A Parameterized Complexity View on Collapsing k-Cores

Junjie Luo[1]
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Berlin, Germany
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China
junjie.luo@campus.tu-berlin.de, luojunjie@amss.ac.cn

Hendrik Molter[2]
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Berlin, Germany
h.molter@tu-berlin.de

Ondřej Suchý[3]
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Prague, Czech Republic
ondrej.suchy@fit.cvut.cz

Abstract

We study the NP-hard graph problem Collapsed k-Core where, given an undirected graph G and integers b, x, and k, we are asked to remove b vertices such that the k-core of remaining graph, that is, the (uniquely determined) largest induced subgraph with minimum degree k, has size at most x. Collapsed k-Core was introduced by Zhang et al. [AAAI 2017] and it is motivated by the study of engagement behavior of users in a social network and measuring the resilience of a network against user drop outs. Collapsed k-Core is a generalization of r-Degenerate Vertex Deletion (which is known to be NP-hard for all $r \geq 0$) where, given an undirected graph G and integers b and r, we are asked to remove b vertices such that the remaining graph is r-degenerate, that is, every its subgraph has minimum degree at most r.

We investigate the parameterized complexity of Collapsed k-Core with respect to the parameters b, x, and k, and several structural parameters of the input graph. We reveal a dichotomy in the computational complexity of Collapsed k-Core for $k \leq 2$ and $k \geq 3$. For the latter case it is known that for all $x \geq 0$ Collapsed k-Core is W[P]-hard when parameterized by b. We show that Collapsed k-Core is W[1]-hard when parameterized by b and in FPT when parameterized by $(b + x)$ if $k \leq 2$. Furthermore, we show that Collapsed k-Core is in FPT when parameterized by the treewidth of the input graph and presumably does not admit a polynomial kernel when parameterized by the vertex cover number of the input graph.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis, Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases r-Degenerate Vertex Deletion, Feedback Vertex Set, Fixed-Parameter Tractability, Kernelization Lower Bounds, Graph Algorithms, Social Network Analysis

Digital Object Identifier 10.4230/LIPIcs.IPEC.2018.7

Related Version A full version is available at [18], `https://arxiv.org/abs/1805.12453`.

[1] Supported by CAS-DAAD Joint Fellowship Program for Doctoral Students of UCAS.
[2] Supported by the DFG, project MATE (NI 369/17).
[3] Supported by grant 17-20065S of the Czech Science Foundation.

13th International Symposium on Parameterized and Exact Computation (IPEC 2018).
Editors: Christophe Paul and Michał Pilipczuk; Article No. 7; pp. 7:1–7:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements This research was initiated at the annual research retreat of the Algorithmics and Computational Complexity (AKT) group of TU Berlin, held in Darlingerode, Germany, from March 19 till March 23, 2018. The authors would like to thank Anne-Sophie Himmel for initial discussions leading to the results in this paper.

1 Introduction

In recent years, modelling user engagement in social networks has received substantial interest [23, 24]. A popular assumption is that a user engages in a social network platform if she has at least a certain number of contacts, say k, on the platform. Further, she is inclined to abandon the social network if she has less than k contacts [2, 7, 8, 11, 19, 25]. In compliance with this assumption, a suitable graph-theoretic model for the "stable" part of a social network is the so-called *k-core* of the social network graph, that is, the largest induced subgraph with minimum degree k [21].[4]

Now, given a stable social network, that is, a graph with minimum degree k, the departure of a user decreases the degree of her neighbors in the graph by one which then might be smaller than k for some of them. Following our assumption these users now will abandon the network, too. This causes a cascading effect of users dropping out (collapse) of the network until a new stable state is reached. From an adversarial perspective a natural question is how to maximally destabilize a competing social network platform by compelling b users to abandon the network. This problem was introduced as Collapsed k-Core by Zhang et al. [25] and the decision version is formally defined as follows.

Collapsed k-Core
Input: An undirected graph $G = (V, E)$, and integers b, x, and k.
Question: Is there a set $S \subseteq V$ with $|S| \leq b$ such that the k-core of $G - S$ has size at most x?

In the mentioned motivation, one would aim to minimize x for a given b and k. Alternatively, we can also interpret this problem as a measure for resilience agains user drop outs of a social network by determining the smallest b for a given k and x.

Related Work. In 2017 Zhang et al. [25] showed that Collapsed k-Core is NP-hard for any $k \geq 1$ and gave a greedy algorithm to compute suboptimal solutions for the problem. However, for $x = 0$ and any fixed k, solving Collapsed k-Core is equivalent to finding b vertices such that after removing said vertices, the remaining graph is $(k-1)$-degenerate[5]. This problem is known as r-Degenerate Vertex Deletion and it is defined as follows.

r-Degenerate Vertex Deletion
Input: An undirected graph $G = (V, E)$, and integers b and r.
Question: Is there a set $S \subseteq V$ with $|S| \leq b$ such that $G - S$ is r-degenerate?

It is easy to see that Collapsed k-Core is a generalization of r-Degenerate Vertex Deletion. In 2010 Mathieson [20] showed that r-Degenerate Vertex Deletion is NP-complete and W[P]-complete when parameterized by the budget b for all $r \geq 2$ even if the input graph is already $(r+1)$-degenerate and has maximum degree $2r+1$. In the mid-90s Abrahamson et al. [1] already claimed W[P]-completeness for r-Degenerate Vertex Deletion with $r = 2$ when parameterized by b under the name Degree 3 Subgraph

[4] Note that the k-core of a graph is uniquely determined.
[5] A graph G is r-degenerate if every subgraph of G has a vertex with degree at most r [10].

Annihilator. For $r = 1$, the problem is equivalent to Feedback Vertex Set and for $r = 0$ it is equivalent to Vertex Cover, both of which are known to be NP-complete and fixed-parameter tractable when parameterized by the solution size [9, 12]. The aforementioned results concerning r-Degenerate Vertex Deletion in fact imply the hardness results shown by Zhang et al. [25] for Collapsed k-Core.

