On r-Simple k-Path and Related Problems Parameterized by k/r

Gregory Gutin* Magnus Wahlström† Meirav Zehavi‡

Abstract

Abasi et al. (2014) introduced the following two problems. In the r-Simple k-Path problem, given a digraph G on n vertices and positive integers r, k, decide whether G has an r-simple k-path, which is a walk where every vertex occurs at most r times and the total number of vertex occurrences is k. In the (r, k)-Monomial Detection problem, given an arithmetic circuit that succinctly encodes some polynomial P on n variables and positive integers k, r, decide whether P has a monomial of total degree k where the degree of each variable is at most r. Abasi et al. obtained randomized algorithms of running time $4^{(k/r)\log r} \cdot n^{\mathcal{O}(1)}$ for both problems. Gabizon et al. (2015) designed deterministic $2^{\mathcal{O}((k/r)\log r)} \cdot n^{\mathcal{O}(1)}$-time algorithms for both problems (however, for the (r, k)-Monomial Detection problem the input circuit is restricted to be non-canceling). Gabizon et al. also studied the following problem. In the p-Set (r, q)-Packing problem, given a universe V, positive integers p, q, r, and a collection $\mathcal{H}$ of sets of size p whose elements belong to V, decide whether there exists a subcollection $\mathcal{H}'$ of $\mathcal{H}$ of size q where each element occurs in at most r sets of $\mathcal{H}'$. Gabizon et al. obtained a deterministic $2^{O((pq/r)\log r)} \cdot n^{\mathcal{O}(1)}$-time algorithm for p-Set (r, q)-Packing.

The above results prove that the three problems are *single-exponentially* fixed-parameter tractable (FPT) when parameterized by the product of *two* parameters, that is, k/r and $\log r$, where $k = pq$ for p-Set (r, q)-Packing. Abasi et al. and Gabizon et al. asked whether the $\log r$ factor in the exponent can be avoided. Bonamy et al. (2017) answered the question for (r, k)-Monomial Detection by proving that unless the Exponential Time Hypothesis (ETH) fails there is no $2^{o((k/r)\log r)} \cdot (n + \log k)^{\mathcal{O}(1)}$-time algorithm for (r, k)-Monomial Detection, i.e. (r, k)-Monomial Detection is highly unlikely to be single-exponentially FPT when parameterized by k/r alone. The question remains open for r-Simple k-Path and p-Set (r, q)-Packing.

We consider the question from a wider perspective: are the above problems FPT when parameterized by k/r only, i.e. whether there exists a computable function f such that the problems admit a $f(k/r)(n + \log k)^{\mathcal{O}(1)}$-time algorithm? Since r can be substantially larger than the input size, the algorithms of Abasi et al. and Gabizon et al. do not even show that any of these three problems is in XP parameterized by k/r alone. We resolve the wider question by (a) obtaining a $2^{\mathcal{O}((k/r)^2\log(k/r))} \cdot (n + \log k)^{\mathcal{O}(1)}$-time algorithm for r-Simple k-Path on digraphs and a $2^{\mathcal{O}(k/r)} \cdot (n + \log k)^{\mathcal{O}(1)}$-time algorithm for r-Simple k-Path on undirected graphs (i.e., for undirected graphs we answer the original question in affirmative), (b) showing that p-Set (r, q)-Packing is FPT (in contrast, we prove that p-Multiset (r, q)-Packing is W[1]-hard), and (c) proving that (r, k)-Monomial Detection is para-NP-hard even if only two distinct variables are in polynomial P and the circuit is non-canceling. For the special case of (r, k)-Monomial Detection where k is polynomially bounded by the input size (which is in XP), we show W[1]-hardness. Along the way to solve p-Set (r, q)-Packing, we obtain a polynomial kernel for any fixed p, which resolves a question posed by Gabizon et al. regarding the existence of polynomial kernels for problems with relaxed disjointness constraints. All our algorithms are deterministic.

*Royal Holloway, University of London, UK. g.gutin@rhul.ac.uk

†Royal Holloway, University of London, UK. magnus.wahlstrom@rhul.ac.uk

‡Ben-Gurion University, Beersheba, Israel. meiravze@bgu.ac.il

1 Introduction

Abasi et al. [1] introduced the following extension of the Directed k-Path problem called the Directed r-Simple k-Path problem: given an n-vertex digraph G and positive integers k, r,[1] decide whether G has an r-simple k-path, that is, a walk where every vertex occurs at most r times and the total number of vertex occurrences is k. At first glance, one may think that the time complexity of any algorithm for solving Directed r-Simple k-Path is an increasing function in r. However, Abasi et al. showed that this is not the case by designing a randomized algorithm of running time $4^{(k/r)\log r} \cdot n^{\mathcal{O}(1)}$. Their algorithm was obtained by a simple reduction to the (r,k)-Monomial Detection problem in which the input consists of an arithmetic circuit that succinctly encodes some n-variable polynomial P, and positive integers k, r. The goal is to decide whether P has a monomial of total degree k, where the degree of each variable is at most r. Abasi et al. proved that (r,k)-Monomial Detection can be solved by a randomized algorithm with time complexity $4^{(k/r)\log r} \cdot n^{\mathcal{O}(1)}$. Gabizon et al. [20] derandomized these two randomized algorithms, though at the expense of increasing the constant factor in the exponent and restricting the input of the (r,k)-Monomial Detection problem to non-canceling circuits.[2] Both algorithms of Gabizon et al. run in time $2^{\mathcal{O}((k/r)\log r)} \cdot n^{\mathcal{O}(1)}$. Gabizon et al. [20] also studied the p-Set (r,q)-Packing problem in which the input consists of an n-element universe V, positive integers p, q, r, and a collection $\mathcal{H}$ of sets of size p whose elements belong to V. The goal is to decide whether there exists a subcollection $\mathcal{H}'$ of $\mathcal{H}$ of size q where each element occurs in at most r sets of $\mathcal{H}'$. Gabizon et al. designed an algorithm for p-Set (r,q)-Packing of running time $2^{\mathcal{O}((k/r)\log r)} \cdot n^{\mathcal{O}(1)}$, where $k = pq$. In other words, the above results show that the three problems are *single-exponentially* fixed-parameter tractable (FPT) when parameterized by the product of *two* parameters, k/r and $\log r$.

The motivation behind the relaxation of disjointness constraints is to enable finding *substantially better (larger)* solutions at the expense of allowing elements to be used multiple (but bounded by r) times. For example, for any choice of k, r, Abasi et al. [1] presented digraphs that have at least one r-simple k-path but do not have even a single (simple) path on $4\log_r k$ vertices. Thus, even if we allow each vertex to be visited at most twice rather than once, already we can gain an *exponential* increase in the size of the output solution. The same result holds also for undirected graphs.[3] In addition, Abasi et al. [1] showed that the relaxation does not make the problem easy: both Undirected r-Simple k-Path and Directed r-Simple k-Path are shown to be NP-hard with $k = (2r-1)n+2$. From this, we observe that NP-hardness holds for a wide variety of choices of r, ranging for r being any fixed constant to r being super-exponential in n (e.g., $r = 2^{n^c}$ for any fixed constant $c \geq 1$). In addition, NP-hardness holds when $k/r = k$ as well as when $k/r = \mathcal{O}(\log^{1/c} k)$ for any fixed constant $c \geq 1$.

