Exact Parameterized Multilinear Monomial Counting via k-Layer Subset Convolution and k-Disjoint Sum

Dongxiao Yu[1], Yuexuan Wang[2], Qiang-Sheng Hua[2,1], and Francis C.M. Lau[1]

[1] Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong, P.R. China
[2] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, P.R. China
{dxyu,fcmlau}@cs.hku.hk, {qshua,wangyuexuan}@tsinghua.edu.cn

Abstract. We present new algorithms for exact multilinear k-monomial counting which is to compute the sum of coefficients of all degree-k multilinear monomials in a given polynomial P over a ring R described by an arithmetic circuit C. If the polynomial can be represented as a product of two polynomials with degree at most $d < k$, our algorithm can solve this problem in $O^*(\binom{n}{\downarrow d})$ time, where $\binom{n}{\downarrow d} = \sum_{i=0}^{d}\binom{n}{i}$. O^* omits a polynomial factor in n. For the general case, the proposed algorithm takes time $O^*(\binom{n}{\downarrow k})$. In both cases, our results are superior to previous approaches presented in [Koutis, I. and Williams, R.: Limits and applications of group algebras for parameterized problems. *ICALP*, pages 653-664 (2009)]. We also present a polynomial space algorithm with time bound $O^*(2^k\binom{n}{k})$. By reducing the $\#k$-path problem and the $\#m$-set k-packing problem to the exact multilinear k-monomial counting problem, we give algorithms for these two problems that match the fastest known results presented in [2].

1 Introduction

The multilinear k-monomial detection problem is a well studied problem from the algorithmic standpoint. A number of its applications in solving combinatorial parameter problems have been presented [12,16,13]. The multilinear k-monomial detection technique has been shown to be superior in solving parameter problems to older approaches such as color coding [1] and divide-and-color [5,11]. In [13], Koutis and Williams studied the technique for solving a related counting problem—exact multilinear k-monomial counting—and showed its power in solving parameter counting problems by reducing the $\#m$-set k-packing problem to this counting problem. With the reduction, they gave a faster algorithm for the $\#m$-set k-packing problem.

Definition 1. *(Exact multilinear k-monomial counting [13]) Given a (commutative) arithmetic circuit C describing an n-variate polynomial P over a ring R, compute the sum of the coefficients of degree-k multilinear monomials in P.*

B. Fu and D.-Z. Du (Eds.): COCOON 2011, LNCS 6842, pp. 74–85, 2011.

An arithmetic circuit C over a ring R and the set of variables $x_1, \ldots, x_n$ is a directed acyclic graph. Every node with in-degree zero is called an input gate and is labeled by either a variable x_i or an element in R. Every other gate is labeled by either $+$ or $\times$, for an addition gate or a product gate respectively. The size of the circuit C, denoted by $|C|$, is defined as the number of gates in C. If there is no confusion, for each operator gate, we call its in-neighbors input gates.

As will be shown later in this paper (see Section 5.1), the $\#k$-path problem can be reduced to the multilinear k-monomial counting problem. Hence, the $\#W[1]$-hardness of the $\#k$-path problem with respect to the parameter k [8] implies that it is unlikely that an $O(f(k)poly(n))$ time algorithm for the exact multilinear k-monomial counting problem can be found. By making use of an algebraic method, Koutis and Williams in [13] gave an algorithm with running time $O^*(2^d\binom{n}{d}+n^{\lfloor k/2 \rfloor})$ for a special case where the polynomial P is a product of two polynomials P_1, P_2 with degrees at most d. Their algorithm performs much worse for the general case, with a time complexity of $O^*(2^k\binom{n}{k})$. However, some recent results indicate that one could do better. For example, Björklund et al. in [16] gave an $O^*(\binom{n}{\lceil k/2 \rceil})$ algorithm for the $\#k$-path problem. In this paper, we affirmatively show that an improvement can be achieved, via the following result.

Theorem 1. *Let $P, Q \in R[X]$ be n-variate polynomials over a ring R described by arithmetic circuits and P, Q contain only multilinear monomials of degree at most d. The sum of coefficients of all multilinear k-terms in $P \cdot Q$ can be computed in $O^*(\binom{n}{\downarrow d})$ time, where $\binom{n}{\downarrow d} = \sum_{i=0}^{d}\binom{n}{d}$.*

For the general case, our algorithm operates in time $O^*(\binom{n}{\downarrow k})$ which is also superior to the result in [13]. Furthermore, based on the novel circuit constructed (see Section 3) which combines the k-layer subset convolution, we can give a polynomial space algorithm with running time $O^*(2^k\binom{n}{k})$.

The application of the multilinear monomial counting technique has partly been demonstrated in [13]. In this work, we further illustrate the advantages and applicability of this new technique. By reducing the $\#k$-path problem and the $\#m$-set k-packing problem to the multilinear k-monomial counting problem, our new proposed approach can yield parameterized algorithms for these two problems matching the state-of-the-art results in [2].

