Going Far From Degeneracy

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway
fomin@ii.uib.no

Petr A. Golovach
Department of Informatics, University of Bergen, Norway
Petr.Golovach@uib.no

Daniel Lokshtanov
Department of Computer Science, University of California Santa Barbara, USA
daniello@ucsb.edu

Fahad Panolan
Department of Computer Science and Engineering, IIT Hyderabad, India
fahad@iith.ac.in

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India
saket@imsc.res.in

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

Abstract

An undirected graph G is *d-degenerate* if every subgraph of G has a vertex of degree at most d. By the classical theorem of Erdős and Gallai from 1959, every graph of degeneracy $d > 1$ contains a cycle of length at least $d + 1$. The proof of Erdős and Gallai is constructive and can be turned into a polynomial time algorithm constructing a cycle of length at least $d + 1$. But can we decide in polynomial time whether a graph contains a cycle of length at least $d + 2$? An easy reduction from Hamiltonian Cycle provides a negative answer to this question: Deciding whether a graph has a cycle of length at least $d + 2$ is NP-complete. Surprisingly, the complexity of the problem changes drastically when the input graph is 2-connected. In this case we prove that deciding whether G contains a cycle of length at least $d + k$ can be done in time $2^{\mathcal{O}(k)}|V(G)|^{\mathcal{O}(1)}$. In other words, deciding whether a 2-connected n-vertex G contains a cycle of length at least $d + \log n$ can be done in polynomial time. Similar algorithmic results hold for long paths in graphs. We observe that deciding whether a graph has a path of length at least $d + 1$ is NP-complete. However, we prove that if graph G is connected, then deciding whether G contains a path of length at least $d + k$ can be done in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$. We complement these results by showing that the choice of degeneracy as the "above guarantee parameterization" is optimal in the following sense: For any $\varepsilon > 0$ it is NP-complete to decide whether a connected (2-connected) graph of degeneracy d has a path (cycle) of length at least $(1 + \varepsilon)d$.

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

Keywords and phrases Longest path, longest cycle, fixed-parameter tractability, above guarantee parameterization

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.47

Related Version The full version of the paper is available at `https://arxiv.org/abs/1902.02526`.

Funding The research leading to these results has received funding from the Research Council of Norway via the projects "CLASSIS" and "MULTIVAL" and from the European Research Council (ERC) via grant LOPPRE, reference 819416.

27th Annual European Symposium on Algorithms (ESA 2019).
Editors: Michael A. Bender, Ola Svensson, and Grzegorz Herman; Article No. 47; pp. 47:1–47:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements We thank Nikolay Karpov for communicating to us the question of finding a path above the degeneracy bound and Proposition 15.

1 Introduction

The classical theorem of Erdős and Gallai [11] says that

▶ **Theorem 1** (Erdős and Gallai [11]). *Every graph with n vertices and more than $(n-1)\ell/2$ edges ($\ell \geq 2$) contains a cycle of length at least $\ell+1$.*

Recall that a graph G is *d-degenerate* if every subgraph H of G has a vertex of degree at most d, that is, the minimum degree $\delta(H) \leq d$. Respectively, the *degeneracy* of graph G, is $\mathrm{dg}(G) = \max\{\delta(H) \mid H \text{ is a subgraph of } G\}$. Since a graph of degeneracy d has a subgraph H with at least $d \cdot |V(H)|/2$ edges, by Theorem 1, it contains a cycle of length at least $d+1$. Let us note that the degeneracy of a graph can be computed in polynomial time, see e.g. [28], and thus by Theorem 1, deciding whether a graph has a cycle of length at least $d+1$ can be done in polynomial time. In this paper we revisit this classical result from the algorithmic perspective.

We define the following problem.

Longest Cycle Above Degeneracy

Input:	A graph G and a positive integer k.
Task:	Decide whether G contains a cycle of length at least $\mathrm{dg}(G)+k$.

Let us first sketch why Longest Cycle Above Degeneracy is NP-complete for $k=2$ even for connected graphs. We can reduce Hamiltonian Cycle to Longest Cycle Above Degeneracy with $k=2$ as follows. For a connected non-complete graph G on n vertices, we construct connected graph H from G and a complete graph K_{n-1} on $n-1$ vertices as follows. We identify one vertex of G with one vertex of K_{n-1}. Thus the obtained graph H has $|V(G)|+n-2$ vertices and is connected; its degeneracy is $n-2$. Then H has a cycle with $\mathrm{dg}(H)+2=n$ vertices if and only if G has a Hamiltonian cycle.

Interestingly, when the input graph is 2-connected, the problem becomes fixed-parameter tractable being parameterized by k. Let us recall that a connected graph G is (vertex) 2-connected if for every $v \in V(G)$, $G-v$ is connected. Our first main result is the following theorem.

▶ **Theorem 2.** *On 2-connected graphs* Longest Cycle Above Degeneracy *is solvable in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.*

Similar results can be obtained for paths. Of course, if a graph contains a cycle of length $d+1$, it also contains a simple path on $d+1$ vertices. Thus for every graph G of degeneracy d, deciding whether G contains a path on $\mathrm{dg}(G)+1$ vertices can be done in polynomial time. Again, it is easy to show that it is NP-complete to decide whether G contains a path with $d+2$ vertices by a reduction from Hamiltonian Path. The reduction is very similar to the one we sketched for Longest Cycle Above Degeneracy. The only difference that this time graph H consists of a disjoint union of G and K_{n-1}. The degeneracy of H is $d=n-2$, and H has a path with $d+2=n$ vertices if and only if G contains a Hamiltonian path. Note that graph H used in the reduction is not connected. However, when the input graph G is connected, the complexity of the problem changes drastically. We define

LONGEST PATH ABOVE DEGENERACY

Input: A graph G and a positive integer k.
Task: Decide whether G contains a path with at least $\mathrm{dg}(G) + k$ vertices.

The second main contribution of our paper is the following theorem.

▶ **Theorem 3.** *On connected graphs* LONGEST PATH ABOVE DEGENERACY *is solvable in time* $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

Let us remark that Theorem 2 does not imply Theorem 3, because Theorem 2 holds only for 2-connected graphs.

We also show that the parameterization lower bound $\mathrm{dg}(G)$ that is used in Theorems 2 and 3 is tight in some sense. We prove that for any $0 < \varepsilon < 1$, it is NP-complete to decide whether a connected graph G contains a path with at least $(1 + \varepsilon)\mathrm{dg}(G)$ vertices and it is NP-complete to decide whether a 2-connected graph G contains a cycle with at least $(1 + \varepsilon)\mathrm{dg}(G)$ vertices.

