Patching Colors with Tensors

Cornelius Brand
Saarland University (MMCI), Saarland Informatics Campus, Germany
cbrand@mmci.uni-saarland.de

Abstract

We describe a generic way of exponentially speeding up algorithms which rely on Color-Coding by using the recently introduced technique of Extensor-Coding (Brand, Dell and Husfeldt, STOC 2018). To demonstrate the usefulness of this "patching" of Color-Coding algorithms, we apply it *ad hoc* to the exponential-space algorithms given in Gutin et al. (Journal Comp. Sys. Sci. 2018) and obtain the fastest known deterministic algorithms for, among others, the k-internal out-branching and k-internal spanning tree problems. To realize these technical advances, we make qualitative progress in a special case of the detection of multilinear monomials in multivariate polynomials: We give the first deterministic fixed-parameter tractable algorithm for the k-multilinear detection problem on a class of arithmetic circuits that may involve cancellations, as long as the computed polynomial is promised to satisfy a certain natural condition.

Furthermore, we explore the limitations of using this very approach to speed up algorithms by determining exactly the dimension of a crucial subalgebra of extensors that arises naturally in the instantiation of the technique: It is equal to F_{2k+1}, the kth odd term in the Fibonacci sequence. To determine this dimension, we use tools from the theory of Gröbner bases, and the studied algebraic object may be of independent interest.

We note that the asymptotic bound of $F_{2k+1} \approx \phi^{2k} = O(2.619^k)$ curiously coincides with the running time bound on one of the fastest algorithms for the k-path problem based on representative sets due to Fomin et al. (JACM 2016). Here, ϕ is the golden ratio.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Graph algorithms analysis; Computing methodologies → Algebraic algorithms

Keywords and phrases Color-Coding, Extensor-Coding, internal out-branching, colorful problems, algebraic algorithms, multilinear detection, deterministic algorithms, exterior algebra

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.25

Related Version Omitted proofs and details will appear in the author's thesis [8].

Acknowledgements I want to thank Sarah Berger, Markus Bläser, Radu Curticapean, Holger Dell, Thore Husfeldt, Christian Ikenmeyer, Fahad Panolan, Petteri Kaski, Mikko Koivisto and Meirav Zehavi for helpful discussions and valuable suggestions. I also thank the anonymous referees for their proposed improvements of the manuscript.

1 Introduction

The research in parameterized algorithms has brought forth a vast toolbox of techniques to tackle an ever-growing list of hard computational problems, and provided solutions for those problems in a multitude of shapes and flavors. Recently, Brand, Dell and Husfeldt [9] added another item to the box: A technique called *Extensor-Coding*, based on the properties of the so-called *exterior algebra*, a fundamental object in multilinear algebra.

Extensor-Coding allows us to understand a variety of key approaches in the design of parameterized algorithms in a common language, including the celebrated and by now classic *Color-Coding*-approach of Alon, Yuster and Zwick [1], the *algebraic fingerprints* due to Koutis [24] and Koutis and Williams [25], the representative families from Fomin et al. [19, 18], and the labeled walks of Björklund et al. [5].

While this yields unification on a conceptual level, the main technical advance made in [9] was with respect to the approximate counting of solutions in the longest path (or k-path) problem. Despite the fact that the technique suggests a straightforward algebraic and *deterministic* algorithm for the k-path problem, its running time was only very slightly (infinitesimally, in fact) below the older bound achieved by Chen et al. [10], and thus at significant distance from Zehavi's current record bound [36]. In this paper, we will identify several problems where Extensor-Coding *does* improve over the state-of-the-art. To this end, we start from a novel and natural variant of the ubiquitous multilinear detection problem on polynomials computed by arithmetic circuits that is more general than the very well-studied case of cancellation-free circuits. This leads us to a rather generic approach of patching, if you will, algorithms involving Color-Coding, by replacing the Colors used in the algorithm by Extensors, i.e., elements of the exterior algebra.[1] This then yields an exponential speed-up. While this improvement is certainly incremental in nature, the novelty of the approach laid out in this work is of a more conceptual nature. This is discussed further below.

On the other hand, we demonstrate an algebraic barrier, limiting the potential progress that can be made by applying Extensor-Coding in a more or less straightforward manner – just as we do it in this paper for the considered algorithmic problems. This barrier is based on the dimension of a certain subalgebra of the exterior algebra, which presents a lower bound for the cost of computation in the algebra and hence for the entire algorithm. It is noteworthy that the resulting bound, which is exactly the $(2k + 1)$th Fibonacci number on an underlying vector space of dimension k, asymptotically coincides with the square of the golden ratio, which also appears in the approach of Fomin et al. to the k-path problem, which is based on the very different, purely combinatorial method of representative families [19]. One might argue that this connection is not entirely surprising, since the approaches can be used to solve the same problem and the representative families in fact have their origin in exterior algebra. This, however, is an argument that only bears on a superficial level: It is by no means clear how the very algebraic object studied in this article relates in any way to the use of the exterior algebra in the representative families-approach, and indeed, one can consider uncovering such a precise connection as an intriguing avenue for further research. Therefore, one may regard studying this algebraic object as a first step to tighten the ties between combinatorics and algebra in the study of rather general parameterized algorithms, and not just for the k-path problem.

1.1 Contributions

1.1.1 New and Improved Algorithms

The algorithmic problems we study are the following: The *k-internal out-branching (k-IOB)* problem asks for the existence of an out-branching (also called a directed spanning tree or arborescence, among others) with at least k internal, i.e., non-leaf, nodes. The *k-internal spanning (k-IST)* tree problem is formulated analogously for undirected graphs. A subset of edges in an edge-colored graph is *k-colorful* if it contains edges of at least k distinct colors, and the definitions of the problems *k-colorful perfect matching* and *k-colorful out-branching* are self-evident.

Our algorithmic advances in these problems are borne by progress in another domain: The *k-multilinear detection (k-MLD)* problem has as an input a multivariate polynomial represented through an arithmetic circuit, and asks whether the polynomial contains a

[1] Extensors are often also referred to as antisymmetric tensors, hence the title.

monomial of degree k in which no variable appears with degree more than one. We focus our attention on the restriction of the problem to those instances where a multilinear monomial may only appear with positive coefficient. Since there is no known way to efficiently test this property, this turns the problem into a promise problem. The promise, however, is typically satisfied in combinatorial applications, where the studied polynomials are usually *multivariate generating functions* of the sought combinatorial objects. The decisive subtlety here is that, while the *polynomial* that is computed by the input circuit may not have negative coefficients in its multilinear part, the circuit *itself* may very well contain negative constants and make use of cancellations, and, in particular, it need *not* be monotone. We will make heavy use of this property in the above application problems, and the above point is decisive here for the following reason: These problems can be expressed using determinantal generating functions, and by a theorem of Jerrum and Snir, determinants do *not* have monotone circuits of subexponential size, such that cancellations are actually crucial for their efficient computation. To the best of our knowledge, this is the first fixed-parameter tractable algorithm for the problem.

