Detours in Directed Graphs

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway

Petr A. Golovach
Department of Informatics, University of Bergen, Norway

William Lochet
Department of Informatics, University of Bergen, Norway

Danil Sagunov
St. Petersburg Department of V.A. Steklov Institute of Mathematics, Russia
JetBrains Research, Saint Petersburg, Russia

Kirill Simonov
Algorithms and Complexity Group, TU Wien, Austria

Saket Saurabh
Institute of Mathematical Sciences, HBNI, Chennai, India
Department of Informatics, University of Bergen, Norway

Abstract

We study two "above guarantee" versions of the classical Longest Path problem on undirected and directed graphs and obtain the following results. In the first variant of Longest Path that we study, called Longest Detour, the task is to decide whether a graph has an (s,t)-path of length at least $\text{dist}_G(s,t)+k$ (where $\text{dist}_G(s,t)$ denotes the length of a shortest path from s to t). Bezáková et al. [7] proved that on undirected graphs the problem is fixed-parameter tractable (FPT) by providing an algorithm of running time $2^{\mathcal{O}(k)} \cdot n$. Further, they left the parameterized complexity of the problem on directed graphs open. Our first main result establishes a connection between Longest Detour on directed graphs and 3-Disjoint Paths on directed graphs. Using these new insights, we design a $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ time algorithm for the problem on directed planar graphs. Further, the new approach yields a significantly faster FPT algorithm on undirected graphs.

In the second variant of Longest Path, namely Longest Path above Diameter, the task is to decide whether the graph has a path of length at least $\text{diam}(G)+k$ ($\text{diam}(G)$ denotes the length of a longest shortest path in a graph G). We obtain dichotomy results about Longest Path above Diameter on undirected and directed graphs. For (un)directed graphs, Longest Path above Diameter is NP-complete even for $k=1$. However, if the input undirected graph is 2-connected, then the problem is FPT. On the other hand, for 2-connected directed graphs, we show that Longest Path above Diameter is solvable in polynomial time for each $k \in \{1,\dots,4\}$ and is NP-complete for every $k \geq 5$. The parameterized complexity of Longest Detour on general directed graphs remains an interesting open problem.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases longest path, longest detour, diameter, directed graphs, parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.29

Related Version *Full Version*: `https://arxiv.org/abs/2201.03318`

Funding *Fedor V. Fomin*: supported by the Research Council of Norway via the project BWCA (grant no. 314528).
Petr A. Golovach: supported by the Research Council of Norway via the project BWCA (grant no. 314528).

William Lochet: supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 819416).
Danil Sagunov: supported by Leonhard Euler International Mathematical Institute in Saint Petersburg (agreement no. 075-15-2019-1620).
Kirill Simonov: supported by the Austrian Science Fund (FWF) via project Y1329 (Parameterized Analysis in Artificial Intelligence).
Saket Saurabh: supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 819416) and Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18.

1 Introduction

In the Longest Path problem, we are given an n-vertex graph G and an integer k. (Graph G could be undirected or directed.) The task is to decide whether G contains a path of length at least k. Longest Path is a fundamental algorithmic problem that played one of the central roles in developing parameterized complexity [46, 9, 2, 36, 40, 13, 12, 41, 51, 26, 26, 25, 42, 8]. To further our algorithmic knowledge about the Longest Path problem, Bezáková et al. [7] introduced a novel "above guarantee" parameterization for the problem. For a pair of vertices s, t of an n-vertex graph G, let $\text{dist}_G(s, t)$ be the distance from s to t, that is, the length of a shortest path from s to t. In this variant of Longest Path, the task is to decide whether a graph has an (s, t)-path of length at least $\text{dist}_G(s, t) + k$. The difference with the "classical" parameterization is that instead of parameterizing by the path length, the parameterization is by the offset k.

Longest Detour **Parameter:** k
Input: A graph G, vertices $s, t \in V(G)$, and an integer k.
Task: Decide whether there is an (s, t)-path in G of length at least $\text{dist}_G(s, t) + k$.

Since the length of a shortest path between s and t can be found in linear time, such a parameterization could provide significantly better solutions than parameterization by the path length. Bezáková et al. [7] proved that on undirected graphs the problem is fixed-parameter tractable (FPT) by providing an algorithm of running time $2^{\mathcal{O}(k)} \cdot n$. Parameterized complexity of Longest Detour on directed graphs was left as the main open problem in [7]. Our paper makes significant step towards finding a solution to this open problem.

Our results. Our first main result establishes a connection between Longest Detour and another fundamental algorithmic problem p-Disjoint Paths. Recall that the p-Disjoint Paths problem is to decide whether p pairs of *terminal* vertices (s_i, t_i), $i \in \{1, \dots, p\}$, in a (directed) graph G could be connected by pairwise internally vertex disjoint (s_i, t_i)-paths. We prove (the formal statement of our result is given in Theorem 4) that if $\mathcal{C}$ is a class of (directed) graphs such that p-Disjoint Paths admits a polynomial time algorithm on $\mathcal{C}$ for $p = 3$, then Longest Detour is FPT on $\mathcal{C}$. Moreover, the FPT algorithm for Longest Detour on $\mathcal{C}$ is single-exponential in k (running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$).

Unfortunately, our result does not resolve the question about parameterized complexity of Longest Detour on directed graphs. Indeed, Fortune, Hopcroft, and Wyllie [29] proved that p-Disjoint Paths is NP-complete on directed graphs for every fixed $p \geq 2$. However, the new insight helps to establish the tractability of Longest Detour on planar directed graphs, whose complexity was also open. The theorem of Schrijver from [48] states that p-Disjoint Paths could be solved in time $n^{\mathcal{O}(p)}$ when the input is restricted to planar directed graphs. (This result was improved by Cygan et al. [17] who proved that p-Disjoint Paths parameterized by p is FPT on planar directed graphs.) Pipelined with our theorem, it immediately implies that Longest Detour is FPT on planar directed graphs.

