Manipulating Node Similarity Measures in Networks

Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
palash.dey@cse.iitkgp.ac.in

Sourav Medya
Kellogg School of Management
Northwestern University
sourav.medya@kellogg.northwestern.edu

ABSTRACT

Node similarity measures quantify how similar a pair of nodes are in a network. These similarity measures turn out to be an important fundamental tool for many real world applications such as link prediction in networks, recommender systems etc. An important class of similarity measures are *local similarity measures*. Two nodes are considered similar under local similarity measures if they have large overlap between their neighboring set of nodes. Manipulating node similarity measures via removing edges is an important problem. This type of manipulation, for example, hinders effectiveness of link prediction in terrorists networks. All the popular computational problems formulated around manipulating similarity measures turn out to be NP-hard. We, in this paper, provide fine grained complexity results of these problems through the lens of parameterized complexity. In particular, we show that some of these problems are fixed parameter tractable (FPT) with respect to various natural parameters whereas other problems remain intractable (W[1]-hard and W[2]-hard in particular). Finally we show the effectiveness of our proposed FPT algorithms on real world datasets as well as synthetic networks generated using Barabasi-Albert and Erdos-Renyi models.

KEYWORDS

Network design, Parameterized complexity, Node similarity

ACM Reference Format:
Palash Dey and Sourav Medya. 2020. Manipulating Node Similarity Measures in Networks. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Analyzing social networks for uncovering hidden information has a wide range of applications in artificial intelligence (see [5, 29, 32, 35] and references therein for a variety of applications). One of the fundamental tools for social network analysis is the notion of a node similarity measure in networks. A node similarity measure is a function which quantifies the similarity between pairs of nodes of a given network. One such popular measure is the Jaccard similarity [20, 31]. The Jaccard similarity between two nodes u and v is the ratio of the number of common neighbors of u and v by the total number of nodes which are neighbor to at least one of u or v. The Jaccard similarity measure belongs to a broad class of measures called *local* similarity measures [38]. A similarity measure is called local if the similarity between two nodes depends only on their neighborhood (nodes which have an edge with at least one of the pair of nodes).

One of the most useful applications of network similarity measures is link prediction — given a network, predict the edges which are likely to be added in the network in future [1, 21, 36, 42]. The link prediction problem has a variety of applications from uncovering hidden links in a covert network [22] to recommendation systems [4, 7, 33].

While node similarity being important in many applications, we aim to find adversarial ways for manipulating node similarity measures in networks. The main questions is the following: *how hard is it for an adversary to manipulate a network to achieve distorted node similarity*? An important application includes finding malicious activities/users in covert networks. The in covert networks can hide links [11] to disturb the node similarities that are useful to predict links between potentially harmful actors or to recognize the actors of similar threat. Another way to look at the problem would be to find the critical edges that should be retained to measure the node similarities correctly.

Zhou et al. [41] first proposed the problem and observed that the effectiveness of network similarity measures can be hampered substantially using various kind of attacks such as edge deletion. They also showed that the underlying computational task of manipulating any such measure is NP-complete.

We abstract out the main computational challenge of manipulating any local similarity measure into three graph combinatorial problems that are independent of any particular similarity measure. The similarity between any two nodes decreases via edge deletion for several similarity measures [41] only when the number of their common neighbors decreases. Thus, the fundamental task becomes reducing the number of common neighbors of some given targeted pair of nodes in an optimal way. The advantage of abstracting out the network similarity manipulation problems into graph combinatorial problems is its universality. All our results, both algorithms and structural, apply uniformly for all local node similarity measures.

In our first problem, called Eliminating Similarity, the input consists of a graph, a set of targeted pairs of nodes, a candidate set C of edges which can be removed and a budget k. We need to compute if there exist k edges in C whose removal ensures that every pair of given nodes has disjoint neighborhood. In covert networks, one might sustain small amount of similarity for uninterrupted and faster communication while suppressing the actual similarity. Being motivated from this fact, we define the Reducing Total Similarity problem with same input as the previous one. Given an additional an integer threshold t and we aim to find if there exist k edges in C whose removal ensures that the sum of the number of common neighbors in the targeted pairs of nodes is at most t. However, when the sum of the number of common neighbors is reasonably

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

Problem	Parameter	Results
Reducing Total Similarity	k	W[1]-hard even for stars [Theorem 4.2]
	$\|\mathcal{S}\|$	FPT [Theorem 3.6]
	Δ	$O(2^{\Delta}\text{poly}(n))$ [Theorem 3.10]
	δ	para-NP-hard [Theorem 4.8]
Eliminating Similarity	k	$O(1.2738^{k} + km)$ [Theorem 3.2], 2 approximation [Corollary 3.3], $2-\varepsilon$ inapproximable under UGC [Corollary 4.3]
	$\|\mathcal{S}\|$	FPT [Theorem 3.7]
	Δ	$O(2^{\Delta}\text{poly}(n))$ [Theorem 3.9]
	δ	para-NP-hard [Corollary 4.9]
Reducing Maximum Similarity	k	W[2]-hard [Theorem 4.6]
	$\|\mathcal{S}\|$	FPT [Theorem 3.8]
	Δ	para-NP-hard [Theorem 4.7]
	δ	para-NP-hard [Corollary 4.9]

Table 1: Summary of results. We denote the maximum degree by Δ, the average degree by δ, the maximum number of edges we can remove by k, and the set of pairs of vertices between whom we wish to reduce similarity by $\mathcal{S}$.

small (compared with the number of pairs given), for few targeted pairs, the number of common neighbors (and thus similarity) could be high. We address this in the Reducing Maximum Similarity problem: the goal is to find at most k edges in C to remove such that the number of common neighbors between any targeted pair of nodes is at most t.