We prove the following deterministic, exponential-space[2] record time bounds, and defer the reader to Sect. 4 for a formal statement of the theorem.

▶ **Theorem 1** (Informal). *There are deterministic algorithms to solve*

1. *the k-internal out-branching problem and the k-internal spanning tree problem in time $3.21^k \cdot \mathrm{poly}(n)$, and*
2. *the k-colorful perfect matching problem on planar graphs and the k-colorful out-branching problem in time $4^k \cdot \mathrm{poly}(n)$.*

Furthermore, there is a deterministic algorithm that solves the restriction of the multilinear detection problem to circuits computing polynomials with positive coefficients in their multilinear part (as laid out above) in time $4^k \cdot \mathrm{poly}(n)$ on skew arithmetic circuits, and in time $2^{\omega k} \cdot \mathrm{poly}(n) < 5.19^k \cdot \mathrm{poly}(n)$ on general circuits, where ω is the exponent of matrix multiplication.

▶ Remark 2. The bound of 5.19 is not competitive; indeed it is easy to prove that an exponential basis of 4.312 can be obtained using a derandomization of Color-Coding, and Pratt [28] gives a randomized algorithm achieving 4.075. Note, however, that our bound depends on ω, and one can make the point (albeit moot in the foreseeable future) of this dependency making our bound potentially competitive.

▶ Remark 3. Skewness, i.e., the syntactic restriction of each multiplication gate having an input as an operand seems rather strong at a first glance. At a second glance, this impression does not hold up: Without concerning ourselves with the technicalities of algebraic complexity theory, suffice it to say that the polynomials that can be computed by efficient skew circuits are precisely those that are efficient projections of determinants, and determinantal generating functions are known for a variety of combinatorial objects. Equivalently, they are those polynomials that are computed by efficient algebraic branching programs (which is a widely studied and very natural computational model).

The algorithms for the problems of detecting a k-internal as well as k-colorful out-branching (and spanning tree) are established, as demonstrated by Björklund et al. , via the Directed Matrix-Tree Theorem [7]. We reuse the meticulous and very careful analysis of the

[2] We are aware of the doubtful practical usefulness of exponential-space algorithms in some settings and ask the skeptical reader to think of our results as being motivated theoretically.

parallel monomial sieving technique by Gutin et al. [22]. We can relatively easily replace the employed pseudorandom objects by extensors. This suggests a rather generic way in which to speed up algorithms based on Color-Coding by instead using Extensor-Coding.

As far as the k-colorful planar matching problem is concerned, as in [22], we use a Pfaffian computation. However, we cannot perform the square root extraction that is necessary to make use of the determinantal identities for the Pfaffian, but rely on a result of Flarup et al. [15] for a direct computation of Pfaffians with skew arithmetic circuits.

Let us conclude the discussion of our algorithmic contributions with a comment on their relevance. As far as combinatorial problems are concerned, the obtained technical improvements are incremental. However, focusing on these individual results is, in a way, a red herring: The important point is that we can obtain speed-ups in sophisticated algorithms for well-studied problems in a simple, *ad hoc* manner by replacing colors with extensors. The only requirement here is the, as argued, rather mild one of the combinatorial objects in question being described through a determinantal formula. This is not primarily a technical, but a *conceptual* insight that is meant to guide the design of deterministic exponential-space algorithms in the future, and to encourage revisiting existing bounds using extensors. To demonstrate how to go about this, we do so for k-IOB, a very well-studied problem where the current state-of-the-art was obtained through a sequence of incremental improvements, each more involved than the previous one.

A similar case can be made for the monomial detection algorithms, which are obtained in an admittedly straightforward manner by an application of the Extensor-Coding method, which was certainly available at the time of writing of [9]. The takeaway here, however, is that polynomials with positive coefficients computed by efficient skew circuits are not some arbitrary, but instead very natural class of polynomials, namely those that arise as determinantal generating functions for combinatorial objects, which are of central interest in algebraic combinatorics.

1.1.2 Algebraic Limitations

Broadly speaking, when aiming towards deterministic decision algorithms, Extensor-Coding essentially works by evaluating a multivariate polynomial associated with the input over an exterior algebra. This algebra is of dimension 4^k, where k will typically be the parameter of the input instance (while it formally is half the dimension of the underlying vector space).

In this way, it is similar to a method introduced by Koutis [24]. It differs, however, in the points at which the polynomial associated with the input instance is evaluated. While Koutis, and later, Williams [33], rely on random evaluation points, in [9], certain carefully constructed vectors are used. One can readily observe that "evaluating a polynomial" involves, on an arithmetic level, nothing but multiplications and additions. In particular, if one evaluates a polynomial over any algebraic structure that is closed under multiplication and addition, one will always obtain again an element of this algebraic structure after evaluation. Turning this around, one can always restrict one's attention to the closure of (i.e., the substructure generated by) the set of evaluation points. In particular, it might be far easier to actually implement the arithmetic operations only over this substructure than over the entire structure.

In this paper, we will examine the subalgebra generated by the aforementioned special evaluation points (i.e., the smallest set closed under addition, scaling with a field constant, and multiplication containing all the evaluation points). Note now that the dimension of an algebra provides a trivial lower bound on the cost of general computation in it, simply because one has to write down its elements' coordinates at some point. Conversely, the dimension itself can be used to derive a trivial upper bound for the cost of computation as well, just by a look-up of the structural constants of the algebra, but this bound is generally far from optimal. For example, naive multiplication of degree-d polynomials (which can be modeled as objects of a $O(d)$-dimensional algebra over some field) takes around $O(d^2)$ field operations, while (over compatible fields) the Fast Fourier Transform method yields the classic bound of $O(d \log d)$. A running time of the latter form, i.e., quasilinear in the dimension of the algebra, is of course the best one can hope for. Note however that, in general, it is a highly nontrivial, and sometimes impossible, task to come up with such fast multiplication algorithms in any kind of algebraic structure.

We prove that, surprisingly (for reasons that will be expanded on later), the dimension of the subalgebra generated by the evaluation points used in Extensor-Coding is of dimension *exponentially smaller* than 4^k, i.e., the dimension of the full algebra. At first, this seems to open up a tantalizing new point of attack on one of the most prominent open problems in the area of parameterized algorithms: To exhibit a *deterministic* algorithm for the k-path problem that matches the running time of $2^k \cdot \text{poly}(n)$ for the best *randomized* algorithms [33, 24]. Now, *a priori*, we could hope for the studied subalgebra to be of dimension 2^k (but not much smaller, by a result of Koutis-Williams [26]). Then, a quasilinear multiplication algorithm would give a bound of $2^k \cdot \text{poly}(k)$ field operations, and (assuming all coefficients stay of moderate size) hence produce a deterministic algorithm solving the problem in time $2^k \cdot \text{poly}(n)$.