Related work. HAMILTONIAN PATH and HAMILTONIAN CYCLE problems are among the oldest and most fundamental problems in Graph Theory. In parameterized complexity the following generalizations of these problems, LONGEST PATH and LONGEST CYCLE, were heavily studied. The LONGEST PATH problem is to decide, given an n-vertex (di)graph G and an integer k, whether G contains a path of length at least k. Similarly, the LONGEST CYCLE problem is to decide whether G contains a cycle of length at least k. There is a plethora of results about parameterized complexity (we refer to the book of Cygan at al. [9] for the introduction to the field) of LONGEST PATH and LONGEST CYCLE (see, e.g., [4, 5, 7, 6, 12, 14, 22, 23, 24, 32]) since the early work of Monien [29]. The fastest known randomized algorithm for LONGEST PATH on undirected graph is due to Björklund et al. [4] and runs in time $1.657^k \cdot n^{\mathcal{O}(1)}$. On the other hand very recently, Tsur gave the fastest known deterministic algorithm for the problem running in time $2.554^k \cdot n^{\mathcal{O}(1)}$ [31]. Respectively for LONGEST CYCLE, the current fastest randomized algorithm running in time $4^k \cdot n^{\mathcal{O}(1)}$ was given by Zehavi in [33] and the best deterministic algorithm constructed by Fomin et al. in [13] runs in time $4.884^k \cdot n^{\mathcal{O}(1)}$.

Our theorems about LONGEST PATH ABOVE DEGENERACY and LONGEST CYCLE ABOVE DEGENERACY fit into an interesting trend in parameterized complexity called "above guarantee" parameterization. The general idea of this paradigm is that the natural parameterization of, say, a maximization problem by the solution size is not satisfactory if there is a lower bound for the solution size that is sufficiently large. For example, there always exists a satisfying assignment that satisfies half of the clauses or there is always a max-cut containing at least half the edges. Thus nontrivial solutions occur only for the values of the parameter that are above the lower bound. This indicates that for such cases, it is more natural to parameterize the problem by the difference of the solution size and the bound. The first paper about above guarantee parameterization was due to Mahajan and Raman [26] who applied this approach to the MAX SAT and MAX CUT problem. This approach was successfully applied to various problems, see e.g. [1, 8, 16, 17, 18, 19, 20, 25, 27].

For LONGEST PATH, the only successful above guarantee parameterization known prior to our work was parameterization above shortest path. More precisely, let s, t be vertices of an undirected graph G. Clearly, the length of any (s, t)-path in G is lower bounded by the shortest distance, $d(s, t)$, between these vertices. Based on this observation, Bezáková et al. in [3] introduced the LONGEST DETOUR problem that asks, given a graph G, two vertices

s, t, and a positive integer k, whether G has an (s,t)-path with at least $d(s,t)+k$ vertices. They proved that for undirected graphs, this problem can be solved in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. On the other hand, the parameterized complexity of Longest Detour on directed graphs is still open. For the variant of the problem where the question is whether G has an (s,t)-path with *exactly* $d(s,t)+k$ vertices, a randomized algorithm with running time $2.746^k \cdot n^{\mathcal{O}(1)}$ and a deterministic algorithm with running time $6.745^k \cdot n^{\mathcal{O}(1)}$ were obtained [3]. These algorithms work for both undirected and directed graphs. Parameterization above degeneracy is "orthogonal" to the parameterization above the shortest distance. There are classes of graphs, like planar graphs, that have constant degeneracy and arbitrarily large diameter. On the other hand, there are classes of graphs, like complete graphs, of constant diameter and unbounded degeneracy.

Our approach. Our algorithmic results are based on classical theorems of Dirac [10], and Erdős and Gallai [11] on the existence of "long cycle" and "long paths" and can be seen as non-trivial algorithmic extensions of these classical theorems. Let $\delta(G)$ be the minimum vertex degree of graph G.

▶ **Theorem 4** (Dirac [10]). *Every n-vertex 2-connected graph G with minimum vertex degree $\delta(G) \geq 2$, contains a cycle with at least $\min\{2\delta(G), n\}$ vertices.*

▶ **Theorem 5** (Erdős and Gallai [11]). *Every connected n-vertex graph G contains a path with at least $\min\{2\delta(G)+1, n\}$ vertices.*

Theorem 4 is used to prove Theorem 2 and Theorem 5 is used to prove Theorem 3.

We give a high-level overview of the ideas used to prove Theorem 2. The ideas behind the proof of Theorem 3 are similar. Let G be a 2-connected graph of degeneracy d. If $d = \mathcal{O}(k)$, we can solve Longest Cycle Above Degeneracy in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ by making use of one of the algorithms for Longest Cycle. Assume from now that $d \geq c \cdot k$ for some constant c, which will be specified in the proof. Then we find a d-core H of G (a connected subgraph of G with the minimum vertex degree at least d). This can be done in linear time by one of the known algorithms, see e.g. [28]. If the size of H is sufficiently large, say $|V(H)| \geq d+k$, we use Theorem 4 to conclude that H contains a cycle with at least $|V(H)| \geq d+k$ vertices.

The most interesting case occurs when $|V(H)| < d+k$. Suppose that G has a cycle of length at least $d+k$. It is possible to prove that there is also a cycle of length at least $d+k$ that hits the core H. We do not know how many times and in which vertices of H this cycle enters and leaves H, but we can guess these terminal points. The interesting property of the core H is that, loosely speaking, for any "small" set of terminal points, inside H the cycle can be rerouted in such a way that it will contain all vertices of H.

A bit more formally, we prove the following structural result. We define a system of segments in G with respect to $V(H)$, which is a family of internally vertex-disjoint paths $\{P_1, \ldots, P_r\}$ in G (see Figure 1). Moreover, for every $1 \leq i \leq r$, every path P_i has at least 3 vertices, its endpoints are in $V(H)$ and all internal vertices of P_i are in $V(G) \setminus V(H)$. Also the union of all the segments is a forest with every connected component being a path.

We prove that G contains a cycle of length at least $k+d$ if and only if

- either there is a path of length at least $k+d-|V(H)|$ with endpoints in $V(H)$ and all internal vertices outside H, or
- there is a system of segments with respect to $V(H)$ such that the total number of vertices outside H used by the paths of the system, is within the interval $[k+d-|V(H)|, 2\cdot(k+d-|V(H)|)]$.