1.1 Contribution

We study parameterized complexity of our problems and also derive few results on approximation algorithms for some of our problems as a by product of our techniques. We consider three parameters: (i) the maximum number k of edges that can be removed, (ii) The number $|\mathcal{S}|$ of pairs of nodes whose similarity we wish to reduce or eliminate, (iii) the maximum degree of any vertex, and (iv) average degree of the graph. We exhibit FPT algorithms for all our problems parameterized by $|\mathcal{S}|$. Whereas, for the parameter k, we show that only the Eliminating Similarity problem admits an FPT algorithm; the other two problems are intractable with respect to this parameter. We summarize our contribution in this paper in Table 1. We also show the following results.

- ▷ Suppose, in the Eliminating Similarity problem, there exists a set $\mathcal{W}$ of (important) vertices such that the set $\mathcal{S}$ of pair of vertices between which we wish to eliminate similarity is the set of all pair of vertices in $\mathcal{W}$ (that is, $\mathcal{S} = \{\{u,v\} : u,v \in \mathcal{W}, u \neq v\}$). Then we show that the Eliminating Similarity problem is polynomial time solvable [Theorem 3.4].
- ▷ Experiments: We evaluate our proposed FPT algorithm on two synthetic and four real datasets from different genres. We show that our FPT algorithms produce significantly better results than some standard baselines. Moreover, our algorithms are efficient and significantly faster than the baselines.

We observe that the Reducing Maximum Similarity problem is the hardest followed respectively by Reducing Total Similarity and Eliminating Similarity. We believe that the novelty of our work lies in abstracting out the essence of the computational task of manipulating node similarity measures via edge deletion into three generic (independent of the similarity measure under consideration) problems and providing a comprehensive complexity theoretic landscape of these problems under the lens of parameterized complexity.

2 PRELIMINARIES

2.1 Similarity Measures:

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be any undirected and unweighted graph. If not mentioned otherwise, we denote the number of vertices and the number of edges by n and m respectively. For any vertex $v \in \mathcal{V}$, its neighborhood $\mathcal{N}(v)$ is defined to be the set of all vertices which shares an edge with v, that is, we define $\mathcal{N}(v) = \{u \in \mathcal{V} : \{u,v\} \in \mathcal{E}\}$. A similarity measure $\textsc{sim}$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a function $\textsc{sim} : \mathcal{V} \times \mathcal{V} \longrightarrow \mathbb{R}_{\geqslant 0}$. We usually do not talk about similarity of a vertex with itself and thus, for convenience, we define $\textsc{sim}(v,v) = 0$ for every $v \in \mathcal{V}$. We call a similarity function $\textsc{sim}$ *local* if, for every pair of vertices $u,v \in \mathcal{V}$, (i) $\textsc{sim}(u,v)$ depends only on $\mathcal{N}(u)$ and $\mathcal{N}(v)$, (ii) $\textsc{sim}(u,v) = 0$ if and only if $\mathcal{N}(u) \cap \mathcal{N}(v) = \emptyset$. The popular Jaccard similarity is local: The Jaccard similarity between u and v is $\textsc{sim}(u,v) = \frac{|\mathcal{N}(u)\cap\mathcal{N}(v)|}{|\mathcal{N}(u)\cup\mathcal{N}(v)|}$. The Adamic/Adar index is another important example of local similarity measure. Here, the similarity between two nodes u and v is defined as $\textsc{sim}(u,v) = \sum_{x\in\mathcal{N}(u)\cap\mathcal{N}(v)} \frac{1}{\log|\mathcal{N}(x)|}$. If we restrict ourselves to local similarity measures, reducing similarity (by deleting edges) between two nodes under any local similarity measure boils down to reducing their common neighbors. With this observation, we define generic computational problems which captures the essence of the computational challenge of reducing similarity between two nodes under any local similarity measure.

2.2 Problem Definition

In first problem, we are given a set of pairs of nodes and we need to compute the set of minimum number of edges whose removal ensures that there is no common neighbor between any of the given pair of nodes.

*Definition 2.1 (*Eliminating Similarity*).* Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a target set $\mathcal{S} = \{\{x, y\} | x \in \mathcal{V}, y \in \mathcal{V}\}$ of pairs of vertices, a subset $C \subseteq \mathcal{E}$ of candidate edges which can be deleted, and an integer k denoting the maximum number of edges that can be deleted, compute if there exists a subset $\mathcal{F} \subseteq C$ such that $|\mathcal{F}| \leqslant k$ and no pair of vertices in $\mathcal{S}$ has any common neighbor in the graph $\mathcal{G} \setminus \mathcal{F}$. We denote an arbitrary instance of this problem by $(\mathcal{G}, \mathcal{S}, C, k)$.

We generalize the Eliminating Similarity problem where the goal is to find a given budget number of edges such that the sum of the number of common neighbors in the given pairs of vertices is below some threshold. Formally, it is defined as follows.

*Definition 2.2 (*Reducing Total Similarity*).* Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a target set $\mathcal{S} = \{\{x, y\} | x \in \mathcal{V}, y \in \mathcal{V}\}$ of pairs of vertices, a subset $C \subseteq \mathcal{E}$ of candidate edges which can be removed, an integer k denoting the maximum number of edges that can be deleted, and an integer t denoting the target sum of number of common neighbors between vertices in $\mathcal{S}$, compute if there exists a subset $\mathcal{F} \subseteq C$ such that $|\mathcal{F}| \leqslant k$ and sum of number of common neighbors between pairs of vertices in $\mathcal{S}$ is at most t in the graph $\mathcal{G} \setminus \mathcal{F}$. We denote an arbitrary instance of this problem by $(\mathcal{G}, \mathcal{S}, C, k, t)$.

