$ by 1 if there is no monomial $x_a x_b$ in p_{ij} such that x_a, x_b are mapped to the same partition Y or Z.

Note that $f_{n,2k}$ is a degree $2k$ polynomial in n variables. When n and k are clear from the context, we use f to denote $f_{n,2k}$. Let $\mathbb{G} = \mathbb{F}(\{\zeta_M \mid M \in \mathcal{M}\} \cup \{\omega_{i,j} \mid 1 \leq i < j \leq n/k\})$, i.e the rational function field of the polynomial ring $\mathbb{F}[\{\zeta_M \mid M \in \mathcal{M}\} \cup \{\omega_{i,j} \mid 1 \leq i < j \leq n/k\}]$.

We note that by definition, f is multilinear and can be computed by a depth-4 circuit, parameterized by the degree, $2k$. As there are $k^{O(k)}$ perfect matchings in $\mathcal{M}$, and a circuit of size $(kn)^{(O(1))}$ can compute the polynomial $\prod_{(i,j)\in M}(1 + p_{ij})$, the size of the depth-4 circuit computing f is at most $k^{O(k)}n^{O(1)}$, which is fpt with k as the parameter and n as the size of the input.

In the remainder of the section, we argue that the polynomial family f defined above has almost full rank under every partition $\varphi : X \to Y \cup Z$, such that $|Y| = |Z| = |X|/2$. Now, the partition function divides the set of variables X into two equal halves, but it might not divide the individual sets $V_i \cup V_j$, the set of variables on which the edge polynomial p_{ij} is defined, in two equal halves. In that case, we define a new quantity, imbalance, as follows.

Definition 3.1. Consider an equi-partition function $\varphi : X \to Y \cup Z$. A set $V \subset X$ is said to be ℓ-unbalanced with respect to φ if $\frac{|V|}{2} - |\varphi(V) \cap Z| = \ell = |\varphi(V) \cap Y| - \frac{|V|}{2}$.

It may be noted that ℓ can be a positive or negative accordingly as $|\varphi(V) \cap Y| > |\varphi(V) \cap Z|$ or otherwise. Our first observation is, even if the set $V = V_i \cup V_j$ is ℓ-unbalanced for $\ell < n/4k$ for all edges (i, j), $\mathsf{rank}_\varphi(p_{ij})$ remains large:

Lemma 3.2. If $V_i \cup V_j$ is ℓ-unbalanced with respect to a partition $\varphi : X \to Y \cup Z$, then $\mathsf{rank}_\varphi(p_{ij}) = \Omega(n/2k - |\ell|)$.

Proof:
Without loss of generality, we consider the case $\ell > 0$. As already defined, $V_i \cup V_j = \{x_{(i-1)n/2k+1}, \ldots, x_{jn/2k}\}$ and we denote p_{ij} as p for the rest of the proof. Let us assume, for the ease of calculations, $V_i \cup V_j = \{x_1, \ldots, x_{n/k}\}$. Let φ be such that for all $x_q \in V_i \cup V_j$,

$$\varphi(x_q) = \begin{cases} y_q, & \text{if } q \leq n/2k + \ell, \\ z_{q-(n/2k+\ell)} & \text{otherwise.} \end{cases} \tag{1}$$

Since $p = p_{ij}$ is a quadratic polynomial, the rows of M_{p^φ} are indexed by degree at most one monomials $\emptyset, y_1, \ldots y_{n/2k+\ell}$ and degree two monomials of the form $y_i y_j, 1 \leq i < j \leq n/2k + \ell$. Similarly, the columns are indexed by $\emptyset, z_1, \ldots, z_{n/2k-\ell}$ and degree two monomials $z_i z_j, 1 \leq i < j \leq n/2k - \ell$.

We claim that the rows and columns indexed by degree 2 monomials will contribute at most 2 to the rank. This is because all row-indexing monomials $y_i y_j$ will have a non-zero entry $\omega_{i,j}$ only corresponding to the first column indexed by $\emptyset$. Similarly, all column-indexing monomials $z_i z_j$ have one non-zero entry along the first row, indexed by $\emptyset$. The first row and the first column can together contribute a rank of at most 2.

Now, it is required to show that the submatrix of M_{p^φ} with rows indexed by $\emptyset, y_1, \ldots y_{n/2k+\ell}$ and columns indexed by $\emptyset, z_1, \ldots, z_{n/2k-\ell}$ has rank $\Omega(n/2k - |\ell|)$.

The $(y_i, z_j)^{\text{th}}$ entry of M_{p^φ} contains $\omega_{i,n/2k+j}$. The sub-matrix of M_{p^φ} on rows and columns indexed by degree-1 monomials $\emptyset, y_1, \ldots y_{n/2k+\ell}$ and $\emptyset, z_1, \ldots, z_{n/2k-\ell}$ has dimension $n/2k + \ell$ by $n/2k - \ell$. By suitably substituting the formal variables $\omega_{i,n/2k+j}$ with values from $\mathbb{F}$, we can ensure that the submatrix of M_{p^φ} is of full column-rank when ℓ is positive. Thus $\mathsf{rank}_\varphi(p) = \Omega(n/2k - \ell)$. Therefore, over any edge (i, j) in G and any φ, the polynomial p_{ij} has rank $\Omega(n/2k - |\ell|)$.

Now, to complete the proof we need to show that the argument above works even when φ does not satisfy (1). In this case, we can re-index the variables in $V_i \cup V_j$ so that the partition φ satisfies (1) after the re-indexing and proceed with the argument above. This re-indexing only induces a permutation of the rows and columns the matrix M_{p^φ} and hence does not affect $\mathsf{rank}_\varphi(p)$. This completes the proof.

□

Before we proceed with proof of the required lower bound on the rank of f under any partition, we define imbalance on each variable set V_i, denoted by $D(V_i)$. If the imbalance on $V_i \cup V_j$ for an edge (i, j) is ℓ, we want $D(V_i), D(V_j)$ to be such that $\ell = D(V_i) + D(V_j)$.

Definition 3.3. For a partition $\varphi : X \to Y \cup Z$, the imbalance on a set V_i is defined as

$$D(V_i) \stackrel{\mathsf{def}}{=} |\varphi(V_i) \cap Y| - \frac{|V_i|}{2}.$$

Let $Y_i = \varphi(V_i) \cap Y,\ Z_i = \varphi(V_i) \cap Z$. We know $\forall i \in [2k],\ |V_i| = \frac{n}{2k}$. So, $D(V_i) = |Y_i| - \frac{n}{4k}$ is the imbalance of φ on V_i.

Under a partition φ, in the extreme cases, all variables in V_i are mapped to Y, or none of them are. Hence, $|Y_i| \in [0, \frac{n}{2k}]$, since $|V_i| = n/2k$. It follows that $D(V_i) \in [\frac{-n}{4k}, \frac{n}{4k}]$.

We are now ready to give the rank bound on the polynomial family f.

Theorem 3.4. For the parameterized multilinear polynomial family $f = (f_{n,2k})_{n,k \geq 0}$ such that $f(X) = \sum_{M \in \mathcal{M}} \zeta_M \prod_{(i,j) \sim M} (1 + p_{ij})$, we have,

$$\mathsf{rank}_\varphi(f_{n,2k}) = \Omega\left(\frac{n^k}{(2k)^{2k}}\right),$$

for every equi-partition $\varphi : X \to Y \cup Z$ and $k > 3$.

Proof:
Let us fix φ to be an arbitrary equi-partition of X. Note that by the definition of f, it is enough to show that for all equi-partitions φ, there exists an optimal matching N and $f_N = \prod_{(i,j)\in N}(1 + p_{ij})$ such that f_N is of full rank i.e., $\mathsf{rank}_\varphi(f_N) = \Omega(\frac{n^k}{(2k)^{2k}})$. Then we set $\zeta_N = 1$ and $\zeta_M = 0$ for all other $M \in \mathcal{M}$, so that f is of almost full rank.

Since f_N is multilinear, it is enough to prove that $\forall (i,j) \in N$, $\mathsf{rank}_\varphi(p_{ij}) = \Omega(\frac{n}{k^2})$. This would imply that our optimal perfect matching N is such that all edges (i,j) in N have very low imbalance under φ. Our argument is a construction of the required matching N.