As an open problem, both Abasi et al. and Gabizon et al. asked whether it is possible to avoid an exponential dependency on $\log r$. In other words, they asked whether the above problems are single-exponentially FPT when parameterized by k/r alone.[4] To answer this question for (r,k)-Monomial Detection, Bonamy et al. [11] proved that the running time of the algorithms of Abasi et al. [1] and of Gabizon et al. [20] for (r,k)-Monomial Detection are optimal under the Exponential Time Hypothesis (ETH): Unless ETH fails there is no $2^{o((k/r)\log r)} \cdot (n+\log k)^{\mathcal{O}(1)}$-time algorithm for (r,k)-Monomial Detection even if $r = \Theta(k^\sigma)$ for any $\sigma \in [0,1)$. The question remains open for Directed r-Simple k-Path and p-Set (r,q)-Packing.

We consider the question from a wider perspective of parameterized complexity: are the above problems FPT when parameterized by k/r only, i.e. whether there exists a computable

[1] Note that k, r can be substantially larger than n.

[2] A non-cancelling circuit has only variables at its leaves and only addition and multiplication gates.

[3] Undirected r-Simple k-Path can be viewed as the special case of Directed r-Simple k-Path where every pair of vertices has either no arc or arcs in both directions.

[4] The interpretation of k/r is a tight lower bound on the number of distinct elements any solution must use.

function f such that the problems admit a $f(k/r)(n + \log k)^{\mathcal{O}(1)}$-time algorithm?

Note that the above algorithms by Abasi et al. and Gabizon et al. are not even XP-algorithms in the parameter k/r because r (encoded in binary) can be much larger than the size of the problem instance under consideration. In particular, even when $k/r = 1$, these algorithms can run in time exponential in the input size. In addition, note that all three problems are easily seen to be FPT when parameterized by k/r and r simultaneously, since algorithms that run in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ immediately follow by simple modifications of known algorithms for the corresponding non-relaxed versions. When r is large enough, the running times of $2^{\mathcal{O}((k/r)\log r)} \cdot n^{\mathcal{O}(1)}$ of the algorithms by Abasi et al. and Gabizon et al. are superior. Here, the $\log r$ factor in the exponent naturally arises, and seems to be perhaps unavoidable. To see this, first consider the very special case where the input contains only $\mathcal{O}(k/r)$ distinct elements. Then, we can store *counters* that keep track of how many times each element is used. Our array of counters would have $2^{\mathcal{O}((k/r)\log r)}$ possible configurations, hence a running time of $2^{\mathcal{O}((k/r)\log r)} \cdot n^{\mathcal{O}(1)}$ is trivial. However, counters are completely prohibited when dependence on r is forbidden, which already renders this extreme special case non-obvious. In fact, a running time of $f(k/r) \cdot (n + \log k)^{\mathcal{O}(1)}$ not only disallows using such an array of counters, but it forbids the usage of *even a single counter*. Thus, one might expect that all three problems are W[1]-hard with respect to k/r.

Our Contribution. We resolve the parameterized complexity of all three problems, namely Directed r-Simple k-Path, p-Set (r, q)-Packing and (r, k)-Monomial Detection, with respect to the parameter k/r. Our main contribution consists of a $2^{\mathcal{O}((k/r)^2 \log(k/r))} \cdot (n + \log k)^{\mathcal{O}(1)}$-time algorithm for Directed r-Simple k-Path and a $2^{\mathcal{O}(k/r)} \cdot (n + \log k)^{\mathcal{O}(1)}$-time algorithm for Undirected r-Simple k-Path.[5] For Undirected r-Simple k-Path, this answers the question posed by Abasi et al. [1] and Gabizon et al. [20], and reiterated by Bonamy et al. [11] and Socala [30]. The proofs are discussed in Sections 3 and 4. (As also noted in previous works, it is easily seen that even when k is polynomial in n, none of the three problems can be solved in time $2^{o(k/r)} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails.) In addition, we show that p-Set (r, q)-Packing is FPT based on the representative set method. The proof is outlined in Section 5. Along the way to prove this result, we obtain a polynomial kernel for any fixed p, which resolves another question posed by Gabizon et al. regarding the existence of polynomial kernels for problems with relaxed disjointness constraints whose sizes are decreasing functions of r. We remark that all of our algorithms are deterministic, and are based on ideas completely different from those of Abasi et al. [1] and of Gabizon et al. [20].

Next, we introduce an extension of p-Set (r, q)-Packing to multisets called the p-Multiset (r, q)-Packing problem. In p-Multiset (r, q)-Packing, $\mathcal{H}$ consists of multisets and in $\mathcal{H}'$ no element of V has more than r occurrences in total (i.e., if a multiset H in $\mathcal{H}'$ contains t copies of element $v \in V$, all other multisets of $\mathcal{H}'$ can have at most $r - t$ occurrences of v in total). We prove that p-Multiset (r, q)-Packing parameterized by k/r is W[1]-hard. Using this result, we also prove that (r, k)-Monomial Detection parameterized by k/r is W[1]-hard even if k is polynomially bounded in the input length, the number of distinct variables is k/r, and the circuit is non-canceling. Moreover, we show that (r, k)-Monomial Detection is para-NP-hard even if only two distinct variables are in polynomial P and the circuit is non-canceling. We discuss both hardness results for (r, k)-Monomial Detection in Section 5.

Related Work. Agrawal et al. [2] showed the power of relaxed disjointness conditions in the context of a problem that otherwise admits no polynomial kernel. Specifically, Agrawal et al. studied the Disjoint Cycle Packing problem: given a graph G and an integer k, decide whether G has k vertex-disjoint cycles. It is known that this problem does not admit

[5] Recall that n is the number of vertices in the input (di)graph.

a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [10]. The main result by Agrawal et al. concerns a relaxation of Disjoint Cycle Packing where every vertex can belong to at most r cycles (rather than at most one cycle). Agrawal et al. showed that this relaxation reveals a spectrum of upper and lower bounds. In particular, they obtained a (non-polynomial) kernel of size $\mathcal{O}(2^{(k/r)^2}k^{7+(k/r)}\log^3 k)$ when $(k/r) = o(\sqrt{k})$. Note that the size of the kernel depends on k.