Unfortunately, we share the fate of Tantalos:

▶ **Theorem 4** (Informal). *The dimension of the subalgebra of the exterior algebra over the complex vector space of dimension $2k$ generated by the evaluation points used in Extensor-Coding is of dimension exactly F_{2k+1}, which is the $(2k+1)$th Fibonacci number, and this is asymptotically bounded as $2.618^k \leq F_{2k+1} \leq 2.619^k$.*

Again, it will be stated formally in Sect. 5. Somewhat unluckily, this does also not suffice to improve the state-of-the-art for k-path, even if we had a sufficiently fast multiplication algorithm, since Zehavi's [36] intricate algorithm has an exponential basis of 2.597 in its running time.

This result can be interpreted in two ways: On the one hand, as laid out above, it can be seen as a negative result, ruling out one approach for solving the k-path problem deterministically and fast. On the other hand, while a quasilinear multiplication algorithm for the subalgebra might be a lot to hope for, even an algorithm using, say, $2.619^{\omega k/2} < 3.136^k$ field operations would mean significant improvements for all the problems studied in this paper, including in particular deterministic multilinear detection for a subclass of circuits with cancellations. It is not at all clear how to implement multiplication over this algebra faster than trivially, and we leave this as a challenging open problem.

Let us comment on the bound of $2.619^{k\omega/2}$, which might at first seem arbitrary. This is simply the bound one would obtain from a matrix representation of dimension at most d, i.e., having dimension $\sqrt{2.619^{k/2}} \times \sqrt{2.619^{k/2}}$. For this reason, it appears in the recent breakthrough work of Umans [32] and Ching-Yun Hsu and Umans [12] on generalized Fast Fourier Transforms on arbitrary finite groups. Therefore, our result nurtures the hope for even further, significant improvement for a variety of parameterized problems, conditional on the development of fast multiplication algorithms over this algebraic structure.

Furthermore, we exhibit a whole family of subalgebras that enjoy all the desired properties of the subalgebra that is used in [9], and that obey the same upper bound on their dimension, while the lower bound argument does not easily carry over, and it is an exciting possibility that, somewhere among these subalgebras, there is one that has even lower dimension.

As for our algorithmic contributions, let us comment on the relevance of the algebraic barrier, and in particular, why this highly special algebraic object is of wider interest. Given the apparent connection to the work of Fomin et al. as well as the recent insights of Pratt [28], the study of algebraic approaches may have much closer ties to the classic combinatorial techniques than is currently visible. Establishing the dimension bounds as done here is clearly necessary and important in order to uncover these ties; ties which, eventually, might make the study of parameterized algorithms and their limitations appealing to a more diverse audience even outside of parameterized algorithms and complexity, in a way similar (even if not as ambitious by far) to how algebraic geometers found interest in complexity theory through its geometric reformulation. Furthermore, as stated, exhibiting fast multiplication algorithms for this problem would directly imply massive speed-ups for problems that are amenable to be "patched" using extensors.

1.2 Related work

The k-internal out-branching and spanning tree problems have attracted a significant amount of attention over the last years [17, 22, 6, 7, 27, 38, 37, 16, 13, 11, 21, 14, 29]. The current deterministic record bounds for all the aforementioned graph problems were recently given in Gutin et al. [22], using monomial sieving in combination with Color-Coding and suitable pseudorandom objects. The bounds they obtain (in the exponential-space setting) for the k-internal out-branching and spanning tree problems are $3.41^k \cdot \mathrm{poly}(n)$, and $4.32^k \cdot \mathrm{poly}(n)$ for the k-colorful perfect matching and out-branching problems.

The detection of k-multilinear terms in polynomials computed by arithmetic circuits lies at the heart of the design of the fastest randomized algorithms for a host of parameterized problems, such as the longest path problem on directed graphs, the k-tree problem, the t-dominating set problem and the m-dimensional k-matching problem, with a record bound of $2^k \cdot \mathrm{poly}(n)$ for randomized k-multilinear detection [26, 33]. The crux is that, for the arithmetic circuits in these algorithms, it is required that they be *monotone*, i.e., do not involve any cancellations of terms. On this class of monotone arithmetic circuits, the k-multilinear detection problem can be solved deterministically in time $3.85^k \cdot \mathrm{poly}(n)$, using the combinatorial notion of representative sets [18].

Recently, the first fixed-parameter tractable *randomized* algorithms were developed for the problem on general arithmetic circuits [9], which has sparked further work in the area, announcing a polynomial-space version and $4.08^k \cdot \mathrm{poly}(n)$ as a new record bound on general circuits [3, 2, 28].

Due to the particularity of the algebraic object studied for the dimension barrier, it is highly doubtful whether any work directly related to it has been done – at the least, we are not aware of such work. However, the idea of proving dimension lower bounds in order to limit the use of certain parameterized algorithmic techniques was already present in work by Koutis and Williams [26]. Their work, however, gives a more general lower bound and relies on entirely different technical ideas, making it akin to our work only in a broader conceptual sense.

Omitted proofs and details will appear in [8], which we call the full version.

2 Preliminaries

In what follows, $2 \leq \omega < 2.374$ [30, 34] is the exponent of matrix multiplication.

An *arithmetic circuit* is a directed acyclic graph with a single vertex of out-degree 0, called the *output*, a set of vertices with in-degree 0 which are labeled with variables or complex numbers, called the *inputs* and labels $+$ and $\times$ on all vertices that are not inputs.

A vertex labeled with $+$ or $\times$ is, respectively, called a $+$-gate or $\times$-gate. All $\times$-gates are required to have in-degree at most two. We call an arithmetic circuit *skew* if, for all $\times$-gates, at least one of the arcs ending at the gate comes from an input. The *polynomial computed by an arithmetic circuit* is defined in the obvious inductive way: It is equal to the label of the inputs, and for a gate that is not an input, it is defined as the result of the operation indicated by the label of the gate, applied to its two inputs.

Given a directed graph $D = (V, A)$, we a call a subgraph B of D *out-branching* if B is an oriented tree with exactly one vertex r of in-degree zero, the *root*, and B is spanning. Vertices of out-degree zero are called *leaves*, and vertices that are not leaves are *internal*. For an integer k, an out-branching is *k-internal* if it contains at least k internal vertices.

A set of arcs is called a *matching* if no vertex of V is contained in two arcs, and a matching is *perfect* if every vertex is contained in some arc of the matching.

