A Game Theoretic Approach For Core Resilience

Sourav Medya[1] , **Tiyani Ma**[2] , **Arlei Silva**[3] and **Ambuj Singh**[3]
[1]Northwestern University
[2]University of California Los Angeles
[3]University of California Santa Barbara
sourav.medya@kellogg.northwestern.edu, tianyima@cs.ucla.edu, {arlei,ambuj}@cs.ucsb.edu

Abstract

K-cores are maximal induced subgraphs where all vertices have degree at least k. These dense patterns have applications in community detection, network visualization and protein function prediction. However, k-cores can be quite unstable to network modifications, which motivates the question: *How resilient is the k-core structure of a network, such as the Web or Facebook, to edge deletions?* We investigate this question from an algorithmic perspective. More specifically, we study *the problem of computing a small set of edges for which the removal minimizes the k-core structure of a network*. This paper provides a comprehensive characterization of the hardness of the k-core minimization problem (KCM), including innaproximability and parameterized complexity. Motivated by these challenges, we propose a novel algorithm inspired by Shapley value—a cooperative game-theoretic concept— that is able to leverage the strong interdependencies in the effects of edge removals in the search space. We efficiently approximate Shapley values using a randomized algorithm with probabilistic guarantees. Our experiments show that the proposed algorithm outperforms competing solutions in terms of k-core minimization while being able to handle large graphs. Moreover, we illustrate how KCM can be applied in the analysis of the k-core resilience of networks.

1 Introduction

K-cores play an important role in revealing the higher-order organization of networks. A k-core [Seidman, 1983] is a maximal induced subgraph where all vertices have internal degree of at least k. These cohesive subgraphs have been applied to model users' engagement and viral marketing in social networks [Bhawalkar *et al.*, 2015]. Other applications include anomaly detection [Shin *et al.*, 2016], community discovery [Peng *et al.*, 2014], and visualization [Carmi *et al.*, 2007]. However, the k-core structure can be quite unstable under network modification. For instance, removing only a few edges might lead to the collapse of the core structure of a graph. This motivates the k-core minimization problem: *Given a graph G and constant k, find a small set of b edges for which the removal minimizes the size of the k-core structure [Zhu* et al.*, 2018].*

We motivate k-core minimization using the following applications: (1) *Monitoring:* Given an infrastructure or technological network, which edges should be monitored for attacks [Laishram *et al.*, 2018]? (2) *Defense:* Which communication channels should be blocked in a terrorist network in order to destabilize its activities [Perliger and Pedahzur, 2011]? and (3) *Design:* How to prevent unraveling in a social or biological network by strengthening connections between pairs of nodes [Bhawalkar *et al.*, 2015]?

Consider a specific application of k-cores in online social networks (OSNs). OSN users tend to perform activities (e.g., joining a group, playing a game) if enough of their friends do the same [Burke *et al.*, 2009]. Thus, strengthening critical links between users is key to the popularity, and even survival, of the network [Farzan *et al.*, 2011]. This scenario can be modeled using k-cores. Initially, everyone is engaged in the k-core. The removal of a few links (e.g., unfriending, unfollowing) might not only cause a couple of users to leave the network but produce a mass exodus due to cascading effects. This process can help us to understand the decline and death of OSNs such as Friendster [Garcia *et al.*, 2013].

K-core minimization (KCM) can be motivated both from the perspective of a centralized agent who protects the structure of a network or an adversary that aims to disrupt it. Moreover, our problem can also be applied to measure network resilience [Laishram *et al.*, 2018].

There is no polynomial time algorithm that achieves a constant-factor approximation for KCM. This behavior differs from more popular problems in graph combinatorial optimization, such as submodular optimization, where a simple greedy algorithm provides constant-factor approximation guarantees. The algorithm proposed in this paper applies the concept of *Shapley values* (SVs), which, in the context of cooperative game theory, measure the contribution of players in coalitions [Shapley, 1953]. Our algorithm selects edges with largest Shapley value to account for the joint effect (or cooperation) of multiple edges in the solution set.

Recent papers have introduced the KCM problem [Zhu *et al.*, 2018] and its vertex version [Zhang *et al.*, 2017], where the goal is to delete a few vertices such that the k-core structure is minimized. However, our work provides a stronger theoretical analysis and more effective algorithms that can be applied to both problems. In particular, we show that our algorithm outperforms the greedy approach proposed by a recent work [Zhu *et al.*, 2018].

Our main contributions are summarized as follows:

- We study the k-core minimization (KCM) problem, which consists of finding a small set of edges, removal of which minimizes the size of the k-core structure of a network.
- We show that KCM is NP-hard, even to approximate by a constant for $k \geq 3$. We also discuss the parameterized complexity of KCM and show the problem is $W[2]$-hard and para-NP-hard for different parameters.
- Given the above inapproximability result, we propose a randomized Shapley Value based algorithm that efficiently accounts for the interdependence among the candidate edges for removal.
- We show the accuracy and efficiency of our algorithm using several datasets. Moreover, we illustrate how KCM can be applied to profile the resilience of real networks.

Due to space limitations some details, proofs, and results are in an extended version [Medya *et al.*, 2019].