Our Contribution. We complete the parameterized complexity landscape of Collapsed k-Core with respect to the parameters b, k, and x. Specifically, we correct errors in the literature [1, 20] concerning the W[P]-completeness of r-Degenerate Vertex Deletion when parameterized by b for $r = 2$. We clarify the parameterized complexity of Collapsed k-Core for $k \leq 2$ by showing W[1]-hardness for parameter b and fixed-parameter tractability for the combination of b and x. Together with previously known results, this reveals a dichotomy in the computational complexity of Collapsed k-Core for $k \leq 2$ and $k \geq 3$.

We present two single exponential linear time FPT algorithms, one for Collapsed k-Core with $k = 1$ and one for $k = 2$. In both cases the parameter is $(b + x)$. In particular, the algorithm for $k = 2$ runs in $O(1.755^{x+4b} \cdot n)$ time which means that it solves Feedback Vertex Set in $O(9.487^b \cdot n)$ time (here, b is the solution size of Feedback Vertex Set). Cao [5] independently developed a very similar algorithm for Feedback Vertex Set. In comparison, we additionally generalize the algorithm for Collapsed k-Core and give a more thorough running time analysis. To the best of our knowledge, despite of its simplicity this algorithm improves many previous linear time parameterized algorithm for Feedback Vertex Set [13, 17]. However, recently a linear time computable polynomial kernel for Feedback Vertex Set was shown [14], which together with e.g. [16] yields an even faster linear time parameterized algorithm.

Furthermore, we conduct a thorough parameterized complexity analysis with respect to structural parameters of the input graph. On the positive side, we show that Collapsed k-Core is fixed-parameter tractable when parameterized by the treewidth of the input graph and show that it presumably does not admit a polynomial kernel when parameterized by either the vertex cover number or the bandwidth of the input graph. We also show that the problem is fixed-parameter tractable when parameterized by the combination of the cliquewidth of the input graph and b or x. Further results include W[1]-hardness when parameterized by the clique cover number of the input graph and para-NP-hardness for the domination number of the input graph.

Due to space contraints, results marked with ($\star$) are (in parts) deferred to a full version [18].

2 Hardness Results from the Literature

In this section, we gather and discuss known hardness results for Collapsed k-Core. Recall that that Collapsed k-Core with $x = 0$ is the same problem as r-Degenerate Vertex Deletion with $r = k - 1$. Hence, the hardness of Collapsed k-Core was first established by Mathieson [20] who showed that r-Degenerate Vertex Deletion is NP-complete and W[P]-complete when parameterized by the budget b for all $r \geq 2$ even if the input graph is already $(r + 1)$-degenerate and has maximum degree $2r + 1$. However, in the proof of Mathieson [20] the reduction is incorrect for the case $r = 2$. Abrahamson et al. [1] claim W[P]-completeness for $r = 2$ but their reduction is also flawed. We provide counterexamples for both cases and show how to adjust the reduction of Mathieson [20] in a full version [18].

▶ **Theorem 1** (Corrected from [20]) ($\star$). *For any $r \geq 2$ r-Degenerate Vertex Deletion is NP-hard and W[P]-complete when parameterized by b, even if the degeneracy of the input graph is $r + 1$ and the maximum degree of the input graph is $2r + 1$.*

The following observation shows that the hardness result by Mathieson [20] (Theorem 1) easily transfers to Collapsed k-Core (also in the cases where $x \neq 0$).

▶ **Observation 2** (⋆). *Let $x' > 0$ be a positive integer. There is a reduction which transforms instances (G, b, x, k) of* Collapsed k-Core *with $x = 0$ into equivalent instances (G', b, x', k) of* Collapsed k-Core.

With that, we arrive at the following corollary.

▶ **Corollary 3.** Collapsed k-Core *is NP-hard and W[P]-hard when parameterized by b for all $x \geq 0$ and $k \geq 3$, even if the degeneracy of the input graph is $\max\{k, x-1\}$ and the maximum degree of the input graph is $\max\{2k-1, x-1\}$.*

Note that r-Degenerate Vertex Deletion is known to be NP-hard for all $r \geq 0$.[6] Hence, we also know that Collapsed k-Core is NP-hard for $k \leq 2$ and all $x \geq 0$. However, the parameterized complexity with respect to b is open in this case. We settle this in the next section.

3 Algorithms and Complexity for $k = 1$ and $k = 2$

In this section we investigate the parameterized complexity of Collapsed k-Core for the case that $k \leq 2$. Since Theorem 3 only applies for $k \geq 3$ we first show in the following that the problem is W[1]-hard with respect to the combination of b and $(n - x)$ for all $k \geq 1$. Furthermore, we present two algorithms; one that solves Collapsed k-Core with $k = 1$ and one for the $k = 2$ case. Both algorithms run in single exponential linear FPT-time with respect to the parameter combination $(b + x)$.

We first give a parameterized reduction from Clique to Collapsed k-Core. Note that since this hardness result holds for the combination of b and the dual parameter of x, it is incomparable to Theorem 3 even for $k \geq 3$.

▶ **Proposition 4** (⋆). Collapsed k-Core *is W[1]-hard when parameterized by the combination of b and $(n-x)$ for all $k \geq 1$, even if the input graph is bipartite and $\max\{2, k\}$-degenerate.*

Proof. We reduce from W[1]-hard problem Clique [9], where given a graph $G = (V, E)$ and an integer p, the task is to decide whether G contains a clique of size at least p. Let (G, p) be an instance of Clique and k be a given constant. We build an instance (G', b, x, k) of Collapsed k-Core as follows. We can assume that $p \leq |V(G)|$, as otherwise we can output a trivial no-instance. We further assume that each vertex of G has degree at least $p + 1$. Vertex of degree less than $p - 1$ is not part of a clique of size at least p, while for all vertices of degree $p - 1$ or p we can check in $O(n \cdot p^3)$ time whether there is a clique of size at least p containing any of them.

We let $V(G') = V \cup U \cup W$, where $V = V(G)$ are the vertices of G, $U = \{u_e^i \mid e \in E, i \in \{1, \ldots, k\}\}$, and $W = \{w_e^i \mid e \in E, i \in \{1, \ldots, k-1\}\}$. We also let $E(G') = E_U \cup E_W$, where $E_U = \{\{v, u_e^i\} \mid e \in E, i \in \{1, \ldots, k\}, v \in e\}$, and $E_W = \{\{u_e^i, w_e^{i'}\} \mid e \in E, i \in \{1, \ldots, k\}, i' \in \{1, \ldots, k-1\}\}$. We actually only introduce the sets W and E_W if $k \geq 2$.