1.1 Related Work

In this work, in order to compute faster the sum of coefficients of all degree-k multilinear monomials in the given polynomial, we transform the given polynomial circuit into a new circuit consisting of some novel combinatorial operation gates. The idea of applying combinatorial techniques to algebraic circuits first appeared in a recent paper [14] on designing polynomial space exact algorithms for several combinatorial problems. However, the underlying idea in [14] is to perform some combinatorial operations in a given algebraic framework rather

than to intentionally accelerate the computation in a polynomial circuit by introducing some novel gates, such as k-layer subset convolution.

The $\#k$-path problem is to count the number of simple paths of length k in a graph G of n vertices V and m edges E. As a generalization of the classical Hamiltonion path problem, the decision version of the k-path problem has been well studied [1,5,11,12,16]. However, for the counting version, existing research is much less extensive possibly because the parameterized problem of counting k-paths appears to be harder than its decision version. The $\#W[1]$-hardness was proved by Flum and Grohe in [8], which implies that it is unlikely to find an $O(f(k)poly(n))$ algorithm for the $\#k$-path problem. Recently, Vassilevska and Williams [15] have shown that k-edge paths can be counted in time $O^*((\frac{k}{2})!\binom{n}{k/2})$. To the best of our knowledge, the fastest known result for $\#k$-path problem was given by Björklund et al. [2] with running time $O^*(\binom{n}{\lceil k/2 \rceil})$.

The situations of set packing problems were similar. Deciding whether a given family of N subsets of an n-element universe contains a k-packing (no two sets in the k sets intersect) is also known to be $W[1]$-hard [7]. For the m-set k-packing problem where each subset contains exactly m elements, Jia et al. [9] showed that the decision problem is fixed parameter tractable with respect to the total size mk of the packing. Koutis [12] and Chen et al. [5] gave faster algorithms with running time $O^*(c^{mk})$ for some constant c. For the counting m-set k-packing problem, Koutis and Williams [13] gave an algorithm with time complexity $O^*(n^{\lceil mk/2 \rceil})$ by reducing the problem to the exact multilinear k-monomial counting problem. Based on the inclusion-exclusion principle and the meet-in-the-middle approach, Björklund et al. in [2] gave the fastest known result with time complexity $O^*(\binom{n}{\lceil mk/2 \rceil})$.

2 Preliminaries

2.1 Zeta Transform and Möbius Inversion

Given a universe set $U = \{1, 2, \ldots, n\}$ and a collection F of subsets of U, let $f : F \longrightarrow R$. The *up-zeta transform* $f\zeta^{\uparrow}$ [2] for all $X \subseteq U$ is defined by

$$f\zeta^{\uparrow}(X) = \sum_{X \subseteq Y} f(Y). \tag{1}$$

The *down-zeta transform* $f\zeta^{\downarrow}$ [4] for all $X \subseteq U$ is defined by

$$f\zeta^{\downarrow}(X) = \sum_{Y \subseteq X} f(Y). \tag{2}$$

Given the down-zeta transform $f\zeta^{\downarrow}$, the original function f may be recovered via the *Möbius inversion* formula

$$f(Y) = \sum_{X \subseteq Y} (-1)^{|Y \setminus X|} f\zeta^{\downarrow}(X). \tag{3}$$

The *fast up-zeta transform* [2], the *fast down-zeta transform* [17,10] and the *fast Möbius inversion* [3] are efficient algorithms for computing the up-zeta transform, the down-zeta transform and the Möbius inversion respectively. For a collection F of subsets of U, define $\downarrow F$ as the set of subsets of U having supersets in F. Based on the "trimmed" fast zeta transform presented in [4][2] which only considers the subsets in $\downarrow F$, the following lemma holds.

Lemma 1. *There exist algorithms that construct, given $G \subseteq 2^U$ and $F \subseteq 2^U$ as input, an R-arithmetic circuit with input gates for $f : F \to R$ and output gates that evaluate to*
(1) $f\zeta^{\uparrow} : G \to R$, with time $O^*(|\downarrow F| + |G|)$;
(2) $f\zeta^{\downarrow} : G \to R$, with time $O^*(|F| + |\downarrow G|)$;

2.2 Subset Convolution

Definition 2 (Subset convolution [3]). *Let U be a set of n elements and R be a ring, and let f and g be functions that associate with every subset $S \subseteq U$ an element of the ring R. The operator subset convolution $*_R$ over R for all $S \subseteq U$ is defined as follows,*

$$f *_R g(S) = \sum_{X \subseteq S} f(X)g(S \setminus X). \tag{4}$$

In [3], the authors presented an algorithm called *fast subset convolution* to solve the subset convolution problem in $O^*(2^n)$ time. The whole algorithm can be divided into three parts: First, it calculates the ranked down-zeta transform of f, g defined in the following formula using the fast down-zeta transform independently for $k = 0, 1, 2, \ldots, n$ in $O^*(2^n)$ time.