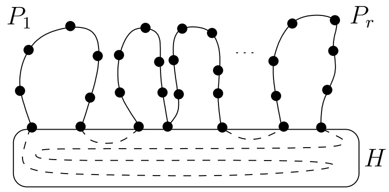

Figure 1 Reducing Longest Cycle Above Degeneracy to finding a system of segments $P_1, \dots, P_r$; complementing the segments into a cycle is shown by dashed lines.

The proof of this structural result is built on Lemma 8, which describes the possibility of routing in graphs of large minimal degree. The crucial property is that we can complement any system of segments of bounded size by segments inside the core H to obtain a cycle that contains all the vertices of H as is shown in Figure 1.

Since $|V(H)| > d$, the problem of finding a cycle of length at least $k+d$ in G boils down to one of the following tasks. Either find a path of length $c' \cdot k$ with all internal vertices outside H, or find a system of segments with respect to $V(H)$ such that the total number of vertices used by the paths of the system is $c'' \cdot k$, here c' and c'' are the constants to be specified in the proof. In the first case, we can use one of the known algorithms to find in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ such a long path. In the second case, we can use color-coding to solve the problem.

Organization of the paper. In Section 2 we give basic definitions and state some known fundamental results. Sections 3–4 contain the proof of Theorems 3 and 2. In Section 3 we state structural results that we need for the proofs and in Section 4 we complete the proofs. In Section 5, we give the complexity lower bounds for our algorithmic results. We conclude the paper in Section 6 by stating some open problems.

2 Preliminaries

We consider only finite undirected graphs. For a graph G, we use $V(G)$ and $E(G)$ to denote its vertex set and edge set, respectively. Throughout the paper we use $n = |V(G)|$ and $m = |E(G)|$. For a graph G and a subset $U \subseteq V(G)$ of vertices, we write $G[U]$ to denote the subgraph of G induced by U. We write $G - U$ to denote the graph $G[V(G) \setminus U]$; for a single-element set $U = \{u\}$, we write $G - u$. For a vertex v, we denote by $N_G(v)$ the *(open) neighborhood* of v, i.e., the set of vertices that are adjacent to v in G. For a set $U \subseteq V(G)$, $N_G(U) = (\bigcup_{v \in U} N_G(v)) \setminus U$. The *degree* of a vertex v is $d_G(v) = |N_G(v)|$. The *minimum degree* of G is $\delta(G) = \min\{d_G(v) \mid v \in V(G)\}$. A *$d$-core* of G is an inclusion maximal induced connected subgraph H with $\delta(H) \geq d$. Every graph of degeneracy at least d contains a d-core and that can be found in linear time (see [28]). A vertex u of a connected graph G with at least two vertices is a *cut vertex* if $G - u$ is disconnected. A connected graph G is 2-*connected* if it has no cut vertices. An inclusion maximal induced 2-connected subgraph of G is called a *biconnected component* or *block*. Let $\mathcal{B}$ be the set of blocks of a connected graph G and let C be the set of cut vertices. Consider the bipartite graph $Block(G)$ with the vertex set $\mathcal{B} \cup C$, where $(\mathcal{B}, C)$ is the bipartition, such that $B \in \mathcal{B}$ and $c \in C$ are adjacent if and only if $c \in V(B)$. The block graph of a connected graph is always a tree (see [21]).

A path in a graph is a self-avoiding walk. Thus no vertex appears in a path more than once. A cycle is a closed self-avoiding walk . For a path P with end-vertices s and t, we say that the vertices of $V(P) \setminus \{s, t\}$ are *internal*. We say that G is a *linear forest* if each component of G is a path. The *contraction* of an edge xy is the operation that removes the vertices x and y together with the incident edges and replaces them by a vertex u_{xy} that is adjacent to the vertices of $N_G(\{x,y\})$ of the original graph. If H is obtained from G by contracting some edges, then H is a *contraction* of G.

We summarize below some known algorithmic results which will be used as subroutines by our algorithm.

▶ **Proposition 6.** Longest Path *and* Longest Cycle *are solvable in time* $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

We also need the result about the variant of Longest Path with fixed end-vertices. In the (s,t)-Longest Path, we are given two vertices s and t of a graph G and a positive integer k. The task is to decide, whether G has an (s,t)-path with at least k vertices. Using the results of Bezáková et al. [2], we immediately obtain the following.

▶ **Proposition 7.** (s,t)-Longest Path *is solvable in time* $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

3 Segments and rerouting

In this section we define systems of segments and prove structural results about them. These combinatorial results are crucial for our algorithms for Longest Path Above Degeneracy and Longest Cycle Above Degeneracy.

The following rerouting lemma is crucial for our algorithms.

▶ **Lemma 8.** *Let G be an n-vertex graph and k be a positive integer such that $\delta(G) \geq \max\{5k-3, n-k\}$. Let $\{s_1,t_1\},\ldots,\{s_r,t_r\}$, $r \leq k$, be a collection of pairs of vertices of G such that (i) $s_i,t_i \notin \{s_j,t_j\}$ for all $i \neq j$, $i,j \in \{1,\ldots,r\}$, and (ii) there is at least one index $i \in \{1,\ldots,r\}$ such that $s_i \neq t_i$. Then there is a family of pairwise vertex-disjoint paths $\mathcal{P} = \{P_1,\ldots,P_r\}$ in G such that each P_i is an (s_i,t_i)-path and $\bigcup_{i=1}^{r} V(P_i) = V(G)$, that is, the paths cover all vertices of G.*

Proof. We prove the lemma in two steps. First we show that there exists a family $\mathcal{P}'$ of pairwise vertex-disjoint paths connecting all pairs $\{s_i,t_i\}$. Then we show that if the paths of $\mathcal{P}'$ do not cover all vertices of G, it is possible to enlarge a path such that the new family of paths covers more vertices.

We start by constructing a family of vertex-disjoint paths $\mathcal{P}' = \{P_1,\ldots,P_r\}$ in G such that each $P_i \in \mathcal{P}'$ is an (s_i,t_i)-path. We prove that we can construct paths in such a way that each P_i has at most 3 vertices. Let $T = \bigcup_{i=1}^{r}\{s_i,t_i\}$ and $S = V(G) \setminus T$. Notice that $|S| \geq n-2k \geq \delta(G)+1-2k \geq 3k-2$. We consecutively construct paths of $\mathcal{P}'$ for $i \in \{1,\ldots,r\}$. If $s_i = t_i$, then we have a trivial (s_i,t_i)-path. If s_i and t_i are adjacent, then edge s_it_i forms an (s_i,t_i)-path with 2 vertices. Assume that $s_i \neq t_i$ and $s_it_i \notin E(G)$. The already constructed paths contain at most $r-1 \leq k-1$ vertices of S in total. Hence, there is a set $S' \subseteq S$ of at least $2k-1$ vertices that are not contained in any of already constructed paths. Since $\delta(G) \geq n-k$, each vertex of G has at most $k-1$ non-neighbors in G. By the pigeonhole principle, there is $v \in S'$ such that $s_iv, t_iv \in E(G)$. Then we can construct the path $P_i = s_ivt_i$.