Let us consider an arbitrary matching $M \in \mathcal{M}$. We construct N from M. For that purpose, we need to analyse each edge $e = (i,j)$ in the matching M. Hence, we associate a weight to all edges e with respect to φ such that $\mathsf{wt}(e) = |D(V_i) + D(V_j)|$. The weight of the matching M denoted by $\mathsf{wt}(M)$, is the sum of the weights of the edges in M, i.e., $\mathsf{wt}(M) = \sum_{e\in M} \mathsf{wt}(e)$.

In the following, we give an iterative procedure, that given M, produces a matching N with the required properties. The procedure obtains a new matching of smaller weight than the given matching in each iteration. The crucial observation then is that matchings that are weight optimal with respect to the procedure outlined below indeed have the required property.

We say that a matching N is *good* with respect to φ, if $\forall\ e = (i,j) \in N$, the weight does not exceed a threshold t, i.e., $\mathsf{wt}(e) \leq t = n/2k - n/(2k(k-1))$. Note that if M is good then for every edge $(i,j) \in M$, we have $V_i \cup V_j$ is ℓ-unbalanced for some ℓ with $|\ell| \leq n/2k - n/(2k(k-1))$. Then, by Lemma 3.2 we have $\mathsf{rank}_\varphi(f_M) \geq (n/(2k(k-1)))^k$. In that case, the matching M is the optimal matching N we desire.

Suppose the matching M is not good. Let $e = (i,j) \in M$ be a *bad edge* such that $\mathsf{wt}(e) > n/2k - n/2k(k-1)$. If there are multiple bad edges, e is chosen such that $\mathsf{wt}(e)$ is the maximum, breaking ties arbitrarily. Note that we can assume that $D(V_i)$ and $D(V_j)$ are of the same sign for $\mathsf{wt}(e)$ to have the highest value. Without loss of generality, assume that both $D(V_i)$ and $D(V_j)$ to be non-negative, i.e., $\mathsf{wt}(e) = D(V_i) + D(V_j)$. Since φ is an equi-partition, we have

$$\sum_{m\in[2k]} D(V_m) = \sum_{m\in[2k]} \Big(|Y_m| - \frac{n}{4k}\Big) = \sum_{m\in[2k]} |Y_m| - \frac{n}{2} = 0$$

$$\implies \sum_{m\in[2k]\setminus\{i,j\}} D(V_m) = -\mathsf{wt}(e)$$

$$\implies \sum_{e'\in M\setminus\{e\}} \mathrm{sgn}(e')\mathsf{wt}(e') = -\mathsf{wt}(e) < \frac{-n}{2k} + \frac{n}{2k(k-1)},$$

where $\mathrm{sgn}(e)$ is ± 1 depending on the sign of $\mathsf{wt}(e)$. By averaging, there is an edge $e_1 \in M$ such that,

$$\mathrm{sgn}(e_1)\mathsf{wt}(e_1) = -\frac{\mathsf{wt}(e)}{(k-1)} < \frac{-n}{2k(k-1)} + \frac{n}{2k(k-1)^2}.$$

Suppose $e_1 = (i_1, j_1)$. The idea is that on swapping the end-points (i, j) of the bad edge e with that of the edge e_1, (i_1, j_1), we will get a new matching M' with two new edges, all other edges being from M. We claim that this matching M' has lesser total weight $\mathsf{wt}(M')$ than $\mathsf{wt}(M)$. We can repeat this process to reduce weights of bad edges in M' till we obtain a matching N where all edges are good.

For the ease of analysis, let $D(V_i) = a$, $D(V_j) = b$, $D(V_{i_1}) = c$, $D(V_{j_1}) = d$. The new matching is constructed based on the values of a, b, c and d. Since $c + d = \mathsf{wt}(e_1) < 0$, it must be that either both c, d are negative, or any one of them is negative, i.e., $c < 0, d \geq 0$ or $c \geq 0, d < 0$. Here, we discuss both these cases.

Case 1 Suppose $c, d < 0$. Then, $|a+b|+|c+d| > |a+c|+|b+d|$. We replace the edges (i, j) and (i_1, j_1) by $(i, i_1), (j, j_1)$ to get a new matching M'. We have $\mathsf{wt}(M') < \mathsf{wt}(M)$.

Case 2 Either $c \geq 0$ and $d < 0$ or $c < 0$ and $d \geq 0$. Without loss of generality, assume that $c \geq 0$ and $d < 0$. We argue even if c is positive, it is smaller than at least one of the values a, b, thus making swapping end-points of e with e_1 yield a better matching.

We know, the least value d can have is $-n/4k$. Suppose $c > \frac{n}{4k} - \frac{n}{2k(k-1)} + \frac{n}{2k(k-1)^2}$. Then we have $d < \frac{-n}{2k(k-1)} + \frac{n}{2k(k-1)^2} - c < \frac{-n}{4k}$ which is impossible. Therefore, we have $c \leq \frac{n}{4k} - \frac{n}{2k(k-1)} + \frac{k}{2k(k-1)^2}$.

So, if $c > a, b$, then $a + b < 2c \leq \frac{n}{2k} - \frac{n}{k(k-1)} + \frac{n}{k(k-1)^2}$. For $k > 3$, this is a contradiction since $\mathsf{wt}(e) = a + b > \frac{n}{2k} - \frac{n}{2k(k-1)}$, and this lower bound seems to be higher than the upper bound. Hence, we consider the following sub-cases:

Subcase (i) $a > c$. Then $a + b > c + b$, replace the edges (i, j) and (i_1, j_1) with the edges (i, j_1) and (i_1, j) to get the new matching M'.

Subcase (ii) $b > c$. Then $a + b > a + c$, replace (i, j) and (i_1, j_1) with the edges (i, i_1) and (j, j_1) to get the new matching M'.

The second case above also implies that $|b+d| \geq 0$ and $|a+c| \geq \frac{n}{2k} - \frac{n}{2k(k-1)} + \frac{k}{2k(k-1)^2}$. We know $|a + b| \leq n/2k$. Therefore, the least decrease in total weight of matching is:

$$\begin{aligned}
|a + b| + |c + d| - |a + c| - |b + d| =& \frac{n}{2k} + \left(\frac{n}{2k(k-1)} - \frac{n}{2k(k-1)^2}\right) \\
& - \left(\frac{n}{2k} - \frac{n}{2k(k-1)} + \frac{k}{2k(k-1)^2}\right) - 0 \\
=& \frac{n}{k(k-1)} - \frac{n}{k(k-1)^2} = \frac{2t}{(k-1)}.
\end{aligned}$$

For the new matching M' obtained from M as above, we have one of the following properties:

- It has smaller total weight than M, i.e., $\mathsf{wt}(M') < \mathsf{wt}(M)$, or

- If M has a unique maximum weight edge, then the weight of any edge in M' is strictly smaller than that in M, i.e., $\max_{e' \in M'} \mathsf{wt}(e') < \mathsf{wt}(e)$, or
- The number of edges that have maximum weight in M' is strictly smaller than that in M, i.e., $|\{e'' \mid \mathsf{wt}(e'') = \max_{e' \in M'} \mathsf{wt}(e')\}| < |\{e'' \mid \mathsf{wt}(e'') = \max_{e' \in M} \mathsf{wt}(e')\}|$.

Since all of the invariants above are finite, by repeating the above procedure a finite number of times we get a matching $N \in \mathcal{M}$ such that any of the above steps are not applicable. That is, for every $e' \in N$, $\mathsf{wt}(e') \leq n/2k - n/2k(k-1)$. In fact, the largest value of $\mathsf{wt}(e) = n/2k$, and least decrease is $\mathsf{wt}(M) - \mathsf{wt}(M') \geq t/(k-1) = n/2k(k-1) - n/2k(k-1)^2$ by averaging. So, in at most k iterations for each edge, i.e., $O(k^2)$ iterations, we will obtain a matching where all edges have zero imbalance. The matching we need has low imbalance t for every edge, so our algorithm will obtain N from M in $O(k^2)$ iterations.