In a solution to the Reducing Total Similarity problem, it may be possible that, although the sum of common neighbors between given pairs of vertices is below some threshold, there exist pairs of vertices in the given set which still share many neighbors. In the Reducing Maximum Similarity problem defined below, the goal is to find k edges to delete such that the maximum overlap between neighbors does not go beyond a threshold.

*Definition 2.3 (*Reducing Maximum Similarity*).* Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a target set $\mathcal{S} = \{\{x, y\} | x \in \mathcal{V}, y \in \mathcal{V}\}$ of pairs of vertices, a subset $C \subseteq \mathcal{E}$ of candidate edges which can be deleted, an integer k denoting the maximum number of edges that can be deleted, and an integer t denoting the target maximum of number of common neighbors between vertices in $\mathcal{S}$, compute if there exists a subset $\mathcal{F} \subseteq C$ such that $|\mathcal{F}| \leqslant k$ and the number of common neighbors between every pair of vertices in $\mathcal{S}$ is at most t in the graph $\mathcal{G} \setminus \mathcal{F}$. We denote an arbitrary instance of this problem by $(\mathcal{G}, \mathcal{S}, C, k, t)$.

The following complexity theoretic relationship among our problems is straight forward.

Proposition 2.4. Eliminating Similarity *polynomial time many-to-one reduces to* Reducing Total Similarity. Eliminating Similarity *polynomial time many-to-one reduces to* Reducing Maximum Similarity.

If not mentioned otherwise, we use k to denote the number of edges that we are allowed to remove, $\mathcal{S}$ to denote the target set.

3 ALGORITHMIC RESULTS

In this section, we present our polynomial time and FPT algorithms. Our first result shows that Eliminating Similarity is fixed parameter tractable parameterized by the number k of edges that we are allowed to delete. Our algorithm reduces any arbitrary instance of Eliminating Similarity to the Vertex Cover problem which is defined as follows.

*Definition 3.1 (*Vertex Cover*).* Given a graph $\mathcal{G}$ and an integer k, compute if there exists a subset $\mathcal{W}$ of at most k vertices in $\mathcal{G}$ such that every edge has at least one end point in $\mathcal{W}$.

We know that there is an FPT algorithm for the Vertex Cover problem running in time $O(1.2738^k + kn)$ where k is the vertex cover we are seeking and n is the number of vertices in $\mathcal{G}$ [8].

Theorem 3.2. *There exists an algorithm for* Eliminating Similarity *which runs in time* $O(1.2738^k + km)$.

Proof. Let $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, k)$ be an arbitrary instance of Eliminating Similarity. For every pair $\{x, y\} \in \mathcal{S}$ and every $u \in \mathcal{V}$ with $\{\{u, x\}, \{u, y\}\} \in \mathcal{E}$, if $|\{\{u, x\}, \{u, y\}\} \cap C| = 0$, then we output no; if $|\{\{u, x\}, \{u, y\}\} \cap C| = 1$ and $k \geqslant 1$, then we remove the edge $\{\{u, x\}, \{u, y\}\} \cap C$ and decrease k by 1. If no pair of vertices in $\mathcal{S}$ have common neighbor in the current graph, then we output yes. Otherwise let us assume without loss of generality that, in the current instance $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, k)$, for every pair $\{x, y\} \in \mathcal{S}$ and every $u \in \mathcal{V}$ with $\{\{u, x\}, \{u, y\}\} \in \mathcal{E}$, we have $\{\{u, x\}, \{u, y\}\} \subseteq C$.

We now reduce the Eliminating Similarity instance to a Vertex Cover $(\mathcal{G}' = (\mathcal{V}', \mathcal{E}'), k')$. We have $\mathcal{V}' = \{v_e : e \in \mathcal{E}\}$. For any two vertices $v_e, v_f \in \mathcal{V}'$, we have an edge between them in $\mathcal{G}'$, that is $\{v_e, v_f\} \in \mathcal{E}'$, if there exist vertices $a, b, c \in \mathcal{V}$ such that $e = \{a, b\}$, $f = \{b, c\}$, and $\{a, c\} \in \mathcal{S}$. We define $k' = k$. We now claim that $(\mathcal{G}, \mathcal{S}, C, k)$ is a yes instance of Eliminating Similarity if and only if $(\mathcal{G}', k')$ is a yes instance of Vertex Cover.

Let $(\mathcal{G}, \mathcal{S}, C, k)$ be a yes instance of Eliminating Similarity. Then there exists a subset $\mathcal{F} \subseteq \mathcal{E}$ of edges in $\mathcal{G}$ with $|\mathcal{F}| \leqslant k$ such that no pair of vertices in $\mathcal{S}$ has a common neighbor in $\mathcal{G} \setminus \mathcal{F}$. We claim that $\mathcal{W} = \{v_e : e \in \mathcal{F}\}$ is a vertex cover of $\mathcal{G}'$. Suppose not, then there exists at least one edge $\{v_e, v_f\} \in \mathcal{E}'$ which $\mathcal{W}$ does not cover. Then, by the construction of $\mathcal{G}'$, we have three vertices $a, b, c \in \mathcal{V}$ such that $e = \{a, b\}$, $f = \{b, c\}$, and $\{a, c\} \in \mathcal{S}$. This contradicts our assumption that no pair of vertices in $\mathcal{S}$ has a common neighbor in $\mathcal{G} \setminus \mathcal{F}$. Hence $\mathcal{W}$ forms a vertex cover of $\mathcal{G}'$. Moreover the size of $\mathcal{W}$ is k which is same as k'. Hence the Vertex Cover instance is a yes instance.

