SIAM J. DISCRETE MATH.
Vol. 34, No. 3, pp. 1587–1601

GOING FAR FROM DEGENERACY*

FEDOR V. FOMIN†, PETR A. GOLOVACH†, DANIEL LOKSHTANOV‡, FAHAD PANOLAN§, SAKET SAURABH¶, AND MEIRAV ZEHAVI‖

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 $2^{\mathcal{O}(k)} \cdot |V(G)|^{\mathcal{O}(1)}$. 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 $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. 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 $(1 + \varepsilon)d$.

Key words. longest path, longest cycle, fixed-parameter tractability, above guarantee parameterization

AMS subject classifications. 05C85, 68R10

DOI. 10.1137/19M1290577

1. Introduction.

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

THEOREM 1 (see 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.

*Received by the editors September 30, 2019; accepted for publication (in revised form) April 29, 2020; published electronically July 20, 2020. A preliminary version of this paper appeared as an extended abstract in the proceedings of ESA 2019.
https://doi.org/10.1137/19M1290577
Funding: The research was funded by the Research Council of Norway via the projects CLASSIS (grant 249994) and MULTIVAL (grant 263317), Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18, and by the European Research Council (ERC) via grant LOPPRE (reference 819416).
†Department of Informatics, University of Bergen, Bergen 5020, Norway (Fedor.Fomin@uib.no, Petr.Golovach@uib.no).
‡Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA 93106 (daniello@ucsb.edu).
§Department of Computer Science and Engineering, IIT Hyderabad, Kandi 502285, India (fahad@iith.ac.in).
¶Institute of Mathematical Sciences, HBNI, Chennai 400094, India (saket@imsc.res.in).
‖Ben-Gurion University, Beersheba, 8410501 Israel (meiravze@bgu.ac.il).

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 noncomplete graph G on n vertices, we construct the connected graph H from the disjoint union of G and the complete graph K_{n-2} on $n-2$ vertices by making one vertex of G adjacent to all the vertices of K_{n-2}. 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 (FPT) being parameterized by k (we refer to the book of Cygan et al. [9] for the formal definition of the notion). 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 is that this time graph H consists of the 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 now define the following.

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 Theorem 3, which is obtained as a corollary of Theorem 2.

Theorem 3. *On connected graphs* Longest Path Above Degeneracy *is solvable in time* $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

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 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 et 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, 33]) since the early work of Monien [29]. The fastest known randomized algorithm for Longest Path on an 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 $2.554^k \cdot n^{\mathcal{O}(1)}$ [32]. Respectively, for Longest Cycle, the current fastest randomized algorithm running in time $4^k \cdot n^{\mathcal{O}(1)}$ was given by Zehavi [34], and the best deterministic algorithm constructed by Fomin et al. [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 an assignment for the values of the variables of a Boolean formula in the conjunctive normal form that satisfies at least half of the clauses or there is always an edge-cut of a graph containing at least half of 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 the above guarantee parameterization was from Mahajan and Raman [26], who applied this approach to the Max Sat and Max Cut problems. 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 the 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 can be seen as nontrivial algorithmic extensions of classical theorems of Dirac [10] and Erdős and Gallai [11]. In particular, to show Theorem 2, we use the famous Dirac's theorem.

THEOREM 4 (see 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.*

We give a high-level overview of the ideas used to prove Theorem 2. 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 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 order 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. Consider the terminal points, that is, the vertices in which this cycle enters and leaves H. 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 three vertices, its end-vertices 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.

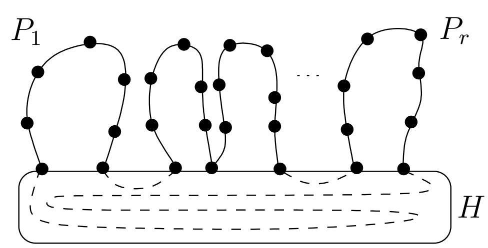

FIG. 1. *Reducing* LONGEST CYCLE ABOVE DEGENERACY *to finding a system of segments* $P_1, \ldots, P_r$ *(complementing the segments into a cycle is shown by dashed lines).*

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

- either there is a path with at least $d + k - |V(H)|$ internal vertices whose end-vertices are 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 $[d + k - |V(H)|, 2 \cdot (d + k - |V(H)|)]$.