For an edge-colored graph G, directed or undirected, we call a subgraph S of G *k-colorful*, if the edge set of S contains at least k differently colored edges.

2.1 A Review of Extensor-Coding

Although not a strict prerequisite, some knowledge of the technique and its uses for the longest path problem make the following certainly more digestible. An elementary exposition of the employed algebraic objects can be found in the original work [9]. However, we will demonstrate the technique in the appropriate brevity. This should, in principle, suffice to understand everything that happens here. For a more mathematically concise and thorough treatment of the algebraic background material, we encourage the interested reader to consult any given textbook on algebra, for example the one by Birkhoff and Mac Lane [4].

Let us first review some basic algebraic notions. A complex unital associative *algebra* is a ring A that is simultaneously a complex vector space, such that scalar multiplication is compatible with the multiplication in the ring. That is, for any $\lambda \in \mathbb{C}$ and $a, b \in A$, it holds that $\lambda(ab) = (\lambda a)b = a(\lambda b)$ and there is an element $1 \in A$ with $1a = a = a1$ for all $a \in A$. Since complex unital associative algebras are the only types of algebras we will encounter henceforth, we understand all these properties whenever we speak merely of an algebra. The dimension of an algebra A is the dimension of A as a complex vector space.

2.1.1 The Exterior Algebra

Let V be the complex vector space $\mathbb{C}^k$, endowed with its canonical basis $\{e_1, \ldots, e_k\}$. We can now consider the set of formal, *non-commutative* polynomials $\mathbb{C}\langle e_1, \ldots, e_k\rangle$ that can be formed in the “indeterminates” (or *generators*) $e_1, \ldots, e_k$.

Let us define a kind of multiplication on these generators that is denoted as $\wedge$, called the *wedge product*. The wedge product operation on the generators is defined to satisfy the anti-commutativity relation: $e_i \wedge e_j = -e_j \wedge e_i$ for all $1 \leq i, j \leq k$. For $i = j$, this means that $e_i \wedge e_i = -e_i \wedge e_i$, which over $\mathbb{C}$ implies $e_i \wedge e_i = 0$ for all i.

By repeatedly applying this rule, we can understand the multiplication of more than two generators as follows. Let $1 \leq i_1, \ldots, i_t \leq k$ be natural numbers, and consider the wedge product $e_{i_1} \wedge \cdots \wedge e_{i_t}$. Exhaustive application of the rules now implies that this product becomes zero if $i_j = i_{j'}$ for any distinct j, j'. Otherwise, it is equal to $\operatorname{sgn}(\sigma) e_{i_{\sigma(1)}} \wedge \cdots \wedge e_{i_{\sigma(t)}}$, where σ is the permutation that brings the sequence $i_1, \ldots, i_t$ in ascending order.

This multiplication is extended to linear combinations of wedge products of the generators by distributivity and bilinearity. The vector space generated by all such wedge products of generators is called the *exterior algebra* over V and is denoted by $\Lambda(V)$. The vector space $\Lambda(V)$ is turned into an algebra by equipping it with the wedge product as multiplication. We have seen that any reordering of the factors either leaves the original wedge product intact or leads to a sign change. Hence, the set of ordered wedge products $\{e_{i_1} \wedge \cdots \wedge e_{i_t} \mid 0 \leq t \leq k, i_1 < \cdots < i_t\}$ form a linear basis of $\Lambda(V)$. This shows that $\Lambda(V)$ is of dimension 2^k. For any set $S \subseteq \{1, \ldots, k\}$, we write e_S to denote $e_{i_1} \wedge \cdots \wedge e_{i_{|S|}}$, where the elements of S are assumed to be in ascending order, i.e., $i_1 < \cdots < i_{|S|}$. We remark that we can regard V as a linear subspace of $\Lambda(V)$ via a vector's basis representation, i.e., $(x_1, \ldots, x_k)^T \in \mathbb{C}^k$ corresponds uniquely to $x_1 e_{\{1\}} + \cdots + x_j e_{\{k\}}$.

Let us quickly comment on the cost of computation in $\Lambda(V)$. It is clear that addition of two elements of $\Lambda(V)$ can be performed using 2^k arithmetic operations, namely by pointwise addition, over $\mathbb{C}$. Clearly, this can be performed in time polynomial in the length of the basis coefficients of the two summands.

For a restricted but crucial variant of multiplication in $\Lambda(V)$, called *skew multiplication*, we recall the following observation:

▶ **Proposition 5** ([9], Sect. 2.3)**.** *Given an element $x \in \Lambda(\mathbb{C}^k)$ and $y \in \mathbb{C}^k \subseteq \Lambda(\mathbb{C}^k)$ as a list of basis coefficients, their product $x \wedge y \in \Lambda(\mathbb{C}^k)$ as a list of basis coefficients can be computed using $2^k \cdot \operatorname{poly}(k)$ arithmetic operations over $\mathbb{C}$.*

Additionally, if the bitlength of coefficients of x and y is bounded in by τ, then their product can be computed in $2^k \cdot \operatorname{poly}(\tau)$ bit operations.

As for general multiplication, let us record what Włodarczyk [35] showed implicitly:

▶ **Theorem 6** ([35])**.** *Given two elements $x, y \in \Lambda(\mathbb{C}^k)$ as a list of basis coefficients, their product $x \wedge y \in \Lambda(\mathbb{C}^k)$ as a list of basis coefficients can be computed using $2^{\omega k/2} \cdot \operatorname{poly}(k)$ arithmetic operations over $\mathbb{C}$.*

Additionally, if the bitlength of coefficients of x and y is bounded by τ, then their product can be computed in $2^{\omega k/2} \cdot \operatorname{poly}(\tau)$ bit operations.

Since the connection to the exterior algebra in [35] is not made explicitly, we give a self-contained (and somewhat simpler) proof of Włodarczyk's result, adapted to our terminology, in the full version.

▶ Remark 7. The best known way to compute a general wedge product is due to insights of Włodarczyk: This goes by reducing the wedge product to the product in a so-called Clifford algebra, which is an algebra that, filtered by degree, gives rise to the exterior algebra as its associated graded algebra. This is surprising from a mathematical perspective: The exterior algebra is an especially degenerate Clifford algebra, and yet the latter is used to compute in the former. Finding other, possibly faster approaches for the computation of wedge products is an independent research direction that we feel worth pursuing in the future, given that the wedge product is the epitomic operation in all of multilinear algebra.