As required, for every edge $(i,j) \in N$, we have $\mathsf{rank}_\varphi(p_{ij}) = \Omega(n/2k(k-1))$ and $\mathsf{rank}_\varphi(f_N) = \Omega(n^k/(2k)^{2k})$. By the construction of the polynomial and Lemma 2.5, we have $\mathsf{rank}_\varphi(f) \geq \max_{M \in \mathcal{M}}\{\mathsf{rank}_\varphi(f_M)\} = \Omega(n^k/(2k)^{2k})$.

□

A high rank sum of three ROFs

In [21], Kayal et al. showed that there is a polynomial that can be written as sum of three ROFs such that any ROABP computing it requires exponential size. The lower bound proof in [21] is based on the construction of a polynomial using three edge disjoint perfect matchings on n vertices.

The construction of this polynomial, as a crucial ingredient, used a 3-regular mildly explicit family of expander graphs defined in [28]. Let $\mathcal{G} = (G(q))_{q>0, \text{ prime}}$ be a family of 3 regular expander graphs where a vertex x in $G(q)$ is connected to $x+1, x-1$ and x^{-1} where all of the operations are modulo q. When q is clear from the context, we denote $G(q)$ by G.

The *double cover* of G is the bipartite graph $G' = (V_1, V_2, E')$ where V_1, V_2 are copies of V and for all pair of vertices $\{u, v\}$ from V such that $u \in V_1, v \in V_2$ and $u \in V_2, v \in V_1$, it holds that $(u,v) \in E' \iff (u,v) \in E$. As no vertex in G has a self-loop, any edge of the form (u,u) is not present in E, and hence, E'.

Figure 1 gives an example of a double cover G' of a graph $G = (V, E)$ where $V = \{1, \ldots, 8\}$ and $E = \{(1,2), (3,4), (5,6), (7,8)\}$.

It is known from [28] that the set of edges in E' can be viewed as the union of 3 edge disjoint perfect matchings. In [21], Kayal et al. construct a polynomial for each of these matchings and the hard polynomial is obtained by taking the sum of these three polynomials. This polynomial has degree $n/2$ and therefore is unsuitable in the parameterized context. We construct a polynomial h from the same graph G' also based on three edge-disjoint perfect matchings as in [21], but our polynomial has degree k. Suppose $M_1 \cup M_2 \cup M_3 = E'$ be the edge-disjoint perfect matchings. We divide the $n/2$ edges in each of the M_i into $k/2$ bags of n/k edges each. Though the hardness of the polynomial follows for an arbitrary division

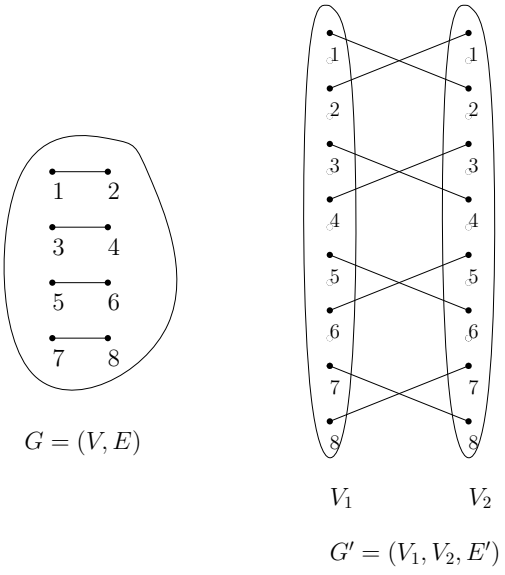

Figure 1. Example construction of double cover G' from $G = (V, E)$, where $V = \{1, \ldots, 8\}$.

of edges in M_i into bags, we consider a special kind of partition, which will be useful in the proof of Theorem 4.2.

Consider the subgraph induced by $M_1 \cup M_2$ which is a union of vertex disjoint cycles $C_1, \ldots, C_\ell$, $\ell \geq 1$. In fact, $C_1, \ldots, C_\ell$ is a cycle cover of G'. Since M_1 and M_2 are disjoint, minimum length of C_i is at least four. For $1 \leq i \leq \ell$, we write the cycle $C_i = (p_i, x_{i,1}, \ldots, x_{i,r}, p_i)$ where $r \geq 3$ and $(p_i, x_{i,1}), (x_{i,1}, x_{i,2}), \ldots, (x_{i,r-1}, x_{i,r}), (x_{i,r}, p_i)$ are the edges in it. The variable p_i is called the *pivot* of C_i. Pivots for the cycles are chosen so that M_3 has no matching between any two pivots. For $1 \leq i \leq \ell$, the cycle C_i induces a natural ordering on the variables involved in it, i.e., $p_i, x_{i,1}, \ldots, x_{i,r}$. We fix an ordering of the cycles $C_1, \ldots, C_\ell$ induced by the ascending order of the indices of their pivots. For $i \in \{1, 2\}$, let B_{i1} be the first $n/2k$ edges from M_i that appear according to the order induced by the cycles $C_1, \ldots, C_\ell$. To be more concrete, suppose $C_1 = (p_1, x_{1,1}, \ldots, x_{1,r_1}, p_1)$ and $C_2 = (p_2, x_{2,1}, \ldots, x_{2,r_1}, p_2)$. If $r_1 > n/k - 1$, $B_{1,1}$ contains the edges $(p_1, x_{1,1}), (x_{1,3}, x_{1,4}), (x_{1,5}, x_{1,6}), \ldots, (x_{1,n/k-1}, x_{1,n/k})$. If $r_1 < n/k$ then, $B_{1,1}$ consist of all edges of M_1 in C_1 and then edges from C_2C_3 etc., such that total number of edges is $n/2k$. Now B_{i2} is the next n/k edges from M_i. Repeating this, we get the partition $M_i = B_{i1} \cup B_{i2} \cup \cdots \cup B_{ik/2}$. Choose a partition $M_3 = B_{31} \cup \ldots \cup B_{3k/2}$ of the edges of M_3 such that the pivots appear in the order given by the cycle cover $C_1, \ldots, C_\ell$. Now, each edge $(u, v) \in M_i$ corresponds to the monomial $x_u x_v$. For every bag B_{ij}, we define the bilinear term $h_{ij} = \sum_{(u,v) \in B_{ij}} x_u x_v$ and the polynomial h_i corresponding to each M_i is defined as $h_i = \prod_{j \in [\frac{k}{2}]} h_{ij}$.

The final polynomial is the following:

$$h(x_1, \ldots, x_n) = \sum_{i \in [3]} w_i \left(\prod_{j \in [\frac{k}{2}]} \sum_{(u,v) \in B_{ij}} x_u x_v \right), \tag{2}$$

where w_1, w_2 and w_3 are formal variables, w_i corresponding to the matching M_i. To analyse the hardness of this polynomial, we need the notion of bichromatic edges.

Definition 3.5. For a partition $\varphi : X \to Y \cup Z$, and an edge $(u, v) \in M_i$, (u, v) is said to be *bichromatic* with respect to φ if either $\varphi(x_u) \in Y$ and $\varphi(x_v) \in Z$, or $\varphi(x_u) \in Z$ and $\varphi(x_v) \in Y$.

For a set of edges A over $\{x_1, \ldots, x_n\}$ let $\mathsf{be}_\varphi(A)$ be the number edges in A that are bichromatic with respect to φ. For a graph $G = (V, E)$, let $\mathsf{be}_\varphi(G) = \mathsf{be}_\varphi(E)$.

Let X be the variable set corresponding to vertices in G, i.e., $X = \{x_1, \ldots, x_n\}$. Fixing an equi-partition φ, we will view it as a coloring of variables in X in the color represented by Y or Z. The degree-2 monomials in the polynomial can now be viewed as a monochromatic edge if both end-points are of the same color, and a bichromatic edge otherwise. As seen before, in a particular term h_{ij}, each bichromatic edge contributes 1 towards the rank of the partial derivative matrix of h_{ij} and all monochromatic edges together contribute a maximum of 2. This is the idea we will use to prove the desired property of the polynomial h in the following theorem.

Theorem 3.6. Let h be the polynomial defined in Equation 2. Then there is a constant $c > \frac{23+20\epsilon}{114}$ for a fixed $\epsilon > 0$ such that for every equi-partition $\varphi : X \to Y \cup Z$, over the rational function field $\mathbb{F}(w_1, w_2, w_3)$, we have

$$\mathsf{rank}_\varphi(h) \geq \left(\frac{n}{k}\right)^{ck}.$$

Proof:
Let us fix an arbitrary partition $\varphi : X \to Y \cup Z$. By the expander property of G (see [21]), the number of edges from Y to Z is lower bounded by $E(Y, Z) \geq \frac{(2+10^{-4})}{2} \cdot |Y| = \frac{(1+\epsilon)n}{2}$ for a fixed $\epsilon > 0$.(See [21] for details.)