The proof of this structural result is built on Lemma 2, which describes the possibility of routing in graphs of large minimal degree. The crucial property is that we can complement any system of segments with bounded number of terminal points 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 $d + k$ in G boils down to one of the following tasks. Either find a path of length at least

$d + k - |V(H)| + 1$ with its end-vertices in H and all internal vertices outside H, or find a system of segments with respect to $V(H)$ such that the total number of internal vertices used by the paths of the system is at least $d + k - |V(H)|$ and is upper bounded by $2(d + k - |V(H)|)$. 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 this paper. In section 2 we give basic definitions and state some known fundamental results. Sections 3–4 contain the proof of Theorems 2 and 3. 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 this paper we use $n = |V(G)|$ to denote the *order* of the input graph G and $m = |E(G)|$ to denote its *size*. 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 $\delta(G) = \min\{d_G(v) \mid v \in V(G)\}$. For a graph property Π, it is said that H is an *inclusion maximal* induced subgraph of G satisfying P if H satisfies Π, but the property does not hold for any induced subgraph H' of G with $V(H) \subset V(H')$. 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 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 *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* P in G is a connected subgraph of G with at most two vertices of degree at most one whose remaining vertices have degree two. We use $P = v_1 \cdots v_k$ to denote the path with the vertices $v_1, \ldots, v_k$ and the edges $v_{i-1}v_i$ for $i \in \{1, \ldots, k\}$; the vertices v_1 and v_k are the *end-vertices* of P, and $v_2, \ldots, v_{k-1}$ are the *internal* vertices. For a path P with end-vertices s and t, we say that P is an (s,t)-path. We say that G is a *linear forest* if each connected component of G is a path. A *cycle* in G is a connected subgraph whose vertices have degree two. We denote by $v_0v_1 \cdots v_k$, where $v_0 = v_k$, to denote the cycle with the vertices $v_1, \ldots, v_k$ and the edges $v_{i-1}v_i$ for $i \in \{1, \ldots, k\}$. 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 1 (see [13, 34]). LONGEST CYCLE *is 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.

Proposition 2 (see [13]). *(s,t)-Longest Path is solvable in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.*

We use the following reduction of Longest Path to Longest Cycle.

Lemma 1. *Let G' be the graph obtained from a graph G by adding a new vertex v and making it adjacent to every vertex of G. Then G has a path with at least k vertices if and only if G' has a cycle with at least $k+1$ vertices. Moreover, G' is a connected graph with $\mathrm{dg}(G') = \mathrm{dg}(G) + 1$, and if G is connected, then G' is 2-connected.*

Proof. Let $P = v_1 \cdots v_k$ be a path in G. Clearly, $C = vv_1 \cdots v_k v$ is a cycle in G'. For the opposite direction, assume that G' has a cycle with at least $k+1$ vertices. If C is a cycle in G, then the deletion of any edge of C gives a path with at least $k+1$ vertices. Assume that C is not a cycle of G. Then $v \in V(C)$ and C can be written as $vv_1 \cdots v_\ell v$, where $\ell \geq k$ and $v_1, \dots, v_\ell \in V(G)$. Then $P = v_1 \cdots v_\ell$ is a path in G with at least k vertices. To see that $\mathrm{dg}(G') = \mathrm{dg}(G) + 1$, it is sufficient to observe that for every $U \subseteq V(G)$, $\delta(G'[U \cup \{v\}]) = \delta(G[U]) + 1$. Trivially, G' is connected and it is easy to see that G' is 2-connected whenever G is connected. ∎

In particular, Lemma 1 implies that G has a path with at least $\mathrm{dg}(G)+k$ vertices if and only if G' has a cycle with at least $\mathrm{dg}(G') + k$ vertices.

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 algorithm for Longest Cycle Above Degeneracy. We start with the following rerouting lemma.