$$f\zeta^{\downarrow}(k, X) = \sum_{Y \subseteq X, |Y|=k} f(Y). \tag{5}$$

Second, it calculates $f\zeta^{\downarrow} \circledast g\zeta^{\downarrow}(k, S)$ defined as below for all $k = 0, 1, \ldots, n$ and $S \subseteq U$.

$$f\zeta^{\downarrow} \circledast g\zeta^{\downarrow}(k, S) = \sum_{j=0}^{k} f\zeta^{\downarrow}(j, S)g\zeta^{\downarrow}(k - j, S). \tag{6}$$

Finally, $f *_R g$ can be expressed as the Möbius inversion of $f\zeta^{\downarrow} \circledast g\zeta^{\downarrow}$.

$$f *_R g(S) = \sum_{X \subseteq S} (-1)^{|S \setminus X|} f\zeta^{\downarrow} \circledast g\zeta^{\downarrow}(|S|, X). \tag{7}$$

Then $f *_R g$ can be obtained by taking advantage of the fast Möbius inversion in $O^*(2^n)$ time. For more details, please refer to [3].

3 Faster Algorithms for Exact Multilinear k-Monomial Counting

3.1 k-Disjoint Sum

Definition 3. *(k-disjoint sum) Given a universe set U and two collections F_1 and F_2 of subsets of U, α and β are functions defined on F_1 and F_2, respectively. The goal is to compute the sum $\alpha \boxtimes \beta$ of products $\alpha(S_1)\beta(S_2)$ over all disjoint pairs of subsets (S_1, S_2) such that $|S_1|+|S_2| = k$ in the Cartesian product $F_1 \times F_2$.*

An obvious way for computing the k-disjoint sum will take $[\binom{n}{\downarrow k}]^2$ time in the worst case if we just simply compare each possible pair of sets in $F_1 \times F_2$, where $\binom{n}{\downarrow k} = \sum_{i=0}^{k} \binom{n}{k}$. By introducing a new convolution and generalizing the idea in [2], we present next a new way to complete the computation efficiently.

Let $F \subseteq 2^U$ and $f : F \longrightarrow R$. The *ranked up-zeta transform* $f\zeta^{\uparrow}(t, X)$ for every $t = 0, 1, \ldots, n$ and $X \subseteq U$ is defined by

$$f\zeta^{\uparrow}(t, X) = \sum_{X \subseteq S, |S|=t} f(S). \tag{8}$$

It is easy to see that the ranked up-zeta transform can be computed by performing the fast up-zeta transform independently for every t via setting the function value as 0 for all sets $S \in F$ with $|S| \neq t$.

For two ranked up-zeta transforms $\alpha\zeta^{\uparrow}$ and $\beta\zeta^{\uparrow}$, define the convolution $\alpha\zeta^{\uparrow} \circledast \beta\zeta^{\uparrow}$ for all $k = 0, 1, \ldots, n$ and $X \subseteq U$ by

$$\alpha\zeta^{\uparrow} \circledast \beta\zeta^{\uparrow}(k, X) = \sum_{k_1=0}^{k} \alpha\zeta^{\uparrow}(k_1, X)\beta\zeta^{\uparrow}(k - k_1, X). \tag{9}$$

Note that the convolution operation is over the rank parameter rather than the subset parameter.

Lemma 2. *k-disjoint sum can be computed in $O(kn\binom{n}{\downarrow k})$ time.*

Proof.

$$\begin{aligned}
\alpha \boxtimes \beta &= \sum_{A \in F_1} \sum_{B \in F_2} [A \cap B = \emptyset][|A| + |B| = k]\alpha(A)\beta(B) \\
&= \sum_{A \in F_1} \sum_{B \in F_2} \sum_{X} (-1)^{|X|}[X \subseteq A \cap B][|A| + |B| = k]\alpha(A)\beta(B) \\
&= \sum_{X} (-1)^{|X|} \sum_{A \in F_1} \sum_{B \in F_2} [X \subseteq A][X \subseteq B][|A| + |B| = k]\alpha(A)\beta(B) \\
&= \sum_{X} (-1)^{|X|} \sum_{k_1=0}^{k} \sum_{A \in F_1} [X \subseteq A][|A| = k_1]\alpha(A) \sum_{B \in F_2} [X \subseteq B][|B| = k - k_1]\beta(B) \\
&= \sum_{X} (-1)^{|X|} \alpha\zeta^{\uparrow} \circledast \beta\zeta^{\uparrow}(k, X).
\end{aligned} \tag{10}$$

The second equation comes from the fact that every nonempty set has exactly the same number of subsets of even size and subsets of odd size. By the above formula, we can evaluate $\alpha \boxtimes \beta$ by first computing the ranked up-zeta transforms $\alpha\zeta^{\uparrow}$ and $\beta\zeta^{\uparrow}$ for every $t = 0, 1, \ldots, k$ over F_1 and F_2, respectively, then taking the convolution (9), and finally evaluating the formula (10).