An easy, but fundamental standard observation is that, for any $t \leq k$, the t-factor wedge multiplication map

$$V^t \to \Lambda(V), \; x_1, \ldots, x_t \mapsto x_1 \wedge \cdots \wedge x_t \in \Lambda(V) \tag{1}$$

can be written down in coordinates as the determinants of the $t \times t$-minors of the matrix $(x_1 \mid x_2 \mid \ldots \mid x_t)$ obtained as the juxtaposition of $x_1, \ldots, x_t$:

$$x_1 \wedge \cdots \wedge x_t = \sum_{S \in \binom{[k]}{t}} \det(x_1 \mid x_2 \mid \ldots \mid x_t)_S e_S \,,$$

where $(x_1 \mid x_2 \mid \ldots \mid x_t)_S$ is the $t \times t$-minor of the matrix $(x_1 \mid x_2 \mid \ldots \mid x_t)$ indexed at the rows S. In particular, this entails that the k-fold wedge product is just the determinant map:

$$x_1 \wedge \cdots \wedge x_k = \det(x_1 \mid x_2 \mid \ldots \mid x_t) e_{[k]} \,.$$

2.1.2 Lifts

Consider the direct sum of vector spaces $V \oplus V \cong \mathbb{C}^{2k}$, and let $\iota_1, \iota_2 : V \to V \oplus V$ be the canonical embeddings of V into $V \oplus V$ as the left and right summand, respectively. In coordinates, this corresponds respectively to pre- or appending k zeros to a vector. Consider the *lift mapping* $\Lambda(V) \to \Lambda(V \oplus V)$, $x \mapsto \overline{x} := \iota_1(x) \wedge \iota_2(x)$. We now have (see also [9, Sect. 3.4]) the following.

$$\overline{x_1} \wedge \cdots \wedge \overline{x_k} = \pm \det(x_1 \mid \cdots \mid x_k)^2 e_{[2k]},$$

where the sign only depends on k.

2.1.3 Vandermonde Codings

Determinants vanish on singular matrices. To prevent unwanted vanishing of determinants later on, we will use a set of vectors that are, in a sense, maximally linearly independent: Vandermonde vectors. To this end, consider the mapping

$$\phi : \ \mathbb{C} \to V, \ c \mapsto (1, c, c^2, \ldots, c^{k-1})$$

and its lifted variant

$$\overline{\phi} : \mathbb{C} \to V \oplus V, \ c \mapsto \overline{\phi(c)}\,.$$

This has the nice property that, on the image of ϕ (or $\overline{\phi}$ for that matter), the k-fold wedge product map as in Eq. (1) is zero only when two factors are *equal*. In particular, the mapping $\mathbb{C}^k \to \Lambda(V \oplus V)$, $(c_1, \ldots, c_k) \mapsto \overline{\phi}(c_1) \wedge \cdots \wedge \overline{\phi}(c_k)$ is zero exactly on the set of points that have at least two coordinates equal. Indeed, the coordinates of this map witness this in the clearest way possible: It is the well-known Vandermonde-determinant $\prod_{i<j}(c_i - c_j)$. An analogous statement of course holds for $\overline{\phi}$.

3 Monomial Detection Problems

As a kind of warm-up, we will give a rather direct, but very useful first application to a special kind of monomial detection problem. The general problem presents itself as follows: As input, it obtains a multivariate (and now again commutative) polynomial $f \in \mathbb{C}[X_1, \ldots, X_n]$ in n indeterminates. Our task is now to decide whether or not f contains a monomial of degree k such that no indeterminate appears twice. A variation of this is to ask whether f contains a monomial such that at least k distinct indeterminates appear in it. The degree of hardness of this problem obviously hinges on the way in which f is represented. When f is given in its sparse representation as a list of monomials and coefficients, then deciding this question is trivial.

This not the case, however, if f is represented by an arithmetic circuit, which is the setting we are interested in. Indeed, we are interested only in a semantically defined subclass of arithmetic circuits: Those that compute a polynomial f such that every multilinear monomial that appears in f does so with positive coefficient. As stated before, it is crucial to note that this does *not* mean a monotonicity restriction for the input circuit, and it may well involve cancellations of terms and negative constants. Let us formally define the set of circuits we are interested in:

▶ **Definition 8.** *Let C be an arithmetic circuit that computes a polynomial f. We call C* combinatorial *if f has non-negative coefficients on its multilinear part, and C can be evaluated over $\mathbb{Z}$ at numbers of absolute value at most τ using* poly(τ) *bit operations.*

We remark that the last condition of this definition is a barely concealed crutch to avoid having to think about subtleties regarding a possible doubly exponential blowup of inputs in general arithmetic circuits, which are irrelevant in our applications. For a very similar reason, we only speak about evaluation over $\mathbb{Z}$; namely, in order to ignore potential issues with representations of complex numbers.

3.1 Multilinear Detection

As promised, we will now start out with an easy application of Extensor-Coding. Speaking of promises, it is again in order to remark that the problems discussed henceforth are promise problems, in the sense that there is no known efficient method of checking whether an input circuit satisfies the condition of being combinatorial.

▶ **Theorem 9.** *There is a deterministic algorithm that, given a combinatorial arithmetic circuit C of size s and an integer k, decides whether or not the polynomial computed by C contains a multilinear monomial of degree k in time $2^{\omega k} \cdot \mathrm{poly}(s) < 5.19^k \cdot \mathrm{poly}(s)$.*

Proof. Consider the lifted Vandermonde coding $\overline{\phi}$, and assume that the polynomial computed by C is n-variate.

The algorithm then simply evaluates C at $(\overline{\phi}(1), \dots, \overline{\phi}(n))$, and outputs "yes" if and only if the coefficient of $e_{[2k]}$ in the resulting element of $\Lambda(V \oplus V)$ is non-zero.

The running time is immediate from the definition and Theorem 6, and correctness can be seen as follows. We first have to take care of the fact that our algebra is not commutative, strictly speaking, but nevertheless we evaluate a commutative polynomial over it. This is remedied either by a standard degree-k homogenization argument on C, and, depending on k, a subsequent single sign correction of the result. Alternatively, one can observe that the signs are consistent across each degree individually, and monomials of different degrees are linearly independent, so that the different signature arising when evaluation over the image of $\overline{\phi}$ will not make a difference. From the fact mentioned in discussion about Vandermonde codings, every monomial containing an indeterminate twice will vanish. The parts of degree less than k do not enter into the coefficient of $e_{[2k]}$, and parts of degree more than k will go to zero anyways. Now, let L be the set of multilinear monomials in the polynomial computed by f. Each such monomial m identifies a subset of $\{1, \dots, n\}$, and by abuse of notation, we will not distinguish between a monomial and a subset. Furthermore, we denote with c_m the coefficient of m, which is non-negative by the assumption on C, and let V_m be the $2k \times 2k$ matrix $(\overline{\phi}(i))_{i \in m}$. Then the coefficient of $e_{[2k]}$ can be seen to be equal to $\sum_{m \in L} c_m \cdot \det(V_m)^2$, which is non-zero if and only if one of the determinants is non-zero. This in turn happens if and only if there is a multilinear monomial of degree k in the polynomial computed by C. ◀