Besides establishing parameterized complexity of Longest Detour on planar directed graphs our theorem has several advantages over the previous work even on undirected graphs. By the seminal result of Robertson and Seymour [47], p-Disjoint Paths is solvable in $f(p) \cdot n^3$ time on undirected graphs for some function f of p only. Therefore on undirected graphs p-Disjoint Paths is solvable in polynomial time for every fixed p, and for $p = 3$ in particular. Later the result of Robertson and Seymour was improved by Kawarabayashi, Kobayashi, and Reed [38] who gave an algorithm with quadratic dependence on the input size. Pipelined with our result, this brings us to a Monte Carlo randomized algorithm solving Longest Detour on undirected graphs in time $10.8^k \cdot n^{\mathcal{O}(1)}$. Our algorithm can be derandomized, and the deterministic algorithm runs in time $45.5^k \cdot n^{\mathcal{O}(1)}$. While the algorithm of Bezáková et al. [7] for undirected graphs runs in time $\mathcal{O}(c^k \cdot n)$, that is, is single-exponential in k, the constant c is huge. The reason is that their algorithm exploits the Win/Win approach based on excluding graph minors. More precisely, Bezáková et al. proved that if a 2-connected graph G contains as a minor, a graph obtained from the complete graph K_4 by replacing each edge by a path with k edges, then G has an (s,t)-path of length at least $\text{dist}_G(s,t) + k$. Otherwise, in the absence of such a graph as a minor, the treewidth of G is at most $32k + 46$. Combining this fact with an FPT 3-approximation algorithm [11], running in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$, to compute the treewidth of a graph, brings us to a graph of treewidth at most $96k + \mathcal{O}(1)$. Finally, solving Longest Detour on graphs of bounded treewidth by one of the known single-exponential algorithms, see [18, 10, 27], will result in running time $3^{96k} \cdot n^{\mathcal{O}(1)}$. Thus on undirected graphs, our algorithm reduces the constant c in the base of the exponent from 3^{96} down to 10.8!

Our second set of results addresses the complexity of the problem strongly related to Longest Detour. The length of a longest shortest path in a graph G is denoted by *diameter of* G, $\text{diam}(G)$. Thus every graph G has a path of length at least $\text{diam}(G)$. But does it have a path of length longer than $\text{diam}(G)$? This leads to the following parameterized problem.

Longest Path above Diameter **Parameter:** k
Input: A graph G and an integer k.
Task: Decide whether there is a path in G of length at least $\text{diam}(G) + k$.

As in Longest Detour, the parameterization is by the offset k. When (s,t) is a pair of diametral vertices in G, the length of the shortest (s,t)-path in G is the diameter of G. However, this does not allow to reduce Longest Path above Diameter to Longest Detour– if there is a path of length $\text{diam}(G) + k$ in G, it is not necessarily an (s,t)-path. Moreover, such a path might connect two vertices with a much smaller distance between them than $\text{diam}(G)$. In fact, our hardness results for Longest Path above Diameter are based precisely on instances where the target path has this property: its length is very close to $\text{diam}(G)$, but much larger than the shortest distance between its endpoints. Thus, the lower bounds we obtain for Longest Path above Diameter are not applicable to Longest Detour.

We obtain the following dichotomy results about Longest Path above Diameter on undirected and directed graphs. For undirected graphs, Longest Path above Diameter is NP-complete even for $k = 1$. However, if the input undirected graph is 2-connected, that is, it remains connected after deleting any of its vertices, then the problem is FPT. For directed graphs, the problem is also NP-complete even for $k = 1$. However, the situation is more complicated and interesting on 2-connected directed graphs. (Let us remind that a strongly connected digraph G is 2-connected or strongly 2-connected, if for every vertex $v \in V(G)$, graph $G - v$ remains strongly connected.) In this case, we show that LONGEST PATH ABOVE DIAMETER is solvable in polynomial time for each $k \in \{1, \dots, 4\}$ and is NP-complete for every $k \geq 5$.

Our approach. A natural way to approach LONGEST DETOUR on directed graphs would be to mimic the algorithm for undirected graphs. By the result of Kawarabayashi and Kreutzer [39], every directed graph of sufficiently large directed treewidth contains a sizable directed grid as a "butterfly minor". However, as reported in [6], there are several obstacles towards applying the grid theorem of Kawarabayashi and Kreutzer for obtaining a Win/Win algorithm. After several unsuccessful attempts, we switched to another strategy.

We start the proof of Theorem 4 by checking whether G has an (s, t)-path of length $\text{dist}_G(s, t) + \ell$ for $k \leq \ell < 2k$. This can be done in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ by calling the algorithm of Bezáková et al. [7] that finds an (s, t)-path in a directed G of length *exactly* $\text{dist}_G(s, t) + \ell$. If such a path is not found, we conclude that if (G, k) is a yes-instance, then G contains an (s, t)-path of length at least $\text{dist}_G(s, t) + 2k$.

Next, we check whether there exist two vertices v and w reachable from s such that $\text{dist}_G(s, w) - \text{dist}_G(s, v) \geq k$ and G has pairwise disjoint (s, w)-, (w, v)-, and (v, t)-paths. If such a pair of vertices exists, we obtain a solution by concatenating disjoint (s, w)-, (w, v)-, and (v, t)-paths. This is the place in our algorithm, where we require a subroutine solving 3-DISJOINT PATHS.

When none of the above procedures finds a detour, we prove a combinatorial claim that allows reducing the search of a solution to a significantly smaller region of the graph. This combinatorial claim is the essential part of our algorithm. More precisely, we show that there are two vertices u and x, and a specific induced subgraph H of G (depending on u and x) such that G has an (s, t)-path of length at least $\text{dist}_G(s, t) + k$ if and only if H has an (u, x)-path of length at least ℓ for a specific $\ell \leq 2k$ (also depending on u and x). Moreover, given u, in polynomial time, we can find a feasible domain for vertex x, and for each choice of x, we can also determine ℓ and construct H in polynomial time. Then we apply the algorithm of Fomin et al. [28] to check whether H has an (u, x)-path in H of length at least ℓ.

Our strategy for LONGEST PATH ABOVE DIAMETER is different. For undirected graphs, the solution turns out to be reasonably simple. It easy to show that LONGEST PATH ABOVE DIAMETER is NP-complete for $k = 1$ by reducing HAMILTONIAN PATH to it. When an undirected graph G is 2-connected, and the diameter is larger than $k + 1$, then G always contains a path of length at least $d + k$. If the diameter is at most k, it suffices to run a LONGEST PATH algorithm to show that the problem is FPT. For directed graphs, a similar reduction shows that the problem is NP-complete for $k = 1$. However, for 2-strongly-connected directed graphs, the situation is much more interesting. It is not too difficult to prove that when the diameter of a 2-strongly-connected digraph is sufficiently large, it always contains a path of length $\text{diam}(G) + 1$. With much more careful arguments, it is possible to push this up to $k = 4$. Thus for each $k \leq 4$, the problem is solvable in polynomial time. For $k = 5$ we can construct a family of 2-strongly-connected digraphs of arbitrarily large diameter that do not have a path of length $\text{diam}(G) + 5$. These graphs become extremely useful as gadgets that we use to prove that the problem is NP-complete for each $k \geq 5$.

Related work. There is a vast literature in the field of parameterized complexity devoted to Longest Path [46, 9, 2, 36, 40, 13, 12, 41, 51, 26, 26, 8]. The surveys [25, 42] and the textbook [16, Chapter 10] provide an overview of the advances in the area.

Longest Detour was introduced by Bezáková et al. in [7]. They gave an FPT algorithm for undirected graphs and posed the question about detours in directed graphs. Even the existence of a polynomial time algorithm for Longest Detour with $k = 1$, that is, deciding whether a directed graph has a path longer than a shortest (s, t)-path, is open. For the related Exact Detour problem, deciding whether there is a detour of length *exactly* $\text{dist}_G(s, t) + k$ is FPT both on directed and undirected graphs [7].