Here we only need to consider the sets X with cardinality at most t for each $t \leq k$ in the process of computing the ranked up-zeta transforms $\alpha\zeta^{\uparrow}$ and $\beta\zeta^{\uparrow}$, since other sets have no supersets with cardinality t in F_1 and F_2. By Lemma 1, we can obtain $\alpha\zeta^{\uparrow}$ and $\beta\zeta^{\uparrow}$ for all $t \leq k$ and $|X| \leq t$ in $\sum_{t=0}^{k} O(n\binom{n}{\downarrow t}) = O(kn\binom{n}{\downarrow k})$ by performing the fast up-zeta transform. For the evaluation of formula (9), it takes $O(k)$ time to get the convolution for each X with $|X| \leq k$. Since the number of sets which need to be considered is $O(\binom{n}{\downarrow k})$, the time spent in this process is $O(k\binom{n}{\downarrow k})$. Finally, the time needed in evaluating the formula (10) is $O(n\binom{n}{\downarrow k})$ using the fast Möbius inversion in a layered manner (see the following Lemma 4). Then we can get our result. □

The following special case allows us to present a stronger result for the k-disjoint sum problem.

Lemma 3. *If for each set S in F_1 and F_2, $|S| \leq d < k$, then k-disjoint sum can be computed in $O(dn\binom{n}{\downarrow d})$ time.*

Proof. Note that we only need to consider the sets $X \subseteq U$ with cardinality at most d, since other sets have no supersets in F_1 and F_2. Based on this fact, we redefine the convolution as follows.

$$\alpha\zeta^{\uparrow} \circledast \beta\zeta^{\uparrow}(k, X) = \sum_{k_1=k-d}^{d} \alpha\zeta^{\uparrow}(k_1, X)\beta\zeta^{\uparrow}(k - k_1, X).$$

With this new convolution and the same other operations as in the proof for Lemma 2, we get the result for Lemma 3. □

3.2 Algorithms for Exact Multilinear k-Monomial Counting

By observing the process of the fast subset convolution, we have the following Lemma 4.

Lemma 4. *The subset convolution $f *_R g$ for all subsets $S \subseteq U$ with $|S| \leq k$ can be computed in $O^*(\binom{n}{\downarrow k})$ time, where $\binom{n}{\downarrow k} = \sum_{i=0}^{k} \binom{n}{i}$.*

Proof. Clearly, we only need to consider these sets S with $|S| \leq k$ in F_1 and F_2 under the given condition. First, note that $f\zeta^{\downarrow}(t, S), g\zeta^{\downarrow}(t, S)$ for all $t \leq k$ and $|S| \leq k$ can be obtained in $O^*(\binom{n}{\downarrow k})$ time by Lemma 1. Based on this, for all $t \leq k$ and $|S| \leq k$, $f\zeta^{\downarrow} \circledast g\zeta^{\downarrow}(t, S)$ can be computed in $O(k\binom{n}{\downarrow k})$ time. Then as shown in [3], we can perform the fast subset convolution in a layered manner, i.e., computing $f *_R g(S)$ for all sets S with $|S| = i$ in the ith-layer using the fast Möbius inversion. It is easy to get that the time needed in the ith-layer is $O^*(\binom{n}{i})$. Then we get our result. □

Note that, in the following sections, we call the subset convolution which only computes the values for $S \subseteq U$ with $|S| \leq k$ a k-layer subset convolution.

As an extension, we consider the subset convolution for more than two functions. For each $S \subseteq U$, let $f_1 *_R \cdots *_R f_m(S) = \sum_{\cup_{i=1}^m S_i = S, \sum_{i=1}^m S_i = |S|} f_1(S_1) \cdots f_m(S_m)$, where $\{f_i\}$ are functions defined on 2^U. Clearly, this can be computed by performing $O(\log k)$ fast subset convolutions. We can get a similar result to Lemma 4 for this generalized subset convolution with the same time complexity. In the following, we will not distinguish these two kinds of subset convolutions.

For sets A and B, the set $A[B]$ is the set of all functions $f : B \to A$. Let U be a set of elements. Define U_k as the collection of all subsets of U with cardinality at most k. We denote $\oplus$ as the addition over $R[U_k]$, i.e., for any $f_i \in R[U_k]$, $i = 1, \ldots, m$, $\oplus$ computes the sum $\sum_{i=1}^m f_i$, where R is the ring. In addition, we still denote the operator k-layer subset convolution over R as $*_R$. A circuit C over $(R[U_k], \oplus, *_R)$ means that each operator gate in C is either an addition gate or a k-layer subset convolution gate over $R[U_k]$. For sets A and B such that A has a 0 element, a singleton is an element f of $A[B]$ such that $f(x) = 0$ for each element $x \in B$ and $x \neq b \in B$.