Finally we set $b = p$ and $x = n - (p + (2k-1)\binom{p}{2})$, where $n = |V(G')|$ is the number of vertices of graph G'.

[6] Theorem 1 states NP-hardness of r-Degenerate Vertex Deletion for $r \geq 2$. Recall that for $r = 1$ r-Degenerate Vertex Deletion is equivalent to Feedback Vertex Set and for $r = 0$ it is equivalent to Vertex Cover, both of which are known to be NP-hard [12].

Algorithm 1: Algorithm for Collapsed k-Core with $k = 1$.

SolveRec(G, S, Q, b, x)
if $|S| > b$ *or* $|Q| > b + x$ **then return** No solution;
Let G' be the 1-core of $G \setminus S$.
if $|V(G')| \leq x$ **then return** S;
if $V(G') \subseteq Q$ **then return** No solution;
Let v be the vertex with the highest degree in G' which is not in Q
$T \leftarrow$ SolveRec($G, S \cup \{v\}, Q, b, x$)
if $T \neq$ No solution **then return** T;
else return SolveRec($G, S, Q \cup \{v\}, b, x$);

We claim that (G', b, x, k) is a yes-instance of Collapsed k-Core if and only if (G, p) is a yes-instance of Clique. We defer the proof of this claim to a full version [18]. ◀

Now we proceed with the algorithm for Collapsed k-Core with $k = 1$. While there is a simple algorithm with $O(3^{x+b}(m+n))$ running time[7] for this case, we consider an algorithm with the slightly worse running time as stated, since we then generalize this algorithm to the case $k = 2$ with some modifications.

▶ **Proposition 5** ($\star$). Collapsed k-Core *with* $k = 1$ *can be solved in* $O(2^{x+2b}(m+n))$ *time and in* $O(2^{O(b+\sqrt{bx})} \cdot n)$ *time. Assuming the Exponential Time Hypothesis, there is no* $2^{o(b)\cdot f(x)} n^{O(1)}$ *time algorithm for* Collapsed k-Core *with* $k = 1$*, for any function* f*.*

Algorithm: We present a recursive algorithm (see Algorithm 1 for pseudocode) that maintains two sets S and Q. The recursive function is supposed to return a solution to the instance, whenever there is a solution B containing all of S and there is no solution containing S and anything of Q. If some of the conditions is not met, then the function should return "No solution". In other words, S is the set of deleted vertices and Q is the set of vertices the algorithm has decided not to delete in the previous steps but may be collapsed in the future. Hence, the solution to the instance, or the information that there is none, is obtained by calling the recursive function with both sets S and Q empty.

The correctness proof and running time analysis of Algorithm 1 is deferred to a full version [18]. In the remainder of the section, we show how to adapt this algorithm for Collapsed k-Core with $k = 2$.

▶ **Theorem 6.** Collapsed k-Core *with* $k = 2$ *can be solved in* $O(1.755^{x+4b} \cdot n)$ *time and in* $O(2^{O(b+\sqrt{bx})} \cdot n)$ *time. Assuming the Exponential Time Hypothesis, there is no* $2^{o(b)\cdot f(x)} n^{O(1)}$ *time algorithm for* Collapsed k-Core *with* $k = 2$*, for any function* f*.*

The above theorem in particular yields an $O(9.487^b \cdot n)$ algorithm for Feedback Vertex Set.

Algorithm: Our algorithm for $k = 2$ is similar to Algorithm 1 with two main differences (see Algorithm 2 for pseudocode). First $|Q| > b + x$ is replaced by $|Q| > 3b + x$. Second when selecting the maximum degree vertex from $V(G') \setminus Q$, we need to make sure that this vertex

[7] An informal description of the algorithm: We use an initially empty set X that should contain vertices of the remaining 1-core. We branch over edges where both endpoints are not in X and either remove one of the endpoints or put both endpoint into X. Then we branch over all edges that have exactly one endpoint in X and either remove the other endpoint or put the other endpoint into X as well.

Algorithm 2: Algorithm for COLLAPSED k-CORE with $k = 2$.

1 SOLVEREC2(G, S, Q, b, x)
2 **if** $|S| > b$ *or* $|Q| > 3b + x$ **then return** No solution;
3 Let G' be the 2-core of $G \setminus S$.
4 **if** $|V(G')| \leq x$ **then return** S;
5 **if** $V(G') \subseteq Q$ **then return** No solution;
6 Let v be the vertex with the highest degree in G' which is not in Q
7 **if** $\deg_{G'}(v) \leq 2$ **then**
8   Let $C_1, \ldots, C_r$ be the connected components of G' not containing vertices of Q, ordered such that $|V(C_1)| \geq |V(C_2)| \geq \ldots \geq |V(C_r)|$;
9   Let $r' \leftarrow \min\{r, b - |S|\}$;
10   **if** $|V(G')| - \sum_{i=1}^{r'} |V(C_i)| \leq x$ **then**
11   **for** $i = 1, \ldots, r'$ **do** Select an arbitrary vertex from C_i and add it to S;
12   **return** S
13   **end**
14   **else return** No solution;
15 **end**
16 $T \leftarrow$ SOLVEREC2$(G, S \cup \{v\}, Q, b, x)$;
17 **if** $T \neq$ No solution **then return** T;
18 **else return** SOLVEREC2$(G, S, Q \cup \{v\}, b, x)$;

has degree greater than 2. Otherwise, either we can directly select vertices from $V(G') \setminus Q$ to break cycles in G' and get a 2-core of size at most x, or the algorithm rejects this branch.

We start by claiming that Algorithm 2 has the following running time.

▶ **Lemma 7** ($\star$). *Algorithm 2 runs in $O(1.755^{x+4b} \cdot n)$ time and in $O(2^{O(b+\sqrt{bx})} \cdot n)$ time.*

Next we claim the following conditional lower bound on the running time for any algorithm for COLLAPSED k-CORE with $k = 2$.

▶ **Lemma 8** ($\star$). *Assuming the Exponential Time Hypothesis, there is no $2^{o(b) \cdot f(x)} n^{O(1)}$ time algorithm for* COLLAPSED k-CORE *with $k = 2$, for any function f.*

Before showing the correctness of Algorithm 2, we give the following lemmata which will be helpful in the correctness proof. The following lemma claims that, except for some specific connected components, we can limit the solution to contain vertices of degree at least three.