Another problem related to our work is Long (s, t)-Path. Here for vertices s and t of a graph G, and integer parameter k, we have to decide whether there is an (s, t)-path in G of length at least k. A simple trick, see [16, Exercise 5.8], allows to use color-coding to show that Long (s, t)-Path is FPT on undirected graph. For directed graphs the situation is more involved, and the first FPT algorithm for Long (s, t)-Path on directed graphs was obtained only recently [28]. The proof of Theorem 4 uses some of the ideas developed in [28].

Both Longest Detour and Longest Path above Diameter fit into the research subarea of parameterized complexity called "above guarantee" parameterization [44, 1, 15, 31, 32, 33, 34, 35, 43, 45]. Besides the work of Bezáková et al. [6], several papers study parameterization of longest paths and cycles above different guarantees. Fomin et al. [23] designed parameterized algorithms for computing paths and cycles longer than the girth of a graph. The same set of the authors in [22] studied FPT algorithms that finds paths and cycles above degeneracy. Fomin et al. [24] developed an FPT algorithm computing a cycle of length $2\delta + k$, where δ is the minimum vertex degree of the input graph. Jansen, Kozma, and Nederlof in [37] looked at parameterized complexity of Hamiltonicity below Dirac's conditions. Berger, Seymour, and Spirkl in [5], gave a polynomial time algorithm that, with input a graph G and two vertices s, t of G, that decides whether there is an *induced* (s, t)-path that is longer than a shortest (s, t)-path. All these algorithms for computing long paths and cycles above some guarantee are for undirected graphs.

The remaining part of this paper is organized as follows. In Section 2, we give preliminaries. In Section 3, we prove our first main result establishing connections between 3-Disjoint Paths and Longest Detour (Theorem 4). Section 4 is devoted to Longest Path above Diameter. The concluding Section 5 provides open questions for further research.

2 Preliminaries

Parameterized Complexity. We refer to the recent books [16, 20] for the detailed introduction to Parameterized Complexity. Here we just remind that the computational complexity of an algorithm solving a parameterized problem is measured as a function of the input size n of a problem and an integer *parameter* k associated with the input. A parameterized problem is said to be *fixed-parameter tractable* (or FPT) if it can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some function $f(\cdot)$.

Graphs. Recall that an undirected graph is a pair $G = (V, E)$, where V is a set of vertices and E is a set of unordered pairs $\{u, v\}$ of distinct vertices called *edges*. A directed graph $G = (V, A)$ is a pair, where V is a set of vertices and A is a set of ordered pairs (u, v) of distinct vertices called *arcs*. Note that we do not allow loops and multiple edges or arcs. We use $V(G)$ and $E(G)$ ($A(G)$, respectively) to denote the set of vertices and the set of edges (set of arcs, respectively) of G. We write n and m to denote the number of vertices and edges (arcs, respectively) if this does not create confusion. For a (directed) graph G and a subset $X \subseteq V(G)$ of vertices, we write $G[X]$ to denote the subgraph of G induced by X. For a set of vertices S, $G - S$ denotes the (directed) graph obtained by deleting the vertices of S, that is, $G - S = G[V(G) \setminus S]$. We write $P = v_1 \cdots v_k$ to denote a *path* with the vertices $v_1, \ldots, v_k$ and the edges $\{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}$ (arcs $(v_1, v_2), \ldots, (v_{k-1}, v_k)$, respectively); v_1 and v_k are the *end-vertices* of P and the vertices $v_2, \ldots, v_{k-1}$ are *internal*. We say that P is an (v_1, v_k)*-path*. The *length* of P, denoted by $\text{length}(P)$, is the number of edges (arcs, respectively) in P. Two paths are *disjoint* if they have no common vertex and they are *internally disjoint* if no internal vertex of one path is a vertex of the other. For a (u, v)-path P_1 and a (v, w)-path P_2 that are internally disjoint, we denote by $P_1 \circ P_2$ the *concatenation* of P_1 and P_2. A vertex v is *reachable* from a vertex u in a (directed) graph G if G has a (u, v)-path. For $u, v \in V(G)$, $\text{dist}_G(u, v)$ denotes the *distance* between u and v in G, that is, the minimum number of edges (arcs, respectively) in an (u, v)-path. An undirected graph G is *connected* if for every two vertices u and v, G has a (u, v)-path. A directed graph G is *strongly-connected* if for every two vertices u and v both u is reachable form v and v is reachable from u. For a positive integer k, an undirected (directed, respectively) graph G is k*-connected* (k*-strongly-connected*, respectively) if $|V(G)| \geq k$ and $G - S$ is connected (strongly-connected, respectively) for every $S \subseteq V(G)$ of size at most $k - 1$. For a directed graph G, by G^T we denote the *transpose* of G, i.e. G^T is a directed graph defined on the same set of vertices and the same set of arcs, but the direction of each arc in G^T is reversed.

We use several known parameterized algorithms for finding long paths. First of all, let us recall the currently fastest deterministic algortihm for LONGEST PATH on directed graphs due to Tsur [50].

▶ **Proposition 1** ([50]). *There is a deterministic algorithm for* LONGEST PATH *with running time* $2.554^k \cdot n^{\mathcal{O}(1)}$.

We also need the result of Fomin et al. [28] for the LONG DIRECTED (s, t)-PATH problem. This problem asks, given a directed graph G, two vertices $s, t \in V(G)$, and an integer $k \geq 0$, whether G has an (s, t)-path of length at least k.

▶ **Proposition 2** ([28]). LONG DIRECTED (s, t)-PATH *can be deterministically solved in time* $4.884^k \cdot n^{\mathcal{O}(1)}$.

Clearly, both results holds for the variant of the problem on undirected graphs.

Finally, we use the result of Bezáková et al. [7] for the variant of LONGEST DETOUR whose task is, given a (directed) graph G, two vertices $s, t \in V(G)$, and an integer $k \geq 0$, decide whether G has an (s, t)-path of length *exactly* $\text{dist}_G(s, t) + k$.

▶ **Proposition 3** ([7]). *There is a bounded-error randomized algorithm that solves* EXACT DETOUR *on undirected graphs in time* $2.746^k \cdot n^{\mathcal{O}(1)}$ *and on directed graphs in time* $4^k \cdot n^{\mathcal{O}(1)}$. *For both undirected and directed graphs, there is a deterministic algorithm that runs in time* $6.745^k \cdot n^{\mathcal{O}(1)}$.

3 An FPT algorithm for finding detours

In this section we show the first main result of our paper.