2 Problem Definition

We assume $G(V, E)$ to be an undirected and unweighted graph with sets of vertices V ($|V| = n$) and edges E ($|E| = m$). Let $d(G, u)$ denote the degree of vertex u in G. An induced subgraph, $H = (V_H, E_H) \subset G$ is such that if $u, v \in V_H$ and $(u, v) \in E$ then $(u, v) \in E_H$. The k-core [Seidman, 1983] of a network is defined below.

Definition 1. ***k-Core:*** *The k-core of a graph G, denoted by $C_k(G) = (V_k(G), E_k(G))$, is defined as the maximal induced subgraph that has vertices with degree at least k.*

K-core decomposition can be performed in time $O(m)$ by recursively removing vertices with degree lower than k [Batagelj and Zaveršnik, 2011].

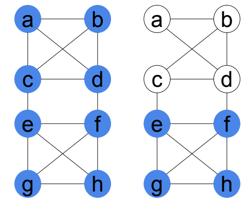
Let $G^B = (V, E \setminus B)$ be the modified graph after deleting a set B of b edges. Deleting an edge reduces the degree of two vertices and possibly removes other vertices from the k-core. For instance, the vertices in a simple cycle are in the 2-core but deleting any edge from the graph moves all the vertices to the 1-core. Let $N_k(G) = |V_k(G)|$ and $M_k(G) = |E_k(G)|$ be the number of nodes and edges respectively in $C_k(G)$.

Definition 2. ***Reduced k-Core:*** *A reduced k-core, $C_k(G^B)$ is the k-core in G^B, where $G^B = (V, E \setminus B)$.*

Example 1. *Figures 1 shows an initial graph, G, and modified graph G^B (where $B = \{(a, c)\}$). In G, all the nodes are in the 3-core. Deleting (a, c) brings the vertices a and c to the 2-core and thus b and d also go to the 2-core.*

Definition 3. ***K-Core Minimization (KCM):*** *Given a candidate edge set Γ, find the set, $B \subset \Gamma$ of b edges to be removed such that $N_k(G^B)$ is minimized, or, $f_k(B) = N_k(G) - N_k(G^B)$ is maximized.*

Fig. 1a shows an initial graph, G, where the nodes are in the 3-core. Deleting (a, c) and (e, g) brings all the vertices to the 2-core, whereas deleting (e, c) and (d, f) has no effect on the 3-core structure (assuming $b = 2$). Clearly, the importance of the edges varies in affecting the k-core upon their removal. Next, we discuss strong inapproximability results for the KCM problem along with parameterized complexity.



(a) Initial (b) Modified

Figure 1: Example of the changes in the core structure via deletion of an edge: (a) All the nodes are in the 3-core. (b) In the modified graph, the nodes $\{a, b, c, d\}$ are in the 2-core.

2.1 Inapproximability

The hardness of the KCM problem stems from the fact that there is a combinatorial number of choices of edges from the candidate set, and there might be strong dependencies in the effects of edge removals. KCM is proved to be NP-hard in [Zhu *et al.*, 2018]. We show a stronger result: KCM is NP-hard to approximate within any constant factor.

Theorem 1. *The KCM problem is NP-hard to approximate within a constant-factor for all $k \geq 3$.*

Proof. We sketch the proof for $k = 3$ (similar for $k > 3$).

Let $SK(U, \mathcal{S}, P, W, q)$ be an instance of the Set Union Knapsack Problem [Goldschmidt *et al.*, 1994], where $U = \{u_1, \ldots u_{n'}\}$ is a set of items, $\mathcal{S} = \{S_1, \ldots S_{m'}\}$ is a set of subsets ($S_i \subseteq U$), $p : \mathcal{S} \rightarrow \mathbb{R}_+$ is a subset profit function, $w : U \rightarrow \mathbb{R}_+$ is an item weight function, and $q \in \mathbb{R}_+$ is the budget. For a subset $\mathcal{A} \subseteq \mathcal{S}$, the weighted union of set $\mathcal{A}$ is $W(\mathcal{A}) = \sum_{e \in \cup_{t \in \mathcal{A}} S_t} w_e$ and $P(\mathcal{A}) = \sum_{t \in \mathcal{A}} p_t$. The problem is to find a subset $\mathcal{A}^* \subseteq S$ such that $W(\mathcal{A}^*) \leq q$ and $P(\mathcal{A}^*)$ is maximized. SK is NP-hard to approximate within a constant factor [Arulselvan, 2014].