▶ **Lemma 9** ($\star$). *For any instance $(G, b, x, 2)$ of* COLLAPSED k-CORE *with $k = 2$, where G is a 2-core and does not contain a cycle as a connected component, if there is a solution B for $(G, b, x, 2)$, then there is also a solution B' for $(G, b, x, 2)$ which contains only vertices with degree larger than 2.*

The next lemma helps to show that line 14 is correct.

▶ **Lemma 10** ($\star$). *If line 14 of Algorithm 2 applies, and for every $q \in Q$ there no solution containing whole $S \cup \{Q\}$, then there is no solution containing the whole S.*

Next, we show that the second part of line 2 of Algorithm 2 is correct.

▶ **Lemma 11.** *If Q is of size more than $3b + x$, then there is no solution containing whole S and no vertex from Q.*

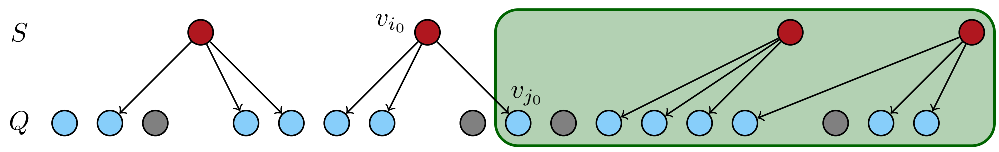

Figure 1 Illustration of function f. Vertices in Q are separated into two parts: gray vertices from X and blue vertices from $Q' = Q \setminus X$. Every vertex in S is mapped to a set of three consecutive blue vertices in Q' from the right to the left. Graph G_{j_0} with the property that $i > \max\{j \mid j \in f(i)\}$ for every i with $v_i \in B \cap V(G_{j_0})$ is contained in the green box.

Proof. Suppose for contradiction that there is a set Q of size at least $3b + x + 1$ and a solution B such that $|B| \le b$, $S \subseteq B$ and $B \cap Q = \emptyset$. Let $v_1, v_2, \ldots, v_{r'}$ be the vertices of the set $S \cup Q$ in the order as they were added to the set by successive recursive calls. Moreover, if $B \setminus S$ is empty, then let $r = r'$. Otherwise, let $B \setminus S = \{v_{r'+1}, \ldots, v_r\}$, i.e., in both cases $\{v_1, \ldots, v_r\} = B \cup Q$. Without loss of generality, we can assume that for every vertex $v_i \in B$, v_i is contained in the 2-core of $G \setminus (B \cap \{v_1, v_2, \ldots, v_{i-1}\})$, since otherwise v_i must come from $B \setminus S$, then $B' = B \setminus \{v_i\}$ is also a solution such that $|B'| \le b$, $S \subseteq B'$ and $B' \cap Q = \emptyset$. Let G' be the 2-core of $G \setminus B$ and let $X = V(G') \cap Q$. Since B is a solution, we know that $|X| \le x$.

We construct a function f that maps every vertex of B to a set of three consecutive vertices of $Q \setminus X$ (see also Figure 1). First let v_i be the vertex in B with the largest i. We let $f(i)$ be the set $\{j_1, j_2, j_3\}$, where $j_1 = \max\{j \mid v_j \in Q \setminus X\}$, $j_2 = \max\{j < j_1 \mid v_j \in Q \setminus X\}$, and $j_3 = \max\{j < j_2 \mid v_j \in Q \setminus X\}$. Now let v_i be the vertex from B with the largest i such that $f(i)$ was not set yet and $v_{i'}$ be the vertex from B with the least i' such that $i' > i$. We set $f(i) = \{j_1, j_2, j_3\}$, where $j_1 = \max\{j < \min\{k \in f(i')\} \mid v_j \in Q \setminus X\}$, $j_2 = \max\{j < j_1 \mid v_j \in Q \setminus X\}$, and $j_3 = \max\{j < j_2 \mid v_j \in Q \setminus X\}$. Since the set $Q \setminus X$ contains at least $3b + 1$ vertices, while B contains at most b, this way we find a mapping for every vertex in B, keeping the images of different vertices disjoint. Moreover, denote $p = |Q \setminus X| - 3|B| \ge 1$. There remains p vertices in $Q \setminus X$ not being in the union of images of f, let us denote them $v_{q_1}, \ldots, v_{q_p}$.

For $t \in \{1, \ldots, r\}$ let G_t be the 2-core of the graph $G \setminus (B \cap \{v_1, \ldots, v_{t-1}\})$. For $v \in V(G_t)$ let $\deg_t(v)$ be the degree of the vertex v in G_t. By the way we selected v_t we again know that $\deg_t(v_t) \ge \deg_t(v_{t'}) \ge \deg_{t'}(v_{t'})$ for every $t < t' \le r'$ and $\deg_t(v_t) \ge \deg_t(v_{t'}) \ge \deg_{t'}(v_{t'})$ for every $t \le r' < t'$. Note also that $\deg_t(v_t) \ge 3$ for all $v_t \in Q$.

If for every vertex $v_i \in B$ we have $i > \max\{j \mid j \in f(i)\}$, then $\deg_i(v_i) \le \deg_j(v_j)$ for vertex v_i in B and every $j \in f(i)$. Together with $\deg_t(v_t) \ge 3$ for all $v_t \in Q$, we have $\deg_i(v_i) - \sum_{j \in f(i)} \deg_j(v_j) + 6 \le 0$ for every vertex v_i in B. Let $V = B \uplus Q' \uplus X \uplus Y \uplus Z$, where $Q' = Q \setminus X$, Y is the set of collapsed vertices not contained in Q and $Z = V(G') \setminus X$ the part of the core of $G \setminus B$ not in Q. Now we transform graph G into a partial directed graph by considering the collapsing process (see also Figure 2). More precisely, for every edge in G, except for edges which have two endpoints in $X \cup Z$, we will assign it a direction. To this end, we just need to give an order of vertices in $V(G) \setminus (X \cup Z)$. We define this order based on the time vertices being deleted or collapsed. It may happen that several vertices in Q' or Y collapse at the same time. For this situation, we just order these vertices according to an arbitrary but fixed order. Since $k = 2$, every collapsed vertex in $Q' \cup Y$ has at most one outgoing edge. Let us consider the number, denoted by N, of edges of the form $\overrightarrow{v_i v_j}$ such that the head v_j is in Q'. Since every vertex in Q' has at most one outgoing edge, we have $N \ge \sum_{v_j \in Q'} \deg_j(v_j) - |Q'|$.