Lemma 2. *Let G be an n-vertex graph, and let k be a positive integer such that $\delta(G) \geq \max\{5k-3, n-k\}$. Let $\{s_1,t_1\}, \dots, \{s_r,t_r\}$, $r \leq k$, be a collection of pairs of vertices of G such that* (i) *$s_i, t_i \notin \{s_j, t_j\}$ for all $i \neq j$, $i,j \in \{1,\dots,r\}$, and* (ii) *there is at least one index $i \in \{1,\dots,r\}$ such that $s_i \neq t_i$. Then there is a family of pairwise vertex-disjoint paths $\mathcal{P} = \{P_1,\dots,P_r\}$ 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.*

Note that we allow $\{s_i, t_i\}$ to be a pair of the same vertices, and in this case P_i is a single-vertex trivial path. Notice also that condition (ii) ensures that at least one path is nontrivial.

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 $\mathcal{P}' = \{P_1,\dots,P_r\}$ 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 three 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 $i \in \{1,\dots,r\}$. 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 two 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 the already constructed paths. Since $\delta(G) \geq n-k$, each vertex of G has at most $k-1$ nonneighbors 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 $\mathcal{P}' = \{P_1, \dots, P_r\}$ of vertex-disjoint (s_i, t_i)-paths in G. Among all such families, let us select a family $\mathcal{P} = \{P_1, \dots, P_r\}$ 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 $k - 1$ nonneighbors 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 $u_1v_1, \dots, u_kv_k$ of G such that for every $i \in \{1, \dots, k\}$, 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 nonequal $i, j \in \{1, \dots, k\}$. Let $w \in V(G) \setminus (\bigcup_{i=1}^{r} V(P_i))$. We again use the observation that w has at most $k - 1$ nonneighbors in G and, therefore, there is $j \in \{1, \dots, k\}$ 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 1 (terminal segments). *We say that a path P in G is a* 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 terminal T-segments for $T = V(H)$. So here is the definition.

DEFINITION 2 (system of T-segments). *We say that a set $\{P_1, \dots, P_r\}$ of paths in G is a* system of T-segments *if it satisfies the following conditions:*

(i) *for each $i \in \{1, \dots, r\}$, P_i is a terminal T-segment;*

(ii) *$P_1, \dots, P_r$ are pairwise internally vertex-disjoint; and*

(iii) *the union of $P_1, \dots, P_r$ is a linear forest.*

Let us remark that we do not require that the end-vertices of the paths $\{P_1, \dots, P_r\}$ cover all vertices of T.

The following lemma will be extremely useful for the algorithm solving LONGEST CYCLE ABOVE DEGENERACY. Informally, it shows that if a 2-connected graph G is of large degeneracy but has a small core H, then deciding whether G has a path of length $d + k$ can be reduced to checking whether G either has a sufficiently long path with the internal vertices outside H and the end-vertices in H or has a system of T-segments with terminal set $T = V(H)$ with sufficiently many internal vertices whose total number is $\mathcal{O}(k)$.

LEMMA 3. *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 $\{P_1, \dots, P_r\}$ 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 2, 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 $1 \le i < j \le r$, 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 2, 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 $(d + k)/2 + 1 \ge p$ vertices. The concatenation of P_1, P, and P_2 is an (s,t)-path in G with at least p internal vertices 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 $(d + k)/2 + 1 \ge p$ 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: $|V(C) \cap T| \ge 2$. Since $|V(C)| \ge d$ 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 $1 \le i < j \le \ell$, 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 $p \ge 2$ in this case. We select an inclusion minimal set of indices $I \subseteq \{1, \ldots, \ell\}$ such that $|\bigcup_{i \in I} V(P_i) \setminus T| \ge p$. Notice that because each path has at most $p-1$ internal vertices, $|\bigcup_{i \in I} V(P_i) \setminus T| \le 2p - 2$. Let $I = \{i_1, \ldots, i_r\}$ and $i_1 < \cdots < i_r$. 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 $I = \{1, \ldots, \ell\}$, $\ell \le p$, 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 $(2p - 2) + p \le 3p - 2 < 3k - 2 < d + k$ vertices (the last inequality follows from the fact that $d \ge 5k - 3$); 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 $p \le k$ vertices from outside T. Towards designing an algorithm LONGEST

Cycle Above Degeneracy, we define the following auxiliary problem which can be solved using the well-known color-coding technique.

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

Lemma 4. Segments with Terminal Set *is solvable in time* $2^{\mathcal{O}(p)} \cdot n^{\mathcal{O}(1)}$.