▶ **Theorem 4.** *Let $\mathcal{C}$ be a class of directed graphs such that* 3-DISJOINT PATHS *can be solved in $f(n)$ time time on $\mathcal{C}$. Then* LONGEST DETOUR *can be solved in* $45.5^k \cdot n^{\mathcal{O}(1)} + \mathcal{O}(f(n)n^2)$ *time by a deterministic algorithm and in* $23.86^k \cdot n^{\mathcal{O}(1)} + \mathcal{O}(f(n)n^2)$ *time by a bounded-error randomized algorithm when the input is restricted to graphs from $\mathcal{C}$.*

Proof. Let (G, s, t, k) be an instance of LONGEST DETOUR with $G \in \mathcal{C}$. For $k = 0$, the problem is trivial and we assume that $k \geq 1$. We also have that (G, s, t, k) is a trivial no-instance if t is not reachable from s. We assume from now that every vertex of G is reachable from s. Otherwise, we set $G := G[R]$, where R is the set of vertices of G reachable from s using the straightforward property that every (s, t)-path in G is a path in $G[R]$. Clearly, R can be constructed in $\mathcal{O}(n + m)$ time by the breadth-first search.

Using Proposition 3, we check in $6.745^{2k} \cdot n^{\mathcal{O}(1)}$ time by a deterministic algorithm (in $4^{2k} \cdot n^{\mathcal{O}(1)}$ time by a randomized algorithm, respectively) whether G has an (s, t)-path of length $\text{dist}_G(s, t) + \ell$ for some $k \leq \ell \leq 2k - 1$ by trying all values of ℓ in this interval. We return a solution and stop if we discover such a path. Assume from now that this is not the case, that is, if (G, s, t) is a yes-instance, then the length of every (s, t)-path of length at least $\text{dist}_G(s, t) + k$ is at least $\text{dist}_G(s, t) + 2k$.

We perform the breadth-first search from s in G. For an integer $i \geq 0$, denote by L_i the set of vertices at distance i from s. Let ℓ be the maximum index such that $L_\ell \neq \emptyset$. Because every vertex of G is reachable from s, $V(G) = \bigcup_{i=0}^{\ell} L_i$. We call $L_0, \ldots, L_\ell$ *BFS-levels*.

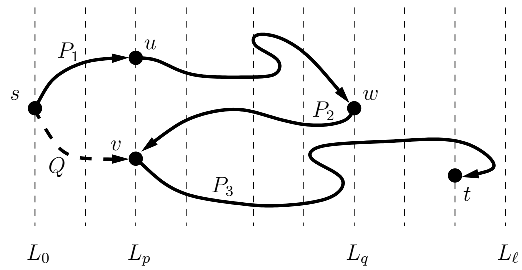

Figure 1 The choice of the BFS-levels L_p and L_q, vertices u, v, and w, and the paths P_1, P_2, and P_3.

Our algorithm is based on structural properties of potential solutions. Suppose that (G, s, t, k) is a yes-instance and let a path P be a solution of minimum length, that is, P is an (s, t)-path of length at least $\text{dist}_G(s, t) + k$ and among such paths the length of P is minimum. Denote by $p \in \{1, \ldots, \ell\}$ the minimum index such that L_p contains at least two vertices of G. Such an index exists, because if $|V(P) \cap L_i| \leq 1$ for all $i \in \{1, \ldots, \ell\}$, then P is a shortest (s, t)-path by the definition of $L_0, \ldots, L_\ell$ and the length of P is $\text{dist}_G(s, t) < \text{dist}_G(s, t) + k$ as $k \geq 1$. Let u be the first (in the path order) vertex of P in L_p and let $v \neq u$ be the second vertex of P that occurs in L_p. Denote by P_1, P_2, and P_3 the (s, u), (u, v), and (v, t)-subpath of P, respectively. Clearly, $P = P_1 \circ P_2 \circ P_3$. Let $q \in \{p, \ldots, \ell\}$ be the maximum index such that P_2 contains a vertex of L_q. Then denote by w the first vertex of P_2 in L_q. See Figure 1 for the illustration of the described configuration. We use this notation for a (hypothetical) solution throughout the proof of the theorem. The following claim is crucial for us.

▷ Claim 5. The length of P_2 is at least k.

Proof of Claim 5. For the sake of contradiction, assume that the length of P_2 is less than k. Let Q be a shortest (s, v)-path in G. By the definition of BFS-levels, $V(Q) \subseteq L_0 \cup \cdots \cup L_p$ and v is a unique vertex of Q in L_p. This implies that Q is internally vertex disjoint with P_3. Note that the length of Q is the same as the length of P_1, because P_1 contains exactly one vertex from each of the BFS levels $L_1, \ldots, L_p$. Then $P' = Q \circ P_3$ is an (s, t)-path and

$$\begin{aligned}\text{length}(P') =& \text{length}(Q) + \text{length}(P_3) = \text{length}(P_1) + \text{length}(P_3)\\ =& \text{length}(P) - \text{length}(P_2) \leq \text{length}(P) - k.\end{aligned}$$

Recall that the length of every (s,t)-path of length at least $\text{dist}_G(s,t) + k$ is at least $\text{dist}_G(s,t) + 2k$. This means that $\text{length}(P) \geq \text{dist}_G(s,t) + 2k$ and, therefore, the length of P' is at least $\text{dist}_G(s,t) + k$, that is, P' is a solution to the considered instance. However, $\text{length}(P') < \text{length}(P)$, because P_2 contains at least one arc. This contradicts the choice of P as a solution of minimum length. This completes the proof of the claim. ◁

By Claim 5, solving Longest Detour on (G,s,t,k) boils down to identifying internally disjoint P_1, P_2, and P_3, where the length of P_2 is at least k.

First, we check whether we can find paths for $q - p \geq k - 1$. Notice that if $q - p \geq k - 1$, then for every internally disjoint (s,w)-, (w,v)-, and (v,t)-paths R_1, R_2, and R_3 respectively, their concatenation $R_1 \circ R_2 \circ R_3$ is an (s,t)-path of length at least $\text{dist}_G(s,t) + k$. Recall that $G \in \mathcal{C}$ and p-Disjoint Paths can be solved in polynomial time on this graph class for $p = 3$. For every choice of two vertices $w, v \in V(G)$, we solve p-Disjoint Paths on the instance $(G, (s,w), (w,v), (v,s))$. Then if there are paths R_1, R_2, and R_3 forming a solution to this instance, we check whether $\text{length}(R_1) + \text{length}(R_2) + \text{length}(R_3) \geq \text{dist}_G(s,t) + k$. If this holds, we conclude that the path $R_1 \circ R_2 \circ R_3$ is a solution to the instance (G,s,t,k) of Longest Detour and return it. Assume from now that this is not the case, that is, we failed to find a solution of this type. Then we can complement Claim 5 by the following observation about our hypothetical solution P.

▷ Claim 6. $q - p \leq k - 2$.

This means that we can assume that $k \geq 2$ and have to check whether we can identify P_1, P_2, and P_3, where $V(P_2) \subseteq \bigcup_{i=p}^{p+k-2} L_i$. For this, we go over all possible choices of u. Note that the choice of u determines p, i.e., the index of the BFS-level containing u. We consider the following two cases for each considered choice of u.