Prior to the work by Gabizon et al. [20], packing problems with relaxed disjointness conditions have already been considered from the viewpoint of parameterized complexity (see, e.g., [25, 15, 27, 28]). Roughly speaking, these papers do not exhibit behaviors where relaxed disjointness conditions substantially (or at all) simplify the problem at hand, but rather provide parameterized algorithms and kernels with respect to k. Here, the work most relevant to us is that by Fernau et al. [15], who studied the p-Set (r,q)-Packing problem. In particular, for any $r \geq 1$, Fernau et al. proved that several very restricted versions of p-Set (r,q)-Packing with $p = 3$ are already NP-hard. Moreover, they obtained a kernel with $\mathcal{O}((p+1)^p k^p)$ vertices.

In addition, we note that Gabizon et al. [20] also studied the Degree-Bounded Spanning Tree problem: given a graph G and integer d, decide whether G has a spanning tree of maximum degree at most d. This problem demonstrates a limitation of the derandomization of Gabizon et al. as the arithmetic circuit required is not non-canceling. Thus, only randomized $2^{\mathcal{O}((n/d)\log d)}$-time algorithm was obtained and designing a deterministic algorithm of such a running time remains an open problem.

Finally, let us remark that k-Path (on both directed and undirected graph) and p-Set q-Packing are both among the most extensively studied problems in Parameterized Complexity. After a long sequence of works during the past three decades, the current best known parameterized algorithms for k-Path have running times $1.657^k n^{\mathcal{O}(1)}$ (randomized, undirected only) [7, 6] (extended in [8]), $2^k n^{\mathcal{O}(1)}$ (randomized) [32] and $2.597^k n^{\mathcal{O}(1)}$ (deterministic) [33, 16, 29]. In addition, k-Path is known not to admit any polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [9].

2 Preliminaries

In the rest of this short version of the paper, we present our contribution in more detail. Due to the space limit, there are almost no proofs; all the proofs and further details are given in the full version of the paper attached as appendix. To allow for easy cross-reference between the short and full versions, we use the same numbers for assertions and definitions in both versions.

A graph is ℓ-colored if each of its vertices is assigned a color from $\{1,...,\ell\}$. For an undirected (directed, resp.) multigraph G, a walk W is an alternating sequence $v_1e_1v_2\dots e_{\ell-1}v_\ell$ such that e_i is an edge between v_i and v_{i+1} (an arc from v_i to v_{i+1}, resp.) for all $i \in \{1,2,\dots,\ell-1\}$. For any $i \in \{1,2,\dots,\ell\}$, v_i is called a *vertex occurrence* or a *vertex visit*, and for all $i \in \{1,2,\dots,\ell-1\}$, $\{v_{i-1},v_i\}$ (resp. (v_{i-1},v_i)) an *edge occurrence* (resp. *arc occurrence*) or an *edge visit* (resp. *arc visit*). For a walk P, the *multisets* of vertex visits and edge (resp. arc) visits are denoted by $V(P)$ and $E(P)$ (resp. $A(P)$), respectively.

An *r-simple path* is a walk where every vertex occurs at most r times. Moreover, an *r-simple k-path* is an r-simple path of size k. Note that a 1-simple path is just a path. A 2-simple path that is a closed walk where every vertex occurs at most once, except for the last and first vertex which occur twice, is just a cycle. Note that by this definition, the first and last vertex of a cycle are well defined. Given vertices $s,t \in V(G)$, an *(s,t)-path* is a path that starts at s and ends at t. Similarly, an *(s,t)-cycle* is a cycle that starts at s and ends at t, in which case $s = t$. To avoid writing some explanations twice, we refer to an (s,s)-cycle also as an (s,s)-path. More generally, an *r-simple (s,t)-path* is an r-simple k-path that starts at s and ends at t.

We use standard terminology and notation in parameterized algorithmics and graph theory, see, e.g., [4, 12, 13, 14].

3 Directed r-Simple k-Path: FPT

In this section, we outline the proof of the following theorem.

Theorem 3.1 Directed r-Simple k-Path *is FPT parameterized by k/r. In particular,* Directed r-Simple k-Path *is solvable in time $2^{\mathcal{O}((\frac{k}{r})^2 \log(\frac{k}{r}))}(n+\log k)^{\mathcal{O}(1)}$ and polynomial space (polynomial in $n + \log k + (k/r)$).*

We begin with two simple claims that reduce the Directed r-Simple Long (s,t)-Path problem to the Directed r-Simple Long (s,t)-Path problem, where we are given a strongly connected digraph G, postive integers k, r, and vertices $s, t \in V(G)$ and the objective is to either *(i)* determine that G has an r-simple k-path or *(ii)* output the largest integer $i \leq k$ such that G has an r-simple (s,t)-path of size i. We observe that Directed r-Simple k-Path can be reduced to the special case of Directed r-Simple Long (s,t)-Path where the input digraph is strongly connected.

Lemma 3.1 *Suppose that* Directed r-Simple Long (s,t)-Path *on strongly connected digraphs can be solved in time $f(k/r) \cdot (n + \log k)^{\mathcal{O}(1)}$ and polynomial space. Then,* Directed r-Simple k-Path *can be solved in time $f(k/r) \cdot (n + \log k)^{\mathcal{O}(1)}$ and polynomial space.*

Lemma 3.2 *Let G be a strongly connected digraph. If any of the following two conditions is satisfied, then G has an r-simple k-path: (a) The graph G has a cycle of length at least k/r, (b) The graph G has a path with at least $2k/r$ vertices.*

The following known proposition asserts that we can efficiently determine whether the input digraph has a long path or a long cycle.

Theorem 3.2 [18, 34] *There exists a deterministic algorithm that given a digraph G, vertices $s, t \in V(G)$, and $k \in \mathbb{N}$, determines in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ and polynomial space whether G has a path from s to t on at least k vertices.*

Thus, from now on, we may assume not only that the input digraph is strongly connected, but that it also has neither a path of size at least $2k/r$ vertices nor a cycle of length at least k/r. Accordingly, we say that an instance (G, k, r, s, t) of Directed r-Simple Long (s,t)-Path is *nice* if G is strongly connected and it has neither a path with at least $2k/r$ vertices nor a cycle of length at least k/r. Moreover, we say that (G, k, r, s, t) is *positive* if G has an r-simple k-path, and otherwise we say that it is *negative*.

The second part of our proof concerns the establishment of an upper bound on the number of distinct (i.e., non-parallel) arcs in at least one r-simple k-path (if at least one such walk exists) or at least one r-simple (s,t)-path of maximum size.

