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

Gregory Gutin†
Magnus Wahlströni‡
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 \(4^{(k/r)}\log r \cdot n^{O(1)}\) for both problems. Gabizon et al. (2015) designed deterministic \(2^{O((k/r)\log r)} \cdot n^{O(1)}\)-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 \(2^{O((pq/r)\log r)} \cdot n^{O(1)}\)-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 \(2^{o((k/r)\log r)} \cdot (n + \log k)^{O(1)}\)-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 \(f(k/r)(n + \log k)^{O(1)}\)-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 \(2^{O((k/r)\log(k/r))} \cdot (n + \log k)^{O(1)}\)-time algorithm for r-Simple k-Path on digraphs and a \(2^{O((k/r) \cdot (n + \log k)^{O(1)}}\)-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.

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 \(4^{(k/r)\log r} \cdot n^{O(1)}\). 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 \(4^{(k/r)\log r} \cdot n^{O(1)}\). 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 \(2^{O((k/r)\log(k/r))} \cdot (n + \log k)^{O(1)}\)-time algorithm for r-Simple k-Path on digraphs and a \(2^{O((k/r) \cdot (n + \log k)^{O(1)}}\)-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.

*Research was Gutin was partially supported by Royal Society Wolfson Research Merit Award. Research of Zehavi was partially supported by Israel Science Foundation (ISF) grant no. 1176/18.
†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, Beerseha, Israel.
meiravz@bgu.ac.il

Note that k, r can be substantially larger than n.
zon et al. [24] 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 \(2^{O((k/r) \log r)} \cdot \sigma(O(1))\). Gabizon et al. [24] 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 \(H\) of sets of size \(p\) whose elements belong to \(V\). The goal is to decide whether there exists a subcollection \(H'\) of \(H\) of size \(q\) where each element occurs in at most \(r\) sets of \(H'\). Gabizon et al. designed an algorithm for p-Set \((r, q)\)-Packing of running time \(2^{O((k/r) \log r)} \cdot \sigma(O(1))\), where \(k = pq\). In other words, the above results show that the three problems are \(\text{single-exponentially}\) 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 \textit{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 \(4 \log r\) vertices. Thus, even if we allow each vertex to be visited at most twice rather than once, already we can gain an \textit{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 \text{NP}-hard with \(k = (2r - 1)n + 2\). From this, we observe that \text{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., \(r = 2^{n^r}\) for any fixed constant \(c \geq 1\)). In addition, \text{NP}-hardness holds when \(k/r = k\) as well as when \(k/r = O(\log^{1/c} k)\) 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 \text{single-exponentially} FPT when parameterized by \(k/r\) alone.\(^4\) To answer this question for \((r, k)\)-Monomial Detection, Bonamy et al. [14] proved that the running time of the algorithms of Abasi et al. [1] and of Gabizon et al. [24] for \((r, k)\)-Monomial Detection are optimal under the Exponential Time Hypothesis (ETH): Unless ETH fails there is no \(2^{o((k/r) \log r)} \cdot (n + \log k)^{O(1)}\)-time algorithm for \((r, k)\)-Monomial Detection even if \(r = \Theta(k^c)\) for any \(c \in \{0, 1\}\). 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 \text{FPT} when parameterized by \(k/r\) only, i.e., whether there exists a computable function \(f\) such that the problems admit a \(f(k/r)(n + \log k)^{O(1)}\)-time algorithm?

Note that the above algorithms by Abasi et al. and Gabizon et al. are not even \text{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 \text{FPT} when parameterized by \(k/r\) and \(r\) simultaneously, since algorithms that run in time \(2^{O((k/r) \log r)} \cdot \sigma(O(1))\) immediately follow by simple modifications of known algorithms for the corresponding non-relaxed versions. When \(r\) is large enough, the running times of \(2^{O((k/r) \log r)} \cdot \sigma(O(1))\) 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 \(O(k/r)\) distinct elements. Then, we can store \textit{counters} that keep track of how many times each element is used. Our array of counters would have \(2^{O((k/r) \log r)}\) possible configurations, hence a running time of \(2^{O((k/r) \log r)} \cdot \sigma(O(1))\) 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 \(f(k/r) \cdot (n + \log k)^{O(1)}\) 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 \(\text{W}[1]\)-hard with respect to \(k/r\).

\textbf{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 \(2^{O((k/r) \log \log (k/r))} \cdot (n + \log k)^{O(1)}\)-time algorithm for Directed \(r\)-Simple \(k\)-Path and a \(2^{O(k/r)} \cdot (n + \log k)^{O(1)}\)-time algorithm for Undirected \(r\)-Simple \(k\)-Path.\(^5\) For Undirected \(r\)-Simple \(k\)-Path, we use results of Gabizon et al. [24] and improve them.

\(^2\)A non-canceling 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.

\(^5\)Recall that \(n\) is the number of vertices in the input \((d)\)graph.
rected \( r \)-Simple \( k \)-Path, this answers the question posed by Abasi et al. [1] and Gabizon et al. [24], and reiterated by Bonamy et al. [14] and Socała [36]. The proofs are discussed in Sections 3 and 4. (As also noted in previous works, it is easily seen that even if \( k \) is polynomial in \( n \), none of the three problems can be solved in time \( 2^{o(k/r)} n^\Theta(1) \) unless the ETH fails.) In addition, we show that \( p \)-Set \((r, q)\)-Packing is FPT based on the representative sets 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. [24].

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, \( H \) consists of multisets and in \( H \) no element of \( V \) has more than \( r \) occurrences in total (i.e., if a multiset \( H \) in \( H \) contains \( t \) copies of element \( v \) \( \in \) \( V \), all other multisets of \( H \) 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 a polynomial kernel unless \( \text{NP} \subseteq \text{coNP/poly} \) [13]. 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 \( O(2^{(k/r)^2} r^{7^{(k/r)} \log^3 k}) \) when \( (k/r) = o(\sqrt{k}) \). Note that the size of the kernel depends on \( k \).

Prior to the work by Gabizon et al. [24], packing problems with relaxed disjointness conditions have already been considered from the viewpoint of parameterized complexity (see, e.g., [30, 20, 33, 34]). 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. [20], 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 \( \text{NP} \)-hard. Moreover, they obtained a kernel with \( O((p + 1) k^h) \) vertices.

In addition, we note that Gabizon et al. [24] 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 a randomized \( 2^\Omega(n/d) \log d \)-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 \( 1.657^k n^{O(1)} \) (randomized, undirected only) [10, 9] (extended in [11]), \( 2^n n^{O(1)} \) (randomized) [38] and \( 2.597^k p^{O(1)} \) (deterministic) [39, 21, 35]. In addition, \( k \)-Path is known not to admit any polynomial kernel unless \( \text{NP} \subseteq \text{coNP/poly} \) [12].

2 Preliminaries

In the rest of this short version of the paper, we present our contribution in more detail. Due to the space limit, we omit most of the proofs; all the proofs and further details are given in the full version of the paper [25]. 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 \( 1 \)-colored if each of its vertices is assigned a color from \( \{1, \ldots, \ell\} \). For an undirected (directed, resp.) multigraph \( G \), a walk \( W \) is an alternating sequence \( v_1 e_1 v_2 \ldots e_{\ell-1} v_\ell \) 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 \( i \in \{1, 2, \ldots, \ell - 1\} \). For any \( i \in \{1, 2, \ldots, \ell\} \), \( v_i \) is called a vertex occurrence or a vertex visit, and for all \( i \in \{1, 2, \ldots, \ell - 1\} \), \( \{v_{i-1}, v_i\} \) (resp. \( \{v_i, v_{i+1}\} \)) 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 \( t \).

Given a directed or undirected multigraph \( G \) and vertices \( s,t \in V(G) \), a walk \( W \) in \( G \) is called an Euler \((s,t)\)-trail if \( W \) visits every edge/arc in \( G \) exactly once, and starts at \( s \) and ends at \( t \). An undirected multigraph \( G \) is called even if for every \( v \in V(G) \), \( d(v) \) is even.

The Feasibility Linear Programming problem (Feasibility ILP) is given by a set \( X \) of variables and a system of linear equations and inequalities with real-valued coefficients and variables from \( X \), and the aim is to decide whether all the linear equations and inequalities (called linear constraints) can be satisfied by an assignment \( \alpha \) of non-negative reals to variables in \( X \). If only integral values are allowed in \( \alpha \), then the problem is called the Feasibility Integer Linear Programming problem (Feasibility ILP).

The following well-known result (cf. Section 6.2 in [15]) will be used in this paper.

**Theorem 2.3.** [29, 26, 23] Feasibility ILP of size \( L \) with \( p \) variables can be solved using \( O(p^{2p+\alpha(p)}L) \) arithmetic operations and space polynomial in \( L \).

We use standard terminology and notation in parameterized algorithmics and graph theory, see, e.g., [6, 7, 15, 17, 18].

Given a multiset \( M \) and an element \( e \in M \), \([e]i\) stands for \( i \) copies of \( e \). The size of a multiset \( M = \{[i_1]e_1, \ldots, [i_p]e_p\} \) is \( \sum_{j=1}^p i_j \).

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

To this end, we consider the Directed \( r \)-Simple \( 2\)-Path problem, where given a digraph \( G \), positive integers \( k, r \), and vertices \( s, t \in V(G) \), 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 first 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. Then, we further observe that the input digraph can be assumed not to contain “long” paths and cycles.

**Lemma 3.1.** Suppose that Directed \( r \)-Simple \( Long \) \((s,t)\)-Path on strongly connected digraphs can be solved in time \( f(k/r) \cdot (n + \log k)^{O(1)} \) and polynomial space. Then, Directed \( r \)-Simple \( k \)-Path can be solved in time \( f(k/r) \cdot (n + \log k)^{O(1)} \) 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.** [22, 40] There exists a deterministic algorithm that given a digraph \( G \), vertices \( s, t \in V(G) \), and \( k \in \mathbb{N} \), determines in time \( 2^{O(k)} \cdot n^{O(1)} \) 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 \( 2k/r \) 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.

### 3.1 Bounding the Number of Distinct Arcs

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

The following well-known theorem is often used in proofs of lemmas.

Theorem 3.3. [7, 17] Let $G$ be a directed multigraph whose underlying undirected graph is connected. Let $s,t \in V(G)$. If $s \neq t$, then there exists an Euler $(s,t)$-trail in $G$ if and only if $d^+(s) = d^-(s) + 1$ and $d^+(t) = d^-(t) + 1$, and the out-degree and in-degree of any other vertex in $G$ are equal. If $s = t$, then there exists an Euler $(s,t)$-trail in $G$ if and only if the out-degree and in-degree of every vertex in $G$ are equal.

Our argument modifies a given walk in a manner that may 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 $Q_{\text{simple}}$ is a subgraph of $P_{\text{simple}}$ that is not equal to $P_{\text{simple}}$.