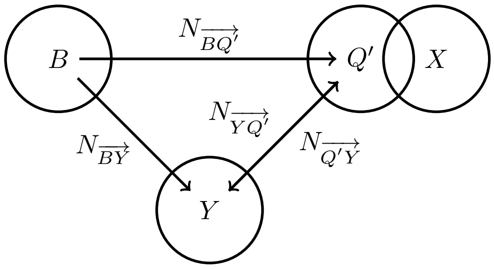

Figure 2 Illustration of the partial directed graph when considering the collapsing process. The set B contains the deleted vertices and X is the set of vertices remaining in the 2-core of $G \setminus B$. The set Q' contains vertices the algorithm has decided not to delete but eventually collapse and Y is the set of other collapsed vertices.

On the other hand, let

- $N_{\overrightarrow{BQ'}}$ be the number of edges going from B to Q';
- $N_{\overrightarrow{BY}}$ be the number of edges going from B to Y;
- $N_{\overrightarrow{YQ'}}$ be the number of edges going from Y to Q';
- $N_{\overrightarrow{Q'Y}}$ be the number of edges going from Q' to Y.

We claim that $N_{\overrightarrow{YQ'}} \leq N_{\overrightarrow{BY}} + N_{\overrightarrow{Q'Y}}$. To show that, denote the number of edges in $G[Y]$ by $N_{\overrightarrow{YY}}$. Since every vertex in Y has at least one incoming edge but at most one outgoing edge, we have $\sum_{v \in Y} \deg^-(v) \leq \sum_{v \in Y} \deg^+(v)$, where $\deg^+(v)$ ($\deg^-(v)$) is the number of incoming (outgoing) edges of vertex v in G, respectively. This means $N_{\overrightarrow{YQ'}} + N_{\overrightarrow{YY}} \leq N_{\overrightarrow{BY}} + N_{\overrightarrow{Q'Y}} + N_{\overrightarrow{YY}}$, therefore, $N_{\overrightarrow{YQ'}} \leq N_{\overrightarrow{BY}} + N_{\overrightarrow{Q'Y}}$, finishing the proof of the claim.

Since the edges which have their heads in Q' have their tails from $B \cup Y \cup Q'$,

$$
\begin{aligned}
N &\leq N_{\overrightarrow{BQ'}} + N_{\overrightarrow{YQ'}} + (|Q'| - N_{\overrightarrow{Q'Y}}) \leq N_{\overrightarrow{BQ'}} + |Q'| + N_{\overrightarrow{BY}} \\
&\leq \sum_{v_i \in B} \deg^-(v_i) + |Q'| \leq \sum_{v_i \in B} \deg_i(v_i) + |Q'|.
\end{aligned}
$$

The last inequality holds since for every vertex $v_i \in B$, v_i is contained in the 2-core of $G \setminus (B \cap \{v_1, v_2, \ldots, v_{i-1}\})$, which means all outgoing edges of v_i are counted in $\deg_i(v_i)$. Then we have $\sum_{v_i \in B} \deg^-(v_i) \leq \sum_{v_i \in B} \deg_i(v_i)$.

$$
\begin{aligned}
0 &\leq \sum_{v_i \in B} \deg_i(v_i) + |Q'| - \Big(\sum_{v_j \in Q'} \deg_j(v_j) - |Q'| \Big) \\
&= \sum_{v_i \in B} \deg_i(v_i) - \sum_{v_j \in Q'} (\deg_j(v_j) - 2) \\
&\leq \sum_{v_i \in B} \deg_i(v_i) - \Big(\sum_{v_i \in B} \sum_{j \in f(i)} (\deg_j(v_j) - 2) + \sum_{i=1}^{p} (\deg_{q_i}(v_{q_i}) - 2) \Big) \\
&< \sum_{v_i \in B} \deg_i(v_i) - \sum_{v_i \in B} \sum_{j \in f(i)} (\deg_j(v_j) - 2) \\
&= \sum_{v_i \in B} \Big(\deg_i(v_i) - \sum_{j \in f(i)} \deg_j(v_j) + 6 \Big) \leq 0
\end{aligned}
$$

which is a contradiction.

The remaining case where there exist a i such that $i < \max\{j \mid j \in f(i)\}$ is analogous and deferred to a full version [18]. ◀

The last part of the proof of Theorem 6 is deferred to a full version [18].

4 Structural Graph Parameters

In this section, we investigate the parameterized complexity of Collapsed k-Core with respect to several structural parameters of the input graph. Theorem 3 already implies hardness for constant values of several structural graph parameters. We expand this picture by observing that the problem remains NP-hard on graphs with a dominating set of size one and by showing that the problem is W[1]-hard when parameterized by the combination of b and the clique cover number of the input graph. On the positive side, we show that the problem is in FPT when parameterized by the treewidth of the input graph or the clique-width of the input graph and k combined with either b, x, $n-x$, or $n-b$. Lastly, we show that the problem presumably does not admit a polynomial kernel for any $k \geq 2$ when parameterized by the combination of b and the vertex cover number of the input graph, or by the combination of b, k, and the bandwidth of the input graph.

We start with an easy observation that we will make use of in most of the hardness results in this section.

▶ **Observation 12** ($\star$). *If (G, b, x, k) is an instance of* Collapsed k-Core *and vertex v is a part of the $(k+b)$-core of G, and $S \subseteq V$ is of size at most b, then either $v \in S$ or v is part of the k-core of $G \setminus S$.*

The following observation yields that we can reduce the size of a dominating set of any instance of Collapsed k-Core to one by introducing a universal vertex. Note that, for example, this only increases the degeneracy by one.

▶ **Observation 13** ($\star$). *Let (G, b, x, k) be an instance of* Collapsed k-Core *and G' be the graph obtained from G by adding a universal vertex, then $(G', b+1, x, k)$ is an equivalent instance of* Collapsed k-Core.