In the following, we will first present a new enumeration based algorithm for the exact multilinear k-monomial counting problem, and then we will give the proof for Theorem 1.

Lemma 5. *Let $P(x_1, \ldots, x_n)$ be a commutative polynomial with coefficients from R, given by an arithmetic circuit C. The sum of coefficients of all multilinear k-terms in P can be computed in time $O^*(\binom{n}{\downarrow k}|C|)$.*

Proof. First we replace each variable x_i with a unique element u_i. Denote the set of elements as $U = \{u_1, u_2, \ldots, u_n\}$. Also denote U_k as the collection of all subsets of U with cardinality at most k. We prove the lemma by constructing a new circuit $C^{'}$ over $(R[U_k], \oplus, *_R)$ based on C. Then we prove that the new circuit $C^{'}$ can correctly output the coefficients of all the multilinear terms of degree at most k in P in $O^*(\binom{n}{\downarrow k}|C|)$ time. The sum can be obtained by adding up the coefficients of all the multilinear k-terms in P.

The new circuit $C^{'}$ is constructed as follows:

(*i*) For each input gate x_i, $i = 1, 2, \ldots, n$, adapt it for inputting a singleton $f_i \in R[U_k]$ whose value is 1 for $\{u_i\}$ and 0 for other elements in U_k; For each input gate outputting a constant number c, adapt it for inputting a singleton $f_i \in R[U_k]$ whose value is c for $\emptyset$ and 0 for other elements in U_k.

(*ii*) For each addition gate, replace it with an addition gate $\oplus$ defined on $R[U_k]$.

(*iii*) For each product gate, replace it with a k-layer subset convolution gate $*_R$.

The output value for each set $S = \{u_{i_1}, \ldots, u_{i_t}\}$ in U_k is the coefficient of its corresponding multilinear monomial $x_{i_1} \cdots x_{i_t}$. We claim that $C^{'}$ will correctly output coefficients of all multilinear monomials with degree at most k. This can be proved by showing that for each gate ϕ in $C^{'}$ and its corresponding gate σ

in C, the value for each set $S = \{u_{i_1}, \ldots, u_{i_t}\} \in U_k$ outputted by ϕ equals the coefficient of the multilinear monomial $x_{i_1} \ldots x_{i_t}$ in the polynomial generated by σ. It is unnecessary here to consider the non-multilinear monomials since the non-multilinear monomials generated by each gate will only lead to non-multilinear monomials in the rest of the computation.

Now, we assign each gate of $C^{'}$ a unique integer from $[1, |C|]$ such that for gates i and j, if $i < j$, there exists no directed edge from j to i, and each input gate will be assigned an integer less than that of any operator gate. Since the circuit is an acyclic graph, such an ordering exists. In the following we will use the assigned number to denote the corresponding gate. We prove our claim by induction according to the ordering. First for input gates, the claim is true according to the definition of the new input gates.

Then for any addition gate λ in $C^{'}$, since the number assigned to each of its input gates is smaller than λ, by assumption, each of its input gates outputs the value for each set $S = \{u_{i_1}, \ldots, u_{i_t}\} \in U_k$ that is equal to the coefficient of the multilinear monomial $x_{i_1} \cdots x_{i_t}$ in the polynomial generated by the corresponding gate in C. Since the coefficient of each multilinear monomial $x_{i_1} \cdots x_{i_t}$ is obtained by summing up the coefficients of the same multilinear monomials in the inputting polynomials, by the definition of our new addition gate, the claim is true for λ.

Finally, for any k-layer subset convolution gate ϑ in $C^{'}$ and its corresponding product gate $\vartheta^{\times}$ in C, assume that the input gates of ϑ are $\vartheta_1, \ldots, \vartheta_m$ and the corresponding input gates of $\vartheta^{\times}$ in C are $\vartheta_1^{\times}, \ldots, \vartheta_m^{\times}$. Let f_i be the function on U_k outputted by ϑ_i and P_i be the polynomial outputted by $\vartheta_i^{\times}$, $1 \leq i \leq m$. Denote the coefficient of the multilinear monomial $Y = x_{j_1} \cdots x_{j_p}$ in the polynomial generated by $\vartheta^{\times}$ as $c(Y)$ and denote the coefficient of Y in P_i as $c_i(Y)$. Surely, Y can be a constant. Then for each multilinear monomial $x_{i_1} \cdots x_{i_t}$, where $t \leq k$, we have

$$c(x_{i_1} \cdots x_{i_t}) = \sum_{\Pi_{i=1}^{m} Y_i = x_{i_1} \cdots x_{i_t}} \Pi_{i=1}^{m} c_i(Y_i). \tag{11}$$