Proof. First, observe that since $G$ has no path of size at least $2k/r$, it holds that $P_{\text{simple}}$ contains at least one cycle. We choose such a cycle $C$ arbitrarily. In what follows, we use the cycle $C$ to modify the walk $P$ in order to obtain a walk $Q$ that has the desired property. To this end, let $\Delta$ be the minimum number of times an arc of $C$ occurs in $P$. Let $H$ be the directed multigraph obtained from $P_{\text{multi}}$ by removing $\Delta$ copies of every arc in $C$ as well as isolated vertices. In addition, let $Q$ be the set of maximal components of $H$ whose underlying undirected multigraphs are connected in the underlying undirected multigraph of $H$. Let $u$ and $v$ denote the first and last (not necessarily distinct) vertices visited by $P$. We consider two subcases depending on $|Q|$.

In the first case, suppose that $|Q| = 1$. Then, the underlying undirected graph of $H$ is connected. Since $P_{\text{multi}}$ has a $(u,v)$-path that visits every arc (that is the path $P$), by Theorem 3.3, every vertex in $P_{\text{multi}}$ has out-degree equal to its in-degree, except for $u$ and $v$ in case $u \neq v$—then, $d^+(u) = d^-(u) + 1$ and $d^+(v) = d^-(v) + 1$. By the definition of $H$, every vertex in $V(G)$ has either both its out-degree and in-degree in $H$ equal to those in $P_{\text{multi}}$ or both its out-degree and in-degree in $H$ smaller by $\Delta$ compared to those in $P_{\text{multi}}$. Thus, every vertex in $H$ has out-degree equal to its in-degree, except for $u$ and $v$ in case $u \neq v$—then, $d^+(u) = d^-(u) + 1$ and $d^+(v) = d^-(v) + 1$. Thus, by Theorem 3.3, $H$ has an Euler trail $Q$. Moreover, since $(G,k,r,s,t)$ is nice, $|A(C)| < k/r$ and therefore $|A(P')| > |A(P)| - \Delta(k/r) \geq k$. Lastly, since the out- and in-degrees of at least one vertex of $C$ was reduced from $\Delta$ in $P_{\text{multi}}$ to 0 in $H$, it holds that $|A(P_{\text{simple}})| > |A(Q_{\text{simple}})|$. Thus, $Q$ is an $r$-simple $k''$-path, for some integer $k'' \geq k$, such that $Q_{\text{simple}}$ is a subgraph of $P_{\text{simple}}$ that is not equal to $P_{\text{simple}}$.

In the second case, suppose that $|Q| \geq 2$. Let $Q_{\text{min}}$ be a component in $Q$ that has minimum number of arcs. Then, $|A(Q_{\text{min}})| < |A(P)|/2$. Let $H^*$ be the directed multigraph obtained from $P_{\text{multi}}$ by removing all the arcs in $Q_{\text{min}}$ as well as isolated vertices. Since $P_{\text{multi}}$ has a $(u,v)$-path that visits every arc (that is the path $P$), by Theorem 3.3, every vertex in $P_{\text{multi}}$ has out-degree equal to its in-degree, except for $u$ and $v$ in case $u \neq v$—then, $d^+(u) = d^-(u) + 1$ and $d^+(v) = d^-(v) + 1$. As in the previous case, every vertex in $V(G)$ has either both its out-degree and in-degree in $H$ equal to those in $P_{\text{multi}}$ or both its out-degree and in-degree in $H$ smaller by $\Delta$ compared to those in $P_{\text{multi}}$. If $u \neq v$, this means that either both $u,v \in V(H^*)$ or both $u,v \notin V(H^*)$. Indeed, as in any directed multigraph, in $Q_{\text{min}}$, the sum of in-degrees of all vertices equals the sum of out-degrees of all vertices. Thus, if $u \in V(Q_{\text{min}}) \setminus V(H^*)$ then $v \in V(Q_{\text{min}}) \setminus V(H^*)$ as well.

However, this means that every vertex in $H^*$ has out-degree equal to its in-degree, except for $u$ and $v$ in case both $u,v \in V(H^*)$ and $u \neq v$—then, $d^+(u) = d^-(u) + 1$ and $d^+(v) = d^-(v) + 1$. Moreover, the underlying undirected graph of $H^*$ is connected (because $H^*$ consists of a collection of components in $Q$ together with the arcs in $C$ that connect their underlying undirected graphs). Thus, by Theorem 3.3, $H^*$ has an Euler trail $Q$. Moreover, $|A(Q)| > \frac{1}{2}|A(P)| \geq k$. In addition, $|A(P_{\text{simple}})| > |A(Q_{\text{simple}})|$ since there is at least one vertex of $Q_{\text{min}}$ that is present in $P_{\text{multi}}$ but not in $H^*$. Thus, $Q$ is an $r$-simple $k''$-path, for some integer $k'' \geq k$, such that $Q_{\text{simple}}$ is a subgraph of $P_{\text{simple}}$ that is not equal to $P_{\text{simple}}$.

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 \(Q_1\) for some integer \(k'' \in \{k, k+1, \ldots, 2k\}\), such that \(Q_1\) is a subgraph of \(P\) that is not equal to \(P\).

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 \(P_{\text{simple}}^{r}\) 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 3.4.** 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. \(P_{\text{simple}}^r\) is an acyclic digraph; 2. For any (not necessarily distinct) \(u, v \in V(P)\), \(P_{\text{simple}}^{\text{u,v}, r}\) has at most one \((u, v)\)-path; 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.

**Proof.** We define a collection of walks \(\mathcal{P}\) as follows: if \((G, k, r, s, t)\) is a positive instance, then \(\mathcal{P}\) is the set of all \(r\)-simple \(k''\)-paths in \(G\) where \(k'' \in \{k, k+1, \ldots, 2k\}\); otherwise, \(\mathcal{P}\) is the set of all \(r\)-simple \((s, t)\)-paths in \(G\) of maximum size. In both cases, \(\mathcal{P} \neq \emptyset\). For any \(\ell \in \mathbb{N}\) and \(\ell\)-simple path \(P\) of size \(\ell\), let \(V(P, r) \setminus \{s, t\}\) contains at most \(\lfloor \ell/r \rfloor\) vertices. Therefore, in the first case, since \(k'' \geq 2k\), every walk in \(\mathcal{P}\) satisfies Property 3. In the second case, every walk \(P \in \mathcal{P}\) contains less than \(k\) vertices (since the instance is negative), therefore \(P\) satisfies Property 3. Thus, it suffices to show that there exists a walk in \(\mathcal{P}\) that satisfies Properties 1 and 2.

Let \(P'\) be the set of walks \(P \in \mathcal{P}\) with minimum number of arcs in \(\mathcal{P}_{\text{simple}}^r\). Moreover, let \(P''\) be the set of walks \(P \in P'\) that maximize \(|V(P, r)|\).

We claim that every walk in \(P''\) satisfies Properties 1 and 2. For this purpose, we consider an arbitrary walk \(P \in P'\). Let \(u\) and \(v\) denote the first and last (not necessarily distinct) vertices visited by \(P\). (If \((G, k, r, s, t)\) is a negative instance, then \(u = s = t\).) Suppose, by way of contradiction, that \(P\) does not satisfy Property 1. Then, \(P_{\text{simple}}^r\) has a directed cycle \(C\). Let \(\Delta\) be the maximum out-degree in \(P_{\text{multi}}\) of a vertex in \(C\). Note that \(\Delta < r\) because \(V(C) \cap V(P, r) = \emptyset\). Let \(H\) be the directed multigraph obtained from \(P_{\text{multi}}\) by adding \(r - \Delta\) copies of every arc in \(C\). Since \(P_{\text{multi}}\) has a \((u, v)\)-path that visits every arc (that is the path \(P\)), by Theorem 3.3, every vertex in \(P_{\text{multi}}\) has out-degree equal to its in-degree, except for \(u\) and \(v\). In case \(u \neq v\), then, \(d^u(u) = d^v(u) + 1\) and \(d^v(v) = d^u(v) + 1\). By our construction of \(H\), it has the same property. Indeed, every vertex in \(V(G)\) has either both its out-degree and in-degree in \(H\) equal to those in \(P_{\text{multi}}\) or both its out-degree and in-degree in \(H\) larger by \(r - \Delta\) compared to those in \(P_{\text{multi}}\). Thus, by Theorem 3.3, \(H\) has an Euler trail \(P\) with the same endpoints as \(P\). Let us consider two cases, depending on the size of \(P''\).

1. First, suppose that \(P''\) is of size at most \(2k'\). Then, \(P'' \in \mathcal{P}\), and since \(P''_{\text{simple}} = P_{\text{simple}}^r\), it further holds that \(P'' \in P'\). However, \(|V(P'', r)| > |V(P, r)|\) because at least one vertex of \(C\) belongs to \(V(P', r)\) but not to \(V(P, r)\) and clearly \(V(P, r) \subseteq V(P'', r)\). Thus, we have a contradiction to the inclusion \(P \in P''\).

2. Second, suppose that \(P''\) is of size larger than \(2k\). We stress that in this case, \((G, k, r, s, t)\) is positive. By Corollary 3.1, \(G\) has an \(r\)-simple \(k''\)-path \(Q_1\) for some integer \(k'' \in \{k, k+1, \ldots, 2k\}\), such that \(Q_1\) is a simple subgraph of \(P''_{\text{simple}}\) that is not equal to \(P_{\text{simple}}^r\). Then, \(Q \in \mathcal{P}\) because, in this case, to be included in \(\mathcal{P}\), a walk does not need to have the same start and end vertices as \(P\). Since \(P_{\text{simple}}^r = P_{\text{simple}}^r\), we have that \(|A(Q_{\text{simple}})| < |A(P_{\text{simple}}^r)| = |A(P_{\text{simple}})|\), which is a contradiction to the inclusion \(P \in P''\).

It remains to argue that \(P\) satisfies Property 2. Suppose, by way of contradiction, that this claim is false. Then, for some vertices \(x, y \in V(P)\), it holds that \(P_{\text{simple}}^{x,y} = P_{\text{simple}}^{x,y}\) has at least two pairwise internally vertex disjoint \((x, y)\)-paths. Denote two such different vertex disjoint paths (chosen arbitrarily) by \(P_{\text{simple}}^{x,y} \) and \(P_{\text{simple}}^{x,y}\) such that \(|A(P_{\text{simple}}^{x,y})| \geq |A(P_{\text{simple}}^{x,y})|\). Note that \(V(P_{\text{simple}}^{x,y}) \setminus \{x, y\} = V(P_{\text{simple}}^{x,y}) \setminus V(P_{\text{simple}}^{x,y}) \neq \emptyset\) and \(A(P_{\text{simple}}^{x,y}) \cap A(P_{\text{simple}}^{x,y}) = \emptyset\). (Note that \(V(P_{\text{simple}}^{x,y}) \setminus V(P_{\text{simple}}^{x,y})\) can be empty since \(P_{\text{simple}}^{x,y}\) can consist of a single arc.) Let \(\Delta_1\) denote the maximum out-degree in \(P_{\text{multi}}\) of a vertex in \(V(P_{\text{simple}}^{x,y}) \setminus \{x, y\}\). In addition, let \(\Delta_2\) denote the minimum number of times an arc of \(A(P_{\text{simple}}^{x,y})\) occurs in \(P\). Now, denote \(\Delta = \min\{r - \Delta_1, \Delta_2\}\). Let \(H\) be the directed multigraph obtained from \(P_{\text{multi}}\) by adding \(\Delta\) copies of every arc of \(P_{\text{simple}}^{x,y}\), and removing \(\Delta\) copies of every...
arc of \( P_2^{x\to y} \) as well as isolated vertices. In addition, let \( Q \) be the set of maximal components of \( \hat{H} \) whose underlying undirected multigraphs are connected in the underlying undirected multigraph of \( H \). We consider two subcases depending on the size of \(|Q|\).