Definition 3.1 *Let (G, k, r, s, t) be an instance of* Directed r-Simple Long (s,t)-Path. *Let P be an r-simple path in G. Let $P_{\rm simple}$ be the subgraph of G that consists of the vertices and edges in G that are visited at least once by P, and let $P_{\sf multi}$ be the directed multigraph obtained from $P_{\rm simple}$ by replacing each arc a by its c_a copies, where c_a is the number of times a is visited on P. Let $V(P, r)$ be the set that contains s, t and every vertex that occurs r times in P, and $P^{-r}_{\rm simple} = P_{\rm simple} - V(P, r)$. For any two (not necessarily distinct) vertices $u, v \in V(P)$, denote $P^{u,v,-r}_{\rm simple} = P_{\rm simple} - (V(P, r) \setminus \{u, v\})$. (In case $u, v \notin V(P, r)$, it holds that $P^{u,v,-r}_{\rm simple} = P_{\rm simple} - V(P, r)$.)*

Our argument modifies a given walk in a manner that might increase its length to keep certain conditions satisfied. To ensure that we never need to handle a walk that is too long, we utilize the following lemma.

Lemma 3.3 *Let (G, k, r, s, t) be a nice instance of* Directed r-Simple Long (s,t)-Path. *Let P be an r-simple k'-path in G for some integer $k' \geq 2k$. Then, G has an r-simple k''-path Q, for some integer $k'' \geq k$, such that $Q_{\rm simple}$ is a subgraph of $P_{\rm simple}$ that is not equal to $P_{\rm simple}$.*

A repeated application of Lemma 3.3 brings us the following corollary.

Corollary 3.1 *Let (G,k,r,s,t) be a nice instance of* Directed r-Simple Long (s,t)-Path. *Let P be an r-simple k'-path in G for some integer $k' \geq 2k$. Then, G has an r-simple k''-path Q, for some integer $k'' \in \{k, k+1, \ldots, 2k\}$, such that Q_{simple} is a subgraph of P_{simple} that is not equal to P_{simple}.*

We now establish that if (G,k,r,s,t) is a positive instance of Directed r-Simple Long (s,t)-Path, then G has an r-simple k'-path for some $k' \in \{k, k+1, \ldots, 2k\}$ such that $V(P,r)$ and P^{-r}_{simple} satisfy three properties regarding their structure. In addition, we establish that if (G,k,r,s,t) is a negative instance of Directed r-Simple Long (s,t)-Path, then at least one r-simple (s,t)-path P in G of maximum size satisfies these three properties as well.

Lemma 3.3 *Let (G,k,r,s,t) be a nice instance of* Directed r-Simple Long (s,t)-Path. *If (G,k,r,s,t) is a positive instance, then G has an r-simple k'-path P for some $k' \in \{k, k+1, \ldots, 2k\}$ that satisfies the following three properties: 1. P^{-r}_{simple} is an acyclic digraph; 2. For any (not necessarily distinct) $u,v \in V(P)$, $P^{u,v,-r}_{\text{simple}}$ has at most one (u,v)-path;*[6] *3. $|V(P,r)| \leq 2k/r+2$. Otherwise (if (G,k,r,s,t) is a negative instance), G has an r-simple (s,t)-path P of maximum size that satisfies these three properties.*

Having Lemma 3.3 at hand, we can already bound the number of distinct arcs by $O((k/r)^3)$. Some additional arguments allow us to make the bound tight.

Lemma 3.7 *Let (G,k,r,s,t) be a nice instance of* Directed r-Simple Long (s,t)-Path. *If (G,k,r,s,t) is positive, then G has an r-simple k-path with fewer than $30(k/r)^2$ distinct arcs. Else, G has an r-simple (s,t)-path of maximum size with fewer than $30(k/r)^2$ distinct arcs.*

The above bound is tight due to the following:

Lemma 3.8 *For any integer $r \in \mathbb{N}_{\geq 2}$, there exists a nice positive instance (G,k,r,s,t) of* Directed r-Simple Long (s,t)-Path *with $k/r = \Theta(r)$ such that every r-simple k-path in G has $\Omega((k/r)^2)$ distinct arcs.*

Knowing that it suffices for us to deal only with walks having a small number of distinct arcs and hence a small number of distinct vertices, we utilize the method of color coding by Alon et al. [3]. For the sake of brevity, we define the following problem. Here, $\mathsf{b}(k/r) = 30(k/r)^2 + 1$. In the Directed Colorful r-Simple Long (s,t)-Path problem, we are given integers $k, r \in \mathbb{N}$, a strongly connected $\mathsf{b}(k/r)$-colored digraph G, and *distinct* vertices $s,t \in V(G)$. The objective is to output an integer i such that *(i)* G has an r-simple (s,t)-path of size i, and *(ii)* for any $j > i$, G does not have a colorful r-simple (s,t)-path of size j. Here, a walk is called *colorful* if every two distinct vertices visited by the walk have distinct colors.

At first glance, it might seem that the objective in the problem definition above could be replaced by the following simpler condition: output the largest integer i such that G has a colorful r-simple (s,t)-path of size i. However, we are not able to resolve this problem, and given the approach of guessing topologies that we define later, having the stronger condition will entail the resolution of a problem as hard as Multicolored Clique and hence lead to a dead-end. Now, we can see that we can focus on our colored variant Directed Colorful r-Simple Long (s,t)-Path.

Lemma 3.9 *Suppose that* Directed Colorful r-Simple Long (s,t)-Path *can be solved in time $g(k/r) \cdot (n + \log k)^{\mathcal{O}(1)}$ and polynomial space. Then,* Directed r-Simple Long (s,t)-Path *on strongly connected digraphs can be solved in time $2^{\mathcal{O}((\frac{k}{r})^2)} \cdot g(k/r) \cdot (n + \log k)^{\mathcal{O}(1)}$ and polynomial space.*

We proceed to define the notion of a topology, which we need in order to sufficiently restrict our search space.

Definition 3.4 *Let $\ell \in \mathbb{N}$. Then, an ℓ-topology is an ℓ-colored digraph with at most ℓ arcs and*

[6] Recall that if $u = v$, by a (u,v)-path we mean a (u,u)-cycle.

without isolated vertices such that each of its vertices has a distinct color. Let $\mathcal{T}_\ell$ denote the set of all ℓ-topologies.

There are not too many topologies.

Lemma 3.10 *Let $\ell \in \mathbb{N}$. Then, $|\mathcal{T}_\ell| = 2^{\mathcal{O}(\ell \log \ell)}$.*

Now, we argue that there exists a walk of the form that we seek that "complies" with at least one of our topologies. We formalize this claim in the following definition and observation.