Now, each perfect matching has $\frac{n}{2}$ edges, so the graph has $\frac{3n}{2}$ edges. By averaging, we get that there is a matching M_i, $1 \leq i \leq 3$ such that the number of bichromatic edges in M_i,

$$\mathsf{be}_\varphi(M_i) \geq \frac{(1+\epsilon)n}{6}. \tag{3}$$

Without loss of generality, suppose $i = 1$. Let $h_1 = \prod_{j \in [\frac{k}{2}]} \sum_{(u,v) \in B_{1j}} x_u x_v$, i.e., the polynomial corresponding to M_1. We need to get an upper bound for $\mathsf{be}_\varphi(M_1)$.

Let us assume that the bichromatic edges in M_1 are distributed evenly across all sets in the partition $B_{11}, \ldots, B_{1k/2}$. Then, for every bag B_{1j} we will have $\mathsf{be}_\varphi(B_{1j}) = \left(\frac{(1+\epsilon)n}{6} \cdot \frac{2}{k}\right) =$

$\frac{(1+\epsilon)n}{3k}$, which will be the same as $\mathsf{rank}_\varphi(h_{1j})$ as each bichromatic edge contributes 1 towards the rank. By sub-multiplicativity, it follows that $\mathsf{rank}_\varphi(h_1) \geq \left(\frac{(1+\epsilon)n}{3k}\right)^{k/2}$.

However, this may not hold in general for M_1, because the bichromatic edges can potentially be distributed in exponential number of ways across the bags B_{1j}. Therefore, it is possible that for some values of j, the bags B_{1j} contain only monochromatic edges. This will reduce the exponent of n in the expression of the rank. Thus, this argument is not the right way to analyse the rank of the polynomial.

Nevertheless, we get a smaller but good enough bound by a simple averaging argument. Let $\mathsf{be}_\varphi(M_i) = \sum_{j\in[\frac{k}{2}]} \mathsf{be}_\varphi(B_{ij})$ and let α denote the number of bags with sufficient number of bichromatic edges, i.e., $\alpha = |\{j \mid \mathsf{be}_\varphi(B_{1,j}) \geq n/20k\}|$. Then, for $(k/2-\alpha)$ bags of M_1, the maximum number of bichromatic edges is upper bounded by $(k/2-\alpha)n/20k$, and the upper bound for each of the α remaining bags is the total number of edges in each bag, n/k. So, we upper bound $\mathsf{be}_\varphi(M_1)$ as follows:

$$\begin{aligned} \mathsf{be}_\varphi(M_1) &\leq \alpha\frac{n}{k} + (k/2-\alpha)\frac{n}{20k} \\ \implies \mathsf{be}_\varphi(M_1) &\leq \alpha\frac{n}{k}\cdot\frac{19}{20} + \frac{n}{40}. \end{aligned} \tag{4}$$

Finally, using the lower bound given by (3) with the upper bound in (4), we have

$$\begin{aligned} \frac{(1+\epsilon)n}{6} &\leq \alpha\frac{n}{k}\cdot\frac{19}{20} + k\cdot\frac{n}{40k} \\ \implies \alpha &\geq \frac{(23+20\epsilon)}{114}k. \end{aligned}$$

Now $\mathsf{rank}_\varphi(\sum_{(u,v)\in B_{1j}} x_u x_v) = \mathsf{be}_\varphi(B_{1j})$ as we have already explained. Hence we have $\mathsf{rank}_\varphi(h_1) \geq (\frac{n}{20k})^\alpha = \left(\frac{n}{k}\right)^{ck}$ for some constant $c > \frac{(23+20\epsilon)}{114}$ as required. On setting $w_1 = 1$ and w_2, w_3 to zero, we obtain the same rank lower bound for h.

□

4. Lower bounds

In this section we prove parameterized lower bounds for some special classes of syntactic multilinear ABPs computing the polynomial families defined in Section 3. In particular, we prove lower bounds for the size of ROABPs, strict interval ABPs and a sum of restricted class of ROPs for computing one or both of the hard polynomials we have defined in the previous section.

The general strategy of the lower bound is to obtain an equi-partition φ of the variables for which any polynomial computed by our chosen model attains the largest rank and then comparing this upper bound, which is usually in terms of the size s of the ABP, with the rank lower bound on the hard polynomials.

4.1. ROABP

In this section we prove a parameterized lower bound for the size of any ROABP computing the polynomials defined in Section 3. The lower bound argument follows from the fact that for any polynomial computed by an ROABP P, there exists an equi-partition φ of variables such that $\mathsf{rank}_\varphi(P)$ is bounded by the size of the ROABP [27].

Theorem 4.1. A ROABP P computing the polynomial family $f = (f_{n,2k})$ requires size

$$S = \Omega(n^k/(2k)^{2k}).$$

Proof:
Let P be an ROABP of size S computing f. Let the layers in P be $L_0, L_1, \ldots, L_n$, such that L_0 contains only the source node s and L_n contains only the terminal node t. We consider the order in which the variables are read from left to right in the ROABP as $x_1, x_2, \ldots, x_n$.

We can define the equi-partition $\varphi : X \to Y \cup Z$ given the above order, such that,

$$\varphi(x_i) = \begin{cases} y_i, \text{ if } i \leq n/2, \\ z_{i-n/2} \text{ otherwise.} \end{cases}$$

Let us consider the $n/2$th layer. By definition of P, the incoming edges to any layer L_i in P are labeled with a linear polynomial in x_i. At any node j in $L_{n/2}$, the paths from s to j are products of linear terms in variables v, $\varphi(v) \in Y$. The sum of these paths, which is computed at j, can be seen as a sub-program $[s, v_j]_P$. Similarly, the sum of the paths from j to t, computed at t, can be denoted by the sub-program $[v_j, t]_P$. Then, we can represent f as

$$f(x_1, \ldots, x_n) = \sum_{j \in L_{n/2}} [s, v_j]_P \cdot [v_j, t]_P.$$

By definition of φ, for all $v_j \in L_{n/2}$, every product $[s, v_j]_P \cdot [v_j, t]_P$ contributes 1 towards the rank of P. This is because every row-indexing monomial in M_{P^φ} corresponding to paths from s to j has non-zero entries corresponding to the same column-indexing monomials, corresponding to paths from j to t. So by Gaussian elimination, the product of sub-programs $[s, v_j]_P$ and $[v_j, t]$ contribute at most 1 to $\mathsf{rank}_\varphi(P)$. The number of such products in the expression for f is at most $|L_{n/2}|$, the number of nodes in the $(n/2)^{\text{th}}$ layer.

Then, $\mathsf{rank}_\varphi(f) \leq |L_{n/2}| \leq S$, S being a loose upper bound on the number of nodes in the $n/2$th layer. By Theorem 3.4, $\mathsf{rank}_\varphi(f) = \Omega(n^k/(2k)^{2k})$. Therefore we have $S = \Omega(n^k/(2k)^{2k})$ as required. □

Combining Theorem 4.1 with Theorem 3.6 we get:

Theorem 4.2. An ROABP computing the family of polynomials h defined in Section 3 requires size $n^{\Omega(k)}$.

Proof:
From the proof of Theorem 4.1, we see that for any size S ROABP computing the polynomial h, the equi-partition φ that maps the first $n/2$ variables to Y and the rest to Z ensures that $\mathsf{rank}_\varphi(h) \leq S$. Then by Theorem 3.6, we have $S = n^{\Omega(k)}$ as required. □

4.2. Separation Between Read-2 and Read-Once Oblivious ABPs

In this section, we show the power of reads of a variable in an oblivious ABP. To be clear, our family of hard polynomials h can be computed efficiently by oblivious ABPs if we allow two reads instead of one for every variable.

Theorem 4.3. The family of polynomials h defined in Section 3 can be computed by read-2 oblivious ABPs of size $n^{O(1)}$.