Proof. Our algorithm for Segments with Terminal Set uses the *color-coding* technique introduced by Alon, Yuster, and Zwick in [2]. As is usual for algorithms of this type, we first describe a randomized Monte Carlo algorithm and then explain how it could be derandomized.

Let (G, T, p, r) be an instance of Segments with Terminal Set.

Notice that if paths $P_1, \dots, P_r$ are a solution for the instance, that is, $\{P_1, \dots, P_r\}$ is a system of T-segments and the total number of internal vertices of the paths is p, then $|\cup_{i=1}^{r} V(P_i)| \leq p + 2r$. If $r > p$, then because each path in a solution should have at least one internal vertex, (G, T, p, r) is a no-instance. Therefore, we can assume without loss of generality that $r \leq p$. Let $q = p + 2r \leq 3p$. We color the vertices of G with q colors uniformly at random. Let $P_1, \dots, P_r$ be paths in G, and let s_i, t_i be the end-vertices of P_i for $i \in \{1, \dots, r\}$. We say that the paths $P_1, \dots, P_r$ together with the ordered pairs (s_i, t_i) of their end-vertices form a *colorful solution* if the following is fulfilled:

(i) $\{P_1, \dots, P_r\}$ is a system of T-segments,
(ii) $|\cup_{i=1}^{r} V(P_i) \setminus T| = p$,
(iii) if $1 \leq i < j \leq r$, $u \in V(P_i)$, and $v \in V(P_j)$, then the vertices u and v have distinct colors unless $i = j - 1$, $u = t_i$, and $v = s_j$ (in this case the colors can be distinct or the same).

It is straightforward to see that any colorful solution is a solution of the original problem. From the other side, if (G, T, p, r) has a solution $P_1, \dots, P_r$, then with probability at least $\frac{q!}{q^q} > e^{-q}$ all distinct vertices of the paths of a solution are colored by distinct colors, and for such a coloring $P_1, \dots, P_r$ is a colorful solution. Since $q \leq 3p$, we have that the probability is lower bounded by e^{-3p}. This implies that if (G, T, p, r) is a yes-instance, then the probability that for a random coloring, no system of segments forming a solution is a colorful solution with respect to the coloring is upper bounded by $1 - e^{-3p}$. This immediately implies that if after trying e^{3p} random colorings there is no colorful solution for any of them, then the probability that (G, T, p, r) is a yes-instance is at most $(1 - e^{-3p})^{e^{3p}} < e^{-1} < 1$.

We construct a dynamic programming algorithm that decides whether there is a colorful solution. Denote by $c \colon V(G) \to \{1, \dots, q\}$ the considered random coloring.

In the first step of the algorithm, for each nonempty $X \subseteq \{1, \dots, q\}$ and distinct $i, j \in X$, we compute the Boolean function $\alpha(X, i, j)$ such that $\alpha(X, i, j) = \mathit{true}$ if and only if there are $s, t \in T$ and an (s, t)-path P such that P is a two-terminal T-segment, $|V(P)| = |X|$, $c(s) = i$, $c(t) = j$ and each vertex of P receives a unique color from X. We define $\alpha(X, i, j) = \mathit{false}$ if $|X| < 3$. For other cases, we use dynamic programming.

To compute $\alpha(X, i, j)$, we do the following auxiliary computations. For each $v \in V(G) \setminus T$ and each nonempty $Y \subseteq X \setminus \{i\}$, we compute the Boolean function $\beta(Y, i, v)$ such that $\beta(Y, i, v) = \mathit{true}$ if and only if there are $s \in T$ and an (s, v)-path

P' such that $V(P')\setminus\{s\}\subseteq V(G)\setminus T$, $c(s)=i$, $|V(P)\setminus\{s\}|=|Y|$, and each vertex of $V(P)\setminus\{s\}$ is colored by a unique color from Y.

We compute $\beta(Y,i,v)$ recursively starting with one-element sets. For every $Y=\{h\}$, where $h\neq i$, and every $v\in V(G)\setminus T$, we set $\beta(Y,i,v)=\mathit{true}$ if $c(v)=h$ and v is adjacent to a vertex of T colored i, and we set $\beta(Y,i,v)=\mathit{false}$ otherwise. For $Y\subseteq\{1,\dots,q\}\setminus\{i\}$ of size at least two, we set $\beta(Y,i,v)=\mathit{true}$ if $c(v)\in Y$ and there is $w\in N_G(v)\setminus T$ with $\beta(Y\setminus\{c(v)\},i,w)=\mathit{true}$, and $\beta(Y,i,v)=\mathit{false}$ otherwise.