Definition 3.5 *Let G be an ℓ-colored digraph, and let P be a colorful r-simple path in G. Let T be an ℓ-topology. We say that P* complies *with T if P_{simple} and T are isomorphic under color preservation, i.e. there exists an isomorphism ϕ between P_{simple} and T such that for all $v \in V(P_{\mathrm{simple}})$, the colors of v and $\phi(v)$ are equal. The function ϕ is called a* witness*.*

Observation 3.1 *Let (G, k, r, s, t) be an instance of* Directed Colorful r-Simple Long (s,t)-Path*. Then, for any colorful r-simple (s,t)-path P, there exists a unique topology $T \in \mathcal{T}_{\mathsf{b}(k/r)}$ with which P complies.*

In light of Observation 3.1, a natural approach to solve Directed Colorful r-Simple Long (s,t)-Path would be to guess a topology, test whether the input digraph has a subgraph isomorphic to it, and then try to answer the question of whether this topology can be extended into an r-simple (s,t)-path. However, the second step of this approach already has a major flaw—for example, if the topology is a clique, then it captures the Multicolored Clique problem. Instead, we will first check whether the topology can be extended to *any* "enriched topology" of an r-simple (s,t)-path that is still *independent of what is the input digraph*. Here, it is crucial that we do not seek all possible extensions, but only one (if any extension exists). This part will be done via integer linear programming (ILP). Notice that we cannot even explicitly write an r-simple (s,t)-path that the enriched topology encodes, since the size of it is already $\mathcal{O}(k)$ (while the input size is only $\mathcal{O}(n + \log k)$), hence checking whether the guess can be realized is slightly tricky. However, we deal with this task later. For now, let us first explain how an enrichment of a topology is defined.

Definition 3.6 Let $\ell, r \in \mathbb{N}$. In addition, let $i, j \in \{1, 2, \ldots, \ell\}$, $i \neq j$. Then, an *r-enriched ℓ-topology with endpoints* i, j is a pair (T, φ) of an ℓ-topology T and a function $\varphi : A(T) \to \{1, 2, \ldots, r\}$ with the following properties: 1. There exist vertices $s = s(T, \varphi) \in V(T)$ and $t = t(T, \varphi) \in V(T)$ colored i and j, respectively; 2. For every vertex $v \in V(T) \setminus \{s, t\}$, it holds that $\displaystyle\sum_{u:(u,v)\in A(T)} \varphi(u,v) = \sum_{u:(v,u)\in A(T)} \varphi(v,u) \leq r$; 3. $\displaystyle\sum_{u:(u,s)\in A(T)} \varphi(u,s) + 1 = \sum_{u:(s,u)\in A(T)} \varphi(s,u) \leq r$; 4. $\displaystyle\sum_{u:(u,t)\in A(T)} \varphi(u,t) = \sum_{u:(t,u)\in A(T)} \varphi(t,u) + 1 \leq r$.

Now, we show how to enrich a topology (if it is possible). For this purpose, we utilize the fact that ILP is FPT when parameterized by the number of variables [24, 22, 19].

Lemma 3.11 *There exists an algorithm that given $\ell, r \in \mathbb{N}$, $i, j \in \{1, 2, \ldots, \ell\}$, $i \neq j$, and an ℓ-topology T, determines in time $\ell^{\mathcal{O}(\ell)} \cdot (\log r)^{\mathcal{O}(1)}$ and polynomial space whether there exists a function $\varphi : A(T) \to \{1, 2, \ldots, r\}$ such that (T, φ) is an r-enriched ℓ-topology with endpoints i, j. In case the answer is positive, the algorithm outputs such a function φ that maximizes $\sum_{e \in A(T)} \varphi(e)$.*

Next, we define what does it mean for a solution to "comply" with an enriched topology.

Definition 3.7 *Let G be an ℓ-colored digraph, and let P be a colorful r-simple (s,t)-path in G. Let $\ell, r \in \mathbb{N}$, i be the color of s, j be the color of t, and (T, φ) be an r-enriched ℓ-topology with endpoints i, j. We say that P* complies with *(T, φ) if P complies with T, and for the function ϕ that witnesses this, for every arc $(u, v) \in P_{\mathrm{simple}}$, the number of copies (u, v) has in P_{multi} is exactly $\varphi(u, v)$.*

Let us now argue that the choice of how to enrich a topology is immaterial as long as at least one enrichment exists (in which case, we also need to compute such an enrichment).

Lemma 3.12 *Let G be an ℓ-colored graph, and let P be a colorful r-simple (s,t)-path in G with $s \neq t$. Let i be the color of s, and j be the color of t. Then, the following conditions hold. 1. There exists an r-enriched ℓ-topology with endpoints i,j with which P complies. 2. Let T be an ℓ-topology with which P complies. Then, for any r-enriched ℓ-topology with endpoints i,j, say (T,φ), there exists an r-simple (s,t)-path in G that complies with (T,φ).*

This lemma motivates a problem definition where the input includes an r-enriched ℓ-topology with endpoints i,j, and we seek an r-simple (s,t)-path in G that complies with it. However, like before, such a problem encompasses Multicolored Clique. Instead, we need a relaxed notion of compliance, which we define as follows.

Definition 3.8 *Let $\ell, r \in \mathbb{N}$. Let (T,φ) be an r-enriched ℓ-topology (T,φ) with endpoints i,j. Let P be an r-simple (s,t)-path in an ℓ-colored digraph G, where i is the color of s and j is the color of t. Then, P* weakly complies *with (T,φ) if the following conditions hold. (a) Every color that occurs in P also occurs in T and vice versa. That is, there exists a unique, surjective (but not necessarily injective) function $\phi : V(P_{\text{simple}}) \to V(T)$ where for all $v \in V(P_{\text{simple}})$, the colors of v and $\phi(v)$ are equal. (b) For every two colors a,b that occur in T, the number of times arcs directed from a vertex colored a to a vertex colored b occur in P is precisely $\varphi(u,v)$ where u and v are the (unique) vertices in T colored a and b, respectively.*

Note that if a walk P complies with (T,φ), then it also weakly complies with (T,φ), but the opposite is not true. In particular, a walk where some distinct vertices have the same color can weakly comply with (T,φ), but it necessarily does not comply with (T,φ).

In the (ℓ,r)-Enriched Topology problem, the input consists of an ℓ-colored digraph G, integers $\ell, r \in \mathbb{N}$, distinct vertices $s,t \in V(G)$, and an r-enriched ℓ-topology (T,φ) with endpoints i,j where i is the color of s and j is the color of t. The objective is to return Yes or No as follows. (i) If G has an r-simple (s,t)-path that complies with (T,φ), then return Yes. (ii) If G has no r-simple (s,t)-path that weakly complies with (T,φ), then return No. (iii) If none of the two conditions above holds, we can return either Yes or No.

The (ℓ,r)-Enriched Topology problem allows us to determine whether there exists an r-simple (s,t)-path in G that weakly complies with (T,φ).

Lemma 3.13 Suppose that (ℓ,r)-Enriched Topology can be solved in time $f(\ell)\cdot(n+\log r)^{\mathcal{O}(1)}$ and polynomial space. Then, Directed Colorful r-Simple Long (s,t)-Path can be solved in time $2^{\mathcal{O}(\mathsf{b}(k/r)\log(\mathsf{b}(k/r)))}\cdot f(\mathsf{b}(k/r))\cdot(n+\log k)^{\mathcal{O}(1)}$ and polynomial space.

It remains to solve the (ℓ,r)-Enriched Topology problem. This can be done by a recursive algorithm.