Considering a larger parameter than, e.g., the size of the dominating set, namely the clique cover number[8], we can show W[1]-hardness, even in combination with b. This can be done with a parameterized reduction from Multicolored Clique parameterized by the solution size.

▶ **Proposition 14** ($\star$). Collapsed k-Core *is W[1]-hard when parameterized by the combination of b and the clique cover number of the input graph.*

On the positive side, we sketch a dynamic program on the tree decomposition of the input graph G which implies that Collapsed k-Core is in FPT when parameterized by the treewidth of the input graph.

▶ **Proposition 15** ($\star$). Collapsed k-Core *is in FPT when parameterized by the treewidth of the input graph.*

Proof Sketch. Let (G, b, x, k) be an instance of Collapsed k-Core. Observe that either $k \leq \text{tw}(G)$ or the k-core of G is (already) empty and we can answer Yes. Hence, for the rest of the proof we assume that $k \leq \text{tw}(G)$. We assume that we are given a nice tree decomposition

[8] The clique cover number of a graph G is the minimum number of induced cliques such that their union contains all vertices of G.

of G [15, 3] and use dynamic programming on the nice tree decomposition of G. The indices of the table are formed for each bag of the decomposition by the number of vertices of the solution already forgotten, the number of vertices in the core already forgotten, a partition of the bag into three set B, X, and Q, an (elimination) order for the vertices in Q, and for each vertex in Q the number of its neighbors in X or higher in the order. This number is always in $0, \dots, k-1$, as otherwise it would not be possible to eliminate the vertex.

The set B represents the partial solution (or rather its intersection with the bag), i.e., the vertices to be deleted. The set X represents the vertices which (are free to) remain in the core. The vertices in Q should collapse after removing the vertices of the solution and the collapse of the vertices preceding them in the order.

There are $3^{\text{tw}(G)} \cdot (\text{tw}(G))^{O(\text{tw}(G))} \cdot k^{\text{tw}(G)} = (\text{tw}(G))^{O(\text{tw}(G))}$ possible indices for each bag. Hence the slightly superexponential running time of $(\text{tw}(G))^{O(\text{tw}(G))} \cdot n^{O(1)}$ follows. ◀

Using monadic second order (MSO) logic formulas, it can be shown that for a smaller structural parameter, namely the cliquewidth of the input graph, there are also positive results. Here however, we can only show fixed-parameter tractability for the combination of the cliquewidth of the input graph with k and either b, x, $n-x$, or $n-b$.

▶ **Proposition 16** ($\star$). Collapsed k-Core *is in FPT when parameterized by the cliquewidth of the input graph combined with k and either b, x, $n-x$, or $n-b$.*

In the remainder of this section, we show that Collapsed k-Core does not admit a polynomial kernel when parameterized by rather large parameter combinations. We first show an OR-cross composition [4, 9] from Cubic Vertex Cover.

▶ **Theorem 17.** *For all $k \geq 2$* Collapsed k-Core *does not admit a polynomial kernel when parameterized by the combination of b and the vertex cover number of the input graph unless NP $\subseteq$ coNP/poly.*

Proof. We apply an OR-cross composition [4, 9] from the NP-hard problem Cubic Vertex Cover [12]. In Cubic Vertex Cover, we are given a 3-regular graph G and an integer s, and the task is to find a vertex subset of size at most s which contains at least one endpoint of each edge of G.

We say an instance of Cubic Vertex Cover is *malformed* if the string does not represent a pair (G, s), where G is a 3-regular graph and s is a positive integer. It is *trivial*, if $s \geq |V(G)|$. We define the equivalence relation $\mathcal{R}$ as follows: all malformed instances are equivalent, all trivial instances are equivalent and two well-formed non-trivial instances (G, s) and (G', s') are $\mathcal{R}$-equivalent if $|V(G)| = |V(G')|$ and $s = s'$. Observe that $\mathcal{R}$ is a polynomial equivalence relation.

Let the input consist of T $\mathcal{R}$-equivalent instances of Cubic Vertex Cover. If the instances are malformed or trivial, we return a constant size no- or yes- instance of Collapsed k-Core, respectively. Let $(G_i, s)_{0 \leq i \leq T-1}$ be well-formed non-trivial $\mathcal{R}$-equivalent instances of Cubic Vertex Cover. Since all instances have the same size of the vertex set, we can assume they share the same vertex set $V = \{v_1, v_2, \dots, v_n\}$. We assume T to be a power of 2, as otherwise we can duplicate some instances. Now we create an instance (G, b, x, k) of Collapsed k-Core for some arbitrary but fixed $k \geq 2$ as follows.

- Set $b = s + 2ks \log T$ and $x = N - N'$, where $N = n + \frac{3}{2}nT + 4ks \log T + k + s + \frac{3}{2}n(k-2)$ is the number of all vertices in graph G we will construct and $N' = s + 2ks \log T + \frac{3}{2}n$.
- First for every vertex v_i in V, create a vertex v_i in G.