Proof:
According to the construction of h, we have three edge-disjoint perfect matchings M_1, M_2 and M_3. We will try to construct an ABP computing the polynomial f with as less number of reads as possible for each variable. We show that two reads per variable is enough.

Firstly, we construct a read-2 oblivious ABP P computing h_1, h_2 as defined in Section 3 and then add edges to P to compute h_3 without having to increase the number of layers in which a variable is read more than 2. Let $C_1, \ldots, C_\ell$ be the cycle cover given by the matchings M_1 and M_2 as used in the definition of h. Let us consider variables in the order in which they appear in each cycle in the cycle cover $C_1, \ldots, C_\ell$. We first construct the polynomials corresponding to the bags B_{11} and B_{21}. This can be done by one scan of the variables corresponding to $C_1, \ldots, C_{i1}$ where C_{i1} is the last cycle intersecting the bags B_{11} or B_{21}. Now we proceed to bags B_{12} and B_{22}. Continuing this way, we obtain an ABP P computing the polynomials h_1 and h_2 such that all variables except the pivots are read once. The pivot variables will however be read twice. Note that size of P is $O(n^2)$.

Example: Figure 2 will demonstrate the construction of the read-2 oblivious ABP P for M_1, M_2 i.e $w_1h_1 + w_2h_2$ where the set of vertices $V = \{1, 2, \ldots, 8\}$ and the perfect matchings $M_1 = \{(1,2),(3,4),(5,6),(7,8)\}$ and $M_2 = \{(1,4),(2,3),(6,7),(5,8)\}$. Hence, they form a cycle cover C_1, C_2 where $C_1 = (1,2,3,4,1)$, $C_2 = (7,8,5,6,7)$. Thus x_1, x_7 are pivot variables and are read twice, while the other variables are read once. The unlabeled edges can be assumed to be labeled by 1.

Now, we consider M_3. Recall that the blocks $B_{31}, \ldots, B_{3n/2k}$ respect the order on pivots given by $C_1, \ldots, C_\ell$. Now we compute the polynomials for each of the blocks in that order. Since M_3 has no edge between the pivots, we can build the polynomial for for B_{31} such that all pivots in B_{31} are aligned with their first read in P. For example, for the monomial px_i in h_{31}, we add edges to P in the following fashion. Suppose L is the layer in P where the variable p is being red for the first time. Now we compute px_i starting from layer L, first reading p and then the variable x_i in the next layer. Recall that P reads x_i only once and hence this read can be done anywhere in the program. One the the polynomial for block

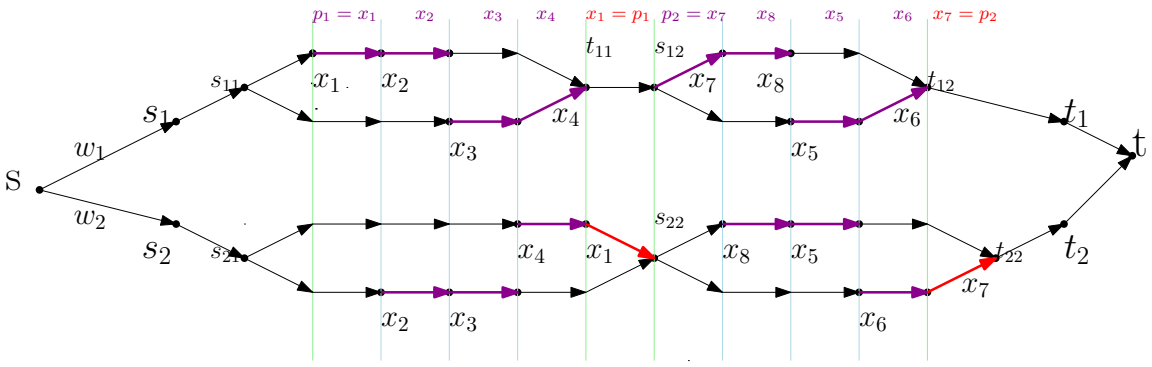

Figure 2. Parameterized read-2 OABP constructed from M_1, M_2

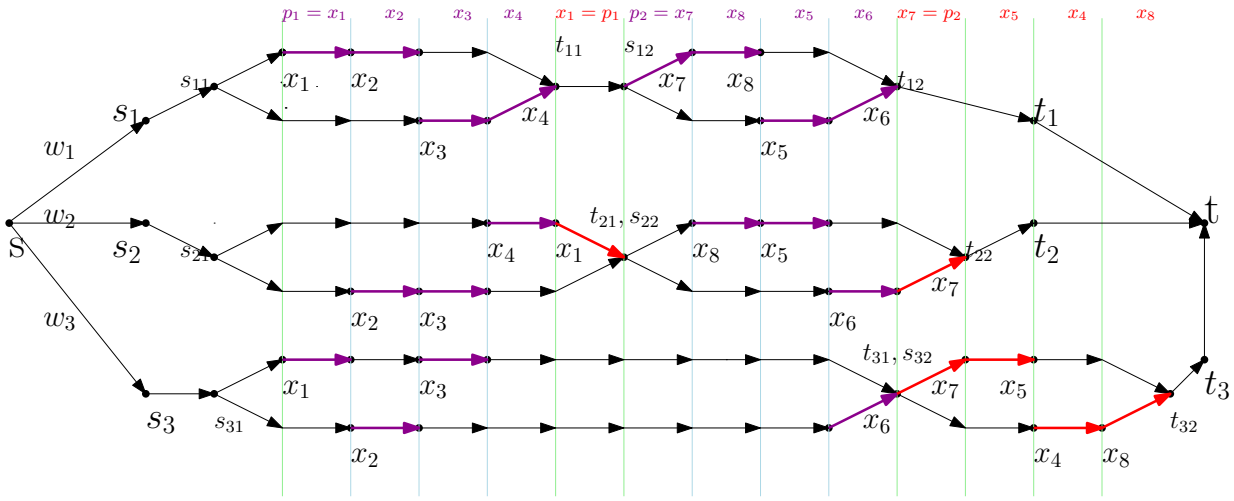

Figure 3. The parameterized read-2 OABP after including the sub-program for M_3

B_{31} is constructed, we proceed to compute h_{32} and so on. Throughout the construction, we only need to care about pivot variables in h_3 and align them suitably so that the number of layers that read the pivot variable is limited to two. Monomials that do not involve any pivot variable can be computed freely as needed so that every variable appears in at most two layers of the resulting program.

Figure 3 below demonstrates the complete construction of a parameterized read-2 oblivious ABP computing the polynomial h when the sub-program corresponding to the third matching $M_3 = \{(1,3),(2,6),(4,8),(5,7)\}$ is incorporated by our algorithm into the construction previously shown in Figure 2.

□

From the above theorem, the separation between read-once and read-twice oblivious ABP is clear.

Corollary 4.4. There is a parameterized polynomial family computable by polynomial size read-2 ABPs such that any ROABP computing it has size $n^{\Omega(k)}$ where k is the parameter.

Proof:
Follows from Theorems 4.2 and 4.3. □

4.3. Strict interval ABPs

In this section we prove a parameterized lower bound against the polynomial family f defined in Section 3 for the size of strict interval ABPs. Without loss of generality, assume that π is the identity permutation. Let P be a π strict-interval ABP computing the polynomial f.

As a crucial ingredient in the lower bound proof, we show that using the standard divide and conquer approach, a strict-interval ABP can be transformed into a depth four circuit with $n^{O(\sqrt{k})}$ blow up in the size. To begin with, we need a simple depth reduction for strict interval ABPs computing degree k polynomials. For that purpose, we first homogenize the strict-interval ABP:

Lemma 4.5. Let P be a syntactic multilinear ABP of size S computing a homogeneous degree k polynomial g on n variables. Then there is a syntactic multilinear ABP P' of depth $k+1$ and size $O(S \cdot k)$ computing g such that:

1. Every node in the i^{th} layer of P' computes a homogeneous degree i polynomial.
2. If P is strict interval then so is P'.