Lemma 3.16 (ℓ,r)-Enriched Topology *can be solved in polynomial time and space, i.e. $(\ell+n+\log r)^{\mathcal{O}(1)}$.*

Finally, we are ready to prove Theorem 2.1.

Proof of Theorem 3.1 By Lemma 3.16, (ℓ,r)-Enriched Topology can be solved in time and space $(\ell+n+\log r)^{\mathcal{O}(1)}$. Thus, by Lemma 3.13, Directed Colorful r-Simple Long (s,t)-Path can be solved in time $2^{\mathcal{O}(\mathsf{b}(k/r)\log(\mathsf{b}(k/r)))}\cdot(n+\log k)^{\mathcal{O}(1)}$ and polynomial space. Substituting $\mathsf{b}(k/r)$, this running time is upper bounded by $2^{\mathcal{O}((k/r)^2\log(k/r))}\cdot(n+\log k)^{\mathcal{O}(1)}$. In turn, by Lemma 3.9, we have that Directed r-Simple Long (s,t)-Path on strongly connected digraphs can be solved in time $2^{\mathcal{O}((k/r)^2\log(k/r))}\cdot(n+\log k)^{\mathcal{O}(1)}$ and polynomial space. Finally, by Lemma 3.1, we conclude that Directed r-Simple k-Path can be solved in time $2^{\mathcal{O}((k/r)^2\log(k/r))}\cdot(n+\log k)^{\mathcal{O}(1)}$ and polynomial space. $\square$

4 Undirected r-Simple k-Path: Single-Exponential Time

In this section, we focus on the proof of the following theorem. As discussed in the introduction, for varied relations between k and r, the running time in this theorem is *optimal* under the ETH.

Theorem 4.1 Undirected r-Simple k-Path *is solvable in time* $2^{\mathcal{O}(\frac{k}{r})}(n+\log k)^{\mathcal{O}(1)}$.

We will first discuss how to prove the following result (which is the main part of our proof).

Lemma 4.1 Undirected r-Simple k-Path *is solvable in time* $2^{\mathcal{O}(\frac{k}{r})}(r+n+\log k)^{\mathcal{O}(1)}$.

Afterwards we will discuss how to bound r. More precisely, let us refer to the special case of Undirected r-Simple k-Path where $r > \sqrt{k}$ as the Special Undirected r-Simple k-Path problem. Then, we focus on the following result.

Lemma 4.2 Special Undirected r-Simple k-Path *is solvable in time* $2^{\mathcal{O}(\frac{k}{r})}(n+\log k)^{\mathcal{O}(1)}$.

Note that if $r \leq \sqrt{k}$, then $k/r = \Omega(\sqrt{k})$, in which case $r \leq \sqrt{k} \leq 2^{O(k/r)}$. Thus, Lemmas 4.1 and 4.2 together imply Theorem 4.1.

Unfortunately, the space limit does not allow us to discuss the proof of Lemma 4.2, which entails (among other arguments) the construction a flow network. Let us outline the proof of Lemma 4.1. Using ideas from the directed case and new ideas, we can prove the following:

Lemma 4.5 *Let (G,k,r) be a nice* Yes*-instance of* Undirected r-Simple k-Path*. Then, G has an r-simple k-path with fewer than $30(k/r)$ distinct edges.*

Having Lemma 4.5 at hand, we could have continued our analysis with simplified arguments of those presented for the directed case and thus obtain an algorithm that solves Undirected r-Simple k-Path in time $2^{\mathcal{O}(\frac{k}{r}\log(\frac{k}{r}))}(n+\log k)^{\mathcal{O}(1)}$ and polynomial space. However, in order to obtain a single-exponential running time bound of $2^{\mathcal{O}(\frac{k}{r})}(n+\log k)^{\mathcal{O}(1)}$, we now take a very different route, which requires a deeper understanding of the structure of a solution. The starting point for this understanding is the following lemma.

Lemma 4.6 *Let (G,k,r) be a nice* Yes*-instance of* Undirected r-Simple k-Path*. Then, G has an r-simple k-path P with fewer than $30(k/r)$ distinct edges, such that the edge multiset of $P_{\rm multi}$ can be partitioned into two multisets, M_1 and M_2, with the following properties: (i) $P_{\rm multi}$ restricted to M_1 is a (simple) spanning tree of $P_{\rm multi}$, and (ii) $P_{\rm multi}$ restricted to M_2 has no even cycle of length at least 4.*

The usefulness in the second property in Lemma 4.6 is primarily due to the following result.

Proposition 4.1 (folklore, see [26, 31]) *A graph with no even cycle is of treewidth at most* 2.

Having Proposition 4.1 at hand, we derive the following corollary to Lemma 4.6.

Corollary 4.1 *Let (G,k,r) be a nice* Yes*-instance of* Undirected r-Simple k-Path*. Then, G has an r-simple k-path P with fewer than $30(k/r)$ distinct edges, such that the edge multiset of $P_{\rm multi}$ can be partitioned into two multisets, M_1 and M_2, with the following properties: (i) $P_{\rm multi}$ restricted to M_1 is a (simple) spanning tree of $P_{\rm multi}$, and (ii) $P_{\rm multi}$ restricted to M_2 is a multigraph of treewidth 2.*

Corollary 4.1 partitions some solution into two parts: a spanning tree and a multigraph of low treewidth. However, for a dynamic programming approach used by us, we need the first part to have some Euler (s,t)-trail rather than just be spanning tree.

It is not hard to derive the following claim from Corollary 4.1.

Lemma 4.9 *Let (G,k,r) be a nice* Yes*-instance of* Undirected r-Simple k-Path*. Then, G has an r-simple k-path P with fewer than $30(k/r)$ distinct edges, such that the edge multiset of $P_{\rm multi}$ can be partitioned into two multisets, M_1 and M_2, with the following properties: (i) $P_{\rm multi}$ restricted to M_1 is a spanning multigraph of $P_{\rm multi}$ with fewer than $60(k/r)$ edges (including* *multiplicities) that has an Eulerian (s,t)-trail where s and t are the end-vertices of P; (ii) P_{multi} restricted to M_2 is a multigraph of treewidth 2.*

Knowing that it suffices for us to deal only with solutions having a small number of distinct vertices (by Lemma 4.9), we utilize the method of color coding to focus on the following problem. Here, $\mathsf{b}(k/r) = 30k/r + 1$. In the UNDIRECTED COLORFUL r-SIMPLE k-PATH problem, we are given integers $k, r \in \mathbb{N}$, and a $\mathsf{b}(k/r)$-colored undirected graph G. The objective is to output No if G has no r-simple k-path (in this case, the input is called a No-instance), and Yes if it has a colorful r-simple k-path with fewer than $30(k/r)$ distinct edges (in this case, the input is called a Yes-instance). If the input is neither a Yes-instance nor a No-instance, the output can be arbitrary.

The proof of the following lemma follows the lines of the proof of Lemma 3.9.