We proved that there is a family $\mathcal{P}' = \{P_1,\ldots,P_r\}$ of vertex-disjoint (s_i,t_i)-paths in G. Among all such families, let us select a family $\mathcal{P} = \{P_1,\ldots,P_r\}$ covering the maximum number of vertices of $V(G)$. If $\bigcup_{i=1}^{r} V(P_i) = V(G)$, then the lemma holds. Assume that $|\bigcup_{i=1}^{r} V(P_i)| < |V(G)|$. Suppose $|\bigcup_{i=1}^{r} V(P_i)| \leq 3k-1$. Since $s_i \neq t_i$ for some i, there is an edge uv in one of the paths. Since $n \geq \delta(G)+1 \geq 5k-2$, there are at least $2k-1$ vertices uncovered by paths of $\mathcal{P}$. Since $\delta(G) \geq n-k$, each vertex of G has at most $k-1$

non-neighbors in G. Thus there is $w \in V(G) \setminus (\bigcup_{i=1}^{r} V(P_i))$ adjacent to both u and v. But then we can extend the path containing uv by replacing uv by the path uwv. The paths of the new family cover more vertices than the paths of $\mathcal{P}$, which contradicts the choice of $\mathcal{P}$.

Suppose $|\bigcup_{i=1}^{r} V(P_i)| \geq 3k$. Because the paths of $\mathcal{P}$ are vertex-disjoint, the union of edges of paths from $\mathcal{P}$ contains a k-matching. That is, there are k edges $u_1v_1, \ldots, u_kv_k$ of G such that for every $i \in \{1, \ldots, k\}$, vertices u_i, v_i are consecutive in some path from $\mathcal{P}$ and $u_i \neq u_j$, $u_i \neq v_j$ for all non-equal $i, j \in \{1, \ldots, k\}$. Let $w \in V(G) \setminus (\bigcup_{i=1}^{r} V(P_i))$. We again use the observation that w has at most $k-1$ non-neighbors in G and, therefore, there is $j \in \{1, \ldots, k\}$ such that $u_jw, v_jw \in E(G)$. Then we extend the path containing u_jv_j by replacing edge u_jv_j by the path u_jwv_j, contradicting the choice of $\mathcal{P}$. We conclude that the paths of $\mathcal{P}$ cover all vertices of G. ◀

Let G be a graph and let $T \subset V(G)$ be a set of terminals. We need the following definitions.

▶ **Definition 9** (Terminal segments). *We say that a path P in G is a* one-terminal T-segment *if it has at least two vertices, exactly one end-vertex of P is in T and other vertices are not in T. Respectively, P is a* two-terminal T-segment *if it has at least three vertices, both end-vertices of P are in T and internal vertices of P are not in T.*

For every cycle C hitting H, removing the vertices of H from C turns it into a set of two-terminal T-segments for $T = V(H)$. So here is the definition.

▶ **Definition 10** (System of T-segments). *We say that a set $\{P_1, \ldots, P_r\}$ of paths in G is a* system of T-segments *if it satisfies the following conditions.*

(i) *For each $i \in \{1, \ldots, r\}$, P_i is a two-terminal T-segment,*
(ii) *$P_1, \ldots, P_r$ are pairwise internally vertex-disjoint, and*
(iii) *the union of $P_1, \ldots, P_r$ is a linear forest.*

Let us remark that we do not require that the end-vertices of the paths $\{P_1, \ldots, P_r\}$ cover all vertices of T. System of segments will be used for solving Longest Cycle Above Degeneracy.

For Longest Path Above Degeneracy we need to modify the definition of a system of T-segments to include the possibility that path can start or end in H.

▶ **Definition 11** (Extended system of T-segments). *We say that a set $\{P_1, \ldots, P_r\}$ of paths in G is an* extended system of T-segments *if the following holds.*

(i) *At least one and at most two paths are one-terminal T-segments and the others are two-terminal T-segments.*
(ii) *$P_1, \ldots, P_r$ are pairwise internally vertex-disjoint and the end-vertices of each one-terminal segment that is in $V(G) \setminus T$ is pairwise distinct with the other vertices of the paths.*
(iii) *The union of $P_1, \ldots, P_r$ is a linear forest and if $\{P_1, \ldots, P_r\}$ contains two one-terminal segments, then the vertices of these segments are in distinct components of the forest.*

The following lemma will be extremely useful for the algorithm solving Longest Path Above Degeneracy. Informally, it shows that if a connected graph G is of large degeneracy but has a small core H, then deciding whether G has a path of length $k + d$ can be reduced to checking whether G has an extended system of T-segments with terminal set $T = V(H)$ such that the total number of vertices used by the system is $\mathcal{O}(k)$.

▶ **Lemma 12.** *Let $d, k \in \mathbb{N}$. Let G be a connected graph with a d-core H such that $d \geq 5k-3$ and $d > |V(H)| - k$. Then G has a path on $d+k$ vertices if and only if G has an extended system of T-segments $\{P_1, \ldots, P_r\}$ with terminal set $T = V(H)$ such that the total number of vertices contained in the paths of the system in $V(G) \setminus V(H)$ is $p = d + k - |V(H)|$.*

Proof. We put $T = V(H)$. Suppose first that G has an extended system $\{P_1, \ldots, P_r\}$ of T-segments and that the total number of vertices of the paths in the system outside T is $p = d + k - |T|$. Let s_i and t_i be the end-vertices of P_i for $i \in \{1, \ldots, r\}$ and assume without loss of generality that for $1 \leq i < j \leq r$, the vertices of P_i and P_j are pairwise distinct with the possible exception $t_i = s_j$ when $i = j - 1$. We also assume without loss of generality that P_1 is a one-terminal segment and $t_1 \in T$ and if $\{P_1, \ldots, P_r\}$ has two one-terminal segments, then the second such segment is P_r and $s_r \in T$. Note that because $|V(H)| > d$, we have that $p = d + k - |V(H)| < k$.