Proof:
Without loss of generality, we assume that P is homogeneous, i.e., for every node v in P, the polynomial $[s, v]_P$ is homogeneous, since homogenization of an ABP, first illustrated by [27], does not blow up the size of the ABP beyond a factor of the degree k. For every node v in P, let $\deg(v)$ be the degree of the polynomial $[s, v]_P$. We give a layer by layer construction of the program P'. Let L_i be the set of all nodes v in P such that $\deg(v) = i$, i.e., $L_i = \{v \mid \deg(v) = i\}$. The program P' has $k+1$ layers in addition to s and t, with i^{th} layer consisting of nodes L_i. The edges are added inductively as follows:

Base Case: Every node in L_0 computes a degree 0 polynomial, i.e., a constant. Add suitable edges from s to nodes in L_0.

Inductive Step: Suppose the branching program has been constructed upto layer L_{i-1} for $i \geq 1$. We add incoming edges to L_i as follows. For every node $v \in L_i$, with an incoming edge (u, v) in P, we have two possibilities,

Case 1 $u \in L_{i-1}$. Then, we add the edge (u, v) to P' with the same label as $\mathsf{label}(u, v)$.

Case 2 $u \in L_i$. In this case, we wait till all the incoming edges of u are processed. Then, for every incoming edge (u', u) in P' we add the edge (u', v) with the label $\mathsf{label}(u', u) \cdot \mathsf{label}(u, v)$.

There is a one-to-one correspondence between the nodes of P and that of P', since we have only moved nodes of P of degree $< i$ from L_i to a suitable layer preceding L_i, ensuring that nodes j in L_i (in P') compute only sub-programs $[s, v_j]_P$ of degree exactly i.

As the polynomial is of degree k, every node in P will result in the creation of at most k new nodes. Hence P' computes the same polynomial as P and the properties 1 and 2 hold as required. □

Using Lemma 4.5 we obtain the desired parameterized version of depth reduction to depth four circuits:

Lemma 4.6. Let $g(x_1, \ldots, x_n)$ be a multilinear polynomial of degree k computed by a syntactic multilinear branching program P of size S. Then

$$g(x_1, \ldots, x_n) = \sum_{i=1}^{T} \prod_{j=1}^{\sqrt{k}} f_{i,j} \tag{5}$$

for some $T = S^{O(\sqrt{k})}$ and $f_{i,j}$ is a degree $\sqrt{k}$ multilinear polynomial computed by a sub-program of P for $i \in \{1, \ldots, T\}$, $j \in \{1, \ldots, \sqrt{k}\}$.

Proof:
The construction follows from a simple divide and conquer subdivision of the program. By Lemma 4.5, we assume that P is homogeneous and has depth $k + 1$.

Let L_i be the set of nodes at layer i of the program P, for $0 \leq i \leq k+1$. We divide P into blocks of $\sqrt{k}$ layers. Each block is a collection of sub-programs between the nodes in the first and last of the $\sqrt{k}$ layers. Let $W = L_{\sqrt{k}} \times L_{2\sqrt{k}} \times \cdots \times L_{(\sqrt{k}-1)\sqrt{k}}$ be the total number of ways in which all these blocks can be aligned with each other. The final polynomial is sum, over all possible alignments, of the product of a sequence of $\sqrt{k}$ sub-programs, one from each block:

$$g(x_1, \ldots, x_n) = \sum_{(i_1, \ldots, i_{\sqrt{k}-1}) \in W} [s, i_1]_P \cdot \prod_{m=1}^{\sqrt{k}-2} [i_m, i_{m+1}]_P \cdot [i_{\sqrt{k}-1}, t]_P \tag{6}$$

Every sub-program is a sum of products of linear polynomials. Hence, looking at the expression, we note that it can be computed by a depth-4 circuit. Considering the number of nodes in each layer to be upper bounded by S, we have $|W| = T = S^{O(\sqrt{k})}$. Thus we are able to expand g as in the statement of the Lemma. □

Now to prove the claimed lower bound for the size of strict interval ABPs, all we need is given a polynomial f computed by an strict interval ABP of size S, an equi-partition φ of X such that $\mathsf{rank}_\varphi(f) << n^k$.

Lemma 4.7. Let f be a polynomial computed by a strict interval ABP of size S. Then there is a partition φ such that $\mathsf{rank}_\varphi(f) \leq S^{O(\sqrt{k})} n^{\sqrt{k}}$.

Proof:
Without loss of generality, assume that P is a strict interval ABP with respect to the identity permutation. Let $\varphi_{\mathsf{mid}} : X \to Y \cup Z$ be a suitable equi-partition such that,

$$\varphi_{\mathsf{mid}}(x_i) = \begin{cases} y_i, \text{ if } i \leq n/2, \\ z_{i-n/2} \text{ otherwise.} \end{cases}$$

We consider the representation for f as in (6). Then for every $1 \leq i \leq T$, for all but one m, we have either $\varphi_{\mathsf{mid}}(\mathsf{var}([i_m, i_{m+1}])) \subseteq Y$ or $\varphi_{\mathsf{mid}}(\mathsf{var}([i_m, i_{m+1}])) \subseteq Z$. Now, considering (5), $\mathsf{rank}_\varphi(f_{ij}) \leq n$ as there are n possibilities for a variable in $\mathsf{var}([i_m, i_{m+1}]_P)$ which is mapped to a different partition by φ_{mid} than the other variables, such that the sub-program $[i_m, i_{m+1}]_P$ contributes a rank of 1.

Therefore, $\mathsf{rank}_{\varphi_{\mathsf{mid}}}([s, i_1]_P \cdot \prod_{m=1}^{\sqrt{k}-2} [i_m, i_{m+1}]_P \cdot [i_{\sqrt{k}-1}, t]_P) \leq n^{\sqrt{k}}$, for every $i_j \in L_{j\sqrt{k}}$.

By sub-additivity of rank_φ, we have $\mathsf{rank}_\varphi(f) \leq S^{O(\sqrt{k})} n^{\sqrt{k}}$ for $\varphi = \varphi_{\mathsf{mid}}$. □

The required lower bound is immediate now.

Corollary 4.8. Any strict-interval ABP computing the polynomial f has size $n^{\Omega(\sqrt{k})}$.

Proof:
Follows from Theorem 3.4 and Lemma 4.7. □

4.4. Rank bound for ROPs by Graph representation

The reader might be tempted to believe that the lower bound arguments in the preceding sections might be applicable to more general models such as sum of ROFs and sum of ROABPs or even multilinear formulas. However, as we have seen in Section 3, there is a sum of three ROFs that has high rank under every partition. Thus our approach using rank_φ as a complexity measure is unlikely to yield lower bounds for even sum of ROFs, which is in contrast to the classical setting, where exponential lower bounds against models such as sum of ROFs and sum of ROABPs follow easily.

In this section, we develop a new method of analyzing rank of degree k polynomials computed by ROFs. We look at the order in which variables appear in the in-order traversal of an ROF. Thus we read the variables in the sum of ROFs with a restricted ordering.

Let $p \in \mathbb{F}[X]$ be the polynomial computed by a ROF Φ. We want to construct a graph $G_p = (X, E_p)$ corresponding to p so that $\mathsf{rank}_\varphi(p)$ can be related to certain parameters of the graph. For this, we add edges or a sequence of edges to the graph according to the type of polynomial each gate computes. We define some types of gates in the formula as follows.

Definition 4.9. (Types of gates in a ROF) Let Φ be a ROF. A gate v in Φ is said to be a *maximal-degree-two gate* if v computes a degree two polynomial, and the parent of v computes a polynomial whose degree is strictly greater than two.

A gate v is said to be a *maximal-degree-one* gate if v computes a linear form and the parent of v computes a polynomial of degree strictly greater than one.

A gate v at depth 1 is said to be a *high degree gate* if the degree of the polynomial computed at v is strictly greater than two.

Let V_2 denote the set of all maximal-degree-two gates in Φ, V_1 denote the set of all maximal-degree-one gates and V_0 denote the set of all high degree gates in Φ at depth one. Let $\mathsf{atomic}(\Phi) = V_0 \cup V_1 \cup V_2$. The following is a straightforward observation:

Observation 1. Let Φ be an ROF and v be a maximal-degree-two gate in Φ. Then the polynomial computed by Φ_v is of the form $\Phi_v = \sum_{i=1}^{s} \ell_{i_1}\ell_{i_2}$, where ℓ_{i_j} $1 \leq i \leq s$, $j \in \{1,2\}$ are variable disjoint linear forms for some $s > 0$ such that each of the ℓ_{i_j} is dependent on at least one variable.