We set $\alpha(X,i,j)=\mathit{true}$ if and only if there are $t\in T$ and $v\in N_G(t)\setminus T$ such that $c(t)=j$ and $\beta(X\setminus\{i,j\},i,v)=\mathit{true}$.

The correctness of computing β and α is proved by standard arguments in a straightforward way. Notice that we can compute the tables of values of β and α in time $2^q\cdot n^{\mathcal{O}(1)}$. First, we compute the values of $\beta(Y,i,v)$ for all $v\in V(G)\setminus T$, $i\in\{1,\dots,q\}$, and nonempty $Y\subseteq\{1,\dots,q\}\setminus\{i\}$. Then we use the already computed values of β to compute the table of values of α.

Next, we use the table of values of α to check whether a colorful solution exists. We introduce the Boolean function $\gamma_0(i,X,\ell,j)$ such that for each $i\in\{1,\dots,r\}$, $X\subseteq\{1,\dots,q\}$, integer $\ell\leq p$, and $j\in X$, $\gamma_0(i,X,\ell,j)=\mathit{true}$ if and only if there are paths $P_1,\dots,P_i$ and ordered pairs $(s_1,t_1),\dots,(s_i,t_i)$ of distinct vertices of T such that each P_h is an (s_h,t_h)-path and the following is fulfilled:

(i) $\{P_1,\dots,P_i\}$ is a system of T-segments,

(ii) $|\cup_{h=1}^{i}V(P_h)\setminus T|=\ell$,

(iii) if $1\leq f<g\leq i$, $u\in V(P_f)$, and $v\in V(P_g)$, then the vertices u and v have distinct colors unless $f=g-1$, $u=t_f$, and $v=s_g$ when the colors could be the same,

(iv) $c(t_i)=j$.

Notice that if $\ell<i$, then $\gamma_0(i,X,\ell,j)=\mathit{false}$. Our aim is to compute $\gamma_0(r,X,p,j)$ for $X\subseteq\{1,\dots,q\}$ and $j\in\{1,\dots,q\}$. Then we observe that a colorful solution exists if and only if there are $X\subseteq\{1,\dots,q\}$ and $j\in\{1,\dots,q\}$ such that $\gamma_0(r,X,p,j)=\mathit{true}$.

If $i=1$ and $\ell\geq 1$, then

$$\gamma_0(1,X,\ell,j)=\Big(\bigvee_{h\in X\setminus\{j\}}\alpha(X,h,j)\Big)\wedge\big(|X|=\ell+2\big). \tag{3.1}$$

For $\ell\geq i>1$, we use the following recurrence:

(3.2)
$$\begin{aligned}\gamma_0(i,X,\ell,j)=&\Big(\bigvee_{j\in Y\subset X,h\in Y\setminus\{j\}}(\alpha(Y,h,j)\wedge\gamma_0(i-1,(X\setminus Y)\cup\{h\},\ell-|Y|+2,h))\Big)\\&\vee\Big(\bigvee_{j\in Y\subset X,h\in Y\setminus\{j\},h'\in X\setminus Y}(\alpha(Y,h,j)\wedge\gamma_0(i-1,X\setminus Y,\ell-|Y|+2,h'))\Big).\end{aligned}$$

The correctness of (3.1) and (3.2) is proved by the standard arguments. Since the size of the table of values of α is $2^q\cdot n^{\mathcal{O}(1)}$ and the table can be constructed in time $2^q\cdot n^{\mathcal{O}(1)}$, we obtain that the values of $\gamma_0(r,X,p,j)$ for $X\subseteq\{1,\dots,q\}$ and $j\in\{1,\dots,q\}$ can be computed in time $3^q\cdot n^{\mathcal{O}(1)}$. To see this, note that we consider all $X\subseteq\{1,\dots,q\}$ and all $Y\subset X$. Hence, the number of considered pairs of sets X and Y is at most 3^q. Therefore, the existence of a colorful solution can be checked in time $3^q\cdot n^{\mathcal{O}(1)}$.