Since the number assigned to each of ϑ's input gates is smaller than ϑ, by assumption, for each multilinear monomial $Y = x_{j_1} \cdots x_{j_p}$, $c_i(Y) = f_i(S_Y)$, where $p \leq k$ and S_Y is the corresponding element set $\{u_{j_1}, \ldots u_{j_p}\}$ of Y. Then

$$\begin{aligned} c(x_{i_1} \cdots x_{i_t}) &= \sum_{\cup_{i=1}^{m} S_i = \{u_{i_1}, \ldots, u_{i_t}\}, \sum_{i=1}^{m} |S_i| = |S|} \Pi_{i=1}^{m} f_i(S_i) \\ &= f_1 *_R \cdots *_R f_m(\{u_{i_1}, \ldots u_{i_t}\}). \end{aligned} \tag{12}$$

Thus the subset convolution value for each set $S = \{u_{i_1} \ldots u_{i_t}\}$ computed by the k-layer subset convolution gate ϑ is equal to the coefficient of the multilinear monomial $x_{i_1} \ldots x_{i_t}$ computed by $\vartheta^{\times}$. The claim is also true for the k-layer subset convolution gates. Hence, $C^{'}$ will correctly output the coefficients of all multilinear monomials with degree at most k.

As for the time complexity, first, the operations in each addition gate of $C^{'}$ can be done in time $O(\binom{n}{\downarrow k})$; second, by Lemma 4, each k-layer subset convolution

gate can finish the computation in $O^*(\binom{n}{\downarrow k})$. Thus, the total time needed is $O^*(\binom{n}{\downarrow k}|C'|) = O^*(\binom{n}{\downarrow k}|C|)$. □

Proof. (The Proof for our main result-Theorem 1) From the proof of Lemma 5, for both P and Q, we can construct circuits C_P and C_Q over $(R[U_d], \oplus, *_R)$ which output values for each set $S = \{u_{i_1}, \ldots, u_{i_t}\}$ equaling to the coefficients of the corresponding multilinear monomial $x_{i_1} \cdots x_{i_t}$ in P and Q, respectively. By Lemma 5, this can be done in $O^*(\binom{n}{\downarrow d})$ time. Let $f_P(S)$ and $f_Q(S)$ be the values for S outputted by P and Q, respectively.

For each multilinear monomial $Y = x_{j_1} \cdots x_{j_p}$, we denote the coefficient of Y in $P \cdot Q$ as $c(Y)$, and denote the coefficients of Y in P and Q as $c_P(Y)$ and $c_Q(Y)$, respectively. Then for the multilinear monomial $x_{i_1} \cdots x_{i_k}$,

$$c(x_{i_1} \cdots x_{i_k}) = \sum_{Y_1 \cdot Y_2 = x_{i_1} \cdots x_{i_k}} c_P(Y_1) \cdot c_Q(Y_2). \tag{13}$$

We know that $c_P(Y_1) = f_P(S_{Y_1})$ and $c_Q(Y_2) = f_Q(S_{Y_2})$, where S_{Y_i} is the corresponding element set of Y_i, $i = 1, 2$. Then

$$c(x_{i_1} \cdots x_{i_k}) = \sum_{S_1 \cup S_2 = \{u_{i_1}, \ldots u_{i_k}\}, S_1 \cap S_2 = \emptyset} f_P(S_1) \cdot f_Q(S_2). \tag{14}$$

Thus the sum c of coefficients of all multilinear k-monomials in $P \cdot Q$ is

$$c = \sum c(x_{i_1} \cdots x_{i_k}) = \sum_{|S_1|+|S_2|=k, S_1 \cap S_2 = \emptyset} f_P(S_1) \cdot f_Q(S_2) = f_P \boxtimes f_Q. \tag{15}$$

Then by Lemma 3 and making use of the algorithm for computing the k-disjoint sum, the sum of coefficients of all multilinear k-terms in $P \cdot Q$ can be obtained in $O^*(\binom{n}{\downarrow d})$ time. Thus we obtain Theorem 1. □

4 A Polynomial Space Algorithm for Exact Multilinear k-Monomial Counting

In this section, we consider a polynomial space algorithm for exact multilinear k-monomial counting based on the circuit constructed in the proof of Lemma 5. The following Lemma 6 is proved in [14].

Lemma 6. *(Theorem 5.1 in [14]) Let V be a set, and let C be a circuit over $(R[2^V], \oplus, *_R)$. Suppose C outputs f, all its inputs are singletons, and m is an integer such that $f(V) \leq m$. Then, given C and m, $f(V)$ can be computed using $O^*(2^{|V|})$ time and $O(|V||C| \log m)$ space.*

Theorem 2. *Let $P(x_1, \ldots, x_n)$ be a commutative polynomial with coefficients from R, given by an arithmetic circuit C. The sum of coefficients of all multilinear k-terms in P can be computed in $O^*(2^k\binom{n}{k})$ time and polynomial space.*