Suppose that $\{P_1, \ldots, P_r\}$ contains one one-terminal segment P_1. Let s_{r+1} be an arbitrary vertex of $T \setminus (\bigcup_{i=1}^{r} V(P_i))$. Notice that such a vertex exists, because $|T \cap (\bigcup_{i=1}^{r} V(P_i))| \leq 2p - 1 < 2k - 1$ and $|T| \geq d + 1 \geq 5k - 3$. Consider the collection of pairs of vertices $\{t_1, s_2\}, \{t_2, s_3\}, \ldots, \{t_r, s_{r+1}\}$. Notice that vertices from distinct pairs are distinct and $t_r \neq s_{r+1}$. By Lemma 8, there are vertex-disjoint paths $P'_1, \ldots, P'_r$ in H that cover T such that P'_i is a (t_i, s_{i+1})-path for $i \in \{1, \ldots, r\}$. By concatenating $P_1, P'_1, P_2, \ldots, P_r, P'_r$ we obtain a path in G with $|T| + p = d + k$ vertices.

Assume now that $\{P_1, \ldots, P_r\}$ contains two one-terminal segments P_1 and P_r. Consider the collection of pairs of vertices $\{t_1, s_2\}, \ldots, \{t_{r-1}, s_r\}$. Notice that vertices from distinct pairs are distinct and there is $i \in \{2, \ldots, r\}$ such that $t_{i-1} \neq s_i$ by the condition (iii) of the definition of an extended system of segments. By Lemma 8, there are vertex-disjoint paths $P'_1, \ldots, P'_{r-1}$ in H that cover T such that P'_i is a (t_i, s_{i+1})-path for $i \in \{1, \ldots, r-1\}$. By concatenating $P_1, P'_1, \ldots, P'_{r-1}, P_r$ we obtain a path in G with $|T| + p = d + k$ vertices.

To show the implication in the opposite direction, let us assume that G has an (x, y)-path P with $d + k$ vertices. We distinguish several cases.

Case 1: $V(P) \cap T = \emptyset$**.** Consider a shortest path P' with one end-vertex $s \in V(P)$ and the second end-vertex $t \in T$. Notice that such a path exists, because G is connected. Denote by P_x and P_y the (s, x) and (s, y)-subpaths of P respectively. Because $d \geq 5k - 3$, $|V(P_x)| \geq k$ or $|V(P_y)| \geq k$. Assume that $|V(P_x)| \geq k$. Then the concatenation of P' and P_x is a path with at least $k + 1$ vertices and it contains a subpath P'' with the end-vertex t with $p + 1$ vertices. We have that $\{P'\}$ is an extended system of T-segments and P'' has p vertices outside T.

Case 2: $V(P) \cap T \neq \emptyset$ **and** $E(P) \cap E(H) = \emptyset$**.** Let $S = V(P) \cap T$. Since H is an induced subgraph of G and $E(P) \cap E(H) = \emptyset$, $|V(P) \setminus S| \geq (d + k)/2 - 1 \geq 3k - 5/2 > 3p - 5/2 \geq 2p - 2$. Then for every $t \in S$, either the (t, x)-subpath P_x of P contains at least p vertices outside T or the (t, y)-subpath P_y of P contains at least p vertices outside T. Assume without loss of generality that P_x contains at least p vertices outside T. Consider the minimal subpath P' of P_x ending at t such that $|V(P') \setminus T| = p$. Then the start vertex s of P' is not in T. Let $\{t_1, \ldots, t_r\} = V(P') \cap T$ and assume that $t_1, \ldots, t_r$ are ordered in the same order as they occur in P' starting from s. In particular, $t_r = t$. Let $t_0 = s$. Consider the paths $P_1, \ldots, P_r$ where P_i is the (t_{i-1}, t_i)-subpath of P' for $i \in \{1, \ldots, r\}$. Since $k > p$, $r \leq k$. We obtain that $\{P_1, \ldots, P_r\}$ is an extended system of T-segments with p vertices outside T.

Case 3: $E(P) \cap E(H) \neq \emptyset$**.** Then there are distinct $s, t \in T \cap V(P)$ such that the (s, t)-subpath of P lies in H. Since P has at least p vertices outside T, there are $s', t' \in V(P) \setminus T$ such that the (s', t')-subpath P' of P is a subpath with exactly p vertices outside T with $s, t \in V(P')$. Let $P_1, \ldots, P_r$ be the family of inclusion maximal subpaths of P' containing the vertices of $V(P')\setminus T$ such that the internal vertices of each P_i are outside T. Observe that since $s \neq t$, the union of these paths is a linear forest with at least two components. We conclude that the set $\{P_1,\ldots,P_r\}$ is a required extended system of T-segments. ◀

The next lemma will be used for solving Longest Cycle Above Degeneracy.

▶ **Lemma 13.** *Let $d,k \in \mathbb{N}$. Let G be a 2-connected graph with a d-core H such that $d \geq 5k-3$ and $d > |V(H)|-k$. Then G has a cycle with at least $d+k$ vertices if and only if one of the following holds (where $p = d+k-|V(H)|$).*

(i) *There are distinct $s,t \in V(H)$ and an (s,t)-path P in G with all internal vertices outside $V(H)$ such that P has at least p internal vertices.*

(ii) *G has a system of T-segments $\{P_1,\ldots,P_r\}$ with terminal set $T=V(H)$ and the total number of vertices of the paths outside $V(H)$ is at least p and at most $2p-2$.*

Proof. We put $T=V(H)$. First, we show that if (i) or (ii) holds, then G has a cycle with at least $d+k$ vertices. Suppose that there are distinct $s,t\in T$ and an (s,t)-path P in G with all internal vertices outside T such that P has at least p internal vertices. By Lemma 8, H has a Hamiltonian (s,t)-path P'. By taking the union of P and P' we obtain a cycle with at least $|T|+p=d+k$ vertices.

Now assume that G has a system of T-segments $\{P_1,\ldots,P_r\}$ and the total number of vertices of the paths outside T is at least p. Let s_i and t_i be the end-vertices of P_i for $i\in\{1,\ldots,r\}$ and assume without loss of generality that for $1\leq i<j\leq r$, the vertices of P_i and P_j are pairwise distinct with the possible exception $t_i=s_j$ when $i=j-1$. Consider the collection of pairs of vertices $\{t_1,s_2\},\ldots,\{t_{r-1},s_r\},\{t_r,s_1\}$. Notice that vertices from distinct pairs are distinct and $t_r\neq s_1$. By Lemma 8, there are vertex-disjoint paths $P'_1,\ldots,P'_r$ in H that cover T such that P'_i is a (t_i,s_{i+1})-path for $i\in\{1,\ldots,r-1\}$ and P'_r is a (t_r,s_1)-path. By taking the union of $P_1,\ldots,P_r$ and $P'_1,\ldots,P'_r$ we obtain a cycle in G with at least $|T|+p=d+k$ vertices.