For the other direction, let $(\mathcal{G}', k')$ is a yes instance of Vertex Cover. Then there exists a vertex cover $\mathcal{W} \subseteq \mathcal{V}'$ of $\mathcal{G}'$ of size at most k'. Let us define $\mathcal{F} = \{e \in \mathcal{E} : v_e \in \mathcal{W}\}$. We claim that there is no pair of vertices in $\mathcal{S}$ which has a common neighbor in $\mathcal{G} \setminus \mathcal{F}$. Suppose otherwise, then there exists a pair $\{a, c\} \in \mathcal{S}$ which has a common neighbor $b \in \mathcal{V}$ in $\mathcal{G} \setminus \mathcal{F}$. However, this implies that $\mathcal{W}$ does not cover the edge $\{v_{\{a,b\}}, v_{\{b,c\}}\}$ in $\mathcal{G}'$ which is a contradiction.

We now describe our FPT algorithm. We first pre-process any instance $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, k)$ of Eliminating Similarity and ensure that for every pair $\{x, y\} \in \mathcal{S}$ and every $u \in \mathcal{V}$ with $\{\{u, x\}, \{u, y\}\} \in \mathcal{E}$, we have $\{\{u, x\}, \{u, y\}\} \subseteq C$. We then construct a corresponding instance $(\mathcal{G}', k')$ of Vertex Cover as described above. We then compute a vertex cover $\mathcal{W}$ (if it exists) of $(\mathcal{G}', k')$ and output the corresponding set of edges of $\mathcal{G}$ to be removed. If the vertex cover instance is a no instance, then we output no. We observe that the number of vertices in the vertex cover instance is the same as the number of edges of the Eliminating Similarity instance. Now the claimed running time of our algorithm follows from the fact that there is an algorithm for the Vertex Cover problem which runs in time $O(1.2738^k + kn)$. □

In the proof of Theorem 3.2, we exhibit an approximation preserving reduction from Eliminating Similarity to Vertex Cover. Since Vertex Cover admits a 2 factor polynomial time approximation algorithm (see for example [34]), we obtain the following result as an immediate corollary of Theorem 3.2.

Corollary 3.3. *There exists a polynomial time algorithm for optimizing k in* Eliminating Similarity *within a factor of* 2.

We next show that Eliminating Similarity is polynomial time solvable if there exists a subset $\mathcal{W} \subseteq \mathcal{V}$ of vertices such that the set $\mathcal{S}$ of given pairs is $\{\{u, v\} : u, v \in \mathcal{W}, u \neq v\}$.

Theorem 3.4. *Suppose, in* Eliminating Similarity*, there exists a set $\mathcal{W}$ of (important) vertices such that the set $\mathcal{S}$ of pair of vertices between which we wish to eliminate similarity is the set of all pair of vertices in $\mathcal{W}$; that is, $\mathcal{S} = \{\{u, v\} : u, v \in \mathcal{W}, u \neq v\}$. Then* Eliminating Similarity *is polynomial time solvable.*

Proof. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the input graph. We build the solution $\mathcal{X}$ (set of edges to be removed) iteratively. The set $\mathcal{X}$ is initialized to $\emptyset$. For every vertex $u \in \mathcal{V} \setminus \mathcal{W}$, we put all the edges between u and every vertex in $\mathcal{W}$ except one edge in $\mathcal{X}$. Let $\mathcal{M}$ be a maximum matching of $\mathcal{G}[\mathcal{W}]$. We put all the edges in $\mathcal{E}[\mathcal{W}] \setminus \mathcal{M}$ in $\mathcal{X}$ where $\mathcal{E}[\mathcal{W}]$ is the set of edges in $\mathcal{G}$ with both end points in $\mathcal{W}$. If the size of $\mathcal{X}$ exceeds the number k of edges that we are allowed to delete, then we output no. Otherwise we output yes with $\mathcal{X}$ being a set of at most k edges whose deletion removes similarity between every pair of vertices in $\mathcal{W}$. Clearly any solution would remove at least $|\mathcal{X}|$ number of edges otherwise either there will be a vertex in $u \in \mathcal{V} \setminus \mathcal{W}$ which has edges to at least 2 vertices in $\mathcal{W}$ (which is a contradiction) or there will be an induced path of length 2 in $\mathcal{G}[\mathcal{W}]$ (which again is a contradiction since there exists pair of vertices in $\mathcal{W}$ having common neighbor). Hence, if the algorithm outputs no, the instance is indeed a no instance. On the other hand, if the algorithm outputs yes, it discovers a set of at most k edges whose removal ensures that no pair of vertices in $\mathcal{W}$ has any common neighbor. This concludes the correctness of the algorithm. □

We next focus on the parameter $|\mathcal{S}|$. We use the following result by Lenstra to design our FPT algorithms.

Lemma 3.5 (Lenstra's Theorem [19]). *There is an algorithm for computing a feasible as well as an optimal solution of an integer linear program which is fixed parameter tractable parameterized by the number of variables.*

We first consider the Reducing Total Similarity problem.