Lemma 4.10 *Suppose that* UNDIRECTED COLORFUL r-SIMPLE k-PATH *can be solved in time $f(k/r) \cdot (r + n + \log k)^{\mathcal{O}(1)}$. Then,* UNDIRECTED r-SIMPLE k-PATH *can be solved in time $2^{\mathcal{O}(k/r)} \cdot f(k/r) \cdot (r + n + \log k)^{\mathcal{O}(1)}$.*

We cannot guess the topology of the spanning multigraph part of a solution in a manner similar to guessing a topology as in the case of digraphs, since trying every possibility already takes times $2^{\mathcal{O}(\frac{k}{r} \log \frac{k}{r})}$. Instead, inspired by the work of Berger et al. [5] (which guesses a degree-sequence of a certain tree), we only guess a so called "occurrence sequence" of the spanning multigraph part of a solution. Let us first define a notion that we call an *occurrence sequence.*

Definition 4.2 *Let $r, k \in \mathbb{N}$. An (r,k)-*occurrence sequence *is a tuple $\overline{\mathbf{d}} = (d_1, \ldots, d_{\mathsf{b}(k/r)})$ that satisfies the following conditions: For all $i \in \{1, 2, \ldots, \mathsf{b}(k/r)\}$, d_i is an integer between 0 and r and $\sum_{i=1}^{\mathsf{b}(k/r)} d_i \leq 2\mathsf{b}(k/r)$. Let $\mathcal{D}_{r,k}$ be the set of all (r,k)-occurrence sequences.*

We now define what structures are good and comply with an occurrence sequence. A multigraph H is called *even* if every vertex in H has even degree.

Definition 4.3 *Let $r, k \in \mathbb{N}$. Let G be a $\mathsf{b}(k/r)$-colored undirected graph. A pair (W, H) of an r-simple path W in G and an even multigraph H whose underlying simple graph is a subgraph of G is q-*good *if the following conditions are satisfied: the treewidth of H is at most 2, every connected component of H has at least one vertex that is visited by W, H is colorful, and the sum of the number of edges visited by W and the number of edges (including multiplicities) of H is $q - 1$. If q is not specified, then $q = k$.*

Definition 4.4 *Let $r, k \in \mathbb{N}$. Let G be a $\mathsf{b}(k/r)$-colored undirected graph, and let $\overline{\mathbf{d}}$ be an (r,k)-occurrence sequence. A good pair (W, H)* complies *with $\overline{\mathbf{d}}$ if for every color $i \in \{1, 2, \ldots, \mathsf{b}(k/r)\}$, the two following conditions are satisfied: the number times W visits vertices colored i is exactly d_i, and the degree of the vertex colored i in H is at most $2(r - d_i)$.*

In the (WALK,TW-2) PARTITION problem, we are given integers $k, r \in \mathbb{N}$, a $\mathsf{b}(k,r)$-colored undirected graph G, and $\overline{\mathbf{d}} \in \mathcal{D}_{r,k}$. The objective is to decide whether there exists a good pair that complies with $\overline{\mathbf{d}}$. We can focus on solving the (WALK,TW-2) PARTITION problem due to

Lemma 4.13 *Suppose that* (WALK,TW-2) PARTITION *can be solved in time $f(k/r) \cdot (r + n + \log k)^{\mathcal{O}(1)}$. Then,* UNDIRECTED COLORFUL r-SIMPLE k-PATH *can be solved in time $2^{\mathcal{O}(k/r)} \cdot f(k/r) \cdot (r + n + \log k)^{\mathcal{O}(1)}$.*

By designing a two-level dynamic programming algorithm, we can prove the following:

Lemma 4.15 (WALK,TW-2) PARTITION *can be solved in time $2^{\mathcal{O}(k/r)} \cdot (r + n + \log k)^{\mathcal{O}(1)}$.*

Proof of Lemma 4.1 Now, we can prove Lemma 4.1 as follows. By Lemma 4.15, (WALK,TW-2) PARTITION can be solved in time in time $2^{\mathcal{O}(k/r)} \cdot (r + n + \log k)^{\mathcal{O}(1)}$. Thus, by Lemma 4.13, UNDIRECTED COLORFUL r-SIMPLE k-PATH can be solved in time $2^{\mathcal{O}(k/r)} \cdot (r + n + \log k)^{\mathcal{O}(1)}$. In turn, by Lemma 4.10, UNDIRECTED r-SIMPLE k-PATH can be solved in time $2^{\mathcal{O}(k/r)} \cdot (r + n + \log k)^{\mathcal{O}(1)}$, which completes the proof. ☐

5 p-Set (r, q)-Packing and (r, k)-Monomial Detection

Recall that in the p-Set (r, q)-Packing problem, the input consists of a ground set V, positive integers p, q, r, and a collection $\mathcal{H}$ of sets of size p whose elements belong to V. The goal is to decide whether there exists a subcollection of $\mathcal{H}$ of size q where each element occurs at most r times. Note that $\mathcal{H}$ can contain copies of the same set, i.e. not all elements of $\mathcal{H}$ are *distinct* sets. Let $\kappa = pq/r$.

Our proof of Theorem 5.2 uses a reduction of a set-packing instance to a situation where the ground set has size bounded by $f(\kappa)$. The reduction uses a tool known as *representative sets* to discard irrelevant parts of the instance. Representative sets have important applications both for FPT algorithms [17] and kernels [23]; see also [12, Ch. 12].

We need only two simple reduction rules: (1) Discard any element that occurs at most r times. Exclude any empty sets, reducing q accordingly; (2) Pad $\mathcal{H}$ to be p-uniform using dummy elements for smaller sets. Compute q disjoint representative sets for the padded version of $\mathcal{H}$ in the uniform matroid $U_{n,\kappa+p}$, and discard any set in $\mathcal{H}$ not contained in any of the resulting representative sets. Using the well-known algorithm for computing representative sets in a uniform matroid, this can be done in polynomial time, and using upper bound on the output of this algorithm (see, e.g., [12, Ch. 12]) we can prove the following:

Lemma 5.2 *Assume that the two rules have been applied exhaustively. Then $n < f(pq/r)$ where $f(\kappa) = \kappa 4^{\kappa}$.*

Our proof of the next lemma uses the fact that ILP is FPT when parameterized by the number of variables.

Lemma 5.3 *An instance of p-Set (r, q)-Packing on a ground set of size n can be solved in time $O(n^{O(pn^p)})$.*

Now we can obtain the main result of this section.

Theorem 5.2 *p-Set (r, q)-Packing parameterized by κ is FPT.*

Proof. We may assume that $p < \kappa$. and our instance of p-Set (r, q)-Packing has been reduced by the two reduction rules above. By Lemma 5.2, $n < \kappa 4^{\kappa}$. Thus, by Lemma 5.3, p-Set (r, q)-Packing parameterized by κ is FPT. □

We observe that the same reduction gives a polynomial kernel when p is a constant.