This leads us to a Monte Carlo algorithm for SEGMENTS WITH TERMINAL SET. We try at most e^{3p} random colorings. For each coloring, we check the existence of a colorful solution. If such a solution exists, we report that we have a yes-instance of the problem. If after trying e^{3p} random colorings we do not find a colorful solution for any of them, we return the answer no. As we already observed, the probability that this negative answer is false is at most $(1-e^{-3p})^{e^{3p}} < e^{-1} < 1$, that is, the probability is upper bounded by the constant $e^{-1} < 1$ that does not depend on the problem size and the parameter. The running time of the algorithm is $(3e)^{3p} \cdot n^{\mathcal{O}(1)}$.

The algorithm can be derandomized, as was explained in [2] (we also refer to [9] for the detailed introduction to the technique), by the replacement of random colorings by a family of *perfect hash functions*. Currently, the best explicit construction of such families was done by Naor, Schulman, and Srinivasan in [30]. The family of perfect hash function in our case has size $e^{3p}p^{O(\log p)} \log n$ and can be constructed in time $e^{3p}p^{O(\log p)} n \log n$ [30]. It immediately gives the deterministic algorithm for SEGMENTS WITH TERMINAL SET running in time $(3e)^{3p}p^{\mathcal{O}(\log p)} \cdot n^{\mathcal{O}(1)}$. ∎

4. Putting it all together: Final proofs.

In this section, we complete the proofs of Theorems 2 and 3. For this, we restate Theorem 2.

THEOREM 2. *On* 2*-connected graphs* LONGEST CYCLE ABOVE DEGENERACY *is solvable in time* $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

Proof. 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 1 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 $F_1, \dots, F_\ell$ be the connected components of $G[S]$. We contract the edges of each connected component $F_1, \dots, F_\ell$ and denote the graph obtained from G by these contractions by G'. We also denote by $u_1, \dots, u_\ell$ the vertices of G' obtained from $F_1, \dots, F_\ell$, respectively. It is straightforward to verify that G' has no cut vertices, that is, G' is 2-connected. For each $i \in \{1, \dots, \ell\}$, consider u_i. This vertex has at least two 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 $u_1v_1, \dots, u_\ell v_\ell$ 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.

Now we can assume that $|V(H)| < d+k$. By Lemma 3, 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 $\{P_1, \dots, P_r\}$ 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 2. To do so, 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 2.

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 4. 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. ∎

Combining Theorem 2 with the reduction from Lemma 1, we immediately obtain the restated Theorem 3.

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

5. Hardness for longest path and cycle above degeneracy. In this section we complement Theorems 3 and 2 by some hardness observations.

PROPOSITION 3. LONGEST PATH ABOVE DEGENERACY *is* NP-*complete even if* $k = 2$ *and* LONGEST CYCLE ABOVE DEGENERACY *is* NP-*complete even for connected graphs and* $k = 2$.

Proof. To show that LONGEST PATH ABOVE DEGENERACY is NP-complete for $k = 2$, consider a noncomplete n-vertex graph G. We construct a copy of the complete $(n-2)$-vertex graph K_{n-2}, pick an arbitrary vertex $v \in V(G)$, and make it adjacent to every vertex of K_{n-2}. Denote by G' the obtained graph. Because G is not a complete graph, $\mathrm{dg}(G') \leq n-2$. Therefore, $\mathrm{dg}(G') = n-2$, because $\mathrm{dg}(K_{n-1}) = n-2$. Observe that G' has a path with $\mathrm{dg}(G') + 2 = n$ vertices if and only if G is Hamiltonian. Since HAMILTONIAN PATH is a well-known NP-complete problem (see [15]), the claim follows. The claim for LONGEST CYCLE ABOVE DEGENERACY follows from the NP-completeness of LONGEST PATH ABOVE DEGENERACY and the reduction from Lemma 1. ∎

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 3 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 4. *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 from HAMILTONIAN PATH which 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 a copy of the complete graph K_p. Denote by $u_1, \ldots, u_p$ its vertices.
- For each $v \in V(G)$, construct an edge vu_1.
- Let $q = \lceil (1+\varepsilon)(p-1) - (n+p) \rceil$. Construct vertices $w_1, \ldots, w_q$ and edges u_1w_1, w_qu_2 and $w_{i-1}w_i$ for $i \in \{2, \ldots, q\}$.