Theorem 3.6. Reducing Total Similarity *parameterized by $|\mathcal{S}|$ has a fixed parameter tractable algorithm.*

Proof. Let $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, k)$ be an arbitrary instance of Reducing Total Similarity. For two vertices u and v of $\mathcal{G}$, we say that u and v are of "same type" if we have the following — for every pair $\{x, y\} \in \mathcal{S}$ of vertices in $\mathcal{S}$, u is a common neighbor of x and y if and only if v is a common neighbor of x and y. Since $|\mathcal{S}| = \ell$, we observe that there can be at most 2^ℓ different types of vertices in $\mathcal{G}$ since types are in one-to-one correspondence with the power set $2^{\mathcal{S}}$ of $\mathcal{S}$. Let $\mathcal{T} = \{\mu(\mathcal{X}) : \mathcal{X} \subseteq \mathcal{S}\}$ be the set of all types in $\mathcal{G}$. For each type $\mu(\mathcal{X}) \in \mathcal{T}$, let $n(\mathcal{X})$ be the number of vertices of type $\mu(\mathcal{X})$ in $\mathcal{G}$— we observe that $n(\mathcal{X})$ can be computed in polynomial time from the given graph $\mathcal{G}$ for every type $\mu(\mathcal{X}) \in \mathcal{T}$. Let v be any vertex in $\mathcal{G}$ of type $\mu(\mathcal{X}) \in \mathcal{T}$. We observe that the vertex v can "participate" in any optimal solution in only $4^{|\mathcal{X}|}$ possible ways: for each pair $\{x, y\} \in \mathcal{X}$ such that the vertex v is a common neighbor of both x and y, exactly one of the following 4 events happen – (i) both the edges $\{v, x\}$ and $\{v, y\}$ belong to the optimal set of edges (call it optimal solution), (ii) only $\{v, x\}$ belongs to the optimal solution, (iii) only $\{v, y\}$ belongs to the optimal solution, and (iv) none of $\{v, x\}$ and $\{v, y\}$ belong to the optimal solution. So vertex of each type $\mu(\mathcal{X}) \in \mathcal{T}$ can "participate" in the optimal solution in at most $4^{|\mathcal{X}|}$ ways; we abstractly define the set of all possible ways of participation of vertices of type $\mu(\mathcal{X})$ by $\mathcal{P}(\mathcal{X})$. We will now formulate the Reducing Total Similarity problem using an integer linear program (ILP). For each type $\mu(\mathcal{X}) \in \mathcal{T}$ and each participation type $P \in \mathcal{P}(\mathcal{X})$, let the variable $\mathcal{Z}(\mathcal{X}; P)$ denote the number of vertices of type $\mu(\mathcal{X})$ which participate in the optimal solution like P. We use the variable $\mathcal{Y}(\{x, y\})$ in ILP to denote the number of common neighbors of x and y for $\{x, y\} \in \mathcal{S}$ after removing the optimal set of edges. For each type $\mu(\mathcal{X})$ and every participation $P \in \mathcal{P}(\mathcal{X})$, let $\lambda(\mathcal{X}, P)$ denote the number of edges which gets removed in P; $\lambda(\mathcal{X}, P)$ is a polynomial time computable fixed (depends on the input graph $\mathcal{G}$ only) quantity. We write the following ILP.

$$\sum_{\{x,y\} \in \mathcal{S}} \mathcal{Y}(\{x, y\}) \leqslant t \tag{1}$$

$$\sum_{\mu(\mathcal{X}) \in \mathcal{T}, P \in \mathcal{P}(\mathcal{X})} \lambda(\mathcal{X}, P) \mathcal{Z}(\mathcal{X}; P) \leqslant k \tag{2}$$

$$\sum_{P \in \mathcal{P}(\mathcal{X})} \mathcal{Z}(\mathcal{X}; P) = n(\mathcal{X}), \qquad \forall \mathcal{X} \in \mathcal{T} \tag{3}$$

$$\mathcal{Y}(\{x, y\}) = \sum_{\mu(\mathcal{X}) \in \mathcal{T}, \{x,y\} \in \mathcal{X}} n(\mathcal{X}) - \sum_{\substack{P \in \mathcal{P}(\mathcal{X}), \text{ at least} \\ \text{one edge on } x \text{ or } y \\ \text{gets removed in } P}} \mathcal{Z}(\mathcal{X}; P) \qquad \forall \{x, y\} \in \mathcal{S} \tag{4}$$

Inequality (1) along with Inequality (4) ensure that the sum of the number of common neighbors between pairs of vertices in $\mathcal{S}$ is at most t. Inequality (2) ensures that the number of edges deleted is at most k. From the discussion above, it follows that the Reducing Total Similarity instance is a yes instance if and only if the above ILP is feasible. Since the number of variables is $O(\ell 2^\ell 4^\ell) = O(\ell 8^\ell)$, the result follows from Lemma 3.5. □

Due to Proposition 2.4, Theorem 3.6 immediately gives us the following corollary.

Theorem 3.7. Eliminating Similarity *parameterized by $|\mathcal{S}|$ has a fixed parameter tractable algorithm.*

Also the idea of Theorem 3.6 can be analogously used to obtain a fixed parameter tractable algorithm for the Reducing Maximum Similarity problem.

Theorem 3.8. Reducing Maximum Similarity *parameterized by* $|\mathcal{S}|$ *has a fixed parameter tractable algorithm.*

Proof. We replace Inequality (1) with the following two inequalities. The rest of the argument is analogous to the proof of Theorem 3.6.

$$\Gamma \leqslant t$$
$$\mathcal{Y}(\{x,y\}) \leqslant \Gamma \qquad \forall \{x,y\} \in \mathcal{S}$$

□

We next consider maximum degree of any node as our parameter. We show that Reducing Maximum Similarity admits an FPT algorithm with respect to maximum degree as parameter.