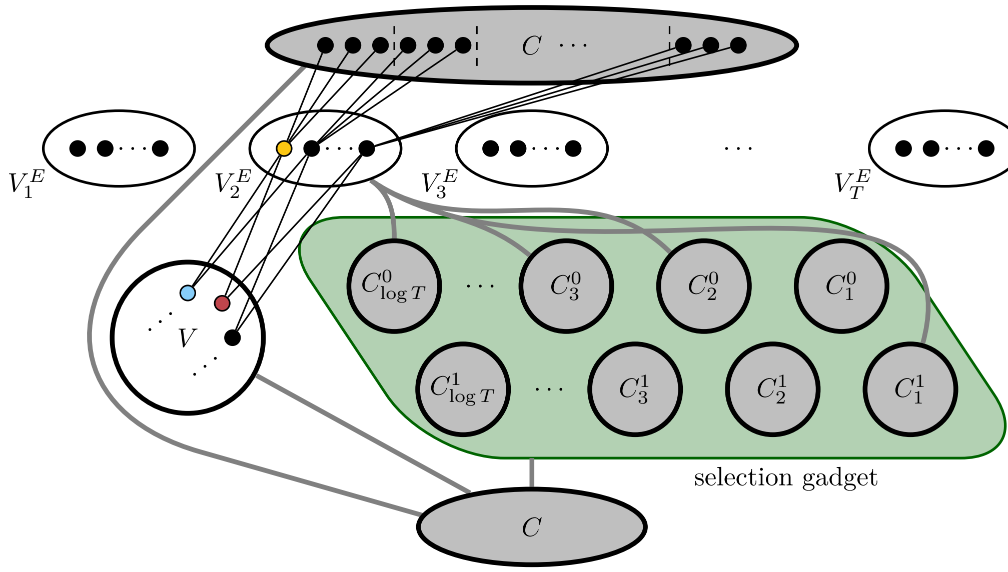

Figure 3 Illustration of the OR-cross composition from Cubic Vertex Cover to Collapsed k-Core with $k = 5$. The selection gadget is all the circles contained in the green box. Every gray edge in this figure means that all vertices in one endpoint of this edge are connected to all vertices in the other endpoint. Every vertex in V_i^E connects to two endpoints of its corresponding edge in G_i. For example, the yellow vertex in V_2^E is connected to the blue and the red vertex in V, which represents the two endpoints of the corresponding edge in G_2. The big clique C is separated into two parts. Every vertex in V_i^E is connected to $k-2$ vertices in the upper part of C. Since $k = 5$, the yellow vertex in V_2^E is connected to three vertices in C. Vertices contained in thick outlined vertex sets form a vertex cover. To keep the picture simple, edges that contain vertices from V_i^E with $i \neq 2$ are not depicted.

- For every edge set $E(G_i)$, create a vertex set V_i^E in G, in which a vertex $v_{p,q}$ represents an edge v_pv_q in $E(G_i)$. Then we have T of these vertex sets and each set has $\frac{3}{2}n$ vertices.
- For every edge v_pv_q in $E(G_i)$, add 2 edges $v_{p,q}v_p$ and $v_{p,q}v_q$ in G.
- Now create the selection gadget in G. It contains $\log T$ pairs of cliques C_i^d ($1 \leq i \leq \log T, d \in \{0,1\}$), and all of them have the same size of $2ks$. For every vertex set V_i^E, let $i = (d_{\log T-1}d_{\log T-2}\dots d_0)_2$ be the binary representation of the index i, where $d_j \in \{0,1\}$ for $0 \leq j \leq \log T - 1$ and we add leading zeros so that the length of the representation is exactly $\log T$. We add edges between all vertices in V_i^E and all vertices in $\bigcup_{j=0}^{\log T-1} C_j^{d_j}$.
- Finally we create a clique C with $|C| = k + b + \frac{3}{2}n(k-2)$, which contains two parts of vertices. The first part contains $k+b$ vertices and each of them connects to all vertices in $V \cup \bigcup_{j=0}^{\log T-1}\bigcup_{d=0}^{1} C_j^d$. The second part of $\frac{3}{2}n(k-2)$ vertices is connected to vertices in V_i^E in the following way. For every vertex $v_{p,q}$ in V_i^E, add edges between $v_{p,q}$ and $k-2$ vertices in C. We make sure that all vertices in the same V_i^E connect to different vertices in C. In other words, every vertex in the second part of C connects to exactly one vertex in every V_i^E.

Notice that the vertex cover number of G is $n+4ks\log T+k+b+\frac{3}{2}n(k-2)$. The construction is illustrated in Figure 3. We now show that at least one instance (G_i, s) is a yes-instance if and only if the instance (G, b, x, k) of Collapsed k-Core constructed above is a yes-instance.

$\Rightarrow$: If (G_i, s) is a yes-instance, which means that there is a vertex subset V^* of size s that covers all edges in G_i, then we delete the corresponding s vertices in G and all vertices in $\bigcup_{j=0}^{\log T-1} C_j^{d_j}$, where $i = (d_{\log T-1}\dots d_0)_2$ is the binary representation of i. So far, we deleted

$s + 2ks\log T$ vertices, and all vertices in V_i^E will collapse, since they just have at most $k-1$ edges remaining, $k-2$ of which connect to vertices in C and at most one to vertices in V. Therefore, the number of remaining vertices is x and instance (G, b, x, k) is a yes-instance.

$\Leftarrow$: If (G, b, x, k) is a yes-instance, we need to show that there is at least one instance which has a vertex cover of size at most s. Let S be the set of deleted vertices of size at most b and let S' be the set of all collapsed vertices. Since $N' = s + 2ks\log T + \frac{3}{2}n$, we have $|S'| \geq \frac{3}{2}n$. In the subgraph of G induced by V, C and $\bigcup_{j=0}^{\log T-1}\bigcup_{d=0}^{1} C_j^d$ all vertices in $V \cup \bigcup_{j=0}^{\log T-1}\bigcup_{d=0}^{1} C_j^d \cup C$ have degree larger than $k+b$. Hence, by Theorem 12 they will not collapse and all collapsed vertices come from $\bigcup_{i=0}^{T-1} V_i^E$.

We show that all collapsed vertices can only come from one single V_i^E for some i. Suppose two vertices v and v' from different sets of V_i^E $(0 \leq i \leq T-1)$ collapse after deleting S, then there is at least one pair of cliques $C_{j_0}^0$ and $C_{j_0}^1$ such that v is connected to all vertices in $C_{j_0}^{d_0}$ for some $d_0 \in \{0,1\}$ and v' is connected to all vertices in $C_{j_0}^{1-d_0}$. To make v collapse, at least $2ks\log T - (k-1)$ vertices from the corresponding cliques in the selection gadget need to be deleted. Then to make v' collapse, at least $2ks - (k-1)$ vertices from $C_{j_0}^{1-d_0}$ need to be deleted. Therefore, at least $2ks\log T + 2ks - 2(k-1)$ vertices need to be deleted, which is strictly more than b. This means that the collapsed vertices come from one single V_i^E. Since $|S'| \geq \frac{3}{2}n$ and $|V_i^E| = \frac{3}{2}n$, we have $S' = V_i^E$ and $S \cap V_i^E = \emptyset$ for some i.