We also obtain the more useful skew variant:

▶ **Theorem 10.** *There is a deterministic algorithm that, given a skew combinatorial arithmetic circuit C of size s and an integer k, decides whether or not the polynomial computed by C contains a multilinear monomial of degree k in time $4^k \cdot \mathrm{poly}(s)$.*

Proof. Follows verbatim like Theorem 9 after replacing Theorem 6 by Proposition 5. ◀

3.2 k-Distinct Detection

We will now turn to the monomial detection problem that will later on be used in applications, namely the *k-distinct detection problem.* Again, the input here is an arithmetic circuit, but this time, the task is to decide whether there exists a monomial containing at least k distinct indeterminates. Using the folklore trick of replacing every variable x_i by $1 + t \cdot x_i$ with a formal indeterminate t, turning a nilpotent variable x_i into an (almost) idempotent one, and then extracting the coefficient of t^k, we obtain:[3]

▶ **Theorem 11.** *There is a deterministic algorithm that, given a skew combinatorial arithmetic circuit C of size s that computes a polynomial that* only *has non-negative coefficients, and an integer k, decides whether or not the polynomial computed by C contains a monomial with at least k different variables in time $4^k \cdot \mathrm{poly}(s)$.*

A proof sketch is given in the full version. Equipped with these observations, we may now turn to our application problems.

4 Graph Problems

We will make use of the classic Directed Matrix-Tree Theorem, following the presentation in [7]. Let us first define the *Laplacian* of a directed graph $G = (D, A)$. To this end, let $X = \{x_a \mid a \in A\}$ be a set of formal indeterminates labeled with the arcs of a graph, and define the matrix $L = (\ell_{uv})_{u,v \in V}$ through

$$\ell_{uv} = \begin{cases} \sum_{w \in V : wu \in A} x_{wu} & \text{if } u = v \\ -x_{uv} & \text{if } uv \in A \\ 0 & \text{if } uv \notin A \end{cases} .$$

After fixing a root $r \in V$, we will consider L_r, the *Laplacian punctured at* r, which is defined as the matrix obtained from L by striking row r and column r. With these definitions in place, we have the following well-known theorem, and just as [7], we refer to the corresponding chapter of Gessel and Stanley in the Handbook of Combinatorics [20] for a proof.

▶ **Theorem 12** (Directed Matrix-Tree Theorem)**.** *Let $G = (D, A)$ be a directed graph. For all $r \in V$, the following holds.*

$$\det L_r = \sum_{\substack{T = (V,B) \text{ is an} \\ \text{out-branching of } G \\ \text{rooted at } r}} \prod_{b \in B} x_b .$$

In other words, the determinant of L_r is the multivariate generating function of the set of out-branchings rooted at r. The important insight is now the following: All known (randomized) efficient algorithms for detecting k-multilinear terms – and, by extension, k-distinct terms – in the polynomial computed by an arithmetic circuit rely on this circuit not involving cancellations in their computation, i.e., they need to be *monotone*.

However, by a theorem of Jerrum and Snir [23], computing $\det L_r$ using such a monotone circuit requires circuits of *exponential size* in n.

On the other hand, there are efficient skew arithmetic circuits (this time with cancellations) for computing the $n \times n$ determinant polynomial:

▶ **Theorem 13** ([31])**.** *There is a family of skew arithmetic circuits $(C_n)_{n \in \mathbb{N}}$ such that C_n computes the $n \times n$ determinant polynomial, and the size $s(n)$ of C_n satisfies $s(n) \leq \mathrm{poly}(n)$. Furthermore, there is an algorithm that, upon input 1^n, outputs a description of C_n in time* $\mathrm{poly}(n)$*, and every circuit can be evaluated over $\mathbb{Z}$ in polynomial time in the length of the input representation.*

[3] We might just replace the indeterminate t by 1, and extract the degree-k term of the result in the exterior algebra. Using an extra indeterminate t allows us to avoid explaining how *degree* is a well-defined concept even over exterior algebras.

In fact, the $n \times n$ determinant polynomial is complete – for a suitable notion of reduction – for the adequately named class VDET of polynomial families computable by poly(n)-sized skew arithmetic circuits, defined *ibid.* Importantly, this together with the Matrix-Tree Theorem 12 also shows that $\det L_r$ is a combinatorial polynomial.

4.1 Out-Branchings

Let us now proceed gently with a first application of what we have gathered so far.

▶ **Theorem 14.** *There is a deterministic algorithm that, given a directed edge-colored graph D on n vertices and an integer k, decides whether D has a k-colorful out-branching in time* $4^k \cdot \mathrm{poly}(n)$.

Proof. First, replace every variable x_a by a fresh variable corresponding to its color $c(a)$, say $y_{c(a)}$, and denote the corresponding symbolic matrix with $L_r(y)$. Since $\det L_r$ is combinatorial and skew, so is $\det L_r(y)$, and we can perform the k-distinct detection from Theorem 11 in the claimed running time. The existence of a monomial with k distinct variables in $\det L_r(y)$ is now clearly equivalent to the existence of a k-colorful out-branching in D. ◀

Theorems 12, 13 and Theorem 11 immediately yield a deterministic algorithm for the problem, running in time $4^k \cdot \mathrm{poly}(n)$. We note that this is already a significant improvement over the time bound of $5.14^k \cdot \mathrm{poly}(n)$ on the runner-up algorithm of [37].

▶ **Proposition 15** (Superseded by [22])**.** *There is a deterministic algorithm that, given a directed graph D on n vertices and an integer k, decides whether D has a k-internal out-branching in time* $4^k \cdot \mathrm{poly}(n)$.

Proof. First, replace every variable x_{uv} be a fresh variable corresponding to its tail, say y_u, and denote the corresponding symbolic matrix with $L_r(y)$. Since $\det L_r$ is combinatorial and skew, so is the n-variate polynomial $\det L_r(y)$, and we can perform the k-distinct detection from Theorem 11 in the claimed running time. It has been observed by Björklund et al. [7] that this is neatly equivalent to the input instance containing a k-internal out-branching. ◀

To speed this up, we can more or less just plug in an Extensor-Coding into the analysis from [22] and set a new record bound for the k-internal out-branching problem. We defer the proof to the full version.

▶ **Theorem 16.** *There is a deterministic algorithm that, given a directed graph D on n vertices and an integer k, decides whether D has a k-internal out-branching in time* $3.21^k \cdot \mathrm{poly}(n)$.