We reduce a version of SK with equal profits and weights (also NP-hard to approximate) to the KCM problem. The graph G' is constructed as follows. For each $u_j \in U$, we create a cycle of m' vertices $Y_{j,1}, Y_{j,2}, \ldots, Y_{j,m'}$ in V and add $(Y_{j,1}, Y_{j,2})$, $(Y_{j,2}, Y_{j,3}), \ldots, (Y_{j,m'-1}, Y_{j,m'}), (Y_{j,m'}, Y_{j,1})$ as edges between them. We also add 5 vertices $Z_{j,1}$ to $Z_{j,5}$ with eight edges where the four vertices $Z_{j,2}$ to $Z_{j,5}$ form a clique with six edges. The other two edges are $(Z_{j,1}, Z_{j,2})$ and $(Z_{j,1}, Z_{j,5})$. Moreover, for each subset S_i we create a set of $O((m')^3)$ vertices (sets $X_{i,*}$ are red rectangles in Figure 2), such that each node has exactly degree 3, and add one more node $X_{i,1}$ with two edges incident to two vertices in $X_{i,*}$ from $X_{i,1}$. In the edge set E, an edge $(X_{i,1}, Y_{j,i})$ will be added if $u_j \in S_i$. Additionally, if $u_j \notin S_i$, the edge $(Y_{j,i}, Z_{j,1})$ will be added to E. Figure 2 illustrates our construction for a set $S_1 = \{u_1, u_2\}, S_2 = \{u_1, u_3\}, S_3 = \{u_2\}$.

In KCM, the number of edges to be removed is the budget, b. The candidate set of edges, Γ is the set of all the edges with form $(Y_{j,1}, Y_{j,2})$. Initially all the nodes in G' are in the 3-core. Our claim is that, for any solution $\mathcal{A}$ of an instance of

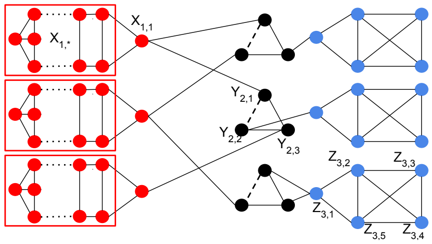


Figure 2: Example construction for hardness reduction from SK where $U = \{u_1, u_2, u_3\}, S = \{S_1, S_2, S_3\}, S_1 = \{u_1, u_2\}, S_2 = \{u_1, u_3\}, S_3 = \{u_2\}$.

SK there is a corresponding solution set of edges, B (where $|B| = b$) in G' of the KCM problem, such that $f_3(B) = P(\mathcal{A}) + b(m'+1)$ if the edges in $\mathcal{A}$ are removed.

The m' nodes in any Y_j and the node $Z_{j,1}$ will be in the 2-core if the edge $(Y_{j,1}, Y_{j,2})$ is removed. So, the removal of any b edges from Γ moves $b(m'+1)$ nodes to the 2-core. But the node $X_{i,1}$ and each node in $X_{i,*}$ ($O((m')^3)$ nodes) will be in the 2-core iff all its neighbours in $Y_{j,i}$s go to the 2-core. Thus, an optimal solution B^* will be $f_3(B^*) = P(\mathcal{A}^*) + b(m'+1)$ where $\mathcal{A}^*$ is the optimal solution for SUKP. For any non-optimal solution B, $f_3(B) = P(\mathcal{A}) + b(m'+1)$ where $\mathcal{A}$ is also non-optimal solution for SUKP. As $P(\mathcal{A}^*)$ is at least $O((m')^3)$ by construction (i.e. $P(\mathcal{A}^*) \gg b(m'+1)$), and $\frac{P(\mathcal{A}^*)}{P(\mathcal{A})}$ cannot be within a constant factor, $\frac{f_3(B^*)}{f_3(B)}$ will also not be within any constant factor. □

Theorem 1 shows that there is no polynomial-time constant-factor approximation for KCM when $k \geq 3$. This contrasts with well-known NP-hard graph combinatorial problems in the literature [Kempe *et al.*, 2003].

2.2 Parameterized Complexity

There are several NP-hard problems (e.g. Vertex Cover) with exact solutions via algorithms that run in time that grows exponentially with the size of the parameter. Thus, if we are only interested in a constant value of the parameter, we can solve the problem in polynomial time. A parameterized problem instance is comprised of an instance X in the usual sense, and a parameter b. The problem is called fixed parameter tractable (FPT) if it is solvable in time $g(b) \times p(|X|)$, where g is an arbitrary function of b and p is a polynomial in the input size $|X|$. We show that KCM is W[2]-hard and para-NP-hard when parameterized by the budget and the core respectively. For details about parameterized complexity classes and reduction techniques, please refer to [Cygan *et al.*, 2015].

Theorem 2. *The KCM problem is $W[2]$-hard parameterized by the budget b.*

We show a parameterized reduction from Set Cover, which is known to be $W[2]$-hard [Bonnet *et al.*, 2016].

Theorem 3. *The KCM problem is para-NP-hard parameterized by k.*

This can be proven from the fact that our problem KCM is NP-hard even for constant k. These parameterized complexity results ensure that the problem KCM is also hard for either parameter b or k. Motivated by these strong hardness

Algorithm 1: Greedy Cut (GC)

Input: G, k, b
Output: B: Set of edges to delete
1 $B \leftarrow \emptyset, max \leftarrow -\infty, \mathscr{G} \leftarrow C_k(G)$
2 **while** $|B| < b$ **do**
3 $\quad e^* \leftarrow \arg\max_{e \in \mathscr{G}(E_k) \setminus B} |computeVS(e = (u,v), \mathscr{G}, k)|$
4 $\quad B \leftarrow B \cup \{e^*\}$
5 $\quad$ LocalUpdate$(e, \mathscr{G}, k)$
6 **return** B

and inapproximability results, we next consider some practical heuristics for the KCM problem.