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 $Q = vu_1w_1 \ldots w_qu_2u_3 \ldots u_p$. 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. Here it is more convenient to modify the above reduction instead of applying Lemma 1 that cannot be used directly. We again reduce from HAMILTONIAN PATH. 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 a copy of the complete graph K_p. Denote by $u_1, \ldots, u_p$ its vertices.
- 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 $w_1, \ldots, w_q$ and edges u_2w_1, w_qu_3 and $w_{i-1}w_i$ for $i \in \{2, \ldots, q\}$.

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 $Q = xu_2w_1 \ldots w_qu_3u_4 \ldots u_py$. 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 Dirac's theorem [10] (Theorem 4) and in the classical theorem of Erdős and Gallai [11] stating that every connected n-vertex graph G contains a path with at least $\min\{2\delta(G) + 1, n\}$ vertices. More precisely, what can be said about the parameterized complexity of the variants of LONGEST 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 [31] for related hardness results. Similar questions can be asked for LONGEST PATH (CYCLE) parameterized above the *average degree* (the *average degree* of a graph G is $\mathrm{ad}(G) = (\sum_{v \in V(G)} d_G(v))/|V(G)| =$

$2|E(G)|/|V(G)|$) using the property that a graph G has a path with at least $\mathrm{ad}(G)+1$ vertices (a cycle with at least $\mathrm{ad}(G)+1$ vertices if $\mathrm{ad}(G) \geq 2$) by the results of Erdős and Gallai [11].

Acknowledgments. We thank Nikolay Karpov for communicating to us the question of finding a path above the degeneracy bound and Proposition 3. We are also grateful to the anonymous reviewer of the journal SIDMA for the helpful suggestion that allowed us to improve the presentation of our results and shorten this paper.

REFERENCES

[1] N. Alon, G. Gutin, E. J. Kim, S. Szeider, and A. Yeo, *Solving MAX-r-SAT above a tight lower bound*, in Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, 2010, pp. 511–517, https://doi.org/10.1137/1.9781611973075.44.

[2] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. ACM, 42 (1995), pp. 844–856, https://doi.org/10.1145/210332.210337.

[3] I. Bezáková, R. Curticapean, H. Dell, and F. V. Fomin, *Finding detours is fixed-parameter tractable*, SIAM J. Discrete Math., 33 (2019), pp. 2326–2345, https://doi.org/10.1137/17M1148566.

[4] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, *Narrow Sieves for Parameterized Paths and Packings*, preprint, https://arxiv.org/abs/1007.1161, 2010.

[5] H. L. Bodlaender, *On linear time minor tests with depth-first search*, J. Algorithms, 14 (1993), pp. 1–23.

[6] J. Chen, J. Kneis, S. Lu, D. Mölle, S. Richter, P. Rossmanith, S.-H. Sze, and F. Zhang, *Randomized divide-and-conquer: Improved path, matching, and packing algorithms*, SIAM J. Comput., 38 (2009), pp. 2526–2547, https://doi.org/10.1137/080716475.

[7] J. Chen, S. Lu, S.-H. Sze, and F. Zhang, *Improved algorithms for path, matching, and packing problems*, in Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, 2007, pp. 298–307.

[8] R. Crowston, M. Jones, G. Muciaccia, G. Philip, A. Rai, and S. 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), Leibniz International Proceedings in Informatics (LIPIcs) 24, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013, pp. 43–54.

[9] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, Cham, 2015.

[10] G. A. Dirac, *Some theorems on abstract graphs*, Proc. London Math. Soc. (3), 2 (1952), pp. 69–81.

[11] P. Erdős and T. Gallai, *On maximal paths and circuits of graphs*, Acta Math. Acad. Sci. Hungar., 10 (1959), pp. 337–356 (unbound insert).

[12] F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh, *Efficient computation of representative families with applications in parameterized and exact algorithms*, J. ACM, 63 (2016), 29, https://doi.org/10.1145/2886094.