Theorem 3.9. *There is an algorithm for* Reducing Total Similarity *with running time* $O(2^{\Delta} poly(n))$ *where* Δ *is the maximum degree of the input graph.*

Proof. Let $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, k, t)$ be an arbitrary instance of Reducing Maximum Similarity. For each pair $\{x,y\} \in \mathcal{S}$, we observe that x and y can have at most Δ common neighbors. Let $\mathcal{N}_{\mathcal{E}}(x)$ and $\mathcal{N}_{\mathcal{E}}(y)$ be the set of edges incident on respectively x and y. Then we have $|\mathcal{N}_{\mathcal{E}}(x) \cup \mathcal{N}_{\mathcal{E}}(y)| \leqslant 2\Delta$. Suppose $\mathcal{V} = \{v_j : j \in [n]\}$ and for $i \in [n]$, we define $\mathcal{V}_i = \{v_j : j \in [i]\}$ and $\mathcal{G}_i = \mathcal{G}[\mathcal{V}_i]$. We now describe a dynamic programming based algorithm. Our dynamic programming table $\mathcal{T}$ is a Boolean table indexed by the set $\{(i, k', t') : i \in [n], k' \in [k], t' = [t]\}$. We define $\mathcal{T}(i, k', t')$ to be true if and only if there exists a set $\mathcal{F}_i \subseteq \mathcal{E}[\mathcal{G}_i]$ of k' edges whose removal from $\mathcal{G}$ makes the total number of common neighbor in $\mathcal{V}_i$ between pairs of vertices in $\mathcal{S}$ to be at most t'. Formally, $\mathcal{T}(i, k', t') = true$ if and only if $\exists \mathcal{F}_i \subseteq \mathcal{E}[\mathcal{G}_i], |\mathcal{F}_i| \leqslant k'$ such that $\sum_{\{x,y\} \in \mathcal{S}} \mathcal{N}(x) \cap \mathcal{N}(y) \cap \mathcal{V}_i' \leqslant t'$ in $\mathcal{G} \setminus \mathcal{F}_i$. We initialize the table entries $\mathcal{T}(1, k', t')$ to be false for every $k' \in [k]$ and $t' \in [t]$ and initialize $\mathcal{T}(1, k', 0)$ to be true for every $k' \in [k]$. To update an entry $\mathcal{T}(i, k', t')$, we guess the edges incident on v_i that will be part of an optimal solution. Formally, we update $\mathcal{T}(i, k', t')$ as follows. For any $X \subseteq N(v_i)$, we define $\Gamma(X, v_i)$ to be $|\{\{x,y\} \in \mathcal{S} : x \in X \text{ or } y \in X\}|$.

$$\mathcal{T}(i, k', t') = \bigvee_{X \subseteq N(v_i)} \mathcal{T}\left(i-1, k' - |X|, t' - \Gamma(X, v_i)\right)$$

For convenience, we define the logical OR of no variables to be false. We output that the Reducing Maximum Similarity instance is a yes instance if and only if $\mathcal{T}(n, k, t)$ is true. The correctness of our algorithm is immediate from our dynamic programming formulation and update rule. We observe that our dynamic programming table has nkt entries each of which can be updated in $O(2^{\Delta}\text{poly}(n))$ time. Hence the running time of our algorithm is $O(2^{\Delta}\text{poly}(n))$. □

Due to Proposition 2.4, Theorem 3.9 immediately implies existence of an FPT algorithm for Eliminating Similarity parameterized by the maximum degree Δ of the graph.

Theorem 3.10. *There is an algorithm for* Eliminating Similarity *with running time* $O(2^{\Delta} poly(n))$ *where* Δ *is the maximum degree of the input graph.*

4 HARDNESS RESULTS

In this section we present our algorithmic hardness results. We begin with showing that the Reducing Total Similarity problem is W[1]-hard parameterized by the number k of edges that we are allowed to delete even for star graphs. For that, we present an fpt-reduction from the Partial Vertex Cover problem parameterized by budget to the Reducing Total Similarity problem. The Partial Vertex Cover problem is defined as follows.

*Definition 4.1 (*Partial Vertex Cover*).* Given a graph $\mathcal{G}$ and two integers k and s, compute if there exist k vertices which cover at least s edges. We denote an arbitrary instance of Partial Vertex Cover by $(\mathcal{G}, k, s)$.

We know that the Partial Vertex Cover problem parameterized by the number k of vertices that we are allowed to choose is W[1]-hard [10, Chapter 13].

Theorem 4.2. Reducing Total Similarity *parameterized by* k *is* W[1]*-hard even for stars.*

Proof. We exhibit an fpt-reduction from Partial Vertex Cover parameterized by the number of vertices that we are allowed to the Reducing Total Similarity problem parameterized by k. Let $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), k, s)$ be an arbitrary instance of Partial Vertex Cover. We construct an instance $(\mathcal{G}' = (\mathcal{V}', \mathcal{E}'), \mathcal{S}, C, k', t)$ of Reducing Total Similarity as follows. Let m be the number of edges in $\mathcal{G}$.

$$\mathcal{V}' = \{u_v : v \in \mathcal{V}\} \cup \{r\}$$
$$\mathcal{E}' = \{\{r, u_x\}, \{r, u_y\} : \{x,y\} \in \mathcal{E}\}$$
$$\mathcal{S} = \{\{u_x, u_y\} : \{x,y\} \in \mathcal{E}\}$$
$$C = \mathcal{E}', k' = k, t = m - s$$

We observe that the number of pairs in $\mathcal{S}$ is m. Also, every pair of vertices in $\mathcal{S}$ has exactly one common neighbor namely r. We claim that the Partial Vertex Cover instance is a yes instance if and only if the Reducing Total Similarity instance is a yes instance.

In one direction, let us assume that the Partial Vertex Cover instance is a yes instance. Let $\mathcal{W} \subseteq \mathcal{V}$ be a subset of vertices which covers at least s edges of $\mathcal{G}$. Then the set $\mathcal{F} = \{\{r, u_v\} : v \in \mathcal{W}\} \subseteq \mathcal{E}'$ of edges makes the common neighborhood of every pair in $\{\{u_x, u_y\} : \{x,y\} \in \mathcal{E}, x \in \mathcal{W} \text{ or } y \in \mathcal{W}\}$ empty. Since $\mathcal{W}$ covers at least s edges in $\mathcal{G}$, it follows that the sum of number of common neighbors between vertices in $\mathcal{S}$ in $\mathcal{G} \setminus \mathcal{F}$ is at most $m - s$ and thus the Reducing Total Similarity instance is a yes instance.