1. First, suppose that \(|Q| = 1\). Then, the underlying undirected graph of \( H \) is connected. Since \( P_{\text{multi}} \) has a \((u,v)\)-path that visits every arc (that is the path \( P \)), by Theorem 3.3, every vertex in \( P_{\text{multi}} \) has out-degree equal to its in-degree, except for \( u \) and \( v \) in case \( u \neq v \)—then, \( d^+(u) = d^-(u) + 1 \) and \( d^-(v) = d^+(v) + 1 \). By the definition of \( H \), every vertex in \( V(G) \) has \( (i) \) both its out-degree and in-degree in \( H \) equal to those in \( P_{\text{multi}} \), or \( (ii) \) both its out-degree and in-degree in \( H \) larger by \( \Delta \) compared to those in \( P_{\text{multi}} \), or \( (iii) \) both its out-degree and in-degree in \( H \) smaller by \( \Delta \) compared to those in \( P_{\text{multi}} \). Thus, every vertex in \( H \) has out-degree equal to its in-degree, except for \( u \) and \( v \) in case \( u \neq v \)—then, \( d^+(u) = d^-(u) + 1 \) and \( d^-(v) = d^+(v) + 1 \). Thus, by Theorem 3.3, \( H \) has an Euler trail \( P' \). Moreover, since \( |A(P'_1)| \geq |A(P'_2)| \), we have that \( |A(P'_1)| \geq |A(P')| \). In addition, \( P'_{\text{simple}} \) is a subgraph of \( P_{\text{simple}} \). We consider three subcases depending on \( \Delta \) and the size of \( P' \).

(a) Suppose that \( \Delta = r - \Delta_1 > \Delta_2 \) and the size of \( P' \) is at most \( 2k \). Then, \( P' \subseteq P'_{\text{multi}} \) and at least one vertex in \( V(P'_1) \setminus V(P'_2) \) has out-degree \( r \) in \( P'_{\text{multi}} \) but not in \( P_{\text{multi}} \). While clearly \( V(P',r) \subseteq V(P',r) \). However, this is a contradiction to the inclusion in \( P'_{\text{multi}} \).

(b) Suppose that \( \Delta = \Delta_2 \) and the size of \( P' \) is at most \( 2k \). Then, \( P' \subseteq P'_{\text{multi}} \) but \( P'_{\text{simple}} \) is not equal to \( P_{\text{simple}} \) (at least one arc of \( P'_2 \) is present in \( P_{\text{simple}} \) but not in \( P'_{\text{simple}} \)). However, this is a contradiction to the inclusion in \( P' \).

(c) Suppose that the size of \( P' \) is larger than \( 2k \). Then, by Corollary 3.1, \( G \) has an \( r \)-simple \( k'' \)-path \( Q \), for some integer \( k'' \in \{k,k+1,\ldots,2k\} \), such that \( Q_{\text{simple}} \) is a subgraph of \( P_{\text{simple}} \) that is not equal to \( P'_{\text{simple}} \). However, this is a contradiction to the inclusion in \( P' \).

2. Now, suppose that \(|Q| \geq 2\). Then, exactly like in the second case in the proof of Lemma 3.3, we derive that \( G \) has an \( r \)-simple \( k'' \)-path \( P' \), for some integer \( k'' \geq k \), such that \( P'_{\text{simple}} \) is a subgraph of \( P_{\text{simple}} \) that is not equal to \( P'_{\text{simple}} \). By Corollary 3.1, this means that \( G \) has an \( r \)-simple \( k'' \)-path \( Q \), for some integer \( k'' \in \{k,k+1,\ldots,2k\} \), such that \( Q_{\text{simple}} \) is a subgraph of \( P'_{\text{simple}} \) that is not equal to \( P'_{\text{simple}} \). However, this is a contradiction to the inclusion \( P \subseteq P' \).

Since both cases led to a contradiction, the proof is complete. \( \square \)

Having Lemma 3.4 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.

**Proof.** Let \( r \in \mathbb{N}_{\geq 2} \). Consider a digraph \( G \) with a vertex \( u \) and \( r \) cycles \( C_i = v_i^1 \ldots v_i^r \) \((i = 1, \ldots, r)\) sharing pairwise only vertex \( u \). For every \( i = 1, \ldots, r \) add to \( G \) a \( 2 \)-cycle \( w^i v_i^1 w^i \), where \( w^1, \ldots, w^r \) are new vertices in \( G \). Let \( P \) be an \( r \)-simple \( k \)-path of \( G \) of maximum size. Observe that \( P \) cannot traverse any \( C_i \) twice (i.e., it cannot visit vertices of any \( C_i \) twice apart from \( v_i^1 \)) along with visiting \( w^r \) \(-r \) times since \( P \) will have more vertex visits if it traverses two cycles \( C_i \) and \( C_j \) instead along with visiting \( w^i \) and \( w^j \) \(-r \) times each. Thus, \( P \) visits \( r \) times \( u \) and each \( v_i^1 \). It visits \( r \) times each \( w^i \) and only once every vertex of \( C_i \) apart from \( v_i^1 \) for all \( i = 1, \ldots, r \). Hence, \( k = r(r+1) + r(r-1) + r(r-1) = \Theta(r^2) \) and \( k/r = \Theta(r) \). Note that \( P \) is an open walk which visits every arc of \( G \) but one. Thus, \( P \) has \( |A(G)| - 1 = r(r+1) + 2r = \Theta((k/r)^2) \) distinct arcs. \( \square \)

### 3.2 Searching for a Solution
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. [5]. For the sake of brevity, we define the following problem. Here, \( b(k/r) = \frac{30(k/r)^2 + 1}{r} \). In the DIRECTED COLORFUL r-SIMPLE Long \((s,t)\)-Path problem, we are given integers \( k,r \in \mathbb{N} \), a \( b(k/r)\)-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 \((s, t)\)-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 Multi-colored Clique and hence lead to a dead-end. Now, we can see that we can focus on our colored variant Directed Colorful \((s, t)\)-Path.

**Lemma 3.9.** Suppose that Directed Colorful \((s, t)\)-Path can be solved in time \( g(k/r) \cdot (n + \log k)^{O(1)} \) and polynomial space. Then, Directed \((s, t)\)-Path on strongly connected digraphs can be solved in time \( 2^{O((k^2)} \cdot g(k/r) \cdot (n + \log k)^{O(1)} \) 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 \( \ell \)-topology is an \( \ell \)-colored digraph with at most \( \ell \) arcs and without isolated vertices such that each of its vertices has a distinct color. Let \( \mathcal{T}_\ell \) denote the set of all \( \ell \)-topologies.

There are not too many topologies.

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

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

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

In light of Observation 3.1, a natural approach to solve Directed Colorful \((s, t)\)-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 Multi-colored 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 \( O(k) \) (while the input size is only \( O(n + \log k) \)), 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 \( \ell \)-topology with endpoints \( i, j \) is a pair \((T, \varphi)\) of an \( \ell \)-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 \( \sum_{u : \{u, v\} \in A(T)} \varphi(u, v) = \sum_{u : \{v, u\} \in A(T)} \varphi(v, u) \leq r \); 3. \( \sum_{u : \{u, s\} \in A(T)} \varphi(u, s) + 1 = \sum_{u : \{s, u\} \in A(T)} \varphi(s, u) \leq r \); 4. \( \sum_{u : \{u, t\} \in A(T)} \varphi(u, t) = \sum_{u : \{t, u\} \in A(T)} \varphi(t, u) + 1 \leq r \).

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 [29, 26, 23].

**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 \( \ell \)-topology, determines in time \( O(\ell, \log r)^{O(1)} \) and polynomial space whether there exists a function \( \varphi : A(T) \to \{1, 2, \ldots, r\} \) such that \((T, \varphi)\) is an \( r \)-enriched \( \ell \)-topology with endpoints \( i, j \). In case the answer is positive, the algorithm outputs such a function \( \varphi \) 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 \( \ell \)-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 $(T, \varphi)$ be an $r$-enriched $\ell$-topology with endpoints $i,j$. We say that $P$ complies with $(T, \varphi)$ if $P$ complies with $T$, and for the function $\phi$ that witnesses this, for every arc $(u,v) \in P_{\text{simple}}$, the number of copies $(u,v)$ has in $P_{\text{multi}}$ 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 $\ell$-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 $\ell$-topology with endpoints $i,j$, with which $P$ complies. 2. Let $T$ be an $\ell$-topology with which $P$ complies. Then, for any $r$-enriched $\ell$-topology with endpoints $i,j$, say $(T, \varphi)$, there exists an $r$-simple $(s,t)$-path in $G$ that complies with $(T, \varphi)$.

This lemma motivates a problem definition where the input includes an $r$-enriched $\ell$-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 $(T, \varphi)$ be an $r$-enriched $\ell$-topology $(T, \varphi)$ with endpoints $i,j$. Let $P$ be an $r$-simple $(s,t)$-path in an $\ell$-colored digraph $G$, where $i$ is the color of $s$ and $j$ is the color of $t$. Then, $P$ weakly complies with $(T, \varphi)$ 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 $\phi : V(P_{\text{simple}}) \rightarrow V(T)$ 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 $(T, \varphi)$, then it also weakly complies with $(T, \varphi)$, but the opposite is not true. In particular, a walk where some distinct vertices have the same color can weakly comply with $(T, \varphi)$, but it necessarily does not comply with $(T, \varphi)$. In the $(\ell,r)$-ENRICHED TOPOLOGY problem, the input consists of an $\ell$-colored digraph $G$, integers $\ell,r \in \mathbb{N}$, distinct vertices $s,t \in V(G)$, and an $r$-enriched $\ell$-topology $(T, \varphi)$ with endpoints $i,j$ where $i$ is the color of $s$ and $j$ is the color of $t$. The objective is to return $\text{Yes}$ or $\text{No}$ as follows. (i) If $G$ has an $r$-simple $(s,t)$-path that complies with $(T, \varphi)$, then return $\text{Yes}$. (ii) If $G$ has no $r$-simple $(s,t)$-path that weakly complies with $(T, \varphi)$, then return $\text{No}$. (iii) If none of the two conditions above holds, we can return either $\text{Yes}$ or $\text{No}$.

The $(\ell,r)$-ENRICHED TOPOLOGY problem allows us to determine whether there exists an $r$-simple $(s,t)$-path in $G$ that weakly complies with $(T, \varphi)$.