Theorem 5.3 *The p-Set (r, q)-Packing problem for constant p has a polynomial-time reduction to a ground set of size $O((q/r)^{p+1})$ and a generalized polynomial kernel of $O((q/r)^{p^2+p} \log r) = O((q/r)^{2(p^2+p)} \log(q/r))$ bits.*

Corollary 5.1 *The p-Set (r, q)-Packing problem for constant p admits a polynomial size kernel.*

The next result shows that if k is not polynomially bounded in the input size, even an XP algorithm for the special case of (r, k)-Monomial Detection where only two distinct variables are present is out of reach. For this purpose, we present a reduction from the Partition problem, which is known to be NP-hard [21]. In this problem, we are given a multiset M of positive integers, and the goal is to determine whether M can be partitioned into two multisets, M_1 and M_2, such that the sum of the integers in M_1 is equal to the sum of the integers in M_2.

Theorem 6.1 (r, k)-Monomial Detection *is para-NP-hard parameterized by k/r even if the number of distinct variables is 2 and the circuit is non-canceling.*

Via a hardness result for an intermediate problem, we also have the following.

Theorem 7.2 (r, k)-Monomial Detection *is W[1]-hard parameterized by k/r even if* (i) *k is polynomially bounded in the input length,* (ii) *the number of distinct variables is k/r, and* (iii) *the circuit is non-canceling.*

References

[1] H. Abasi, N. H. Bshouty, A. Gabizon, and E. Haramaty, *On r-simple k-path*, in Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II, 2014, pp. 1–12. 1, 2

[2] A. Agrawal, D. Lokshtanov, D. Majumdar, A. E. Mouawad, and S. Saurabh, *Kernelization of cycle packing with relaxed disjointness constraints*, in 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy, 2016, pp. 26:1–26:14. 2

[3] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. ACM, 42 (1995), pp. 844–856. 5

[4] J. Bang-Jensen and G. Gutin, *Digraphs - theory, algorithms and applications*, Springer, 2002. 3

[5] A. Berger, L. Kozma, M. Mnich, and R. Vincze, *Time- and space-optimal algorithms for the many-visits TSP*, CoRR, abs/1804.06361 (2018). 9

[6] A. Björklund, *Determinant sums for undirected hamiltonicity*, SIAM J. Comput., 43 (2014), pp. 280–299. 3

[7] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, *Narrow sieves for parameterized paths and packings*, J. Comput. Syst. Sci., 87 (2017), pp. 119–139. 3

[8] A. Björklund, V. Kamat, L. Kowalik, and M. Zehavi, *Spotting trees with few leaves*, SIAM J. Discrete Math., 31 (2017), pp. 687–713. 3

[9] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin, *On problems without polynomial kernels*, J. Comput. Syst. Sci., 75 (2009), pp. 423–434. 3

[10] H. L. Bodlaender, S. Thomassé, and A. Yeo, *Kernel bounds for disjoint cycles and disjoint paths*, Theor. Comput. Sci., 412 (2011), pp. 4570–4578. 3

[11] M. Bonamy, L. Kowalik, M. Pilipczuk, A. Socala, and M. Wrochna, *Tight lower bounds for the complexity of multicoloring*, in 25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria, 2017, pp. 18:1–18:14. 1, 2

[12] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, 2015. 3, 10

[13] R. Diestel, *Graph Theory, 4th Edition*, vol. 173 of Graduate texts in mathematics, Springer, 2012. 3

[14] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013. 3

[15] H. Fernau, A. López-Ortiz, and J. Romero, *Using parametric transformations toward polynomial kernels for packing problems allowing overlaps*, TOCT, 7 (2015), pp. 13:1–13:29. 3

[16] F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh, *Efficient computation of representative families with applications in parameterized and exact algorithms*, J. ACM, 63 (2016), pp. 29:1–29:60. 3

[17] ——, *Efficient computation of representative families with applications in parameterized and exact algorithms*, J. ACM, 63 (2016), pp. 29:1–29:60. 10

[18] F. V. Fomin, D. Lokshtanov, F. Panolan, S. Saurabh, and M. Zehavi, *Long directed (s,t)-path: FPT algorithm*, Inf. Process. Lett. (To Appear), (2018). 4

[19] A. Frank and É. Tardos, *An application of simultaneous diophantine approximation in combinatorial optimization*, Combinatorica, 7 (1987), pp. 49–65. 6

[20] A. Gabizon, D. Lokshtanov, and M. Pilipczuk, *Fast algorithms for parameterized problems with relaxed disjointness constraints*, in Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings, 2015, pp. 545–556. 1, 2, 3

[21] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman, New York, 1979. 10

[22] R. Kannan, *Minkowski's convex body theorem and integer programming*, Math. Oper. Res., 12 (1987), pp. 415–440. 6

[23] S. Kratsch and M. Wahlström, *Representative sets and irrelevant vertices: New tools for kernelization*, in FOCS, IEEE Computer Society, 2012, pp. 450–459. 10

[24] H. Lenstra, *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548. 6

[25] A. López-Ortiz, C. B. Perez, and J. Romero, *Arbitrary overlap constraints in graph packing problems*, Int. J. Found. Comput. Sci., 29 (2018), pp. 101–122. 3

[26] P. Misra, V. Raman, M. S. Ramanujan, and S. Saurabh, *Parameterized algorithms for even cycle transversal*, in Graph-Theoretic Concepts in Computer Science - 38th International Workshop, WG 2012, Jerusalem, Israel, June 26-28, 2012, 2012, pp. 172–183. 8

[27] J. Romero and A. López-Ortiz, *A parameterized algorithm for packing overlapping subgraphs*, in Computer Science - Theory and Applications - 9th International Computer Science Symposium in Russia, CSR 2014, Moscow, Russia, June 7-11, 2014. Proceedings, 2014, pp. 325–336. 3

[28] ——, *The G-Packing with t-Overlap Problem*, in Algorithms and Computation - 8th International Workshop, WALCOM 2014, Chennai, India, February 13-15, 2014, Proceedings, 2014, pp. 114–124. 3

[29] H. Shachnai and M. Zehavi, *Representative families: A unified tradeoff-based approach*, J. Comput. Syst. Sci., 82 (2016), pp. 488–502. 3

[30] A. Socala, *Lower Bounds Under Strong Complexity Assumptions*, PhD thesis, University of Warsaw, 2017. 2

[31] C. Thomassen, *On the presence of disjoint subgraphs of a specified type*, Journal of Graph Theory, 12 (1988), pp. 101–111. 8

[32] R. Williams, *Finding paths of length k in $o^*(2^k)$) time*, Inf. Process. Lett., 109 (2009), pp. 315–318. 3

[33] M. Zehavi, *Mixing color coding-related techniques*, in Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings, 2015, pp. 1037–1049. 3

[34] ——, *A randomized algorithm for Long Directed Cycle*, Inf. Process. Lett., 116 (2016), pp. 419–422. 4