The corresponding results for k-internal spanning trees of undirected graphs follow immediately by standard reductions to the directed case (see [22]).

4.2 Colorful Planar Perfect Matchings

Just like out-branchings, perfect matchings in planar graphs have an efficiently computable multivariate generating function, namely the *Pfaffian* $\mathrm{Pf}\, A$ of a suitable skew-symmetric matrix A. Gutin et al. [22] employ a determinantal identity for the Pfaffian, and do so by evaluating the determinant polynomial in question over the integers in a black-box fashion. Namely, they exploit that $\mathrm{Pf}(A)^2 = \det(A)$. In the very last step, this requires a square root extraction, which is a perfectly viable path over the integers, but not over more general algebras. Therefore, instead of construing $\mathrm{Pf}(A)$ as $\sqrt{\det(A)}$, we rely on an observation of Flarup et al. [15], who show that the Pfaffian has efficient skew circuits. Putting this together, we immediately obtain:

▶ **Theorem 17.** *There is a deterministic algorithm that, given an undirected edge-colored planar graph G on n vertices and an integer k, decides whether G has a k-colorful perfect matching in time $4^k \cdot \mathrm{poly}(n)$.*

5 The Dimension Barrier

In this section, we will explain Theorem 4. The entire proof is deferred to the full version.

5.1 The Subalgebra Generated by Evaluation Points

Let us first elaborate on the fundamental insight that motivates this result and that was alluded to already in the introduction: Let f be a complex polynomial, such as the polynomial computed by a circuit in, say, Theorem 11. During the evaluation of this polynomial f (or rather the circuit computing it) at points from the image of $\overline{\phi}$, only sums and wedge products of elements of the image of $\overline{\phi}$ are ever formed. Note that this is what we do in all our applications: We only plug in points of the form $\overline{\phi}(c)$ for some $c \in \mathbb{C}$. We can say this rigorously and concisely by stating that during such an evaluation, all computation may only occur in the *subalgebra of* $\Lambda(V \oplus V)$ *generated by the image of* $\overline{\phi}$, which is – by definition – the set of all sum-wedge product combinations of the generating set. It is precisely the subalgebra generated by these elements $\overline{\phi}(c)$ that we will study. The fundamental quantity associated with this subalgebra is its dimension, that is, the dimension of the subalgebra as a complex vector space. Let us stress again the argument from the introduction that any potential progress on the technique in its present form hinges on the dimension of this subalgebra: It provides both strict lower bounds and a good guide towards the upper bounds one can hope for when solving problems using Extensor-Coding.

5.2 A Family of Related Subalgebras

The decisive property that makes Vandermonde codings so useful is that any distinct k of them are linearly independent, which is commonly referred to the set of Vandermonde vectors being in general position. This, however, is not only a property enjoyed by Vandermonde vectors. First of all, any finite random set of vectors is in general position almost certainly.

This suggests that instead of considering the subalgebra generated by Vandermonde codings, one might as well just take any n random vectors and proceed with them. Extensive computational experiments have shown, however, that the expected dimension of this subalgebra is almost as high as the dimension of the full algebra. Curiously, it seems to be equal (and in fact this was the only case that ever occurred during all our experiments) to the k-th Catalan number, which grows asymptotically faster than $4^{(1-\varepsilon)k}$ for all $\varepsilon > 0$.

It is worth pondering about this phenomenon for a little while. From a matroid perspective, the Vandermonde codings just correspond to a representation of the k-uniform matroid over an n-element universe. One might now hope that the dimension of the subalgebra generated by the lifts of the columns of this representation matrix is a matroid invariant; however, this is *not* the case. Even more, most representations are just as bad as the worst case.

Remarkably, the extraordinarily well-behaved case of the Vandermonde representation transfers quite directly to a related family of subalgebras. Consider any set $P = \{p_1, \ldots, p_k\}$ of formal univariate polynomials that are linearly independent and of degree less than k. In other words, a basis of the set of polynomials of degree less than k. We can form, for any n, a $k \times n$ evaluation matrix of this set P, where the i-th row is given as $(p_i(1), p_i(2), \ldots, p_i(n))$. If we pick as P the standard monomial basis, this is just the Vandermonde representation.

However, we may as well pick any other basis, and the resulting matrix will again have the property that any subset of k columns is linearly independent. This works, provided the p_i do not have a common root at the evaluation point. We can take care of this easily by just picking a different, appropriate set of evaluation points. Note that this set exists (and in fact, almost certainly, any random set will do) by the fact that the p_i are distinct polynomials. Now, we can again study the subalgebra of $\Lambda(V \oplus V)$ generated by the lifts of the column vectors of this evaluation matrix. Surprisingly, the argument for the upper dimension bound carries over immediately, but we could not find a corresponding proof for the lower bound. This opens up the exciting possibility of finding lower-dimensional algebras in this family, that could lead to even faster algorithms. Note that despite the main motivation for studying this algebra may stem from the flagship k-path problem, due to the connection to multilinear detection, finding these algorithms has impacts for all the problems that reduce to this special case of multilinear detection, including those studied in this paper.

5.3 The Barrier

As announced, we will merely state the relevant theorem. A full, detailed proof can be found in the full version.

▶ **Theorem 18.** *Let $V = \mathbb{C}^k$ and let F_{2k+1} be the $(2k+1)$th Fibonacci number. The subalgebra of $\Lambda(V \oplus V)$ generated by the image $\{\overline{\phi}(c) \mid c \in \mathbb{C}\}$ of $\overline{\phi}$ is of dimension exactly equal to F_{2k+1}.*

Furthermore, for any linear basis P of the univariate polynomials of degree at most k, denote for $c \in \mathbb{C}$ with $P(c)$ the column vector obtained by evaluating all polynomials in P at c. Then, the subalgebra generated by $\{\overline{P(c)} \mid c \in \mathbb{C}\}$ is of dimension at most F_{2k+1}.

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

2 Vikraman Arvind, Abhranil Chatterjee, Rajit Datta, and Partha Mukhopadhyay. Fast Exact Algorithms Using Hadamard Product of Polynomials. *CoRR*, abs/1807.04496, 2018. arXiv:1807.04496.

3 Vikraman Arvind, Abhranil Chatterjee, Rajit Datta, and Partha Mukhopadhyay. Univariate Ideal Membership Parameterized by Rank, Degree, and Number of Generators. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, pages 7:1–7:18, 2018. doi:10.4230/LIPIcs.FSTTCS.2018.7.

4 Garrett Birkhoff and Saunders Mac Lane. *Algebra.* AMS, 1999.

5 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: Fast subset convolution. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 67–74, 2007. doi:10.1145/1250790.1250801.