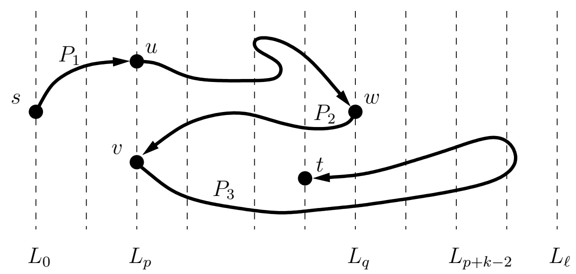

Figure 2 The structure of paths P_1, P_2, and P_3 in Case 1.

Case 1. $t \in L_r$ for some $p \leq r \leq p + k - 2$ (see Figure 2). Then $\text{dist}_G(s,t) = r$ and (G,s,t,k) is a yes-instance if and only if $G[L_p \cup \cdots \cup L_\ell]$ has a (u,t)-path S of length at least $(r-p)+k$, because the (s,u)-subpath of a potential solution should be a shortest (s,u)-path. Since $r - p \leq k - 2$, we have that $(r-p)+k \leq 2k-2$ and we can find S in $4.884^{2k} \cdot n^{\mathcal{O}(1)}$ time by Proposition 2 if it exists. If we obtain S, then we consider an arbitrary shortest (s,u)-path S' in G and conclude that $S' \circ S$ is a solution. This completes Case 1.

Case 2. $t \in L_r$ for some $r \geq p + k - 1$ (see Figure 3). We again consider our hypothetical solution $P = P_1 \circ P_2 \circ P_3$. Let $H = G[L_{p+k-1} \cup \cdots \cup L_\ell]$. Denote by X the set of vertices $x \in V(H)$ such that t is reachable from x in H. Denote by x the first vertex of P_3 in X. Clearly, such a vertex exists because $t \in X$. Moreover, $x \in L_{p+k-1}$ and its predecessor y in P_3 is in L_{p+k-2}. Otherwise, t would be reachable from $y \in V(H)$ in H contradicting the choice of x. Let Q_1 and Q_2 be the (v,y)- and (x,t)-subpaths of P_3. Then $P_3 = Q_1 \circ yx \circ Q_2$. We show one more claim about the hypothetical solution P.

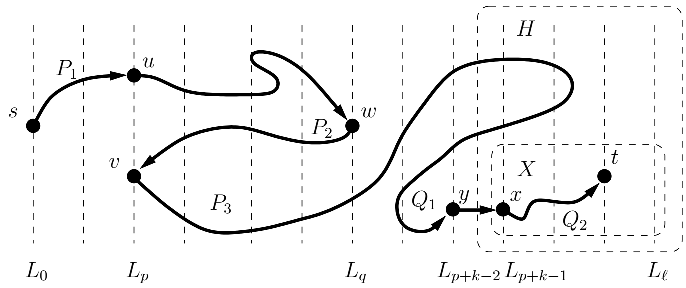

Figure 3 The structure of paths P_1, P_2, and P_3 in Case 2.

▷ Claim 7. $V(Q_1) \cap X = \emptyset$.

Proof of Claim 7. The proof is by contradiction. Assume that $z \in V(Q_1) \cap X$. Then t is reachable from z in H. However, x is the first vertex of P_3 with this property by the definition; a contradiction. ◁

Notice that because $x \in X$, there is an (x,t)-path Q_2' with $V(Q_2') \subseteq X$. By Claim 7, Q_1 and Q_2' are disjoint. Since $X \subseteq L_{p+k-1} \cup \cdots \cup L_\ell$, we have that $(V(P_1) \cup V(P_2)) \cap X = \emptyset$. In particular, Q_2' is disjoint with P_1 and P_2 as well. Let $P_3' = Q_1 \circ yx \circ Q_2'$. By Claim 5, $P' = P_1 \circ P_2 \circ P_3'$ is a solution, because length$(P_2) \geq k$. This allows us to conclude that (G,s,t,k) has a solution (for the considered choice of u) if and only if there is $y \in L_{p+k-2}$ such that

(i) there is $x \in X$ such that $(y,x) \in A(G)$, and
(ii) the graph $G[L_p \cup \cdots \cup L_\ell] - X$ has a (u,y)-path of length at least $2k-2$.

Our algorithm proceeds as follows. We construct the set X using the breadth-first search in $\mathcal{O}(n+m)$ time. Then for every $y \in L_{p+k-2}$ we check (i) whether there is $x \in X$ such that $(y,x) \in A(G)$, and (ii) whether $G[L_p \cup \cdots \cup L_\ell] - X$ has a (u,y)-path S of length at least $2k-2$. To verify (ii), we apply Proposition 2 that allows to perform the check in $4.884^{2k} \cdot n^{\mathcal{O}(1)}$ time. If we find such a vertex y and path S, then to obtain a solution, we consider an arbitrary shortest (s,u)-path S' and an arbitrary (x,t) path S'' in $G[X]$. Then $P' = S' \circ S \circ yx \circ S''$ is a required solution to (G,s,t,k). This concludes the analysis in Case 2 and the construction of the algorithm.

The correctness of our algorithm has been proved simultaneously with its construction. The remaining task is to evaluate the total running time. Recall that we verify in $6.745^{2k} \cdot n^{\mathcal{O}(1)}$ time whether G has an (s,t)-path of length $\text{dist}_G(s,t) + \ell$ for some $k \leq \ell \leq 2k-1$ by a deterministic algorithm, and we need $4^{2k} \cdot n^{\mathcal{O}(1)}$ time if we use a randomized algorithm. Then we construct the BFS-levels in linear time. Next, we consider $\mathcal{O}(n^2)$ choices of v and w and apply the algorithm for 3-Disjoint Paths $(G,(s,w),(w,v),(v,s))$ in $f(n)$ time. If we failed to find a solution so far, we proceed with $\mathcal{O}(n)$ possible choices of u and consider either Case 1 or 2 for each choice. In Case 1, we solve the problem in $4.884^{2k} \cdot n^{\mathcal{O}(1)}$ time. In Case 2, we construct X in $\mathcal{O}(n+m)$ time. Then for $\mathcal{O}(n)$ choices of y, we verify conditions (i) and (ii) in $4.884^{2k} \cdot n^{\mathcal{O}(1)}$ time. Summarizing, we obtain that the total running time is $6.745^{2k} \cdot n^{\mathcal{O}(1)} + \mathcal{O}(f(n)n^2)$. Because $6.745^2 < 45.5$, we have that the deterministic algorithm runs in $45.5^k \cdot n^{\mathcal{O}(1)} + \mathcal{O}(f(n)n^2)$ time. Since $4^2 < 4.884^2 < 23.86$, we conclude that the problem can be solved in $23.86^k \cdot n^{\mathcal{O}(1)} + \mathcal{O}(f(n)n^2)$ time by a bounded-error randomized algorithm. ◀

In particular, combining Theorem 4 with the results of Cygan et al. [17], we obtain the following corollary.