Defining paths and constructing G_p

For a linear form $\ell = \sum_{j=1}^{r} \alpha_{i_j} x_{i_j}$, let $\mathsf{path}(\ell)$ be the simple undirected path comprised of edges (x_{i_1}, x_{i_2}), $(x_{i_2}, x_{i_3}), \ldots, (x_{i_{r-1}}, x_{i_r})$.

In the case when $r = 1$, $\mathsf{path}(\ell)$ is just single vertex. Similarly, for a subset $S \subseteq X$ of variables, let $\mathsf{path}(S)$ denote the path constituted by the edges (x_{i_1}, x_{i_2}), $(x_{i_2}, x_{i_3}), \ldots,$ $(x_{i_{r-1}}, x_{i_r})$ where $S = \{x_{i_1}, \ldots, x_{i_r}\}$, $i_1 < i_2 < \ldots < i_r$.

For two variable disjoint linear forms ℓ and ℓ', let $\mathsf{path}(\ell, \ell')$ be the path obtained by connecting the last vertex in $\mathsf{path}(\ell)$ to the first vertex of $\mathsf{path}(\ell')$ by a new edge.

Now, we define a graph $G_p = (X, E_p)$ where vertices correspond to variables $x_u \in X$ and the set of edges E_p defined as follows.

For each $v \in \mathsf{atomic}(\Phi)$ we add the following edges to E_p:

Case 1 $\Phi_v = \sum_{i=1}^{r} \ell_{i_1}\ell_{i_2}$ for some $r > 0$ add $\mathsf{path}(\ell_{i_1}, \ell_{i_2})$ to G_p for every $1 \leq i \leq t$.

Case 2 $\Phi_v = \prod_{i \in S} x_i$ or $\Phi_v = \sum_{i \in S} c_i x_i$, where $S \subseteq X$, c_is are constants from $\mathbb{F}$, add $\mathsf{path}(S)$ to G_p.

It may be noted that the graph G_p is not unique as it depends on the given minimal ROF Φ computing f. Now that we have an underlying graph, we view the equi-partition φ on the variables X of the polynomial as a coloring, and analyse the rank of the polynomial computed by Φ using the measure of number of bichromatic edges, as defined in Section 3.

Lower bound using G_p

In the following, we show that for a given partition φ, we bound the $\mathsf{rank}_\varphi(p)$ in terms of the number of bichromatic edges $\mathsf{be}_\varphi(G_p)$.

Theorem 4.10. Let $p \in \mathbb{F}[x_1, \ldots, x_n]$ be a multilinear polynomial of degree k computed by a ROF Φ. Then, for any equi-partition $\varphi : X \to Y \cup Z$, $\mathsf{rank}_\varphi(p) \leq (4\mathsf{be}_\varphi(G_p))^{\frac{k}{2}}$.

Proof:
The proof is by induction on the structure of Φ. The base case is when the root gate of Φ is in $\mathsf{atomic}(\Phi)$. We prove the bound for all the three kinds of gates in $\mathsf{atomic}(\Phi)$ and the inductive argument follows.

Consider a gate $v \in \mathsf{atomic}(\Phi)$.

Case 1 $\Phi_v = \sum_{(i,j)\in S} x_i x_j$. If $\varphi(x_i),\ \varphi(x_j)$ are not in the same partition, then each monomial $x_i x_j$ contributes 1 towards $\mathsf{rank}_\varphi(p)$. At the same time, the edge (x_i, x_j) added to E_p is bichromatic, so each monomial contributes 1 towards the measure $\mathsf{be}_\varphi(G_p)$ as well.

Case 2 $\Phi_v = \sum_{(a,b)\in T} \ell_a \ell_b$. If, for some $x_i, x_j \in \mathsf{var}(\ell_a)$, $\varphi(x_i), \varphi(x_j)$ are in different partitions, then the linear form ℓ_a contributes 2 towards $\mathsf{rank}_\varphi(\ell_a)$. If the same holds true for ℓ_b, then $\ell_a \ell_b$ would together contribute 4 towards $\mathsf{rank}_\varphi(p)$ and ≥ 2 towards the measure $\mathsf{be}_\varphi(G_p)$.

Case 3 $\Phi_v = \sum_{i\in W_1} c_i x_i$ or $\Phi_v = \prod_{i\in W_2} x_i$ for some $W_1, W_2 \subseteq X$. The first case has been considered already. For the second case, if $\exists x_a, x_b \in W_2$ such that $\varphi(x_a), \varphi(x_b)$ are in different partitions, the polynomial computed by the gate v will contribute a 1 towards $\mathsf{rank}_\varphi(p)$ and at least 1 towards $\mathsf{be}_\varphi(G_p)$, otherwise it contributes 0 towards both measures.

Thus we have verified that the statement is true when the root gate v of Φ is contained in $\mathsf{atomic}(\Phi)$. Suppose $p = p_1\ \mathsf{op}\ p_2$ for $\mathsf{op} \in \{+, \times\}$ where p_1 and p_2 are variable disjoint and are computed by ROFs. By induction hypothesis, $\mathsf{rank}_\varphi(p_j) \leq (4\mathsf{be}_\varphi(G_{p_j}))^{\frac{k_j}{2}}$ where $k_j = \deg(f_j)$. As $\mathsf{be}_\varphi(G_p) = \mathsf{be}_\varphi(G_{p_1}) + \mathsf{be}_\varphi(G_{p_2})$ and $k = k_1 + k_2$ ($\mathsf{op} = \times$) or $k = \max\{k_1, k_2\}$ ($\mathsf{op} = +$) we have, $\mathsf{rank}_\varphi(f) \leq (4\mathsf{be}_\varphi(G_p))^{\frac{k}{2}}$ as required. □

Recall that the bisection of an undirected graph $G = (V, E)$ is a set $S \subseteq V$ such that $|S| = |V|/2$. The size of a bisection S is the number of edges across S and $\overline{S}$, i.e., $|\{(u, v) \mid (u, v) \in E, u \in S, v \notin S\}|$. The following is an immediate corollary to Theorem 4.10:

Theorem 4.11. Let G be a graph on n vertices such that there is a bisection of G of size $n^{1-\epsilon}$. Suppose $p_1, \ldots, p_s$ be ROFs such that G_{p_i} is a sub-graph of G. Then, if $f = p_1 + \cdots + p_S$ we have $S = (n^{\Omega(k)}/t(k))$, where t is a computable function on k.

Proof:
Let $C = (S, \overline{S})$ be the bisecting cut and $\mathsf{size}(C)$ denote the number of edges across the cut. We fix an equi-partition $\varphi : X \to Y \cup Z$ as follows:

$$\varphi(x_i) \in \begin{cases} Y, & \text{if } i \in S, \\ Z, & \text{otherwise.} \end{cases}$$

Then by Theorem 4.10, $\mathsf{rank}_\varphi(p_i) \leq (4\mathsf{be}_\varphi(G_{p_i}))^{\frac{k}{2}}$. Since G_{p_i} is a sub-graph of G, we have $\mathsf{be}_\varphi(G_p)) \leq \mathsf{size}(C) \leq n^{1-\epsilon}$. Therefore, $\mathsf{rank}_\varphi(p_i) \leq O_k(n^{(1-\epsilon)k/2})$. By sub-additivity, we have $\mathsf{rank}_\varphi(f) \leq SO_k(n^{(1-\epsilon)k/2})$ where O_k is upto a factor that depends only on a function of k. By Theorem 3.4, we get $S = \Omega(n^{\epsilon k/2})$. □

Conclusions

Our results demonstrate the challenges in translating classical arithmetic circuit lower bounds to the parameterized setting, when the degree of the polynomial is the parameter. We get a full rank polynomial that can be computed by depth four arithmetic circuits of fpt size, whereas in the classical setting, full rank polynomials cannot be computed by multilinear formulas of polynomial size [17].

This makes the task of proving parameterized lower bounds for algebraic computation much more challenging task. Given the application of polynomials, whose degree is bound by a parameter, in the design of efficient parameterized algorithms for many counting problems, we believe that this is a worthy research direction to pursue.

Further, we believe that our results are an indication that study of parameterized complexity of polynomials with degree as the parameter could possibly shed more light on the use of algebraic techniques in parameterized algorithms.