On the other direction, let us assume that the Reducing Total Similarity instance is a yes instance. Let $\mathcal{F} \subseteq \mathcal{E}'$ be a set of edges such that the sum of the number of common neighbors between vertices in $\mathcal{S}$ in $\mathcal{G} \setminus \mathcal{F}$ is at most $m - s$. Let us consider $\mathcal{W} = \{v \in \mathcal{V} : \{r, u_v\} \in \mathcal{F}\} \subseteq \mathcal{V}$. It follows that $\mathcal{F}$ covers every edge in $\{\{x,y\} \in \mathcal{E} : \{r, u_x\} \in \mathcal{F} \text{ or } \{r, u_y\} \in \mathcal{F}\}$ which has at least s edges since the sum of number of common neighbors between vertices in $\mathcal{S}$ in $\mathcal{G} \setminus \mathcal{F}$ is at most $m - s$. Hence the Partial Vertex Cover instance is a yes instance.

Since the above reduction is an fpt-reduction, the result follows. □

In the proof of Theorem 4.2, we also exhibit an approximation preserving reduction from Vertex Cover (set $s = m$) to the optimization version of Eliminating Similarity where the goal is to remove a minimum number of edges. Since Vertex Cover is known to be inapproximable within factor $(2 - \varepsilon)$ for any $\varepsilon > 0$ in polynomial time under Unique Games Conjecture (UGC), we immediately have the following corollary (see [34] for example).

Corollary 4.3. *For every $\varepsilon > 0$, there does not exist any polynomial time algorithm for approximating k for* Eliminating Similarity *within a factor of $(2 - \varepsilon)$ unless UGC fails.*

We next show that the Reducing Maximum Similarity problem is W[2]-hard parameterized by k. Towards that, we use the a specialization of the set cover problem where every element of the universe appears in the same number of sets. We first show that the set cover problem with this assumption still continues to be W[2]-hard and then present an FPT-reduction from it to our problem.

*Definition 4.4 (*Uniform Set Cover*).* Given an universe $\mathcal{U}$, a collection $\mathcal{D} \subseteq 2^{\mathcal{U}}$ of subsets of $\mathcal{U}$ such that every element $u \in \mathcal{U}$ appears in the same number of sets in $\mathcal{S}$, and a budget b, compute if there exists a sub-collection $\mathcal{W} \subseteq \mathcal{D}$ such that (i) $|\mathcal{W}| \leqslant b$ and (ii) $\cup_{A \in \mathcal{W}} A = \mathcal{U}$. We denote an arbitrary instance of Uniform Set Cover by $(\mathcal{U}, \mathcal{D}, b)$.

The Set Cover problem is the same as the Uniform Set Cover problem except the fact that elements in the universe $\mathcal{U}$ can belong to any number of sets in $\mathcal{S}$ in Set Cover. It is known that the Set Cover problem parameterized by b is W[2]-hard [10].

Proposition 4.5. Uniform Set Cover *parameterized by b is* W[2]*-hard.*

Proof. We exhibit an fpt-reduction from Set Cover to Uniform Set Cover. Let $(\mathcal{U}, \mathcal{D}, b)$ be an instance of Set Cover. Without loss of generality, we assume that, for every element $u \in \mathcal{U}$, there exists a set $X \in \mathcal{D}$ such that $u \in X$. We construct the following instance $(\mathcal{U}', \mathcal{D}', b')$ of Uniform Set Cover. For any element $u \in \mathcal{U}$, let f_u be the number of sets in $\mathcal{D}$ where u belongs and $|\mathcal{D}| = \ell$.

$$\mathcal{U}' = \mathcal{U}$$
$$\mathcal{D}' = \mathcal{D} \cup_{u \in \mathcal{U}} \uplus_{i=f_u}^{\ell} \{\{u\}\}$$
$$b' = b$$

The equivalence of the two instances are straight forward and we defer its proof to the full version of the paper. □

We now prove that Reducing Maximum Similarity is W[2]-hard parameterized by the number k of vertices that we are allowed to delete. For that, we exhibit an fpt-reduction from the Uniform Set Cover problem parameterized by the budget to the Reducing Maximum Similarity problem parameterized by k.

Theorem 4.6. Reducing Maximum Similarity *parameterized by k is* W[2]*-hard even for bipartite graphs of radius* 2.

Proof. We exhibit an fpt-reduction from Uniform Set Cover to Reducing Maximum Similarity. Let $(\mathcal{U}, \mathcal{S}, b)$ be an arbitrary instance of Uniform Set Cover. Let f be the number of sets that every element in $\mathcal{U}$ belongs. We consider the following instance $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, k, t)$ of Reducing Maximum Similarity.

$$\mathcal{V} = \{r\} \cup \{x_u : u \in \mathcal{U}\} \cup \{y_D : D \in \mathcal{D}\}$$
$$\mathcal{E} = \{\{r, y_D\} : D \in \mathcal{D}\} \cup \{\{x_u, y_D\} : u \in D\}$$
$$\mathcal{S} = \{\{r, x_u\} : u \in \mathcal{U}\}$$
$$C = \mathcal{E}', k = b, t = f - 1$$

We claim that the Uniform Set Cover instance is a yes instance if and only if the Reducing Maximum Similarity instance is a yes instance.