6 Andreas Björklund, Vikram Kamat, Lukasz Kowalik, and Meirav Zehavi. Spotting Trees with Few Leaves. *SIAM J. Discrete Math.*, 31(2):687–713, 2017. doi:10.1137/15M1048975.

7 Andreas Björklund, Petteri Kaski, and Ioannis Koutis. Directed Hamiltonicity and Out-Branchings via Generalized Laplacians. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 91:1–91:14, 2017. doi:10.4230/LIPIcs.ICALP.2017.91.

8 Cornelius Brand. *Paths and Walks, Forests and Planes.* PhD thesis, Saarland University, 2019.

9 Cornelius Brand, Holger Dell, and Thore Husfeldt. Extensor-coding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 151–164, New York, NY, USA, 2018. ACM. doi:10.1145/3188745.3188902.

10 Jianer Chen, Songjian Lu, Sing-Hoi Sze, and Fenghui Zhang. Improved algorithms for path, matching, and packing problems. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 298–307, 2007. URL: `http://dl.acm.org/citation.cfm?id=1283383.1283415`.

11 Nathann Cohen, Fedor V. Fomin, Gregory Z. Gutin, Eun Jung Kim, Saket Saurabh, and Anders Yeo. Algorithm for finding k-vertex out-trees and its application to k-internal out-branching problem. *J. Comput. Syst. Sci.*, 76(7):650–662, 2010. `doi:10.1016/j.jcss.2010.01.001`.

12 Artur Czumaj, editor. *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*. SIAM, 2018. `doi:10.1137/1.9781611975031`.

13 Jean Daligault. *Techniques combinatoires pour les algorithmes paramétrés et les noyaux, avec applications aux problèmes de multicoupe. (Combinatorial Techniques for Parameterized Algorithms and Kernels, with Applications to Multicut.)*. PhD thesis, Montpellier 2 University, France, 2011. URL: `https://tel.archives-ouvertes.fr/tel-00804206`.

14 Henning Fernau, Serge Gaspers, and Daniel Raible. Exact and parameterized algorithms for max internal spanning tree. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 100–111. Springer, 2009.

15 Uffe Flarup, Pascal Koiran, and Laurent Lyaudet. On the Expressive Power of Planar Perfect Matching and Permanents of Bounded Treewidth Matrices. In *Algorithms and Computation, 18th International Symposium, ISAAC 2007, Sendai, Japan, December 17-19, 2007, Proceedings*, pages 124–136, 2007. `doi:10.1007/978-3-540-77120-3_13`.

16 Fedor V Fomin, Serge Gaspers, Saket Saurabh, and Stéphan Thomassé. A linear vertex kernel for maximum internal spanning tree. *Journal of Computer and System Sciences*, 79(1):1–6, 2013.

17 Fedor V. Fomin, Fabrizio Grandoni, Daniel Lokshtanov, and Saket Saurabh. Sharp Separation and Applications to Exact and Parameterized Algorithms. *Algorithmica*, 63(3):692–706, 2012. `doi:10.1007/s00453-011-9555-9`.

18 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Representative Sets of Product Families. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 443–454. Springer, 2014. `doi:10.1007/978-3-662-44777-2_37`.

19 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient Computation of Representative Families with Applications in Parameterized and Exact Algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. `doi:10.1145/2886094`.

20 Ira M. Gessel and Richard P. Stanley. Algebraic Enumeration. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics (Vol. 2)*, pages 1021–1061. MIT Press, Cambridge, MA, USA, 1995. URL: `http://dl.acm.org/citation.cfm?id=233228.233231`.

21 Gregory Gutin, Igor Razgon, and Eun Jung Kim. Minimum leaf out-branching and related problems. *Theoretical Computer Science*, 410(45):4571–4579, 2009.

22 Gregory Z. Gutin, Felix Reidl, Magnus Wahlström, and Meirav Zehavi. Designing deterministic polynomial-space algorithms by color-coding multivariate polynomials. *J. Comput. Syst. Sci.*, 95:69–85, 2018. `doi:10.1016/j.jcss.2018.01.004`.

23 Mark Jerrum and Marc Snir. Some Exact Complexity Results for Straight-Line Computations over Semirings. *J. ACM*, 29(3):874–897, 1982. `doi:10.1145/322326.322341`.

24 Ioannis Koutis. Faster Algebraic Algorithms for Path and Packing Problems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 575–586. Springer, 2008. `doi:10.1007/978-3-540-70575-8_47`.

25 Ioannis Koutis and Ryan Williams. Algebraic fingerprints for faster algorithms. *Commun. ACM*, 59(1):98–105, 2016. `doi:10.1145/2742544`.

26 Ioannis Koutis and Ryan Williams. Limits and Applications of Group Algebras for Parameterized Problems. *ACM T. Algorithms*, 12(3):31:1–31:18, 2016. doi:10.1145/2885499.

27 Wenjun Li, Yixin Cao, Jianer Chen, and Jianxin Wang. Deeper local search for parameterized and approximation algorithms for maximum internal spanning tree. *Inf. Comput.*, 252:187–200, 2017. doi:10.1016/j.ic.2016.11.003.

28 Kevin Pratt. Faster Algorithms via Waring Decompositions. *CoRR*, abs/1807.06194, 2018. arXiv:1807.06194.

29 Elena Prieto and Christian Sloper. Reducing to independent set structure: the case of k-internal spanning tree. *Nordic Journal of Computing*, 12(3):308–318, 2005.

30 Andrew James Stothers. *On the complexity of matrix multiplication.* PhD thesis, The University of Edinburgh, 2010.

31 Seinosuke Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Transactions on Information and Systems*, 75(1):116–124, 1992.

32 Chris Umans. Fast generalized DFTs for all finite groups. *CoRR*, abs/1901.02536, 2019. arXiv:1901.02536.

33 Ryan Williams. Finding paths of length k in $O(2^k)$ time. *Inform. Process. Lett.*, 109(6):315–318, 2009. doi:10.1016/j.ipl.2008.11.004.

34 Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 887–898, 2012. doi:10.1145/2213977.2214056.

35 Michał Włodarczyk. Clifford Algebras Meet Tree Decompositions. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63 of *LIPIcs*, pages 29:1–29:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.29.

36 Meirav Zehavi. Mixing Color Coding-Related Techniques. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 1037–1049, 2015. doi:10.1007/978-3-662-48350-3_86.

37 Meirav Zehavi. Mixing Color Coding-Related Techniques. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA)*, volume 9294, pages 1037–1049. Springer, 2015. doi:10.1007/978-3-662-48350-3_86.

38 Meirav Zehavi. Algorithms for k-Internal Out-Branching and k-Tree in Bounded Degree Graphs. *Algorithmica*, 78(1):319–341, May 2017. doi:10.1007/s00453-016-0166-3.