References

[1] Valiant LG. The Complexity of Computing the Permanent. *Theor. Comput. Sci.*, 1979. **8**:189–201. doi:10.1016/0304-3975(79)90044-6. URL https://doi.org/10.1016/0304-3975(79)90044-6.

[2] Saptharishi R, Chillara S, Kumar M. A survey of lower bounds in arithmetic circuit complexity. Technical report, 2016. URL https://github.com/dasarpmar/lowerbounds-survey/releases.

[3] Shpilka A, Yehudayoff A. Arithmetic Circuits: A Survey of Recent Results and Open Questions. *Foundations and Trends® in Theoretical Computer Science*, 2010. **5**(3—4):207–388. doi:10.1561/0400000039. URL http://dx.doi.org/10.1561/0400000039.

[4] Baur W, Strassen V. The Complexity of Partial Derivatives. *Theor. Comput. Sci.*, 1983. **22**:317–330. doi:10.1016/0304-3975(83)90110-X. URL https://doi.org/10.1016/0304-3975(83)90110-X.

[5] Grochow JA, Mulmuley KD, Qiao Y. Boundaries of VP and VNP. In: Chatzigiannakis I, Mitzenmacher M, Rabani Y, Sangiorgi D (eds.), 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy, volume 55 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016 pp. 34:1–34:14. doi:10.4230/LIPIcs.ICALP.2016.34. URL https://doi.org/10.4230/LIPIcs.ICALP.2016.34.

[6] Downey RG, Fellows MR. Parameterized Complexity. Monographs in Computer Science. Springer, 1999. ISBN 978-1-4612-6798-0. doi:10.1007/978-1-4612-0515-9. URL https://doi.org/10.1007/978-1-4612-0515-9.

[7] Engels C. Why are certain polynomials hard?: A look at non-commutative, parameterized and homomorphism polynomials. Ph.D. thesis, Saarland University, 2016. URL https://nbn-resolving.org/urn:nbn:de:bsz:291-scidok-64387.

[8] Müller M. Parameterized Randomization. Ph.D. thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, 2008. URL https://d-nb.info/993356915/34.

[9] Kabanets V, Impagliazzo R. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Comput. Complex.*, 2004. **13**(1-2):1–46. doi:10.1007/s00037-004-0182-6. URL https://doi.org/10.1007/s00037-004-0182-6.

[10] Björklund A. Exact Covers via Determinants. In: Marion J, Schwentick T (eds.), 27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France, volume 5 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010 pp. 95–106. doi:10.4230/LIPIcs.STACS.2010.2447. URL https://doi.org/10.4230/LIPIcs.STACS.2010.2447.

[11] Amini O, Fomin FV, Saurabh S. Counting Subgraphs via Homomorphisms. *SIAM J. Discrete Math.*, 2012. **26**(2):695–717. doi:10.1137/100789403. URL https://doi.org/10.1137/100789403.

[12] Fomin FV, Lokshtanov D, Raman V, Saurabh S, Rao BVR. Faster algorithms for finding and counting subgraphs. *J. Comput. Syst. Sci.*, 2012. **78**(3):698–706. doi:10.1016/j.jcss.2011.10.001. URL http://dx.doi.org/10.1016/j.jcss.2011.10.001.

[13] Björklund A, Husfeldt T, Taslaman N. Shortest cycle through specified elements. In: Rabani Y (ed.), Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012. SIAM, 2012 pp. 1747–1753. doi:10.1137/1.9781611973099.139. URL https://doi.org/10.1137/1.9781611973099.139.

[14] Gupta A, Kamath P, Kayal N, Saptharishi R. Arithmetic Circuits: A Chasm at Depth Three. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA. IEEE Computer Society, 2013 pp. 578–587. doi:10.1109/FOCS.2013.68. URL https://doi.org/10.1109/FOCS.2013.68.

[15] Fournier H, Limaye N, Malod G, Srinivasan S. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In: Shmoys DB (ed.), Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014. ACM, 2014 pp. 128–135. doi:10.1145/2591796.2591824. URL https://doi.org/10.1145/2591796.2591824.

[16] Kumar M, Saraf S. The Limits of Depth Reduction for Arithmetic Formulas: It's All About the Top Fan-In. *SIAM J. Comput.*, 2015. **44**(6):1601–1625. doi:10.1137/140999220. URL https://doi.org/10.1137/140999220.

[17] Raz R. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 2009. **56**(2):8:1–8:17. doi:10.1145/1502793.1502797. URL https://doi.org/10.1145/1502793.1502797.

[18] Raz R. Separation of Multilinear Circuit and Formula Size. *Theory of Computing*, 2006. **2**(6):121–135. doi:10.4086/toc.2006.v002a006. URL https://doi.org/10.4086/toc.2006.v002a006.

[19] Raz R, Yehudayoff A. Balancing Syntactically Multilinear Arithmetic Circuits. *Comput. Complex.*, 2008. **17**(4):515–535. doi:10.1007/s00037-008-0254-0. URL https://doi.org/10.1007/s00037-008-0254-0.

[20] Chillara S, Engels C, Limaye N, Srinivasan S. A Near-Optimal Depth-Hierarchy Theorem for Small-Depth Multilinear Circuits. In: Thorup M (ed.), 59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018. IEEE Computer Society, 2018 pp. 934–945. doi:10.1109/FOCS.2018.00092. URL https://doi.org/10.1109/FOCS.2018.00092.

[21] Kayal N, Nair V, Saha C. Separation Between Read-once Oblivious Algebraic Branching Programs (ROABPs) and Multilinear Depth Three Circuits. In: Ollinger N, Vollmer H (eds.), 33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France, volume 47 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016 pp. 46:1–46:15. doi:10.4230/LIPIcs.STACS.2016.46. URL https://doi.org/10.4230/LIPIcs.STACS.2016.46.

[22] Chen Z, Fu B. Approximating multilinear monomial coefficients and maximum multilinear monomials in multivariate polynomials. *J. Comb. Optim.*, 2013. **25**(2):234–254. doi:10.1007/s10878-012-9496-5. URL https://doi.org/10.1007/s10878-012-9496-5.

[23] Chen Z, Fu B, Liu Y, Schweller RT. On testing monomials in multivariate polynomials. *Theor. Comput. Sci.*, 2013. **497**:39–54. doi:10.1016/j.tcs.2012.03.038. URL https://doi.org/10.1016/j.tcs.2012.03.038.

[24] Chauhan A, Rao BVR. Parameterized Analogues of Probabilistic Computation. In: Ganguly S, Krishnamurti R (eds.), Algorithms and Discrete Applied Mathematics - First International Conference, CALDAM 2015, Kanpur, India, February 8-10, 2015. Proceedings, volume 8959 of *Lecture Notes in Computer Science*. Springer, 2015 pp. 181–192. doi:10.1007/978-3-319-14974-5_18. URL https://doi.org/10.1007/978-3-319-14974-5_18.

[25] Ghosal P, Prakash O, Rao BVR. On Constant Depth Circuits Parameterized by Degree: Identity Testing and Depth Reduction. In: Cao Y, Chen J (eds.), Computing and Combinatorics - 23rd International Conference, COCOON 2017, Hong Kong, China, August 3-5, 2017, Proceedings, volume 10392 of *Lecture Notes in Computer Science*. Springer, 2017 pp. 250–261. doi:10.1007/978-3-319-62389-4_21. URL https://doi.org/10.1007/978-3-319-62389-4_21.

[26] Arvind V, Raja S. Some Lower Bound Results for Set-Multilinear Arithmetic Computations. *Chicago J. Theor. Comput. Sci.*, 2016. **2016**. URL http://cjtcs.cs.uchicago.edu/articles/2016/6/contents.html.

[27] Nisan N. Lower Bounds for Non-Commutative Computation (Extended Abstract). In: Koutsougeras C, Vitter JS (eds.), Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA. ACM, 1991 pp. 410–418. doi:10.1145/103418.103462. URL https://doi.org/10.1145/103418.103462.

[28] Hoory S, Linial N, Wigderson A. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 2006. **43**(4):439–561. doi:10.1090/S0273-0979-06-01126-8. URL `https://doi.org/10.1090/S0273-0979-06-01126-8`.