3 Algorithms

In this section, we propose efficient heuristics for KCM.

3.1 Baseline: Greedy Cut

For KCM, only the current k-core of the graph, $\mathscr{G}(V_k, E_k) = C_k(G)$ ($|V_k| = N_k$,$|E_k| = M_k$), has to be taken into account. Remaining nodes will already be in a lower-than-k-core and can be removed. We define a vulnerable set $VS_k(e, \mathscr{G})$ as those nodes that would be demoted to a lower-than-k-core if edge e is deleted from the current core graph $\mathscr{G}$. Algorithm 1 (GC) [Zhu *et al.*, 2018] is a greedy approach for selecting an edge set B ($|B| = b$) that maximizes the k-core reduction, $f_k(B)$. In each step, it chooses the edge that maximizes $|VS_k(e, \mathscr{G})|$ (step 3-4) among the candidate edges Γ. The specific procedure for computing $VS_k(e, \mathscr{G})$ (step 3), $LocalUpdate$ and their running times ($O(M_k + N_k)$) are described in the extended version of this paper [Medya *et al.*, 2019]. The overall running time of GC is $O(b|\Gamma|(M_k + N_k))$.

3.2 Shapley Value Based Algorithm

The greedy algorithm discussed in the last section is unaware of some dependencies between the candidates in the solution set. For instance, in Figure 1a, all the edges have same importance (the value is 0) to destroy the 2-core structure. In this scenario, GC will choose an edge arbitrarily. However, removing an optimal set of seven edges can make the graph a tree (1-core). To capture these dependencies, we adopt a cooperative game theoretic concept named Shapley Value [Shapley, 1953]. Our goal is to take into account a coalition of edges (players) and divide the total gain by this coalition equally among the edges inside it.

Shapley Value

The Shapley Value of an edge e in KCM is defined as follows. Let the value of a coalition P be $\mathscr{V}(P) = f_k(P) = N_k(G) - N_k(G^P)$. Given an edge $e \in \Gamma$ and a subset $P \subseteq \Gamma$ such that $e \notin P$, the marginal contribution of e to P is:

$$\mathscr{V}(P \cup \{e\}) - \mathscr{V}(P), \quad \forall P \subseteq \Gamma. \tag{1}$$

Let Ω be the set of all $|\Gamma|!$ permutations of all the edges in Γ and $P_e(\pi)$ be the set of all the edges that appear before e in a permutation π. The Shapley Value of e is the average of its

Algorithm 2: Shapley Value Based Cut (SV)

Input: G, k, b, Γ
Output: B: Set of edges to delete
1 Initialize all Φ'_e as 0, $\forall e \in \Gamma$
2 Generate $S = O(\frac{\log \Gamma}{\epsilon^2})$ random permutations of edges
3 $B \leftarrow \emptyset, \mathscr{G} \leftarrow C_k(G)$
4 **for** $\pi \in S$ **do**
5   **for** $e = (u, v) \in \Gamma$ **do**
6         $\Phi'_e \leftarrow \Phi'_e + (\mathscr{V}(P_e(\pi) \cup \{e\}) - \mathscr{V}(P_e(\pi)))$
7 $\Phi'_e \leftarrow \frac{\Phi'_e}{|S|}, \forall e \in \Gamma$
8 Select top b Φ'_e edges from B
9 **return** B

marginal contributions to the edge set that appears before e in all the permutations:

$$\Phi_e = \frac{1}{|\Gamma|!} \sum_{\pi \in \Omega} \mathscr{V}(P_e(\pi) \cup \{e\}) - \mathscr{V}(P_e(\pi)). \qquad (2)$$

Shapley Values capture the importance of an edge inside a set (or coalition) of edges. However, computing Shapley Value requires considering $O(|\Gamma|!)$ permutations. Next we show how to efficiently approximate the Shapley Value for each edge via sampling.

Approximate Shapley Value Based Algorithm

Algorithm 2 (Shapley Value Based Cut, SV) selects the best b edges according to their approximate Shapley Values based on a sampled set of permutations, S. For each permutation in S, we compute the marginal gains of all the edges. These marginal gains are normalized by the sample size, s. In terms of time complexity, steps 4-6 are the dominating steps and take $O(s|\Gamma|(N_k + M_k))$ time, where N_k and M_k are the number of nodes and edges in $C_k(G)$, respectively. Note that similar sampling based methods have been introduced for different applications [Castro *et al.*, 2009; Maleki *et al.*, 2013] (details are in Section 5).

Analysis

We have introduced a fast sampling algorithm (SV) for k-core minimization using Shapley Values. Here, we study the quality of the approximation provided by SV as a function of the number of samples. We show that our algorithm is nearly optimal with respect to each Shapley Value with high probability. More specifically, given $\epsilon > 0$ and $\delta < 1$, SV takes $p(\frac{1}{\epsilon}, \frac{1}{\delta})$ samples, where p is a polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}$, to approximate the Shapley Values within ϵ error with probability $1 - \delta$.