To show the implication in the other direction, assume that G has a cycle C with at least $d+k$ vertices.

Case 1: $V(C)\cap T=\emptyset$. Since G is a 2-connected graph, there are pairwise distinct vertices $s,t\in T$ and $x,y\in V(C)$ and vertex-disjoint (s,x) and (y,t)-paths P_1 and P_2 such that the internal vertices of the paths are outside $T\cup V(C)$. The cycle C contains an (x,y)-path P with at least $(d+k)/2+1\geq p$ vertices. The concatenation of P_1, P and P_2 is an (s,t)-path in G with at least p internal verices and the internal vertices are outside T. Hence, (i) holds.

Case 2: $|V(C)\cap T|=1$. Let $V(C)\cap T=\{s\}$ for some vertex s. Since G is 2-connected, there is a shortest (x,t)-path P in $G-s$ such that $x\in V(C)$ and $t\in T$. The cycle C contains an (s,x)-path P' with at least $(d+k)/2+1\geq p$ vertices. The concatenation of P' and P is an (s,t)-path in G with at least p internal vertices and the internal vertices of the path are outside T. Therefore, (i) is fulfilled.

Case 3: $|V(C)\cap T|\geq 2$. Since $|V(C)|\geq d$ and $|T|<d$, we have that $V(C)\setminus T\neq\emptyset$. Then we can find pairs of distinct vertices $\{s_1,t_1\}\ldots,\{s_\ell,t_\ell\}$ of $T\cap V(C)$ and segments $P_1,\ldots,P_\ell$ of C such that (a) P_i is an (s_i,t_i)-path for $i\in\{1,\ldots,\ell\}$ with at least one internal vertex and the internal vertices of P_i are outside T, (b) for $1\leq i<j\leq\ell$, the vertices of P_i and P_j are distinct with the possible exception $t_i=s_j$ if $i=j-1$ and, possibly, $t_\ell=s_1$, and (c) $\bigcup_{i=1}^{\ell}V(P_i)\setminus T=V(C)\setminus T$. If there is $i\in\{1,\ldots,\ell\}$ such that P_i has at least p internal vertices, then (i) is fulfilled.

Now assume that each P_i has at most $p-1$ internal vertices; notice that $p \geq 2$ in this case. We select an inclusion minimal set of indices $I \subseteq \{1,\ldots,\ell\}$ such that $|\bigcup_{i\in I} V(P_i) \setminus T| \geq p$. Notice that because each path has at most $p-1$ internal vertices, $|\bigcup_{i\in I} V(P_i) \setminus T| \leq 2p-2$. Let $I = \{i_1,\ldots,i_r\}$ and $i_1 < \ldots < i_r$. By the choice of $P_{i_1},\ldots,P_{i_r}$, the union of $P_{i_1},\ldots,P_{i_r}$ is either the cycle C or a linear forest. Suppose that the union of the paths is C. Then $I = \{1,\ldots,\ell\}$, $\ell \leq p$ and $|V(P) \cap T| = \ell$. Note that because $|V(H)| > d$, we have that $p = d+k-|V(H)| < k$. We obtain that C has at most $(2p-2)+p \leq 3p-2 < 3k-2 < d+k$ vertices (the last inequality follows from the fact that $d \geq 5k-3$); a contradiction. Hence, the union of the paths is a linear forest. Therefore, $\{P_{i_1},\ldots,P_{i_r}\}$ is a system of T-segments with terminal set $T = V(H)$ and the total number of vertices of the paths outside T is at least p and at most $2p-2$, that is, (ii) is fulfilled. ◀

We have established the fact that the existence of a long (path) cycle is equivalent to the existence of an (extended) system of T-segments for some terminal set T with at most $p \leq k$ vertices from outside T. Towards designing algorithms for Longest Path Above Degeneracy and Longest Cycle Above Degeneracy, we define two auxiliary problems which can be solved using well known color-coding technique.

Segments with Terminal Set

Input:	A graph G, $T \subset V(G)$ and a positive integers p and r.
Task:	Decide whether G has a system of segments $\{P_1,\ldots,P_r\}$ w.r.t. T such that the total number of internal vertices of the paths is p.

Extended Segments with Terminal Set

Input:	A graph G, $T \subset V(G)$ and a positive integers p and r.
Task:	Decide whether G has an extended system of segments $\{P_1,\ldots,P_r\}$ w.r.t. T such that the total number of vertices of the paths outside T is p.

▶ **Lemma 14** ($\star$). [1] Segments with Terminal Set *and* Extended Segments with Terminal Set *are solvable in time* $2^{\mathcal{O}(p)} \cdot n^{\mathcal{O}(1)}$.

4 Putting all together: Final proofs

Proof of Theorem 3. Let G be a connected graph of degeneracy at least d and let k be a positive integer. If $d \leq 5k-4$, then we check the existence of a path with $d+k \leq 6k-4$ vertices using Proposition 6 in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. Assume from now that $d \geq 5k-3$. Then we find a d-core H of G. This can be done in linear time using the results of Matula and Beck [28]. If $|V(H)| \geq d+k$, then by Theorem 5, H, and hence G, contains a path with $\min\{2d+1, |V(H)|\} \geq d+k$ vertices. Assume that $|V(H)| < d+k$. By Lemma 12, G has a path with $d+k$ vertices if and only if G has paths $P_1,\ldots,P_r$ such that $\{P_1,\ldots,P_r\}$ is an extended system of T-segments for $T = V(H)$ and the total number of vertices of the paths outside T is $p = d+k-|T|$. Since the number of vertices in every graph is more than its minimum degree, we have that $|T| > d$, and thus $p < k$. For each $r \in \{1,\ldots,p\}$, we verify if such a system exists in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ by making use of Lemma 14. Thus the total running time of the algorithm is $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

[1] Proofs of results marked with ($\star$) are omitted in this extended abstract.

Proof of Theorem 2. Let G be a 2-connected graph of degeneracy at least d and let $k \in \mathbb{N}$. If $d \leq 5k-4$, then we check the existence of a cycle with at least $d+k \leq 6k-4$ vertices using Proposition 6 in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. Assume from now on that $d \geq 5k-3$. Then we find a d-core H of G in linear time using the results of Matula and Beck [28].