▶ **Corollary 8.** *Longest Detour can be solved in* $45.5^k \cdot n^{\mathcal{O}(1)}$ *time by a deterministic algorithm and in* $23.86^k \cdot n^{\mathcal{O}(1)}$ *time by a bounded-error randomized algorithm on planar directed graphs.*

Using the fact that p-Disjoint Paths can be solved in $\mathcal{O}(n^2)$ time by the results of Kawarabayashi, Kobayashi, and Reed [38], we immediately obtain the result for Longest Detour on undirected graphs. However, we can improve the running time of a randomized algorithm by tuning our algorithm for the undirected case.

▶ **Corollary 9.** *Longest Detour can be solved in* $45.5^k \cdot n^{\mathcal{O}(1)}$ *time by a deterministic algorithm and in* $10.8^k \cdot n^{\mathcal{O}(1)}$ *time by a bounded-error randomized algorithm on undirected graphs.*

Proof. The deterministic algorithm is the same as in the directed case. To obtain a better randomized algorithm, we follow the algorithm from Theorem 4 and use the notation introduced in its proof. Let (G, s, t, k) be an instance of Longest Detour with $G \in \mathcal{C}$. We assume without loss of generality that $k \geq 1$ and G is connected. Using Proposition 3, we check in $2.746^{2k} \cdot n^{\mathcal{O}(1)}$ time by a randomized algorithm whether G has an (s, t)-path of length $\text{dist}_G(s, t) + \ell$ for some $k \leq \ell \leq 2k - 1$. If we fail to find a solution this way, we construct the BFS-levels $L_0, \ldots, L_\ell$.

Suppose that (G, s, t, k) is a yes-instance with a hypothetical solution P composed by the concatenation of P_1, P_2, and P_3 as in the proof of Theorem 4. Let also L_p and L_q be the corresponding BFS-levels. Observe that if $q - p \geq k/2$, then $\text{length}(P_2) \geq k$, because for every edge $\{x, y\}$ of G, x and y are either in the same BFS-level or in consecutive levels contrary to the directed case where we may have an arc (x, y) where $x \in L_i$ and $y \in L_j$ for arbitrary $j \in \{0, \ldots, i\}$. Recall that for every choice of two vertices $w, v \in V(G)$, we solve p-Disjoint Paths on the instance $(G, (s, w), (w, v), (v, s))$ and try to find a solution to (G, s, t, k) by concatenating the solutions for these instances of p-Disjoint Paths. If we fail to find a solution this way, we can conclude now that $q - p \leq k/2 - 1$ improving Claim 6. Further, we pick u and consider two cases.

In Case 1, where $t \in L_r$ for some $p \leq r \leq p + k/2 - 1$, we now find a (u, t)-path S in $G[L_p \cup \cdots \cup L_\ell]$ of length at least $(r - p) + k \leq 3k/2$ in $4.884^{3k/2} \cdot n^{\mathcal{O}(1)}$ time. If such a path exists, we obtain a solution.

In Case 2, where $t \in L_r$ for some $r \geq p + k/2$, we consider $H = G[L_{h+1} \cup \cdots \cup L_\ell]$ for $h = p + \lceil k/2 \rceil$ and denote by X the set of vertices of the connected component of H containing X. Then for every $y \in L_h$ we check (i) whether there is $x \in X$ such that $\{y, x\} \in E(G)$, and (ii) whether $G[L_p \cup \cdots \cup L_\ell] - X$ has a (u, y)-path S of length at least $k + \lceil k/2 \rceil$ in $4.884^{3k/2} \cdot n^{\mathcal{O}(1)}$ time. If such a path exists, we construct a solution containing it in the same way as on the directed case.

The running time analysis is essentially the same as in the proof of Theorem 4. The difference is that now we have that $2.746^2 \leq 4.884^{3/2} < 10.80$. This implies that the algorithm runs in $10.8^k \cdot n^{\mathcal{O}(1)}$ time. ◀

4 Longest Path Above Diameter

In this section, we investigate the complexity of Longest Path above Diameter. It can be noted that this problem is NP-complete in general even for $k = 1$.

▶ **Proposition 10.** *Longest Path above Diameter is* NP*-complete for* $k = 1$ *on undirected graphs.*

Proof. Let G be an undirected graph with $n \geq 2$ vertices. We construct the graph G' as follows (see Figure 4).

- Construct a copy of G.
- Construct a vertex u and make it adjacent to every vertex of the copy of G.
- Construct two vertices s and t, and then (s, u) and (u, t) paths P_s and P_t, respectively, of length $n - 1$.

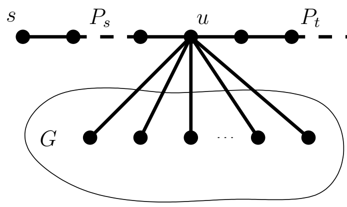

Figure 4 Construction of G'.

Notice that $\text{diam}(G) = \text{length}(P_s) + \text{length}(P_t) = 2n - 2$. It is easy to verify that G' has a path of length $2n - 1$ if and only if G has a path of length $n - 1$, that is, G is Hamiltonian. Because Hamiltonian Path is well-known to be NP-complete [30], we conclude that Longest Path above Diameter is NP-complete for $k = 1$ ◀

Proposition 10 immediately implies that Longest Path above Diameter is NP-complete for $k = 1$ on strongly connected directed graphs as we can reduce the problem on undirected graphs to the directed variant by replacing each edge by the pair of arcs with opposite orientations. Still, it can be observed that the reduction in Proposition 10 strongly relies on the fact that the constructed graph G' has an articulation point u. Hence, it is natural to investigate the problem further imposing connectivity constraints on the input graphs. And indeed, it can be easily seen that Longest Path above Diameter is FPT on 2-connected undirected graphs.

▶ **Observation 11.** *Longest Path above Diameter can be solved in time* $6.523^k \cdot n^{\mathcal{O}(1)}$ *on undirected 2-connected graphs.*

Proof. Let (G, k) be an instance of Longest Path above Diameter where G is 2-connected. If $d = \text{diam}(G) \leq k$, we can solve the problem in time $2.554^{d+k} \cdot n^{\mathcal{O}(1)}$ by using the algorithm of Proposition 1 to check whether G has a path of length $d + k$. Note that $2.554^{d+k} \leq 2.554^{2k} \leq 6.523^k$. Otherwise, if $d > k$, consider a pair of vertices s and t with $\text{dist}_G(s, t) = d$. Because G is 2-connected, by Menger's theorem (see, e.g., [19]), G has a cycle C containing s and t. Since $\text{dist}_G(s, t) = d$ and $d \geq k + 1$, the length of C is at least $d + k + 1$. This implies that C contains a path of length $d + k$. ◀

However, the arguments from the proof of Observation 11 cannot be translated to directed graphs. In particular, if a directed graph G is 2-strongly-connected, it does not mean that for every two vertices u and v, G has a cycle containing u and v. We show the following theorem providing a full dichotomy for the complexity of Longest Path above Diameter on 2-strongly-connected graphs.