We sample uniformly with replacement, a set of permutations S ($|S| = s$) from the set of all permutations, Ω. Each permutation is chosen with probability $\frac{1}{|\Omega|}$. Let Φ'_e be the approximate Shapley Value of e based on S. X_i is a random variable that denotes the marginal gain in the i-th sampled permutation. So, the estimated Shapley Value is $\Phi'_e = \frac{1}{s}\sum_{i=1}^{s} X_i$. Note that $\mathbb{E}[\Phi'_e] = \Phi_e$.

Theorem 4. *Given ϵ $(0 < \epsilon < 1)$, a positive integer ℓ, and a sample of independent permutations $S, |S| = s$, where $s \geq \frac{(\ell+1)\log|\Gamma|}{2\epsilon^2}$; then $\forall e \in \Gamma$:*

$$Pr(|\Phi'_e - \Phi_e| < \epsilon \cdot N_k) \geq 1 - 2|\Gamma|^{-\ell}$$

where N_k denotes the number of nodes in $C_k(G)$.

To prove this, we use *Hoeffding's inequality* [Hoeffding, 1963] and the union bound. The parameter l further enhances the trade-off between number of samples and the corresponding probability. The next result is stronger as it shows a similar bound for a set of edges.

Next, we apply Theorem 4 to analyze the quality of a set B produced by Algorithm 2 (SV), compared with the result of an exact algorithm (without sampling). Let the Shapley Values of the top b edges be $\Phi^o_B = \{\Phi_{O1}, \Phi_{O2}, \Phi_{O3}, ..., \Phi_{Ob}\}$ where $\Phi_{O1} \geq \Phi_{O2} \geq ... \geq \Phi_{Ob}$. The set produced by Algorithm 2 (SV) has Shapley Values $\Phi^a_B = \{\Phi_{A1}, \Phi_{A2}, \Phi_{A3}, ..., \Phi_{Ab}\}$, where $\Phi_{A1} \geq \Phi_{A2} \geq ... \geq \Phi_{Ab}$. We prove the following result regarding the SV algorithm.

Corollary 5. *For any $i, \Phi_{Oi} \in \Phi^o_B$ and $\Phi_{Ai} \in \Phi^a_B$, ϵ $(0 < \epsilon < 1)$, positive integer ℓ, and a sample of independent permutations $S, |S| = s$, where $s \geq \frac{(\ell+1)\log|\Gamma|}{2\epsilon^2}$:*

$$Pr(|\Phi_{Oi} - \Phi_{Ai}| < 2\epsilon \cdot N_k) \geq 1 - 2|\Gamma|^{-\ell}$$

where N_k denotes the number of nodes in $C_k(G)$.

Proof. For all edges $e \in \Gamma$, Theorem 4 shows that $Pr(|\Phi'_e - \Phi_e| < \epsilon \cdot N_k) \geq 1 - 2|\Gamma|^{-\ell}$. So, with probability $1 - 2|\Gamma|^{-\ell}$, $|\Phi'_{Oi} - \Phi_{Oi}| < \epsilon \cdot N_k$ and $|\Phi'_{Ai} - \Phi_{Ai}| < \epsilon \cdot N_k$. As $\Phi'_{Ai} > \Phi'_{Oi}$, $|\Phi_{Oi} - \Phi_{Ai}| < 2\epsilon \cdot N_k$ with the same probability. □

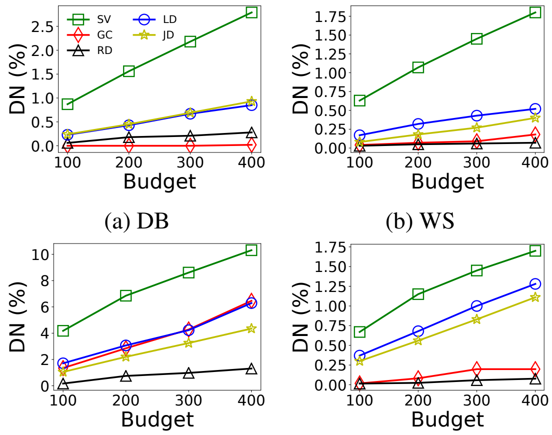
At this point, it is relevant to revisit the hardness of approximation result from Theorem 1 in the light of Corollary 5. First, SV does not directly minimize the KCM objective function (Definition 3). Instead, it provides a score for each candidate edge e based on how different permutations of edges including e minimize the KCM objective under the assumption that such scores are divided fairly among the involved edges. Notice that such assumption is not part of KCM, and thus Shapley Values play the role of a heuristic. Corollary 5, which is a polynomial-time randomized approximation scheme (PRAS) type of guarantee instead of a constant-factor approximation, refers to the exact Shapley Value of the top b edges, and not the KCM objective. We evaluate how SV performs regarding the KCM objective in our experiments.

Dataset Name	$\|V\|$	$\|E\|$	k_{max}	Type
email-Enron (EE)	36K	183K	42	Email
Facebook (FB)	60K	1.5M	52	OSN
web-Stanford (WS)	280K	2.3M	70	Webgaph
DBLP (DB)	317K	1M	113	Co-authorship
Erdos-Renyi (ER)	60K	800K	19	Synthetic

Table 1: Dataset descriptions and statistics. The value of k_{max} (or degeneracy) is the largest k among all the values of k for which there is a k-core in the graph.