We claim that if $|V(H)| \geq d+k$, then H contains a cycle with at least $d+k$ vertices. If H is 2-connected, then this follows from Theorem 4. Assume that H is not a 2-connected graph. By the definition of a d-core, H is connected. Observe that $|V(H)| \geq d+1 \geq 5k-2 \geq 3$. Hence, H has at least two blocks and at least one cut vertex. Consider the block graph $Block(H)$ of H. Recall that the vertices of $Block(H)$ are the blocks and the cut vertices of H and a cut vertex c is adjacent to a block B if and only if $c \in V(B)$. Recall also that $Block(H)$ is a tree. We select an arbitrary block R of H and declare it to be the *root* of $Block(H)$. Let $S = V(G) \setminus V(H)$. Observe that $S \neq \emptyset$, because G is 2-connected and H is not. Let $F_1, \ldots, F_\ell$ be the components of $G[S]$. We contract the edges of each component and denote the obtained vertices by $u_1, \ldots, u_\ell$. Denote by G' the obtained graph. It is straightforward to verify that G' has no cut vertices, that is, G' is 2-connected. For each $i \in \{1, \ldots, \ell\}$, consider u_i. This vertex has at least 2 neighbors in $V(H)$. We select a vertex $v_i \in N_{G'}(u_i)$ that is not a cut vertex of H or, if all the neighbors of u_i are cut vertices, we select v_i be a cut vertex at maximum distance from R in $Block(H)$. Then we contract u_iv_i. Observe that by the choice of each v_i, the graph G'' obtained from G' by contracting $u_1v_1, \ldots, u_\ell v_\ell$ is 2-connected. We have that G'' is a 2-connected graph of minimum degree at least d with at least $d+k$ vertices. By Theorem 4, G'' has a cycle with at least $\min\{2d, |V(G'')|\} \geq d+k$ vertices. Because G'' is a contraction of G, we conclude that G contains a cycle with at least $d+k$ vertices as well.

From now we can assume that $|V(H)| < d+k$. By Lemma 13, G has a cycle with $d+k$ vertices if and only if one of the following holds for $p = d+k-|T|$ where $T = V(H)$.

(i) There are distinct $s, t \in T$ and an (s,t)-path P in G with all internal vertices outside T such that P has at least p internal vertices.

(ii) G has a system of T-segments $\{P_1, \ldots, P_r\}$ and the total number of vertices of the paths outside T is at least p and at most $2p-2$.

Notice that $p \leq k$ (because $d - |T| \leq 0$). We verify whether (i) holds using Proposition 7. To do it, we consider all possible choices of distinct s, t. Then we construct the auxiliary graph G_{st} from G by the deletion of the vertices of $T \setminus \{s,t\}$ and the edges of $E(H)$. Then we check whether G_{st} has an (s,t)-path of length at least $p+1$ in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ applying Proposition 7.

Assume that (i) is not fulfilled. Then it remains to check (ii). For every $r \in \{1, \ldots, p\}$, we verify the existence of a system of T-segments $\{P_1, \ldots, P_r\}$ in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ using Lemma 14. We return the answer *yes* if we get the answer yes for at least one instance of Segments with Terminal Set and we return *no* otherwise.

5 Hardness for Longest Path and Cycle above Degeneracy

In this section we complement Theorems 3 and 2 by some hardness observations.

▶ **Proposition 15** (⋆). [2] Longest Path Above Degeneracy *is* NP*-complete even if* $k = 2$ *and* Longest Cycle Above Degeneracy *is* NP*-complete even for connected graphs and* $k = 2$*.*

[2] Proposition 15 and its proof was pointed to us by Nikolay Karpov.

Recall that a graph G has a path with at least $\mathrm{dg}(G) + 1$ vertices and if $\mathrm{dg}(G) \geq 2$, then G has a cycle with at least $\mathrm{dg}(G) + 1$ vertices. Moreover, such a path or cycle can be constructed in polynomial (linear) time. Hence, Proposition 15 gives tight complexity bounds. Nevertheless, the construction used to show hardness for LONGEST PATH ABOVE DEGENERACY uses a disconnected graph, and the graph constructed to show hardness for LONGEST CYCLE ABOVE DEGENERACY has a cut vertex. Hence, it is natural to consider LONGEST PATH ABOVE DEGENERACY for connected graphs and LONGEST CYCLE ABOVE DEGENERACY for 2-connected graphs. We show in Theorems 3 and 2 that these problems are FPT when parameterized by k in these cases. Here, we observe that the lower bound $\mathrm{dg}(G)$ that is used for the parameterization is tight in the following sense.

▶ **Proposition 16.** *For any $0 < \varepsilon < 1$, it is* NP*-complete to decide whether a connected graph G contains a path with at least $(1+\varepsilon)\mathrm{dg}(G)$ vertices and it is* NP*-complete to decide whether a 2-connected graph G contains a cycle with at least $(1+\varepsilon)\mathrm{dg}(G)$ vertices.*

Proof. Let $0 < \varepsilon < 1$.

First, we consider the problem about a path with $(1 + \varepsilon)\mathrm{dg}(G)$ vertices. We reduce HAMILTONIAN PATH that is well-known to be NP-complete (see [15]). Let G be a graph with $n \geq 2$ vertices. We construct the graph G' as follows.

- Construct a copy of G.
- Let $p = 2\lceil \frac{n}{\varepsilon} \rceil$ and construct p pairwise adjacent vertices $u_1, \dots, u_p$.
- For each $v \in V(G)$, construct an edge vu_1.
- Let $q = \lceil (1 + \varepsilon)(p - 1) - (n + p) \rceil$. Construct vertices $w_1, \dots, w_q$ and edges u_1w_1, w_qu_2 and $w_{i-1}w_i$ for $i \in \{2, \dots, q\}$.

Notice that $q = \lceil (1 + \varepsilon)(p - 1) - (n + p) \rceil = \lceil 2\varepsilon\lceil \frac{n}{\varepsilon} \rceil - n - 1 - \varepsilon \rceil \geq \lceil n - 1 - \varepsilon \rceil \geq 1$ as $n \geq 2$. Observe also that G is connected. We claim that G has a Hamiltonian path if and only if G' has a path with at least $(1+\varepsilon)\mathrm{dg}(G')$ vertices. Notice that $\mathrm{dg}(G') = p - 1$ and $|V(G')| = n+p+q = \lceil (1+\varepsilon)\mathrm{dg}(G') \rceil$. Therefore, we have to show that G has a Hamiltonian path if and only if G' has a Hamiltonian path. Suppose that G has a Hamiltonian path P with an end-vertex v. Consider the path $Q = vu_1w_1 \dots w_qu_2u_3 \dots u_p$. Clearly, the concatenation of P and Q is a Hamiltonian path in G'. Suppose that G' has a Hamiltonian path P. Since u_1 is a cut vertex of G', we obtain that P has a subpath that is a Hamiltonian path in G.