[13] F. V. Fomin, D. Lokshtanov, F. Panolan, S. Saurabh, and M. Zehavi, *Long directed* (s, t)*-path: FPT algorithm*, Inform. Process. Lett., 140 (2018), pp. 8–12, https://doi.org/10.1016/j.ipl.2018.04.018.

[14] H. N. Gabow and S. Nie, *Finding a long directed cycle*, ACM Trans. Algorithms, 4 (2008), 21.

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

[16] S. Garg and G. 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), SIAM, Philadelphia, 2016, pp. 1152–1166, https://doi.org/10.1137/1.9781611974331.ch80.

[17] G. Gutin, E. J. Kim, M. Lampis, and V. Mitsou, *Vertex cover problem parameterized above and below tight bounds*, Theory Comput. Syst., 48 (2011), pp. 402–410.

[18] G. Gutin, L. van Iersel, M. Mnich, and A. Yeo, *Every ternary permutation constraint satisfaction problem parameterized above average has a kernel with a quadratic number of variables*, J. Comput. Syst. Sci., 78 (2012), pp. 151–163.

[19] G. Z. Gutin and V. Patel, *Parameterized traveling salesman problem: Beating the average*, SIAM J. Discrete Math., 30 (2016), pp. 220–238, https://doi.org/10.1137/140980946.

[20] G. Z. Gutin, A. Rafiey, S. Szeider, and A. Yeo, *The linear arrangement problem parameterized above guaranteed value*, Theory Comput. Syst., 41 (2007), pp. 521–538, https://doi.org/10.1007/s00224-007-1330-6.

[21] F. Harary, *A characterization of block-graphs*, Canad. Math. Bull., 6 (1963), pp. 1–6.

[22] F. Hüffner, S. Wernicke, and T. Zichner, *Algorithm engineering for color-coding with applications to signaling pathway detection*, Algorithmica, 52 (2008), pp. 114–132, https://doi.org/10.1007/s00453-007-9008-7.

[23] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith, *Divide-and-color*, in Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG), Lecture Notes in Comput. Sci. 4271, Springer, Berlin, Heidelberg, 2008, pp. 58–67.

[24] I. Koutis, *Faster algebraic algorithms for path and packing problems*, in Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Comput. Sci. 5125, Springer, Berlin, 2008, pp. 575–586.

[25] D. Lokshtanov, N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh, *Faster parameterized algorithms using linear programming*, ACM Trans. Algorithms, 11 (2014), 15, https://doi.org/10.1145/2566616.

[26] M. Mahajan and V. Raman, *Parameterizing above guaranteed values: MaxSat and MaxCut*, J. Algorithms, 31 (1999), pp. 335–354.

[27] M. Mahajan, V. Raman, and S. Sikdar, *Parameterizing above or below guaranteed values*, J. Comput. System Sci., 75 (2009), pp. 137–153.

[28] D. W. Matula and L. L. Beck, *Smallest-last ordering and clustering and graph coloring algorithms*, J. ACM, 30 (1983), pp. 417–427, https://doi.org/10.1145/2402.322385.

[29] B. Monien, *How to find long paths efficiently*, in Analysis and Design of Algorithms for Combinatorial Problems (Udine, 1982), North-Holland Math. Stud. 109, North-Holland, Amsterdam, 1985, pp. 239–254, https://doi.org/10.1016/S0304-0208(08)73110-4.

[30] M. Naor, L. J. Schulman, and A. Srinivasan, *Splitters and near-optimal derandomization*, in Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS 1995), IEEE, 1995, pp. 182–191.

[31] I. Schiermeyer, *Problems remaining NP-complete for sparse or dense graphs*, Discuss. Math. Graph Theory, 15 (1995), pp. 33–41, https://doi.org/10.7151/dmgt.1004.

[32] D. Tsur, *Faster deterministic parameterized algorithm for k-*Path, Theoret. Comput. Sci., 790 (2019), pp. 96–104, https://doi.org/10.1016/j.tcs.2019.04.024.

[33] R. Williams, *Finding paths of length k in $O^*(2^k)$ time*, Inform. Process. Lett., 109 (2009), pp. 315–318.

[34] M. Zehavi, *A randomized algorithm for long directed cycle*, Inform. Process. Lett., 116 (2016), pp. 419–422, https://doi.org/10.1016/j.ipl.2016.02.005.