4 Experiments

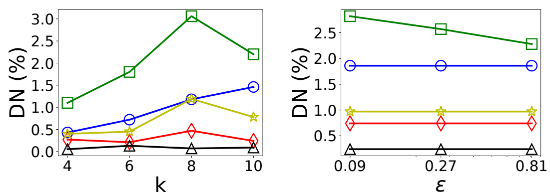
In this section, we evaluate the proposed Shapley Value Based Cut (SV) algorithm for k-core minimization against baseline solutions. In Sec. 4.3, we show how k-core minimization can be applied in analyzing the structural resilience of networks.



(a) DB (b) WS

(c) EE (d) FB

Figure 3: K-core minimization (DN(%)) varying the number of edges in the budget: The Shapley Value based Cut (SV) algorithm outperforms the best baseline (LD) by up to 6 times.



(a) WS (b) WS

Figure 4: K-core minimization (DN(%)) varying (a) the core parameter k; (b) and the sampling error ϵ.

4.1 Experimental Setup

All the experiments were conducted on a 2.59 GHz Intel Core i7-4720HQ machine with 16 GB RAM running Windows 10. Algorithms were implemented in Java.

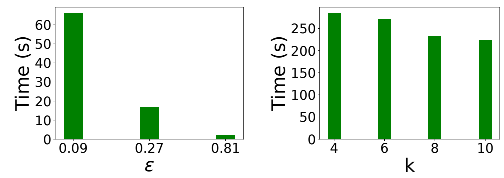
Datasets: The real datasets are available online and are mostly from SNAP[1]. The Facebook dataset is from [Viswanath *et al.*, 2009]. Table 1 shows dataset statistics, including the largest k-core (a.k.a. degeneracy). We also apply a random graph (ER) generated using the Erdos-Renyi model.

Algorithms: Our algorithm, *Shapley Value Based Cut (SV)* is described in Section 3.2. Besides the Greedy Cut (GC) algorithm [Zhu *et al.*, 2018] (Section 3.1), we also consider three more baselines in our experiments. *Low Jaccard Coefficient (JD)* removes the k edges with lowest Jaccard coefficient. Similarly, *Low-Degree (LD)* deletes k edges for which adjacent vertices have the lowest degree. We also apply *Random (RD)*, which simply deletes k edges from the candidate set Γ uniformly at random. Notice that while LD and JD are quite simple approaches for KCM, they often outperform GC.

Quality evaluation metric: Percentage $DN(\%)$ of vertices from the initial graph G that leave the k-core after the deletion of edges in B: $DN(\%) = \frac{N_k(G)-N_k(G^B)}{N_k(G)} \times 100$.

Default parameters: We set the candidate edge set Γ to those edges ($M_k(G)$) between vertices in the k-core $C_k(G)$.

[1] https://snap.stanford.edu



(a) Varying ϵ (b) Varying k

Figure 5: Running times by SV in FB while varying (a) the sampling error ϵ and (b) the core parameter k

Unless stated otherwise, the value of the approximation parameter for SV (ϵ) is 0.05 and the number of samples is $\frac{\log|\Gamma|}{\epsilon^2}$.

4.2 Quality Evaluation

KCM algorithms are compared in terms of quality (DN(%)) for varying budget (b), core value k, and the error of the sampling scheme applied by the SV algorithm (ϵ).

Figure 3 presents the k-core minimization results for $k=5$—similar results were found for $k=10$—using four different datasets. SV outperforms the best baseline by up to six times. This is due to the fact that our algorithm can capture strong dependencies among sets of edges that are effective at breaking the k-core structure. On the other hand, GC, which takes into account only marginal gains for individual edges, achieves worse results than simple baselines such as JD and LD.

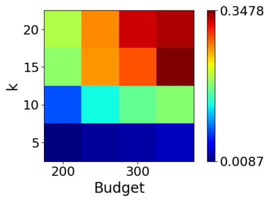
We evaluate the impact of k over quality for the algorithms using WS in Fig. 4a. The budget (b) is set to 400. As in the previous experiments, SV outperforms the competing approaches. However, notice that the gap between LD (the best baseline) and SV decreases as k increases. This is due to the fact that the number of samples decreases for higher k as the number of candidate edge also decreases, but it can be mended by a smaller ϵ. On the other hand, a large value of k leads to a less stable k-core structure that can often be broken by the removal of edges with low-degree endpoints. LD is a good alternative for such extreme scenarios. Similar results were found for other datasets.

The parameter ϵ controls the sampling error of the SV algorithm according to Thm. 4. We show the effect of ϵ over the quality results for WS in Fig. 4b. The values of b and k are set to 400 and 12 respectively. As expected, DN(%) is inversely proportional to the value of ϵ for SV. The trade-off between ϵ and the running time of our algorithm enables both accurate and efficient selection of edges for k-core minimization.