We consider the vertex set S. We know that after deleting S, all vertices in V_i^E collapse. Denote V_I the vertex set of all vertices in $\bigcup_{j=0}^{\log T-1} C_j^{d_j}$, where $i = (d_{\log T-1}\dots d_0)_2$ is the binary representation of i. Since every vertex in V_I is connected to all vertices in V_i^E, to make V_i^E collapse, it is always better to choose vertices from V_I than any other vertex. If S does not contain all vertices from V_I, we can update S by replacing any $|V_I \setminus S|$ vertices in S with vertices in $V_I \setminus S$. Then $V_I \subseteq S$.

Suppose there is a vertex $v_{p,q}$ in V_i^E such that both v_p and v_q are not in $S \cap V$, then S contains at least one vertex v_c in C connected to $v_{p,q}$, as otherwise $v_{p,q}$ has degree at least k and will not collapse. We update S by replacing v_c with v_p. This will not influence the size of S and more importantly, this will not influence the collapsed set $S' = V_i^E$, since v_c in C is connected to only one vertex $v_{p,q}$ in V_i^E, and $v_{p,q}$ will still collapse under the new S. By updating S in the same way for other vertices in V_i^E not covered by vertices in $S \cap V$, we get a vertex set $S \cap V$ which covers all vertices in V_i^E at least once. And $|S \cap V| \leq s$, since $V_I \subseteq S$ and $|V_I| = 2ks\log T$. This corresponds to a vertex cover of size s in G_i. ◀

Lastly, we claim that there is a simple OR-cross composition [4, 9] from Collapsed k-Core onto itself. Note that the parameter combination of the following result is incomparable to that of Theorem 17.

▶ **Proposition 18** (⋆). Collapsed k-Core *does not admit a polynomial kernel when parameterized by the combination of b, k, and the bandwidth of the input graph unless NP ⊆ coNP/poly.*

5 Conclusion

Our results highlight a dichotomy in the computational complexity of Collapsed k-Core for $k \leq 2$ and $k \geq 3$. Along the way, we correct some inaccuracies in the literature concerning the parameterized complexity of Collapsed k-Core with $k = 3$ and $x = 0$ and give a simple single exponential linear time parameterized algorithm for Collapsed k-Core with $k = 2$, which almost matches the simplest known, independently found, single exponential linear time algorithm for Feedback Vertex Set. We leave as an open question whether

Collapsed k-Core with $k = 2$ on graphs of maximum degree 3 is polynomial time solvable similarly as Feedback Vertex Set [6, 22]. This would improve also the running time of our algorithm for Collapsed k-Core with $k = 2$ on general graphs. We further investigate the parameterized complexity with respect to several structural parameters of the input graph. As a highlight we show that Collapsed k-Core does not admit polynomial kernels for rather large parameter combinations. We leave the complexity of Collapsed k-Core when parameterized solely by the cliquewidth of the input graph open.

References

1. Karl A. Abrahamson, Rodney G. Downey, and Michael R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1995.
2. Kshipra Bhawalkar, Jon Kleinberg, Kevin Lewi, Tim Roughgarden, and Aneesh Sharma. Preventing unraveling in social networks: the anchored k-core problem. *SIAM Journal on Discrete Mathematics*, 29(3):1452–1475, 2015.
3. Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A $c^k n$ 5-Approximation Algorithm for Treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.
4. Hans L Bodlaender, Bart MP Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics*, 28(1):277–305, 2014.
5. Yixin Cao. A Naive Algorithm for Feedback Vertex Set. In *Proceedings of the 1st Symposium on Simplicity in Algorithms, (SOSA '18)*, volume 61 of *OASICS*, pages 1:1–1:9. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
6. Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: New measure and new structures. *Algorithmica*, 73(1):63–86, 2015.
7. Rajesh Chitnis, Fedor V. Fomin, and Petr A. Golovach. Parameterized complexity of the anchored k-core problem for directed graphs. *Information and Computation*, 247:11–22, 2016.
8. Rajesh Hemant Chitnis, Fedor V. Fomin, and Petr A. Golovach. Preventing Unraveling in Social Networks Gets Harder. In *Proceedins of the 27th AAAI Conference on Artificial Intelligence (AAAI '13)*, pages 1085–1091. AAAI Press, 2013.
9. Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
10. Reinhard Diestel. *Graph Theory, 5th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2016.
11. David Garcia, Pavlin Mavrodiev, and Frank Schweitzer. Social resilience in online communities: The autopsy of friendster. In *Proceedings of the 1st ACM Conference on Online Social Networks (COSN '13)*, pages 39–50. ACM, 2013.
12. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
13. Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72(8):1386–1396, 2006.
14. Yoichi Iwata. Linear-Time Kernelization for Feedback Vertex Set. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 68:1–68:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2017.68.

15 Ton Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.

16 Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Information Processing Letters*, 114(10):556–560, 2014.

17 Daniel Lokshtanov, MS Ramanujan, and Saket Saurabh. Linear Time Parameterized Algorithms for Subset Feedback Vertex Set. *ACM Transactions on Algorithms (TALG)*, 14(1):7, 2018.

18 Junjie Luo, Hendrik Molter, and Ondřej Suchý. A Parameterized Complexity View on Collapsing k-Cores. *CoRR*, 2018. arXiv:1805.12453.

19 Fragkiskos D. Malliaros and Michalis Vazirgiannis. To stay or not to stay: modeling engagement dynamics in social graphs. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)*, pages 469–478. ACM, 2013.

20 Luke Mathieson. The parameterized complexity of editing graphs for bounded degeneracy. *Theoretical Computer Science*, 411(34-36):3181–3187, 2010.

21 Stephen B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

22 Shuichi Ueno, Yoji Kajitani, and Shin'ya Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1-3):355–360, 1988.

23 Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. Recommending Groups to Users Using User-Group Engagement and Time-Dependent Matrix Factorization. In *Proceedins of the 30th AAAI Conference on Artificial Intelligence (AAAI '16)*, pages 1331–1337. AAAI Press, 2016.

24 Shaomei Wu, Atish Das Sarma, Alex Fabrikant, Silvio Lattanzi, and Andrew Tomkins. Arrival and departure dynamics in social networks. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pages 233–242. ACM, 2013.

25 Fan Zhang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. Finding Critical Users for Social Network Engagement: The Collapsed k-Core Problem. In *Proceedins of the 31st AAAI Conference on Artificial Intelligence (AAAI '17)*, pages 245–251. AAAI Press, 2017.