**Lemma 3.13.** Suppose that $(\ell,r)$-ENRICHED TOPOLOGY can be solved in time $f(\ell)(n + \log r)^{O(1)}$ and polynomial space. Then, DIRECTED COLORFUL $r$-SIMPLE LONG $(s,t)$-PATH can be solved in time $2^{O(b(k/r) \log(b(k/r)))} \cdot f(b(k/r)) \cdot (n + \log k)^{O(1)}$ and polynomial space.

It remains to solve the $(\ell,r)$-ENRICHED TOPOLOGY problem. If we allowed a linear dependency on $k$ in the running time, then this task would have been easier than our actual task, since we could have used the following approach: first, we would have computed some walk $P^*$ that uses every arc $a$ in the input enriched topology exactly $\varphi(e)$ times—note that the size of such $P^*$ can be $\Omega(\ell r)$ (that is, $\Omega(k)$); then, we could have used a simple dynamic programming (DP) computation to check whether the input digraph $G$ contains such a colored walk (where vertices having the same color in $P^*$ are allowed to be mapped to distinct vertices in $G$ as long as these vertices have the same color).

Here, we describe how to attain a logarithmic dependency on $k$. For this purpose, we present a recursive algorithm (combined with DP) that solves $(\ell,r)$-ENRICHED TOPOLOGY. Due to the nature of the recursion, we need to consider an annotated version of $(\ell,r)$-ENRICHED TOPOLOGY, defined as follows. In the ANNOTATED $(\ell,r)$-ENRICHED TOPOLOGY problem, we are given integers $\ell,r \in \mathbb{N}$, an $\ell$-colored digraph $G$, distinct vertices $s,t \in V(G)$, an $r$-enriched $\ell$-topology $(T, \varphi)$ with endpoints $i,j$ where $i$ is the color of $s$ and $j$ is the color of $t$, and a subset $A \subseteq V(G)$. The objective is to return $\text{Yes}$ or $\text{No}$ as follows.

- If $G$ has an $r$-simple $(s,t)$-path that complies with $(T, \varphi)$ and visits every vertex in $A$ at least once, then return $\text{Yes}$.
- If $G$ has no $r$-simple $(s,t)$-path that weakly complies with $(T, \varphi)$ and visits every vertex in $A$ at least once, then return $\text{No}$.
- If none of the two conditions above holds, we can return either $\text{Yes}$ or $\text{No}$.

We define the basis as the case where the topology is a DAG. Then, we make use of the following lemma.

**Lemma 3.14.** There exists an algorithm that, given
an instance \( I = (G, \ell, r, s, t, T, \varphi, A) \) of **Annotated \((\ell, r)\)**-Enriched Topology where \( T \) is a DAG, solves \( I \) in polynomial time and space.

In each step, we “handle” a (directed simple) cycle from the current topology so that at least one of its arcs is eliminated—here, it is crucial to eventually derive a logarithmic dependency on \( k \) that we completely eliminate an arc and not only decrease the value that \( \varphi \) assigns to it. We summarize the result as follows.

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

Finally, we are ready to prove Theorem 3.1.

**Proof of Theorem 3.1.** By Lemma 3.16, \((\ell, r)\)-Enriched Topology can be solved in time and space \((\ell + n + \log r)O(1)\). Thus, by Lemma 3.13, Directed Colorful \( r\)-Simple Long \((s, t)\)-Path can be solved in time \(2^{O(b/r)}(n + \log k)O(1)\) and polynomial space. Substituting \( b(k/r) \), this running time is upper bounded by \(2^{O((k/r)\log(k/r))}(n + \log k)O(1)\). 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 \(2^{O((k/r)^2\log(k/r))}(n + \log k)O(1)\) and polynomial space. Finally, by Lemma 3.1, we conclude that Directed \( r\)-Simple \( k\)-Path can be solved in time \(2^{O((k/r)^2\log(k/r))}(n + \log k)O(1)\) 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 \(2^{O(\frac{\log k}{k})}(n + \log k)O(1)\).

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 \(2^{O(\frac{\log k}{k})}(r + n + \log k)O(1)\).

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 \(2^{O(\frac{\log k}{k})}(n + \log k)O(1)\).

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

Let us outline the proof of Lemma 4.1. Say that an instance \((G, k, r)\) of Undirected \( r\)-Simple \( k\)-Path is nice if \( G \) has no path of length at least \( k/r \). Similarly to the directed case, the existence of such a path implies that \((G, k, r)\) is a Yes-instance and can be tested for in single-exponential time; hence we focus on nice instances.

Using ideas from the directed case adapted to the simpler structure of undirected graphs, 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 \(2^{O(\frac{\log k}{k})}(n + \log k)O(1)\) and polynomial space. However, in order to obtain a single-exponential running time bound of \(2^{O(\frac{\log k}{k})}(n + \log k)O(1)\), 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 \( P_{\text{uni}} \) can be partitioned into two multisets, \( M_1 \) and \( M_2 \), with the following properties: (i) \( P_{\text{uni}} \) restricted to \( M_1 \) is a (simple) spanning tree of \( P_{\text{uni}} \); and (ii) \( P_{\text{uni}} \) restricted to \( M_2 \) has no even cycle of length at least 4.

Proof. Let \( e_1, e_2, \ldots, e_m \) be some ordering of the edges in \( E(G) \). For any walk \( W \), define \((x^W_1, x^W_2, \ldots, x^W_m)\) be the vector where \( x_i \) is equal to the number of times \( e_i \) occurs in \( W \) for all \( i \in \{1, 2, \ldots, m\} \). By Lemma 4.5, \( G \) has an \( r\)-simple \( k\)-path with fewer than \( 30(k/r) \) distinct edges. Among all such \( r\)-simple \( k\)-paths, let \( P \) be one where \((x^P_1, x^P_2, \ldots, x^P_m)\) is lexicographically smallest. Let \( T \) be an arbitrary spanning tree of \( P_{\text{uni}} \), and denote \( M_1 = E(T) \). In addition, denote \( M_2 = E(P_{\text{uni}}) \setminus M_1 \). (Note that \( M_2 \) is a multiset: if an edge \( e \) has \( x \) copies in \( P_{\text{uni}} \), then it has either \( x \) or \( x-1 \) copies in \( M_2 \).) Let \( H \) denote the restriction of \( P_{\text{uni}} \) to \( M_2 \).

We claim that \( H \) has no even cycle of length at least 4. To prove this, suppose by way of contradiction that \( H \) does have some even cycle \( C \) of length at least 4. Let \( C = v_1 - v_2 - v_3 - \cdots - v_q - v_1 \) such that \( e = \{v_1, v_2\} \) is the leftmost edge among the edges in \( E(C) \) according to our predefined ordering of \( E(G) \). Note that \( q \geq 4 \)
is even. In addition, denote $U = \{ \{v_i, v_{i+1}\} : i \in \{1, 2, \ldots, q - 1\}, i \text{ is odd} \}$. Now, define $H'$ as the graph obtained from $H$ by removing one copy of each edge in $U$ and adding one copy of each edge in $E(C) \setminus U$. Then, every vertex has the same degree in $H'$ and in $H$. Let $\tilde{H}$ denote the multigraph obtained by adding one copy of each edge in $M_1$ into $H'$. Then, every vertex has the same degree in $\tilde{H}$ and in $P_{\text{multi}}$. Moreover, $M_1 \subseteq E(\tilde{H})$ means that $H$ has a spanning tree and hence it is connected. Since $P_{\text{multi}}$ has an Euler $(s, t)$-trail for some vertices $s, t \in V(G)$ (this trail is simply $P$), by the undirected version of Theorem 3.3, $\tilde{H}$ also has an Euler $(s, t)$-trail, say $Q$. Then, $Q$ is an $r$-simple $k$-path with the same (or fewer) number of distinct edges as $P$. From our choice of $\{v_1, v_2\}$, it follows that $(x_1^Q, x_2^Q, \ldots, x_{m}^Q)$ is lexicographically smaller than $(x_1^P, x_2^P, \ldots, x_{m}^P)$. However, this contradicts our choice of $P$. □

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

Proposition 4.1. (folklore, see [31, 37]) 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 $P_{\text{multi}}$ can be partitioned into two multisets, $M_1$ and $M_2$, with the following properties: (i) $P_{\text{multi}}$ restricted to $M_1$ is a (simple) spanning tree of $P_{\text{multi}}$, and (ii) $P_{\text{multi}}$ 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. In particular, if some vertices of the first part will have odd degrees, our algorithm will not be able to ensure that each of these vertices will be reused an odd number of times (or even used at all) in the second part.

At this point, the reader may wonder whether we can bound the treewidth of the entire solution (for at least one solution) by a constant. However, it can be proven that for some instances, all solutions correspond to graphs with very high treewidth (in particular, of treewidth that cannot be bounded by a fixed constant).

Lemma 4.7. Let $r \geq 5$. For any constant $c \in \mathbb{N}$, there exists a nice Yes-instance $(G, k, r)$ of Undirected $r$-Simple $k$-Path such that every $r$-simple $k$-path $P$ in $G$ satisfies the following property: the treewidth of $P_{\text{simple}}$ is larger than $c$.

It is not hard to derive the new partition (Lemma 4.9) from Corollary 4.1 and the following simple lemma.

Lemma 4.8. Let $G$ be a multigraph which has an Euler $(s, t)$-trail for some vertices $s, t \in V(G)$. Then, $G$ has a subgraph $H$ with the following properties: 1. Every distinct edge in $G$ occurs at least once in $H$; 2. $H$ has an Euler $(s, t)$-trail; 3. $H$ has only at most $2d$ edges (including multiplicities), where $d$ is the number of distinct edges in $G$.

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 $P_{\text{multi}}$ can be partitioned into two multisets, $M_1$ and $M_2$, with the following properties: (i) $P_{\text{multi}}$ restricted to $M_1$ is a spanning multigraph of $P_{\text{multi}}$ with fewer than $60(k/r)$ edges (including multiplicities) that has an Euler $(s, t)$-trail where $s$ and $t$ are the end-vertices of $P$; (ii) $P_{\text{multi}}$ restricted to $M_2$ is a multigraph of treewidth 2.

4.1 Searching for a Solution

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, $b(k/r) = 30k/r + 1$.

In the Undirected Colorful $r$-Simple $k$-Path problem, we are given integers $k, r \in \mathbb{N}$, and a $b(k/r)$-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 $f(k/r) \cdot (r + n + \log k)^{O(1)}$. Then, Undirected $r$-Simple $k$-Path can be solved in time $2^{O(k/r)} \cdot f(k/r) \cdot (r + n + \log k)^{O(1)}$.

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 $2^{\mathcal{O}(\log^2 k)}$. Instead, inspired by the work of Berger et al. [8] (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 \( \mathbf{a} = (d_1, \ldots, d_{b(k/r)}) \) that satisfies the following conditions: For all \( i \in \{1, 2, \ldots, b(k/r)\} \), \( d_i \) is an integer between 0 and \( r \) and \( \sum_{i=1}^{b(k/r)} d_i \leq 2b(k/r) \). Let \( \mathcal{D}_{r,k} \) be the set of all \((r,k)\)-occurrence sequences.