Consider now the problem about a cycle with at least $(1 + \varepsilon)\mathrm{dg}(G)$ vertices. We again reduce HAMILTONIAN PATH and the reduction is almost the same. Let G be a graph with $n \geq 2$ vertices. We construct the graph G' as follows.

- Construct a copy of G.
- Let $p = 2\lceil \frac{n}{\varepsilon} \rceil$ and construct p pairwise adjacent vertices $u_1, \dots, u_p$.
- For each $v \in V(G)$, construct edges vu_1 and vu_2.
- Let $q = \lceil (1 + \varepsilon)(p - 1) - (n + p) \rceil$. Construct vertices $w_1, \dots, w_q$ and edges u_2w_1, w_qu_3 and $w_{i-1}w_i$ for $i \in \{2, \dots, q\}$.

As before, we have that $q \geq 1$. Notice additionally that $p \geq 3$, i.e., the vertex u_3 exists. It is straightforward to see that G' is 2-connected. We claim that G has a Hamiltonian path if and only if G' has a cycle with at least $(1 + \varepsilon)\mathrm{dg}(G')$ vertices. We have that $\mathrm{dg}(G') = p - 1$ and $|V(G')| = \lceil (1+\varepsilon)\mathrm{dg}(G') \rceil$. Hence, we have to show that G has a Hamiltonian path if and only if G' has a Hamiltonian cycle. Suppose that G has a Hamiltonian path P with end-vertices x and y. Consider the path $Q = xu_2w_1 \dots w_qu_3u_4 \dots u_py$. Clearly, P and Q together form a Hamiltonian cycle in G'. Suppose that G' has a Hamiltonian cycle C. Since $\{u_1, u_2\}$ is a cut set of G', we obtain that C contains a path that is a Hamiltonian path of G. ◀

6 Conclusion

We considered the lower bound $\mathrm{dg}(G)+1$ for the number of vertices in a longest path or cycle in a graph G. It would be interesting to consider the lower bounds given in Theorems 4 and 5. More precisely, what can be said about the parameterized complexity of the variants of Long Path (Cycle) where given a (2-connected) graph G and $k \in \mathbb{N}$, the task is to check whether G has a path (cycle) with at least $2\delta(G)+k$ vertices? Are these problems FPT when parameterized by k? It can be observed that the bound $2\delta(G)$ is "tight". That is, for any $0<\varepsilon<1$, it is NP-complete to decide whether a connected (2-connected) G has a path (cycle) with at least $(2+\varepsilon)\delta(G)$ vertices. See also [30] for related hardness results.

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. Ivona Bezáková, Radu Curticapean, Holger Dell, and Fedor V. Fomin. Finding Detours is Fixed-parameter Tractable. *CoRR*, abs/1607.07737, 2016. URL: `http://arxiv.org/abs/1607.07737`, `arXiv:1607.07737`.
3. Ivona Bezáková, Radu Curticapean, Holger Dell, and Fedor V. Fomin. Finding Detours is Fixed-Parameter Tractable. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, volume 80 of *LIPIcs*, pages 54:1–54:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
4. Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *CoRR*, abs/1007.1161, 2010. URL: `http://arxiv.org/abs/1007.1161`.
5. Hans L. Bodlaender. On Linear Time Minor Tests with Depth-First Search. *J. Algorithms*, 14(1):1–23, 1993. `doi:10.1006/jagm.1993.1001`.
6. 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 J. Comput.*, 38(6):2526–2547, 2009. `doi:10.1137/080716475`.
7. Jianer Chen, Songjian Lu, Sing-Hoi Sze, and Fenghui Zhang. Improved algorithms for path, matching, and packing problems. In *Proceedings of the18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 298–307. SIAM, 2007.
8. 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 *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43–54, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
9. Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
10. G. A. Dirac. Some theorems on abstract graphs. *Proc. London Math. Soc. (3)*, 2:69–81, 1952.
11. P. Erdős and T. Gallai. On maximal paths and circuits of graphs. *Acta Math. Acad. Sci. Hungar*, 10:337–356 (unbound insert), 1959.
12. Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient Computation of Representative Families with Applications in Parameterized and Exact Algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. `doi:10.1145/2886094`.
13. 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`.

14 Harold N. Gabow and Shuxin Nie. Finding a long directed cycle. *ACM Transactions on Algorithms*, 4(1), 2008. doi:10.1145/1328911.1328918.

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

16 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.

17 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.

18 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. *J. Computer and System Sciences*, 78(1):151–163, 2012. doi:10.1016/j.jcss.2011.01.004.

19 Gregory Z. Gutin and Viresh Patel. Parameterized Traveling Salesman Problem: Beating the Average. *SIAM J. Discrete Math.*, 30(1):220–238, 2016.

20 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.

21 Frank Harary. A characterization of block-graphs. *Canad. Math. Bull.*, 6:1–6, 1963.

22 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.

23 Joachim Kneis, Daniel Mölle, Stefan Richter, and Peter Rossmanith. Divide-and-Color. In *Proceedings of the 34th International Workshop Graph-Theoretic Concepts in Computer Science (WG)*, volume 4271 of *Lecture Notes in Computer Science*, pages 58–67. Springer, 2008.

24 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 Comput. Sci.*, pages 575–586. Springer, 2008.

25 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.

26 Meena Mahajan and Venkatesh Raman. Parameterizing above Guaranteed Values: MaxSat and MaxCut. *J. Algorithms*, 31(2):335–354, 1999.

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

28 David W. Matula and Leland L. Beck. Smallest-Last Ordering and clustering and Graph Coloring Algorithms. *J. ACM*, 30(3):417–427, 1983. doi:10.1145/2402.322385.

29 B. Monien. How to find long paths efficiently. In *Analysis and design of algorithms for combinatorial problems (Udine, 1982)*, volume 109 of *North-Holland Math. Stud.*, pages 239–254. North-Holland, Amsterdam, 1985. doi:10.1016/S0304-0208(08)73110-4.

30 Ingo Schiermeyer. Problems remaining NP-complete for sparse or dense graphs. *Discussiones Mathematicae Graph Theory*, 15(1):33–41, 1995. doi:10.7151/dmgt.1004.

31 Dekel Tsur. Faster deterministic parameterized algorithm for k-Path. *CoRR*, abs/1808.04185, 2018. arXiv:1808.04185.

32 Ryan Williams. Finding Paths of Length k in $O^*(2^k)$ Time. *Inf. Process. Lett.*, 109(6):315–318, 2009.

33 Meirav Zehavi. A randomized algorithm for long directed cycle. *Inf. Process. Lett.*, 116(6):419–422, 2016. doi:10.1016/j.ipl.2016.02.005.