▶ **Theorem 12.** *On 2-strongly-connected directed graphs, Longest Path above Diameter with* $k \leq 4$ *can be solved in polynomial time, while for* $k \geq 5$ *it is* NP*-complete.*

In what remains of this section, we give some intuition behind the proof of Theorem 12; the details can be found in the full version of the paper [21]. To show the positive part of the theorem, it is sufficient to consider graphs with diameter greater than some fixed constant, because in graphs with smaller diameter the problem can be solved in linear time. For graphs with a sufficiently large diameter, we show that a path of length diameter plus four always exists. To construct such a path, we take the diameter pair (s, t) and employ 2-strong-connectivity of the graph to find two disjoint (s, t)-paths and two disjoint (t, s)-paths in the graph. We then show that out of the several possible ways to comprise a path out of the parts of these four paths, at least one always obtains a path of desired length. The most non-trivial case of this construction involves constructing two paths of length five, one ending in a vertex u that is at distance three from s and the other starting in a vertex v from which we can reach t using three arcs. We then concatencate these two paths using a specific (u, v)-path inbetween. Since (s, t) is a diameter pair, the length of any (u, v)-path is at least diameter minus six, so the length of the concatenation is at least diameter plus four. The other cases are analyzed in a similar fashion.

For the lower bound part of Theorem 12, the general idea of the proof is similar to that of Proposition 10. We aim to take a path-like gadget graph, then take a sufficiently large Hamiltonian Path instance and connect it to the middle of the gadget. However, while in the general case it suffices to simply take a path graph (Proposition 10), the 2-strongly-connected case is much more technically involved. First, we need a family of gadget graphs that are 2-strongly-connected, have arbitrarily large diameter, but each graph in the family does not have a path longer than diameter plus four. This, in fact, is exactly a counterexample to the positive part of Theorem 12, as the existence of such family of graphs proves that there cannot always be a path of length diameter plus four in a sufficiently large 2-connected directed graph. Additionally, for the reduction we need that graphs in this family behave like paths, specifically that the length of the longest path that ends in the "middle" of the gadget is roughly half of the diameter. Constructing this graph family is a main technical challenge of the theorem. After constructing the gadget graph family the proof is reasonably simple, as we take a 2-connected Hamiltonian Path instance, and connect it to the "middle" of a sufficiently large gadget graph. The connection is done by a simple 4-vertex connector gadget that ensures that the resulting graph is 2-strongly-connected, but only allows for paths that alternate at most once between the gadget graph and the starting instance.

5 Conclusion

We proved that if $\mathcal{C}$ is a class of directed graph such that p-Disjoint Paths is in P on $\mathcal{C}$ for $p = 3$, then Longest Detour is FPT on $\mathcal{C}$. However p-Disjoint Paths is NP-complete on directed graphs for every fixed $p \geq 2$ [29]. This leaves open the question of Bezáková et al. [7] about parameterized complexity of Longest Detour on general directed graphs. Even the complexity (P versus NP) of deciding whether a directed graph contains an (s, t)-path longer than $\mathrm{dist}_G(s, t)$ (the case of $k = 1$) remains open. Notice that Longest Detour is not equivalent to p-Disjoint Paths for $p = 3$ and, therefore, the hardness of p-Disjoint Paths does not imply hardness of Longest Detour.

Our result implies, in particular, that Longest Detour is FPT on planar directed graphs. There are various classes of directed graphs on which p-Disjoint Paths is tractable for fixed p (see, e.g., the book of Bang-Jensen and Gutin [3]). For example, by Chudnovsky, Scott, and Seymour [14], p-Disjoint Paths can be solved in polynomial time for every fixed

p on semi-complete directed graphs. Together with Theorem 4, it implies that LONGEST DETOUR is FPT on semi-complete directed graphs and tournaments. However, from what we know, these results could be too weak in the following sense. Using the structural results of Thomassen [49], Bang-Jensen, Manoussakis, and Thomassen in [4] gave a polynomial-time algorithm to decide whether a semi-complete directed graph has a Hamiltonian (s, t)-path for two given vertices s and t. Thus the real question is whether LONGEST DETOUR is in P on semi-complete directed graphs or tournaments.

The second part of our results is devoted to LONGEST PATH ABOVE DIAMETER. We proved that this problem is NP-complete for general graphs for $k = 1$ and showed that it is in FPT when the input graph is undirected and 2-connected. We established the complexity dichotomy for LONGEST PATH ABOVE DIAMETER for the case of 2-strongly-connected directed graphs by showing that the problem can be solved in polynomial time for $k \leq 4$ and is NP-complete for $k \geq 5$. This naturally leaves an open question for larger values of strong connectivity. The computational complexity of LONGEST PATH ABOVE DIAMETER on t-strongly connected graphs for $t \geq 3$ is open. For a very concrete question, is there a polynomial algorithm for LONGEST PATH ABOVE DIAMETER with $k = 5$ on graphs of strong connectivity 3?

References

1 Noga Alon, Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving MAX-r-SAT above a tight lower bound. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 511–517. SIAM, 2010.

2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

3 Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs - Theory, Algorithms and Applications, Second Edition*. Springer Monographs in Mathematics. Springer, 2009.

4 Jørgen Bang-Jensen, Yannis Manoussakis, and Carsten Thomassen. A polynomial algorithm for hamiltonian-connectedness in semicomplete digraphs. *J. Algorithms*, 13(1):114–127, 1992. doi:10.1016/0196-6774(92)90008-Z.

5 Eli Berger, Paul Seymour, and Sophie Spirkl. Finding an induced path that is not a shortest path, 2020. arXiv:2005.12861.

6 Ivona Bezáková, Radu Curticapean, Holger Dell, and Fedor V. Fomin. Finding detours is fixed-parameter tractable. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 80 of *LIPIcs*, pages 54:1–54:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.54.

7 Ivona Bezáková, Radu Curticapean, Holger Dell, and Fedor V. Fomin. Finding detours is fixed-parameter tractable. *SIAM J. Discret. Math.*, 33(4):2326–2345, 2019. doi:10.1137/17M1148566.

8 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *Journal of Computer and System Sciences*, 87:119–139, 2017. doi:10.1016/j.jcss.2017.03.003.

9 Hans L. Bodlaender. On linear time minor tests with depth-first search. *Journal of Algorithms*, 14(1):1–23, 1993. doi:10.1006/jagm.1993.1001.

10 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

11 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016. doi:10.1137/130947374.