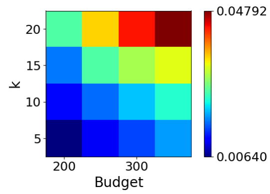
Running Time: Running times for SV varying the sampling error (ϵ) and the core parameter (k) using the FB dataset are given in Figures 5a and 5b, respectively. Even for small error, the algorithm is able to process graphs with tens of thousands of vertices and millions of edges in, roughly, one minute. Running times decay as k increases due to two factors: (1) the size of the k-core structure decreases (2) pruning gets boosted by a less stable core structure.

4.3 Application: k-core Resilience

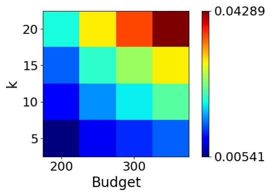
We show how KCM can be applied to profile the resilience of networks. A profile provides a visualization of the resilience



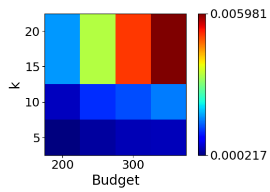
(a) DB



(b) WS



(c) FB



(d) ER

Figure 6: Core resilience for four different networks: (a) DB (co-authorship), (b) WS (Webgraph), (c) FB (social), (d) ER (random). ER and DB are the most and least stable networks, respectively. Tipping points are found for ER and DB.

of the k-core structure of a network for different combinations of k and budget. We apply $DN(\%)$ as a measure of the percentage of the k-core removed by a certain amount of budget—relative to the immediately smaller budget value.

Figure 6 shows the results for co-authorship (DB), Web (WS), social network (FB) and a random (ER) graph. Each cell corresponds to a given k-b combination and the color of cell (X, Y) shows the difference in $DN(\%)$ between $b = Y$ and $b = Y - 100$ for $k = X$. As colors are relative, we also show the range of values associated to the color scheme. We summarize our main findings as follows:

Stability: ER (Figure 6d) is the most stable graph, as can be noticed by the range of values in the profile. The majority of nodes in ER are in the 19-core. DB (Figure 6a) is the least stable, but only when $k > 5$, which is due to its large number of small cliques. The high-core structure of DB is quite unstable, with less than 1% of the network in the 20-core structure after the removal of 500 edges.

Tipping points: We also look at large effects of edge removals within small variations in budget—for a fixed value of k. Such a behavior is not noticed for FB and WS (Figures 6b and 6c, respectively), for which profiles are quite smooth. This is mostly due to the presence of fringe nodes at different levels of k-core structure. On the other hand, ER produced the most prominent tipping points ($k = 15$ and $k = 20$).

5 Previous Work

The k-core decomposition algorithms were introduced in [Seidman, 1983; Batagelj and Zaveršnik, 2011; Bonnet *et al.*, 2016] for different settings. K-cores are applied in community detection [Peng *et al.*, 2014], characterizing the Internet topology [Carmi *et al.*, 2007] and user engagement [Malliaros and Vazirgiannis, 2013]. Bhawalkar et al. [Bhawalkar *et al.*, 2015] and Chitnis et al. [Chitnis *et al.*, 2013] studied the problem of increasing the size of k-core by anchoring a few vertices initially outside of the k-core.

Understanding the behavior of a complex system (e.g. the Internet, a power grid) under different types of attacks and failures has been a popular topic in network science [Cohen *et al.*, 2000]. An overview of different metrics for assessing robustness/resilience is given by [Ellens and Kooij, 2013].

Stability/resilience of k-core: A recent paper [Zhu *et al.*, 2018] proposing the k-core minimization problem is the previous work most related to ours. Compared to the refereed work we: (1) prove stronger inaproximability results for KCM; (2) propose a more effective heuristic based on Shapley Values; and (3) provide a more extensive evaluation of algorithms for KCM using several datasets. Adiga et al. [Adiga and Vullikanti, 2013] studied the stability of high cores in noisy networks. A few studies [Laishram *et al.*, 2018; Zhou *et al.*, 2019] recently introduced a notion of resilience in terms of the stability of k-cores against deletion of random nodes/edges. Another related paper [Zhang *et al.*, 2017] studied the node version of KCM.

Shapley Value (SV) and combinatorial problems: A Shapley Value based algorithm was previously introduced for influence maximization (IM) [Narayanam and Narahari, 2011]. However, notice that IM can be approximated within a constant-factor by a simple greedy algorithm [Kempe *et al.*, 2003]. We apply Shapley Values to solve KCM, which has stronger inapproximability results than IM. Sampling techniques have been for the efficient computation of SV's have also been studied [Castro *et al.*, 2009; Maleki *et al.*, 2013]. Castro et al. [Castro *et al.*, 2009] introduced SV sampling in the context of symmetric and non-symmetric voting games. Maleki et al. [Maleki *et al.*, 2013] provided analyses for stratified sampling specially when the marginal contributions of players are similar. We are able to prove stronger sampling results, bounding the error for the top k edges in terms of Shapley Value (Cor. 5), for KCM.

6 Conclusion

We have studied the k-core minimization (KCM) problem, which consists of finding a set of edges, removal of which minimizes the size of the k-core of a graph. KCM was shown to be NP-hard, even to approximate within a constant. The problem is also W[2]-hard and para-NP-hard parameterized by budget and k, respectively. Given such hardness and inapproximability results, we have proposed an efficient randomized heuristic based on Shapley Value to account for the interdependence in the impact of candidate edges to be removed. We have evaluated our algorithm against baseline approaches using several real graphs, showing that the proposed solution is scalable and outperforms its competitors in terms of quality. We have also illustrated how KCM can be used for profiling the structural resilience of networks.