Proof. As was done in Lemma 5, we associate each variable x_i with a unique element u_i and denote the set of elements as U. Then it is easy to find the sub-circuit $C^{'}$ for computing the polynomial $P^{'} = P(x_1, \ldots, x_k, 0, \ldots, 0)$. Let $U^{'}$ be the set $\{u_1, u_2, \ldots, u_k\}$. Then by Lemma 5, we can construct a new circuit $C^{''}$ based on $C^{'}$ over $(R[2^{U^{'}}], \oplus, *_R)$ with inputs being singletons, and the output is a function on $2^{U^{'}}$ whose values correspond to the coefficients of all multilinear monomial in $P^{'}$. By Lemma 6, the coefficient of $x_1 \cdots x_k$ can be computed in $O^*(2^k)$ time and polynomial space. Similarly, we can compute the coefficients of other multilinear k-terms in P. Thus we get the result. □

5 Applications

In this section, we show how the multilinear monomial counting technique presented in Theorem 1 can be applied to solving parameter counting problems. In particular, in Sections 5.1 and 5.2, we show that the multilinear monomial counting technique can be used to solve the #k-path and the #m-set k-packing problems with time complexities that match the state-of-the-art known results.

5.1 #k-Path

Definition 4. *(#k-path) Given an n-vertices undirected graph G, the #k-path problem is to count the number of simple paths of length k in G.*

Theorem 3. *The number of k-paths in an n-vertices graph G can be obtained in $O^*(\binom{n}{\lceil k/2 \rceil})$ time.*

Proof. Let A be the adjacency matrix of G, and let $x_1, \ldots, x_n$ be variables. Define a matrix $B[i,j] = A[i,j]x_i$. Let $\overrightarrow{\mathbf{1}}$ be the row n-vector of all 1s, and $\overrightarrow{\mathbf{x}}$ be the column vector defined by $\overrightarrow{\mathbf{x}}[i] = x_i$. Define the k-walk polynomial to be $P_k(x_1, \ldots, x_n) = \overrightarrow{\mathbf{1}} \cdot B^{k-1} \cdot \overrightarrow{\mathbf{x}}$. This polynomial is given in [16] for solving the decision version of the k-path problem. Clearly, P_k is the sum of k-monomials representing the k-walks in G and only simple k-paths in G correspond to multilinear k-monomials. Note that each path corresponds to two multilinear k-monomials in P_k that are the same. Then the number of k-paths in G can be obtained by dividing the sum of coefficients of multilinear k-terms in $P_k(x_1, \ldots, x_n)$ by 2.

In the following, we assume that k is even in order to make the analysis simpler. Let $\overrightarrow{P} = \overrightarrow{\mathbf{1}} \cdot B^{k/2}$ and $\overrightarrow{Q} = B^{k-k/2-1} \cdot \overrightarrow{\mathbf{x}}$. It is easy to know that P is a row $n-$vector and Q is a column n-vector. Clearly, $\overrightarrow{P} \cdot \overrightarrow{Q} = P_k(x_1, \ldots, x_n)$. Assume that $\overrightarrow{P} = (P_1 \ \ldots \ P_k)$ and $\overrightarrow{Q} = (Q_1 \ \ldots \ Q_k)^T$, where P_i and Q_i are n-variate polynomials with degree at most $\frac{k}{2}$. Then $P_k = P_1 \cdot Q_1 + \ldots + P_n Q_n$.

Observe that both $\overrightarrow{P}$ and $\overrightarrow{Q}$ can be implemented by circuits of size $O(k(m+n))$. Assume that C_P and C_Q are the circuits for $\overrightarrow{P}$ and $\overrightarrow{Q}$. It is quite easy to find the sub-circuits $C_P^{'}$ and $C_Q^{'}$ for computing P_1 and Q_1. Then by Theorem 1, the sum of coefficients of all multilinear k-terms in $P_i Q_i$ can be obtained in $O^*(\binom{n}{k/2})$ time. Similarly, we can get the sum of coefficients of multilinear k-terms in $P_i Q_i$, $1 \leq i \leq n$. Finally, by summing up the values in the above

step for all P_iQ_i, we can get the sum of coefficients of multilinear k-terms in P_k which can be used to get the number of k-paths in G, and the total time needed is $O^*(\binom{n}{k/2})$. □

5.2 #m-Set k-Packing

Definition 5. *(#m-set k-packing) Given a collection F of N sets, each containing m elements from a universe U of n elements, the #m-set k-packing is to count the number of collections $F' \subseteq F$ such that F' is composed by k mutually disjoint sets.*

Theorem 4. *The number of m-set k-packings can be counted in $O^*(N\binom{n}{\lceil mk/2 \rceil})$ time.*

Proof. We will apply the polynomial presented in [12] for solving the decision version of the m-set k-packing problem which is constructed as follows: (i) assign variables $X = \{x_i\}, i = 1, 2, \ldots, n$ to elements of U; (ii) assign to the set $S_i \in F$ a degree m set-monomial Y_i, defined as the product of the variables corresponding to the elements of S_i. Then define $P_k = (\sum_{i=1}^{N} Y_i)^k$. Clearly, in P_k, each monomial $Y_{i_1}Y_{i_2}\cdots Y_{i_k}$ represents a $k-$tuple of sets $< S_{i_1}, S_{i_2}, \cdots, S_{i_k} >$ and only those $k-$tuples whose sets compose a $m-$set $k-$packing are represented by multilinear $mk-$monomials. Note that each m-set k-packing corresponds to $k!$ multilinear mk-monomials in P_k that are the same. Then the number of m-set k-packings can be obtained by dividing the sum of coefficients of all multilinear mk-terms in P_k by $k!$. In the following, for convenience, we assume that mk is even. The analysis for odd numbers is similar.