Crucially, the number of occurrence sequences is single-exponential:

**Lemma 4.11.** Let \( r, k \in \mathbb{N} \). Then, \( |\mathcal{D}_{r,k}| = 2^{O(k/r)} \).

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 \( b(k/r) \)-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 \( b(k/r) \)-colored undirected graph, and let \( \mathbf{a} \) be an \((r,k)\)-occurrence sequence. A good pair \((W, H)\) complies with \( \mathbf{a} \) if for every color \( i \in \{1, 2, \ldots, b(k/r)\} \), 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 \( 2(r - d_i) \).

Let us now argue that we can focus on seeking a pair \((W, H)\) as in Definition 4.4.

**Lemma 4.12.** Let \( (G, k, r) \) be an instance of Undirected Colorful \( r \)-Simple \( k \)-Path. If \((G, k, r)\) is a Yes-instance, then there exist \( \mathbf{a} \in \mathcal{D}_{r,k} \) and a good pair that complies with \( \mathbf{a} \). Moreover, if there exist \( \mathbf{a} \in \mathcal{D}_{r,k} \) and a good pair that complies with \( \mathbf{a} \), then \((G, k, r)\) is not a No-instance.

Accordingly, we define the following problem. In the \( \text{(Walk,TW-2) Partition} \) problem, we are given integers \( k, r \in \mathbb{N} \), a \( (k, r) \)-colored undirected graph \( G \), and \( \mathbf{a} \in \mathcal{D}_{r,k} \). The objective is to decide whether there exists a good pair that complies with \( \mathbf{a} \). We can focus on solving the \( \text{(Walk,TW-2) Partition} \) problem due to the following.

**Lemma 4.13.** Suppose that \( \text{(Walk,TW-2) Partition} \) can be solved in time \( f(k/r) \cdot (r + n + \log k)^{O(1)} \). Then, Undirected Colorful \( r \)-Simple \( k \)-Path can be solved in time \( 2^{O(k/r)} \cdot f(k/r) \cdot (r + n + \log k)^{O(1)} \).

We solve \( \text{(Walk,TW-2) Partition} \) by designing a two-level DP. We first give a lemma that handles a single connected component of the treewidth-2 multigraph \( H \) that is a member of the pair we aim to find. The proof is based on a standard DP over a tree decomposition with a slight technicality: we do not know the structure of \( H \) and hence we do not have the tree decomposition over which the DP should be performed. Nevertheless, we can repeatedly “guess” the current top bag and hence imitate a standard DP over a (unknown) tree decomposition.

**Lemma 4.14.** There exists an \( 2^{O(k/r)} \cdot (r + n + \log k)^{O(1)} \)-time algorithm that, given an undirected graph \( G \), a set of colors \( C \subseteq \{1, 2, \ldots, b(k/r)\} \), a vertex \( v^* \in V(G) \) whose color belongs to \( C \), and \( \mathbf{a} = (d_1, \ldots, d_{b(k/r)}) \in \mathcal{D}_{r,k} \), outputs the largest integer \( M \) for which there exists a colorful multigraph \( H \) that satisfies the following conditions.

1. For each \( v \in V(H) \), the degree of \( v \) in \( H \) is even and does not exceed \( 2(r - d_i) \) where \( i \) is the color of \( v \).
2. The underlying simple graph of \( H \) is a connected subgraph of \( G \).
3. The treewidth of \( H \) is at most 2.
4. \( v^* \in V(H) \).
5. The number of edges (including multiplicities) in \( H \) is exactly \( M \).
6. Every vertex in \( H \) is colored by a color from \( C \).

Now, we solve the \( \text{(Walk,TW-2) Partition} \) problem.

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

**Proof.** Let \( A \) denote the algorithm in Lemma 4.14. We now describe a DP procedure to solve \( \text{(Walk,TW-2) Partition} \). To this end, let \( (G, k, r, \mathbf{a}) = (d_1, \ldots, d_{b(k/r)}) \) be an instance of \( \text{(Walk,TW-2) Partition} \). We have a DP table \( \mathbf{N} \) with an entry \( \mathbf{N}[v, \mathbf{d}, C] \) for every vertex \( v \in V(G) \), occurrence sequence \( \mathbf{d} = (d_1, \ldots, d_{b(k/r)}) \) such that \( d_i \in \{0, \ldots, d_i\} \) for every \( i \in \{1, \ldots, b(k/r)\} \), and set of colors \( C \subseteq \{1, \ldots, b(k/r)\} \).

The purpose of each entry \( \mathbf{N}[v, \mathbf{d}, C] \) is to store the largest integer \( M \) such that there exists a good pair \((W, H)\) with \( M = |E(H)| \) which complies with \( \mathbf{d} \), where \( v \) is an end-vertex of \( W \), and where the
set of colors of vertices in $H$ is a subset of $C$. The order of computation is non-decreasing with respect to $\sum_{i=1}^{b} d_i^i$.

In the DP basis, we consider every entry $N[v, d, C]$ that satisfies $\sum_{i=1}^{b} d_i^i \leq 1$, and let $c$ be the color of $v$. If $d_c \neq 1$, then $N[v, d, C] = -\infty$. Else $N[v, d, C]$ is the maximum of 0 and the output of algorithm $A$ when called with input $(G, C, v, d)$. For the DP step, we consider every entry $N[v, d, C]$ that satisfies $\sum_{i=1}^{b} d_i^i \geq 2$. Let $c$ be the color of $v$. If $d_c' = 0$, then $N[v, d, C] = -\infty$. Now, suppose that $d_c' \geq 1$. Denote $\overrightarrow{d} = (d_1', \ldots, d_{b(k/r)}')$ where $d_i' = d_i'' - 1$ and $d_i'' = d_i'$ for all $i \in \{1, \ldots, b(k/r)\} \setminus \{c\}$. In addition, for every subset $C' \subseteq C$, let $A_{C'}$ be the output of algorithm $A$ when called with input $(G, C', v, \overrightarrow{d})$. Then,

$$N[v, d, C] = \max_{u \in V(G)} \{N[u, \overrightarrow{d}', C],$$

$$\max_{C' \subseteq C} (A_{C'} + 1 + N[u, \overrightarrow{d}, C', C']) \}.$$ 

After the DP computation is complete, we return $\text{Yes}$ if and only if there exists an entry $N[v, d, C]$ for some $v \in V(G)$ and $C \subseteq \{1, \ldots, b(k/r)\}$ that stores an integer that is at least $k - 1$.

By Lemma 4.14, the running time of our algorithm is $O((k/r) \cdot (r + n + \log k)^{O(1)})$. Moreover, its correctness can be verified by a simple induction on $\sum_{i=1}^{b} d_i^i$. □

**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 $2^{O(k/r)} \cdot (r + n + \log k)^{O(1)}$. Thus, by Lemma 4.13, Undirected Colorful $r$-Simple $k$-Path can be solved in time $2^{O(k/r)} \cdot (r + n + \log k)^{O(1)}$. In turn, by Lemma 4.10, Undirected $r$-Simple $k$-Path can be solved in time $2^{O(k/r)} \cdot (r + n + \log k)^{O(1)}$, which completes the proof. □

### 4.2 Bounding $r$

In what follows, we focus on the proof of Lemma 4.2. Without loss of generality, we implicitly suppose that given an instance $(G, k, r)$ of Special Undirected $r$-Simple $k$-Path, the graph $G$ is connected, else the problem can be solved by considering each connected component separately.

**Bounding the Vertex Cover Number.**

The reason why the case where $r > \sqrt{k}$ is simpler than the general case lies in the following lemma.

**Lemma 4.16.** Let $(G, k, r)$ be an instance of Special Undirected $r$-Simple $k$-Path. If $G$ has a matching of size $[k/r]$, then $(G, k, r)$ is a $\text{Yes}$-instance.

**Proof.** Suppose that $G$ has a matching $M$ of size $s = \lfloor k/r \rfloor$, and denote $M = \{(u_1, v_1), \ldots, (u_s, v_s)\}$. For every $i \in \{1, 2, \ldots, s - 1\}$, let $P_i$ denote an arbitrary path in $G$ from $u_i$ to $u_{i+1}$ (such a path exists since $G$ is assumed to be connected). Consider the following walk:

$$W = u_1 - v_1 - P_1 - u_2 - v_2 - P_2 - \cdots - u_s - v_s = u_1 - v_1 - P_1 - v_2 - P_2 - \cdots - v_s.$$ 

For every $i \in \{1, 2, \ldots, s - 1\}$, let $\text{occ}_i$ denote the maximum of the number of occurrences of $u_i$ in $W$ and the number of occurrences of $v_i$ in $W$. Note that each vertex occurs at most once in each path $P_j$, $j \in \{1, 2, \ldots, s - 1\}$. In particular, $\text{occ}_i \leq s \leq r$ for all $i \in \{1, 2, \ldots, s\}$. To describe our modification of $W$ we need the following notation: for every $i \in \{1, 2, \ldots, s\}$, let $Q_i$ denote the $(u_i, v_i)$-walk that traverses the edge $\{u_i, v_i\}$ several times such that each vertex among $u_i$ and $v_i$ occurs $Q_i$ exactly $r - \text{occ}_i + 1$ times. Now, we modify $W$ as follows:

$$W' = Q_1 - P_1 - Q_2 - P_2 - \cdots - Qs_{-1} - P_{s-1} - Q_s.$$ 

Then, every vertex occurs at most $r$ times in $W'$. Moreover, for every $i \in \{1, 2, \ldots, s\}$, at least one among the vertices $u_i$ and $v_i$ occurs exactly $r$ times in $W'$. Thus, the size of $W'$ is at least $s \cdot r = \lfloor k/r \rfloor \cdot r \geq k$. Thus, $G$ has an $r$-simple $k$-path. □

Since the set of endpoints of any maximal matching is a vertex cover, and a maximal matching can be computed greedily in polynomial time, we derive the following corollary.

**Corollary 4.2.** There exists a polynomial-time algorithm that, given an instance $(G, k, r)$ of Special Undirected $r$-Simple $k$-Path, either correctly concludes that it is a $\text{Yes}$-instance or output a vertex cover of $G$ of size at most $2\lceil k/r \rceil \leq 3k/r$. □

**Color Coding and Vertex Guessing.** We define the following problem. In the Special Undirected Colorful $r$-Simple $k$-Path problem, we are given integers $k, r \in \mathbb{N}$, a $b(k/r)$-colored undirected graph $G$, and a vertex cover $U$ of $G$ of size at most $3k/r$ where each vertex in $U$ has a unique color. The objective is to output $\text{No}$ if $G$ has no $r$-simple $k$-path (in this case, the input is called a $\text{No}$-instance), and $\text{Yes}$ if it has a colorful $r$-simple $k$-path (in this case, the input is called a $\text{Yes}$-instance) that visits every vertex in $U$ and which has fewer than $30(k/r)$ distinct edges. If the input is neither a $\text{Yes}$-instance nor a $\text{No}$-instance, the output can be arbitrary. We have the following result.