12 Jianer Chen, Joachim Kneis, Songjian Lu, Daniel Mölle, Stefan Richter, Peter Rossmanith, Sing-Hoi Sze, and Fenghui Zhang. Randomized divide-and-conquer: Improved path, matching, and packing algorithms. *SIAM Journal on Computing*, 38(6):2526–2547, 2009. doi:10.1137/080716475.

13 Jianer Chen, Songjian Lu, Sing-Hoi Sze, and Fenghui Zhang. Improved algorithms for path, matching, and packing problems. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 298–307. SIAM, 2007.

14 Maria Chudnovsky, Alex Scott, and Paul D. Seymour. Excluding pairs of graphs. *J. Comb. Theory, Ser. B*, 106:15–29, 2014. doi:10.1016/j.jctb.2014.01.001.

15 Robert Crowston, Mark Jones, Gabriele Muciaccia, Geevarghese Philip, Ashutosh Rai, and Saket Saurabh. Polynomial kernels for lambda-extendible properties parameterized above the Poljak-Turzik bound. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 24 of *LIPIcs*, pages 43–54. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. doi:10.4230/LIPIcs.FSTTCS.2013.43.

16 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

17 Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. The planar directed k-vertex-disjoint paths problem is fixed-parameter tractable. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 197–206. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.29.

18 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 150–159. IEEE, 2011.

19 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

20 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

21 Fedor V. Fomin, Petr A. Golovach, William Lochet, Danil Sagunov, Kirill Simonov, and Saket Saurabh. Detours in directed graphs. *CoRR*, abs/2201.03318, 2022. arXiv:2201.03318.

22 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Going far from degeneracy. *SIAM J. Discret. Math.*, 34(3):1587–1601, 2020. doi:10.1137/19M1290577.

23 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Parameterization Above a Multiplicative Guarantee. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.39.

24 Fedor V. Fomin, Petr A. Golovach, Danil Sagunov, and Kirill Simonov. Algorithmic extensions of Dirac's Theorem. *CoRR*, abs/2011.03619, 2020. arXiv:2011.03619.

25 Fedor V. Fomin and Petteri Kaski. Exact exponential algorithms. *Communications of the ACM*, 56(3):80–88, 2013. doi:10.1145/2428556.2428575.

26 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *Journal of ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.

27 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Representative families of product families. *ACM Trans. Algorithms*, 13(3):36:1–36:29, 2017. doi:10.1145/3039243.

28 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Long directed (s, t)-path: FPT algorithm. *Inf. Process. Lett.*, 140:8–12, 2018. doi:10.1016/j.ipl.2018.04.018.

29 Steven Fortune, John E. Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980. doi:10.1016/0304-3975(80)90009-2.

30 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

31 Shivam Garg and Geevarghese Philip. Raising the bar for vertex cover: Fixed-parameter tractability above a higher guarantee. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1152–1166. SIAM, 2016. doi:10.1137/1.9781611974331.ch80.

32 Gregory Gutin, Eun Jung Kim, Michael Lampis, and Valia Mitsou. Vertex cover problem parameterized above and below tight bounds. *Theory of Computing Systems*, 48(2):402–410, 2011. doi:10.1007/s00224-010-9262-y.

33 Gregory Gutin, Leo van Iersel, Matthias Mnich, and Anders Yeo. Every ternary permutation constraint satisfaction problem parameterized above average has a kernel with a quadratic number of variables. *Journal of Computer and System Sciences*, 78(1):151–163, 2012. doi:10.1016/j.jcss.2011.01.004.

34 Gregory Z. Gutin and Viresh Patel. Parameterized traveling salesman problem: Beating the average. *SIAM J. Discrete Math.*, 30(1):220–238, 2016.

35 Gregory Z. Gutin, Arash Rafiey, Stefan Szeider, and Anders Yeo. The linear arrangement problem parameterized above guaranteed value. *Theory Comput. Syst.*, 41(3):521–538, 2007. doi:10.1007/s00224-007-1330-6.

36 Falk Hüffner, Sebastian Wernicke, and Thomas Zichner. Algorithm engineering for color-coding with applications to signaling pathway detection. *Algorithmica*, 52(2):114–132, 2008. doi:10.1007/s00453-007-9008-7.

37 Bart M. P. Jansen, László Kozma, and Jesper Nederlof. Hamiltonicity below Dirac's condition. In *Proceedings of the 45th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 11789 of *Lecture Notes in Computer Science*, pages 27–39. Springer, 2019.

38 Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce A. Reed. The disjoint paths problem in quadratic time. *J. Comb. Theory, Ser. B*, 102(2):424–435, 2012. doi:10.1016/j.jctb.2011.07.004.

39 Ken-ichi Kawarabayashi and Stephan Kreutzer. The directed grid theorem. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 655–664. ACM, 2015. doi:10.1145/2746539.2746586.

40 Joachim Kneis, Daniel Mölle, Stefan Richter, and Peter Rossmanith. Divide-and-color. In *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 4271 of *Lecture Notes in Computer Science*, pages 58–67. Springer, 2006. doi:10.1007/11917496_6.

41 Ioannis Koutis. Faster algebraic algorithms for path and packing problems. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5125 of *Lecture Notes in Computer Science*, pages 575–586. Springer, 2008. doi:10.1007/978-3-540-70575-8_47.

42 Ioannis Koutis and Ryan Williams. Algebraic fingerprints for faster algorithms. *Communications of the ACM*, 59(1):98–105, 2016. doi:10.1145/2742544.

43 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014. doi:10.1145/2566616.

44 Meena Mahajan and Venkatesh Raman. Parameterizing above guaranteed values: MaxSat and MaxCut. *Journal of Algorithms*, 31(2):335–354, 1999. doi:10.1006/jagm.1998.0996.

45 Meena Mahajan, Venkatesh Raman, and Somnath Sikdar. Parameterizing above or below guaranteed values. *Journal of Computer and System Sciences*, 75(2):137–153, 2009. doi:10.1016/j.jcss.2008.08.004.

46 Burkhard Monien. How to find long paths efficiently. In *Analysis and design of algorithms for combinatorial problems*, volume 109 of *North-Holland Math. Stud.*, pages 239–254. North-Holland, Amsterdam, 1985. doi:10.1016/S0304-0208(08)73110-4.

47 Neil Robertson and Paul D. Seymour. Graph minors. XIII. the disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006.

48 Alexander Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM J. Comput.*, 23(4):780–788, 1994. doi:10.1137/S0097539792224061.

49 Carsten Thomassen. Hamiltonian-connected tournaments. *J. Comb. Theory, Ser. B*, 28(2):142–163, 1980. doi:10.1016/0095-8956(80)90061-1.

50 Dekel Tsur. Faster deterministic parameterized algorithm for k-path. *Theor. Comput. Sci.*, 790:96–104, 2019. doi:10.1016/j.tcs.2019.04.024.

51 Ryan Williams. Finding paths of length k in $O^*(2^k)$ time. *Information Processing Letters*, 109(6):315–318, 2009. doi:10.1016/j.ipl.2008.11.004.