Let $P = (\sum_{i=1}^{N} Y_i)^{k/2}$ and $Q = (\sum_{i=1}^{N} Y_i)^{k/2}$. Clearly, the degrees of P, Q are at most $\frac{mk}{2}$ and $P_k = P \cdot Q$. Then P and Q can be implemented by circuits of size $O(N)$. By Theorem 1, the sum of multilinear mk-terms in P_k can be computed in $O^*(N\binom{n}{\lceil mk/2 \rceil})$ time. Then the result is proved. □

6 Conclusions

In this paper, by using a novel circuit design that utilizes k-layer subset convolution and by using the presented fast algorithm for k-disjoint sum, we have proposed new algorithms for the exact multiliner k-monomial counting problem, which can also be used to solve a variety of parameter counting problems. For this basic parameterized monomial counting problem, our results are superior to previous results given in [13]. Also, by reducing the #k-path problem and the #m-set k-packing problem to the exact multilinear k-monomial counting problem, we give algorithms that match the fastest known results presented in [2]. More importantly, as long as we can design an appropriate polynomial, our technique can be used to tackle a large collection of other parameter counting problems, such as the #k-tree problem and its directed counterpart the #k-vertex out-tree problem.

For future work, one natural idea is to extend Theorem 1 to the case where the polynomial is the product of more than two lower degree polynomials. In

addition, the following question is still open: Is there an algorithm for multilinear k-monomial counting that can break the $\binom{n}{k}$ barrier for general circuits? A number of counting problems will benefit from such an algorithm.

Acknowledgements. This work was supported in part by Hong Kong RGC-GRF grant 714009E, HKU Small Project Funding 21476015.47932.14200.420.01, the National Basic Research Program of China Grant 2007CB807900, 2007CB807901, the National Natural Science Foundation of China Grant 61073174, 61033001, 61061130540, and the Hi-Tech research and Development Program of China Grant 2006AA10Z216.

References

1. Alon, N., Yuster, R., Zwick, U.: Color coding. Journal of the ACM 42(4), 844–856 (1995)
2. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Counting paths and packings in halves. In: Fiat, A., Sanders, P. (eds.) ESA 2009. LNCS, vol. 5757, pp. 578–586. Springer, Heidelberg (2009)
3. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Fourier meets möbius: fast subset convolution. In: STOC, pp. 67–74 (2007)
4. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Trimmed moebius inversion and graphs of bounded degree. Theory Comput. Syst. 47(3), 637–654 (2010)
5. Chen, J., Lu, S., Sze, S.-H., Zhang, F.: Improved algorithms for path, matching, and packing problems. In: SODA, pp. 298–307 (2007)
6. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. J. Symbolic Computation 9(3), 251–280 (1990)
7. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Berlin (1999)
8. Flum, J., Grohe, M.: The parameterized complexity of counting problems. SIAM J. Comput. 33, 892–922 (2004)
9. Jia, W., Zhang, C., Chen, J.: An efficient parameterized algorithm for m-set packing. J. Algorithms 50, 106–117 (2004)
10. Kennes, R.: Computational aspects of the Moebius transform of a graph. IEEE Transactions on Systems, Man, and Cybernetics 22, 201–223 (1991)
11. Kneis, J., Mölle, D., Richter, S., Rossmanith, P.: Divide-and-color. In: Fomin, F.V. (ed.) WG 2006. LNCS, vol. 4271, pp. 58–67. Springer, Heidelberg (2006)
12. Koutis, I.: Faster algebraic algorithms for path and packing problems. In: ICALP, pp. 575–586 (2009)
13. Koutis, I., Williams, R.: Limits and applications of group algebras for parameterized problems. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 653–664. Springer, Heidelberg (2009)
14. Lokshtanov, D., Nederlof, J.: Saving space by algebraization. In: STOC, pp. 321–330 (2010)
15. Vassilevska, V., Williams, R.: Finding, minimizing, and counting weighted subgraphs. In: STOC, pp. 455–464 (2009)
16. Williams, R.: Finding paths of length k in $O^*(2^k)$ time. Inf. Process. Lett. 109(6), 315–318 (2009)
17. Yates, F.: The design and analysis of factorial experiments, Technical Communication No. 35, Commonwealth Bureau of Soil Science, Harpenden, UK (1937)