**Lemma 4.17.** Suppose that Special Undirected Colorful $r$-Simple $k$-Path can be solved in time
The presence of a small vertex cover gives rise to the definition of a problem simpler than (Walk,TW-2) Partition, which we will be able to solve while having a poly-logarithmic (rather than polynomial) dependency on $r$. To this end, we need a new definition.

Definition 4.5. Let $r, k \in \mathbb{N}$. Let $G$ be a $b(k/r)$-colored undirected graph, and let $U$ be a vertex cover of $G$. In addition, let $\overline{d} = (d_1, d_2, \ldots, d b(k/r)) \in D_{r,k}$. An $r$-simple path $W$ in $G$ is a $\overline{d}$-fit if for every color $i \in \{1, 2, \ldots, b(k/r)\}$, the number of times $W$ visits vertices colored $i$ is exactly $d_i$. A function $\varphi : E(G) \to \mathbb{N}$ is a $\overline{d}$-fit if (i) for every vertex $v \in V(G)$, $\sum_{e \in E(G)} \varphi(e) = 0$ is an even number upper bounded by $2(r - d_i)$ where $i$ is the color of $v$, and (ii) $\sum_{b(k/r)} d_i + \frac{1}{2} \sum_{e \in E(G)} \varphi(e) \geq k$.

In the (Walk,Edges) Partition problem, we are given integers $k, r \in \mathbb{N}$, a $b(k/r)$-colored undirected graph $G$, a vertex cover $U$ of $G$ of size at most $3k/r$ where each vertex in $U$ has a unique color, and an occurrence sequence $\overline{d} = (d_1, d_2, \ldots, d b(k/r)) \in D_{r,k}$. We denote $d_i \geq 1$ for every color $i$ of a vertex in $U$. The objective is to decide whether there exist both an $r$-simple path $W$ in $G$ that is a $\overline{d}$-fit and a function $\varphi : E(G) \to \mathbb{N}$ that is a $\overline{d}$-fit.

Importantly, the two objects that we seek in the (Walk,Edges) Partition problem are independent of each other (unlike the case of (Walk,TW-2) Partition). Intuitively, the reason why we can allow this independence is precisely because we know that the walk is going to visit every vertex of a vertex cover, and hence no matter what the second object will be, we will necessarily obtain a connected multigraph at the end when we combine the two. This leads to the following result.

Lemma 4.18. Suppose that (Walk,Edges) Partition can be solved in time $f(k/r) \cdot (n + \log k)^{O(1)}$. Then, Special Undirected Colorful $r$-Simple $k$-Path can be solved in time $2^{O(k/r)} \cdot f(k/r) \cdot (n + \log k)^{O(1)}$.

Note that the existence of an $r$-simple path $W$ in $G$ that is a $\overline{d}$-fit can be easily tested by using DP. Indeed, we can just use a simplified version of the DP procedure in the proof of Lemma 4.15 that avoids all calls to the external algorithm from Lemma 4.14 (since these calls only concern the construction of $H$). Thus, we have the following observation.

Observation 4.2. There is a polynomial-time algorithm that, given an instance $(G, k, r, U, \overline{d})$ of (Walk,Edges) Partition, determines whether $G$ has an $r$-simple path $W$ that is a $\overline{d}$-fit.

Flow. Finally, we construct a flow network to prove that the existence of a function $\varphi$ that is a $\overline{d}$-fit can be tested in polynomial time.

Lemma 4.19. There is a polynomial-time algorithm that, given an instance $(G, k, r, U, \overline{d})$ of (Walk,Edges) Partition, determines whether there exists a function $\varphi : E(G) \to \mathbb{N}$ that is a $\overline{d}$-fit.

Proof Sketch. We describe the construction. To describe our algorithm $A$, let $(G, k, r, U, \overline{d})$ be an instance of (Walk,Edges) Partition. For every vertex $v \in V(G)$, denote $c_v = r - d_v$ where $i$ is the color of $v$. In addition, denote $F = \sum_{e \in V(G)} c_v$ and $\ell = 2(k - \sum_{b(k/r)} d_i)$. We construct a flow network $N$ with source $s$ and sink $t$ as follows.

- For every vertex $v \in V(G)$, insert (into $N$) two new vertices, $v_1$ and $v_2$, the arc $(v_1, v_2)$ of infinite (upper) capacity and cost 1, and the arcs $(s, v_1)$ and $(v_2, t)$ both of (upper) capacity $c_v$ and cost 0.
- For every edge $(u, v) \in E(G)$, insert (into $N$) the arcs $(u_1, v_2)$ and $(v_1, u_2)$ both of infinite (upper) capacity and cost 0.

The lower capacity of each arc is simply 0. We seek the minimum cost $C$ required to send $F$ units of (integral) flow from $s$ to $t$ in $N$. This task can be performed in polynomial times [3]. (We stress that $F$ and capacities are represented in binary, and the running time is polynomial in the size of this representation.) After performing this task, algorithm $A$ checks whether $C \leq F - \ell$. If this condition is satisfied, then $A$ accepts, and otherwise it rejects. Clearly, $A$ runs in polynomial time. The proof that our reduction is correct is omitted. □

We are ready to prove Lemma 4.2.

Proof of Lemma 4.2. By Observation 4.2 and Lemma 4.19, (Walk,Edges) Partition is solvable in polynomial time. Thus, by Lemma 4.18, Special Undirected Colorful $r$-Simple $k$-Path can be solved in time $2^{O(k/r)} \cdot (n + \log k)^{O(1)}$. In turn, by Lemma 4.17, this means that Undirected $r$-Simple $k$-Path can be solved in time $2^{O(k/r)} \cdot f(k/r) \cdot (n + \log k)^{O(1)}$. □

5 $p$-Set $(r, q)$-Packing

Recall that in the $p$-Set $(r, q)$-Packing problem, the input consists of a ground set $V$ of size $n$, positive integers $p, q, r$, and a collection $\mathcal{H}$ of (not necessarily distinct) sets of size $q$ 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. Let $\kappa = pq/r$. 

Copyright © 2019 by SIAM
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 \( f(n) \). The reduction uses a tool known as \emph{representative sets} to discard irrelevant parts of the instance. Representative sets have important applications for both FPT algorithms [21] and kernels [28]; see also [15, 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 \( H \) to be \( p \)-uniform using dummy elements for smaller sets. Compute \( q \) disjoint representative sets for the padded version of \( H \) in the uniform matroid \( U_{n, \kappa+p} \), and discard any set in \( H \) not contained in any of the resulting representative sets.

**Lemma 5.1.** \emph{Reduction Rule (2) is sound and leaves at most}

\[
m \leq q\left(\frac{\kappa + p}{p}\right)
\]

sets. The rule can be applied in time polynomial in the input size and \( (\kappa+p{/}p) \).

\textbf{Proof.} Each representative set has size at most \( (\kappa+p{/}p) \), hence the total size of the output is indeed

\[
m' \leq q\left(\frac{\kappa + p}{p}\right).
\]

We argue correctness. Let \( H' \) be the instance produced. Clearly if \( H' \) is positive then so is \( H \). Now assume that \( H \) is positive, and let \( F \subseteq H \), \( |F| = q \), be a solution with maximum intersection with \( H' \). Assume there exists a set \( E \in F \setminus H' \). Let \( X \) be the set of vertices that occur precisely \( r \) times in \( F \), and let \( X' = X \setminus E \). Thus \( |X'| \leq |X| \leq \kappa \). Then each representative family contains at least one set \( E' \) disjoint from \( X' \), i.e., \( q \) alternative sets \( E' \) in total. Since \( |F - E| < q \), for at least one such set \( E' \) it also holds that \( E' \notin F \). Then \( F - E + E' \) is a packing of \( q \) sets, where every element occurs in at most \( r \) sets, and with a larger intersection with \( H' \) than \( F \), which contradicts that \( F \) was maximal. Thus \( F \subseteq H' \), and the output instance is positive. The running time follows from the computation of a representative set. \( \square \)

**Lemma 5.2.** Once the two rules have been applied exhaustively, \( n < f(pq/r) \) where \( f(n) = \kappa 4^n \).

\textbf{Proof.} On the one hand, since every element of the ground set occurs in more than \( r \) sets of the input, there are \( m > rn/p \) sets in the input, hence \( n < mp/r \). On the other hand, by the representative sets reduction we have \( m < q4^{pq/r} \). Then

\[
n < mp/r < q4^{pq/r}p/r = f(pq/r).
\]

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

**Lemma 5.3.** \emph{\( (r, q) \)-Packing can be solved in time \( n^{O(pq/r)} \).}

\textbf{Proof.} Let \( m^* \) denote the number of distinct sets in the input. Then \( m^* = O(n^p) \). It is easy to write FEASIBILITY ILP instance with \( m^* \) variables that encodes the problem, with one variable \( x_E \) for each distinct set \( E \) denoting the number of copies of \( E \) to use in the solution, with constraints that \( x_E \) is bounded by the multiplicity of \( E \) in the input, and a packing constraint for each vertex. Thus, the number of variables, constraints and size of the Feasibility ILP instance are \( m^* \), \( n \) and \( O(nm^*) = O(n^{p+1}) \), respectively. Hence, by Theorem 2.3, we can solve the instance in time \( O(n^{O(pq/r)}) \). \( \square \)

Now we can obtain the main result of this section.

**Theorem 5.2.** \emph{\( (r, q) \)-Packing parameterized by \( \kappa \) is FPT.}

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

We observe that the same reduction gives a polynomial kernel with \( p \) as a constant.

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

\textbf{Proof.} By Lemma 5.1, if the reduction rules have been applied then the number of sets is bounded by

\[
m \leq q\left(\frac{pq/r + p}{p}\right) \leq q(pq/r + 1)^p,
\]

and as in Lemma 5.2 with \( p \) a constant we get \( n = O(m/r) \). Putting them together, we get \( n = O((q/r)^{p+1}) \). This gives the first result. For the latter, we may observe that the reduction produces a multiset where at most

\[
m^* \leq (n+1)^p = O((q/r)^{p+1})
\]
distinct sets are possible (since sets have size at most \( p \)). Hence the instance can be described by giving the multiplicity in the input for each set type, keeping only the first \( r \) copies of each set. This gives a description with \( m^* \log r \) bits. Finally, we note that \( r \leq q \leq m \) and that the input instance of \( (r, q) \)-Packing is coded without multiplicities;
hence $r$ is bounded by the total input size. If the total input size is at least $2^{m^* \log m^*}$ then we can solve the problem completely in polynomial time, otherwise we have $\log r \leq m^* \log m^*$.

We will use the following simple lemma.

**Lemma 5.4.** [4] Let $L, L'$ be a pair of decidable parameterized problems such that $L'$ is in NP, and $L$ is NP-complete. If there is a general kernelization from $L$ to $L'$ producing a generalized kernel of polynomial size, then $L$ has a polynomial-size kernel.