Acknowledgements

Research was funded partially by NSF IIS #1817046 and DTRA HDTRA1-19-1-0017.

References

[Adiga and Vullikanti, 2013] Abhijin Adiga and Anil Kumar S Vullikanti. How robust is the core of a network? In *ECML/PKDD*, pages 541–556. Springer, 2013.

[Arulselvan, 2014] Ashwin Arulselvan. A note on the set union knapsack problem. *Discrete Applied Mathematics*, 169:214–218, 2014.

[Batagelj and Zaveršnik, 2011] Vladimir Batagelj and Matjaž Zaveršnik. Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5(2):129–145, 2011.

[Bhawalkar *et al.*, 2015] 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.

[Bonnet *et al.*, 2016] Édouard Bonnet, Vangelis Th Paschos, and Florian Sikora. Parameterized exact and approximation algorithms for maximum k-set cover and related satisfiability problems. *RAIRO-Theoretical Informatics and Applications*, 50(3):227–240, 2016.

[Burke *et al.*, 2009] Moira Burke, Cameron Marlow, and Thomas Lento. Feed me: motivating newcomer contribution in social network sites. In *SIGCHI*, 2009.

[Carmi *et al.*, 2007] Shai Carmi, Shlomo Havlin, Scott Kirkpatrick, Yuval Shavitt, and Eran Shir. A model of internet topology using k-shell decomposition. *PNAS*, 2007.

[Castro *et al.*, 2009] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009.

[Chitnis *et al.*, 2013] Rajesh Hemant Chitnis, Fedor V Fomin, and Petr A Golovach. Preventing unraveling in social networks gets harder. In *AAAI*, 2013.

[Cohen *et al.*, 2000] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85(21):4626, 2000.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. 2015.

[Ellens and Kooij, 2013] Wendy Ellens and Robert E Kooij. Graph measures and network robustness. *arXiv preprint arXiv:1311.5064*, 2013.

[Farzan *et al.*, 2011] Rosta Farzan, Laura A Dabbish, Robert E Kraut, and Tom Postmes. Increasing commitment to online communities by designing for social presence. In *CSCW*, 2011.

[Garcia *et al.*, 2013] David Garcia, Pavlin Mavrodiev, and Frank Schweitzer. Social resilience in online communities: The autopsy of friendster. In *COSN*, 2013.

[Goldschmidt *et al.*, 1994] Olivier Goldschmidt, David Nehme, and Gang Yu. Note: On the set-union knapsack problem. *Naval Research Logistics (NRL)*, 1994.

[Hoeffding, 1963] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[Kempe *et al.*, 2003] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146, 2003.

[Laishram *et al.*, 2018] Ricky Laishram, Ahmet Erdem Sariyüce, Tina Eliassi-Rad, Ali Pinar, and Sucheta Soundarajan. Measuring and improving the core resilience of networks. In *The Web Conference*, 2018.

[Maleki *et al.*, 2013] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*, 2013.

[Malliaros and Vazirgiannis, 2013] Fragkiskos D Malliaros and Michalis Vazirgiannis. To stay or not to stay: modeling engagement dynamics in social graphs. In *CIKM*, pages 469–478, 2013.

[Medya *et al.*, 2019] Sourav Medya, Tiyani Ma, Arlei Silva, and Ambuj Singh. K-core minimization: A game theoretic approach. *arXiv preprint arXiv:1901.02166*, 2019.

[Narayanam and Narahari, 2011] Ramasuri Narayanam and Yadati Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE TASE*, pages 130–147, 2011.

[Peng *et al.*, 2014] Chengbin Peng, Tamara G Kolda, and Ali Pinar. Accelerating community detection by using k-core subgraphs. *arXiv preprint arXiv:1403.2226*, 2014.

[Perliger and Pedahzur, 2011] Arie Perliger and Ami Pedahzur. Social network analysis in the study of terrorism and political violence. *PS: Political Science & Politics*, 44(1), 2011.

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

[Shapley, 1953] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 1953.

[Shin *et al.*, 2016] Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos. Corescope: Graph mining using k-core analysis—patterns, anomalies and algorithms. In *ICDM*, pages 469–478, 2016.

[Viswanath *et al.*, 2009] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. On the evolution of user interaction in facebook. In *OSN*, 2009.

[Zhang *et al.*, 2017] Fan Zhang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. Finding critical users for social network engagement: The collapsed k-core problem. In *AAAI*, pages 245–251, 2017.

[Zhou *et al.*, 2019] Zhongxin Zhou, Fan Zhang, Xuemin Lin, Wenjie Zhang, and Chen Chen. K-core maximization: An edge addition approach. In *IJCAI*, 2019.

[Zhu *et al.*, 2018] Weijie Zhu, Chen Chen, Xiaoyang Wang, and Xuemin Lin. K-core minimization: An edge manipulation approach. In *CIKM*, pages 1667–1670. ACM, 2018.