Theorem 5.3 and Lemma 5.4 imply the following:

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

Let us complement Theorem 5.3 by showing that the lower bound for $r = 1$ carries over to the parameter $q/r$ for arbitrary values of $r$.

**Theorem 5.4.** The $p$-Set $(r,q)$-Packing problem with fixed value of $p \geq 3$ does not admit a generalized kernel of size $O((q/r)^{\varepsilon})$ for any $\varepsilon > 0$ unless the polynomial hierarchy collapses.

**Proof.** Dell and Marx [16] showed that Perfect $p$-Set Matching (i.e., the variant where $r = 1$ and $n = pq$) does not admit a generalized kernel of $O(q^{\varepsilon})$-bits for any $\varepsilon > 0$ unless the polynomial hierarchy collapses. We show a parameter-preserving reduction from the case of $r = 1$ to the arbitrary case. Let $H$ be the input to an instance of Perfect $p$-Set Matching where $H \subseteq Q$ is a $p$-uniform hypergraph over some ground set $V$, $|V| = n = pq$. We produce an output instance of $p$-Set $(r,q)$-Packing by padding $H$ with $(r-1)n$ sets, each of which is incident with precisely one member of $V$ and which in total cover every element of $V$ precisely $r-1$ times. (We pad these sets with arbitrary dummy elements to produce a $p$-uniform output.) We set $q' = q + (r-1)n$. We claim that the output has a $(q',r)$-packing if and only if $H$ contains a $q$-packing. This is not hard to see. On the one hand, any $q$-packing in $H$ can be padded to a $q'$-packing in the output by including all the packing sets; on the other hand, for any $(q',r)$-packing where some element $v \in V$ is covered by two non-padding sets, we can get a different $(q',r)$-packing by discarding one set from $H$ and replacing it by a further padding set covering $v$. The value of $p$ is unchanged. Finally, since $q < n$ we have $q' = q + (r-1)n < rn$, hence $q'/r < n = pq = O(q)$, and the parameter is only increased by a constant factor. □

6. $(r,k)$-Monomial Detection: para-NP-Hardness

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. 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.

**Proof.** To prove this theorem, we give a reduction from Partition to $(r,k)$-Monomial Detection parameterized by $k/r$. To this end, let $M$ be an instance of Partition. We define our set of variables as $\{x,y\}$ (that is, we have only two variables), and we define a polynomial $POL$ as follows:

$$POL = \sum_{M' \subseteq M} \left( \prod_{n \in M'} x^n \right) \cdot \left( \prod_{n \in M \setminus M'} y^n \right).$$

We define $k = \sum_{n \in M} n$, and $r = k/2$.

We now prove that $M$ is a Yes-instance of Partition if and only if $POL$ has a monomial of degree $k$ where each variable has degree at most $r$. To this end, notice that $M$ is a Yes-instance of Partition if and only if there exists $M' \subseteq M$ such that $\sum_{n \in M'} n = k/2$. Now, for any $M' \subseteq M$, the following statement holds: $\sum_{n \in M'} n = k/2$ if and only if $(\prod_{n \in M'} x^n) \cdot (\prod_{n \in M \setminus M'} y^n)$ is a monomial (of degree $k$) where each variable has degree at most $r$. However, by the definition of $POL$, the latter part of the statement is true if and only if $POL$ has a monomial of degree $k$ where each variable has degree at most $r$.

Next, we show that $POL$ can be encoded by a non-canceling arithmetic circuit of size polynomial in $\log k$. To this end, denote $M = \{n_1, n_2, \ldots, n_\ell\}$ where $\ell = |M|$, and let $n^*$ be the largest number that occurs at least once in $M$. Then, for all $z \in \{x,y\}$ and $i \in \{0,1,\ldots,\lfloor \log_2 n^* \rfloor\}$, we have a gate $g_{z,i}$ defined recursively as follows. First, for all $z \in \{x,y\}$, we set $g_{z,0}$ to be the input gate $z$. Second, for all $z \in \{x,y\}$ and $i \in \{1,2,\ldots,\lfloor \log_2 n^* \rfloor\}$, we set $g_{z,i} = g_{z,i-1}^2$. By simple induction on $i$, for all $z \in \{x,y\}$ and $i \in \{0,1,\ldots,\lfloor \log_2 n^* \rfloor\}$, it holds that $g_{z,i}$ encodes $z^{2^i}$. Now, for all $z \in \{x,y\}$ and
$n \in M$, we have a gate $g_{z,n}$ defined as follows:

$$g_{z,n} = \prod_{i \in \{1,2,\ldots,|\log_2 n+1\}} \hat{g}_{z,i},$$

where $\text{digit}(n,i)$ is the $i$-th least significant digit of $n$ when encoded in binary. Then, for all $z \in \{x,y\}$ and $n \in M$, we have that $g_{z,n}$ encodes $z^n$.

For all $i \in \{1,2,\ldots,\ell\}$, we have gates $h'_i$ and $h_i$ defined recursively as follows. First, we set $h_1 = h'_1 = g_{x,n} + g_{y,n}$. Second, for all $i \in \{2,3,\ldots,\ell\}$, we set $h'_i = g_{x,n} + g_{y,n}$, and $h_i = h_{i-1} \cdot h'_i$. By simple induction on $i$, we have that for all $i \in \{1,2,\ldots,\ell\}$, $h_i$ encodes the following polynomial:

$$\sum_{M' \subseteq \{n_1,n_2,\ldots,n_\ell\}} \left( \prod_{n \in M'} x^n \right) \cdot \left( \prod_{n \in \{n_1,n_2,\ldots,n_\ell\} \setminus M'} y^n \right)$$

Thus, $h_{\ell}$ encodes $\text{POL}$.

Finally, we argue that $(r,k)$-Monomial Detection is $\text{para-NP}$-hard parameterized by $k/r$. Suppose, by way of contradiction, that this claim is false. Then, $(r,k)$-Monomial Detection admits an algorithm, say $\mathcal{A}$, that runs in time $|I|^f(k/r)$ on input $I$ for some function $f$ that depends only on $k/r$. Thus, we can solve any instance $M$ of $\text{PARTITION}$ by using the reduction above to construct (in polynomial time) an equivalent instance $I$ of $(r,k)$-Monomial Detection, and then calling $\mathcal{A}$ with $I$. However, the parameter $k/r$ equals 2 (since $r = k/2$), and hence $|I|^f(k/r) = |I|^\phi(1)$, that is, we solve $\text{PARTITION}$ in polynomial-time. Since $\text{PARTITION}$ is $\text{NP}$-hard, we have reached a contradiction. This completes the proof. □

7 $p$-Multiset $(r,q)$-Packing and $(r,k)$-Monomial Detection: $\text{W}[1]$-Hardness

The first result of this section is that $p$-Multiset $(r,q)$-Packing is $\text{W}[1]$-hard. The proof of this theorem uses a reduction from the Multicolored Clique problem, which is known to be $\text{W}[1]$-hard [32, 19]. In this problem, we are given a vertex-colored graph $G$ and a positive integer $k$, where each vertex has a color in $\{1,2,\ldots,k\}$, and our goal is to decide whether $G$ has a multicolored clique, that is, a clique where each vertex has a distinct color. Later in this section, we show that our theorem implies that a restricted case of $(r,k)$-Monomial Detection is $\text{W}[1]$-hard as well.

**Theorem 7.1.** $p$-Multiset $(r,q)$-Packing is $\text{W}[1]$-hard parameterized by $pq/r$ even if the size of the universe is $pq/r$.

**Proof.** Our source problem is Multicolored Clique. Given an instance $(G,k)$ of Multicolored Clique, we construct an instance $(U,S,p,q,r)$ of $p$-Multiset $(r,q)$-Packing as follows. For each color $i \in \{1,\ldots,k\}$, let $C^i$ be the set of vertices in $G$ whose color is $i$. Let $n$ denote the size of a color class, that is, $n = |C^i|$ for any $i \in \{1,\ldots,k\}$. Moreover, for all $i \in \{1,\ldots,k\}$, denote $C^i = \{v^i_1,v^i_2,\ldots,v^i_n\}$. Define $r = n$, $p = kn$ and $q = k + \binom{k}{2}$. Note that $pq/r = (kn)(k + \binom{k}{2})/n = k(k + \binom{k}{2})$.

The universe $U$ contains the following distinct elements:

- For each color $i \in \{1,\ldots,k\}$, we have an element $c^i$.
- For each pair $(i,j) \in \{1,\ldots,k\} \times \{1,\ldots,k\}$ with $i \neq j$, we have an element $c^i \cdot c^j$ and an element $\hat{c}^i \cdot \hat{c}^j$.
- For each pair $(i,j) \in \{1,\ldots,k\} \times \{1,\ldots,k\}$ with $i < j$, and for each $t \in \{1,\ldots,k-2\}$, we have an element $c^t_{i,j}$. Observe that $|U| = k + 2k(k-1) + \binom{k}{2}(k-2) = k(k + \binom{k}{2}) = pq/r$.

Now, we construct $S$ as follows.

- For each color $i \in \{1,\ldots,k\}$ and for each $x \in \{1,\ldots,n\}$, we insert the multiset $M_x^i = \{x\} \cup \{(n-x)\cdot c^t_{i,j} \mid j \in \{1,\ldots,k\} \setminus \{i\}\}$

Note that $|M_x^i| = n + (k-1)n = p$.

- For each edge $e = \{v^i_x,v^j_y\} \in E(G)$ (where $v^i_x \in C^i$ and $v^j_y \in C^j$) with $i < j$, we insert the multiset

$$M_x^{(i,j)}(x,y) = \left( \bigcup_{t \in \{1,\ldots,k-2\}} \{x\cdot c^t_{i,j},(n-x)\cdot c^t_{i,j}\} \right) \cup \{(n-x)\cdot c^t_{i,j},[x] \cdot \hat{c}^t_{i,j},(n-y)\cdot \hat{c}^t_{i,j},[y] \cdot \hat{c}^t_{i,j}\}.$$  

Note that $|M_x^{(i,j)}(x,y)| = (k-2)n + 2n = p$.

**Proof of Correctness.** In the forward direction, we suppose that we have a multicolored clique $K$ in $G$. Let $v^i_1$ be the (unique) vertex in $C^i$ that belongs to $K$. Then, it holds that the subcollection $S' := \{M_x^{(i)} : i \in \{1,\ldots,k\}\} \cup \{M_x^{(i,j)}(i,j) : (i,j) \in \{1,\ldots,k\} \times \{1,\ldots,k\}, i < j\}$ of $S$ is an $r$-relaxed packing of size $q$. (To see that this claim is true, observe that each element in $U$ occurs in this subcollection precisely $n$ times.)

In the reverse direction, we suppose that we have a subcollection $S'$ of $S$ that is an $r$-relaxed
packing of size $q$. Then, we first observe that $S'$ can contain at most one multiset from $\{M_1, \ldots, M_n\}$ for each $i \in \{1, \ldots, k\}$ (since otherwise the element $c^i$ occurs more than $r$ times), and at most one multiset from $\{M^{(i,j)}_{x,y} : (v_{x}^i, v_{y}^i) \in E(G)\}$ for each $(i,j) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$ with $i < j$ (since otherwise the element $c^{i,j}$ occurs more than $r$ times). Then, because $\lvert S' \rvert = q$, we have that $S'$ contains exactly one multiset from $\{M_1, \ldots, M_n\}$ for each $i \in \{1, \ldots, k\}$, and exactly one multiset from $\{M^{(i,j)}_{x,y} : (v_{x}^i, v_{y}^i) \in E(G)\}$ for each $(i,j) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$ with $i < j$. In particular, this means that it is well defined to let $\varphi(i)$, $i \in \{1, \ldots, k\}$, denote the integer $x$ such that $M^i_x \in S'$. Moreover, it is well defined to let $\varphi(i,j)$, $(i,j) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$ with $i < j$, denote the pair $(x,y)$ such that $M^{(i,j)}_{x,y} \in S'$.

Define $K = G[\{v_1^{(1)}, \ldots, v_k^{(k)}\}]$. Then, we claim that $K$ is a multicolored $k$-clique in $G$. It is clear that $\lvert V(K) \rvert = k$ and that $K$ is multicolored. Thus, it remains to show that for each $(i,j) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$ with $i < j$, it holds that $\{v_{\varphi(i)}^{(i)}, v_{\varphi(j)}^{(j)}\} \in E(G)$. For this purpose, we arbitrarily select $(i,j) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$ with $i < j$. To show that $\{v_{\varphi(i)}^{(i)}, v_{\varphi(j)}^{(j)}\} \in E(G)$, it suffices to show that $\varphi(i,j) = (\varphi(i), \varphi(j))$. Let us denote $\varphi(i,j) = (x,y)$. We only show that $x = \varphi(i)$, since the proof that $y = \varphi(j)$ is symmetric. Suppose, by way of contradiction, that $x \neq \varphi(i)$. We consider two cases.

- First, suppose that $x < \varphi(i)$. Note that $c^{i,x}$ occurs $\varphi(i)$ times in $M^i_{\varphi(i)}$, and it occurs $n - x$ times in $M^{(i,x)}_{\varphi(i)}$. However, $\varphi(i) + (n - x) > n$, which implies that $c^{i,x}$ occurs more than $r$ times in $S'$. Thus, we have reached a contradiction.

- Second, suppose that $x > \varphi(i)$. Note that $c^{x,i}$ occurs $n - \varphi(i)$ times in $M^x_{\varphi(i)}$, and it occurs $x$ times in $M^{(x,i)}_{\varphi(i)}$. However, $(n - \varphi(i)) + x > n$, which implies that $c^{x,i}$ occurs more than $r$ times in $S'$. Thus, we have reached a contradiction.

This completes the proof. □

Our reduction heavily relies on the inclusion of input instances that contain multisets rather than sets. In particular, it does not rule out the possibility that $p$-$SET$ $(r,q)$-$PACKING$ is FPT parameterized by $(pq)/r$—that is, this proof does not contradict Section 5.

As a consequence of Theorem 7.1, we obtain the following theorem.

**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.

**Proof.** The proof of this theorem is based on the standard encoding of set packing problems using multivariate polynomials (see, e.g., [27]). For the sake of completeness, we present the details. By Theorem 7.1, it suffices to give a reduction from $p$-$MULTISET$ $(r,q)$-$PACKING$ with $\lvert U \rvert \leq (pq)/r$. To this end, let $(U,S,p,q,r)$ be an instance of $p$-$MULTISET$ $(r,q)$-$PACKING$ with $\lvert U \rvert \leq (pq)/r$. Since $p$ is the size of each multiset in the input, it is polynomial in the input size. Moreover, $q$ (and hence also $r$) can be assumed to be polynomial in the input size, since if $q > \lvert S \rvert$, then we have a No-instance.

We define our set of variables as $X = \{x_u : u \in U\}$ (that is, we have one variable for each element in $U$), and we define a polynomial $POL$ as follows:

$$
POL = \sum_{s' \subseteq S} \prod_{M \in \mathcal{S}^1 \cup s} \prod_{u \in M} x_u.
$$

Define $k = pq$. For any choice of non-negative integers $d_u$ for each $u \in U$ whose sum is $k$, it holds that $POL$ has $\prod_{u \in U} x_{u}^{d_u}$ as a monomial if and only if there exists a subcollection $S' \subseteq S$ of size $q$ where each element $u \in U$ occurs exactly $d_u$ times. Thus, $(U,S,p,q,r)$ is a Yes-instance of $p$-$MULTISET$ $(r,q)$-$PACKING$ if and only if $POL$ has a monomial (of total degree $k$) where the degree of each variable is at most $r$.

To complete the proof, it remains to show that $POL$ can be encoded by an arithmetic circuit of polynomial size. For this purpose, denote $\mathcal{S} = \{M_1, M_2, \ldots, M_s\}$ where $\ell = \lvert S \rvert$. For each $M \in \mathcal{S}$, we have a gate $g_M$ which is the multiplication $\prod_{u \in M} x_u$. Now, for all $i \in \{1,2,\ldots,\ell\}$ and $j \in \{1,2,\ldots,q\}$, we have a gate $g_{i,j}$ that is defined as follows.

- If $j = 1$, then $g_{i,j} = \sum_{t=1}^{i} g_{M_t}$ for all $i \in \{1,2,\ldots,\ell\}$.
- If $i = 1$ and $j > 1$, then $g_{i,j} = 0$.
- If $i > 1$ and $j > 1$, then $g_{i,j} = g_{i-1,j} + g_{i-1,j-1}$.

The output of the arithmetic circuit is given by $g_{\ell,q}$.

To see that the circuit above encodes $POL$, we claim that for all $i \in \{1,2,\ldots,\ell\}$ and $j \in \{1,2,\ldots,q\}$, it holds that

$$
g_{i,j} = \sum_{s' \subseteq \{M_1,M_2,\ldots,M_s\}} \prod_{M \in \mathcal{S}^1 \cup s} \prod_{u \in M} x_u.
$$
The proof is by induction. In the basis, where \( i = 1 \) or \( j = 1 \), the claim clearly holds. Now, suppose that the claim holds for \( i - 1 \geq 1 \), and let us prove it for \( i \). Then, by the inductive hypothesis,

\[
g_{i,j} = g_{i-1,j} + g_{i-1,j-1} + g_{M_i}
= \sum_{s \subseteq (M_1, \ldots, M_{i-1})} \prod_{\alpha \in s} x_{\alpha} \prod_{\beta \in M_i} x_{\beta} + g_{M_i}
= \sum_{s \subseteq (M_1, \ldots, M_{i-1})} \prod_{\alpha \in s} x_{\alpha} \prod_{\beta \in M_i} x_{\beta}.
\]

This completes the proof. \( \square \)

In light of Theorems 7.1 and 7.2, the reader might wonder whether \( p \)-MULTISET \((r, q)\)-PACKING and the special case of \((r, k)\)-MONOMIAL DETECTION where \( r \) is polynomially bounded by the input size are at least in XP. However, this question has already been resolved positively—the \( 2^{O((k/r) \log r)} \cdot n^{O(1)} \)-time algorithms by Abasi et al. [1] and Gabizon et al. [24] imply that this containment holds. Thus, we have a complete characterization of the parameterized complexity of all problems studied in this paper.

8 Conclusion

In this paper, we considered three problems, DIRECTED \( r \)-SIMPLE \( k \)-PATH and \( p \)-SET \((r, q)\)-PACKING and \((r, k)\)-MONOMIAL DETECTION, parameterized by \( k/r \). We proved that DIRECTED \( r \)-SIMPLE \( k \)-PATH and \( p \)-SET \((r, q)\)-PACKING are FPT, but \((r, k)\)-MONOMIAL DETECTION is para-NP-hard. In particular, we established a \( 2^{O((k/r)^2 \log(k/r))} \cdot \binom{n+log(k)}{k}^{O(1)} \)-time algorithm for \( r \)-SIMPLE \( k \)-PATH on digraphs and a \( 2^{O(k/r)} \cdot \binom{n+log(k)}{k}^{O(1)} \)-time algorithm for \( r \)-SIMPLE \( k \)-PATH on undirected graphs. Our work also resolved an open problem posed by Gabizon et al. concerning the design of polynomial kernels for problems with relaxed disjointness constraints whose size becomes smaller as the relaxation parameter becomes larger.

Let us conclude our paper with a couple of open problems. First, it would interesting to characterize input polynomials \( P \) for which \((r, k)\)-MONOMIAL DETECTION becomes FPT or, at least, find non-trivial sufficient conditions for \( P \) such that the restricted \((r, k)\)-MONOMIAL DETECTION is FPT and both DIRECTED \( r \)-SIMPLE \( k \)-PATH and \( p \)-SET \((r, q)\)-PACKING can be easily reduced to it. Secondly, we would like to point out that the existence of a single-exponential \( FPT \) algorithm for DIRECTED \( r \)-SIMPLE \( k \)-PATH remains an open problem. The question of the existence of a deterministic \( 2^{O((n/d) \log d)} \)-time algorithm for DEGREE-BOUNDED SPANNING TREE, which we did not consider in this study, is also open.

In general, it would be interesting to study the parameterized complexity of other problems with relaxed disjointness constraints parameterized by \( k/r \). Indeed, we believe that much remains to be explored in the realm of problems with relaxed disjointness constraints. Such problems can enable to obtain substantially (sometimes super-exponentially) better solutions at the expense of allowing repetitions, sometimes with the great advantage of a time complexity that diminishes surprisingly fast as \( r \) increases.

References

[1] H. Abasi, N. H. Bshouty, A. Gabizon, and E. Håkmaty, 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, 3, 19
[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. 3
[3] R. K. Abuj, T. L. Magnanti, and J. B. Orlin, Network flows - theory, algorithms and applications, Prentice Hall, 1993. 14
[4] N. Alon, G. Z. Gutin, E. J. Kim, S. Szeider, and A. Yeo, Solving MAX-\( r \)-SAT above a tight lower bound, Algorithmica, 61 (2011), pp. 638–655. 16
[5] N. Alon, R. Yuster, and U. Zwick, Color-coding, J. ACM, 42 (1995), pp. 844–856. 7
[6] J. Bang-Jensen and G. G. (Eds.), Classes of Directed Graphs, Springer, 2018. 4
[7] J. Bang-Jensen and G. Gutin, Digraphs - theory, algorithms and applications, Springer, 2002. 4, 5
[8] A. Berger, L. Kozma, M. Mnich, and R. Vincze, Time- and space-optimal algorithms for the many-visits TSP, CoRR, abs/1804.06361 (2018). 11
[9] A. Björklund, Determinant sums for undirected hamiltonicity, SIAM J. Comput., 43 (2014), pp. 280–299. 3
[10] 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
[11] A. Björklund, V. Kamat, L. Kowalik, and M. Zehavi, Spotting trees with few leaves, SIAM J. Discrete Math., 31 (2017), pp. 687–713. 3

Copyright © 2019 by SIAM