In one direction, let us assume that the Uniform Set Cover instance is a yes instance. Let $\mathcal{W} \subseteq \mathcal{D}$ forms a set cover for $\mathcal{U}$ and $|\mathcal{W}| \leqslant b$. We claim that $\mathcal{F} = \{\{r, y_D\} : D \in \mathcal{W}\} \subseteq \mathcal{E}'$ is a solution for the Reducing Maximum Similarity instance. To see this, we consider any pair $\{r, x_u\} \in \mathcal{S}$. By the definition of f, there are f common neighbors of r and x_u in $\mathcal{G}$. Since $\mathcal{W}$ forms a set cover, the number of common neighbors of r and x_u in $\mathcal{G} \setminus \mathcal{F}$ is at most $f - 1$. We also have $|\mathcal{F}| \leqslant b = k$. This proves that the Reducing Maximum Similarity instance is a yes instance.

On the other direction, let us assume that the Reducing Maximum Similarity instance is a yes instance. Let $\mathcal{F} \subseteq \mathcal{E}'$ be a set of edges such that (i) $|\mathcal{F}| \leqslant k$ and (ii) the number of common neighbors in every pair of vertices in $\mathcal{S}$ is at most $t = f - 1$. We consider the sub-collection $\mathcal{W} = \{D \in \mathcal{D} : \{r, y_D\} \in \mathcal{F}$ or $\{x_u, y_D\} \in \mathcal{F}$ for some $u \in \mathcal{U}\} \subseteq \mathcal{D}$. Since $|\mathcal{F}| \leqslant k$, we have $|\mathcal{W}| \leqslant |\mathcal{F}| \leqslant k = b$. We claim that $\mathcal{W}$ forms a set cover for $\mathcal{U}$. Suppose not, then there exists an element $u \in \mathcal{U}$ that is not covered by $\mathcal{W}$. Then the number of common neighbors of r and x_u in $\mathcal{G} \setminus \mathcal{F}$ is f which is a contradiction. Hence the Uniform Set Cover instance is a yes instance.

Since the above reduction is an fpt-reduction, the result follows. □

Our last result of this section is that the Reducing Maximum Similarity problem is para-NP-hard parameterized by the maximum degree Δ of the graph. For that, we use the well known result that the Vertex Cover problem is NP-complete even for 3 regular graphs [14].

Theorem 4.7. Reducing Maximum Similarity *is* NP*-complete even if the degree of every vertex in the input graph is at most* 7.

Proof. The Reducing Maximum Similarity problem is clearly in NP. To prove NP-hardness, we reduce from an arbitrary instance $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), k)$ of Vertex Cover on 3-regular graph. We construct the following instance $(\mathcal{G}' = (\mathcal{V}', \mathcal{E}'), \mathcal{S}, C, k', t)$ of Reducing Maximum Similarity.

$$\mathcal{V}' = \{x_u, y_u : u \in \mathcal{U}\}$$
$$\mathcal{E}' = \{\{x_u, x_v\}, \{x_u, y_v\}, \{x_v, y_u\} : \{u, v\} \in \mathcal{E}\} \cup \{\{x_u, y_u\} : u \in \mathcal{V}\}$$
$$\mathcal{S} = \{\{y_u, y_v\} : \{u, v\} \in \mathcal{E}\}$$
$$C = \mathcal{E}'$$
$$k' = k, t = 1$$

We now claim that the two instances are equivalent. Since the degree of every vertex in $\mathcal{G}$ is 3, it follows that the degree of any vertex in $\mathcal{G}'$ is at most 7. In one direction let us assume that the VERTEX COVER instance is a YES instance. Let $\mathcal{W} \subseteq \mathcal{V}$ be a vertex cover of cardinality k. We claim that, after removing every edge in the set $\mathcal{F} = \{\{x_w, y_w\} : w \in \mathcal{W}\}$, the number of common neighbors between every pair of vertices in $\mathcal{S}$ is at most 1. Suppose not, then there exists a pair $\{y_u, y_v\}$ which has at least 2 neighbors in $\mathcal{G}' \setminus \mathcal{F}$. Then, it follows that both $\{x_u, y_u\} \notin \mathcal{F}$ and $\{x_v, y_v\} \notin \mathcal{F}$. However this implies that $\mathcal{W}$ does not cover the edge $\{u, v\}$ in $\mathcal{G}$ which contradicts our assumption that $\mathcal{W}$ is a vertex cover for $\mathcal{G}$. Hence the REDUCING MAXIMUM SIMILARITY instance is a YES instance.

For the other direction, let us assume that there exists a subset $\mathcal{F} \subseteq \mathcal{E}'$ of edges in $\mathcal{G}'$ such that, in $\mathcal{G}' \setminus \mathcal{F}$, the number of common neighbors between every pair of vertices in $\mathcal{S}$ is at most 1. Let us consider a subset $\mathcal{W} = \{w \in \mathcal{V}$: an edge incident on x_w belongs to $\mathcal{F}\}$. Since $|\mathcal{F}| \leqslant k' = k$, we have $|\mathcal{W}| \leqslant k$. We claim that $\mathcal{W}$ forms a vertex cover for $\mathcal{G}$. Suppose not, then there exists an edge $\{u, v\} \in \mathcal{E}$ which the set $\mathcal{W}$ does not cover. Then it follows that the both x_u and x_v are common neighbor of the pair $\{y_u, y_v\}$ of vertices. However, this contradicts our assumption about $\mathcal{F}$. Hence $\mathcal{W}$ forms a vertex cover for $\mathcal{G}$ and the VERTEX COVER instance is a YES instance. □

We finally consider the average degree δ of the graph as our parameter. We show that ELIMINATING SIMILARITY is para-NP-hard parameterized by δ.

THEOREM 4.8. ELIMINATING SIMILARITY *is para-*NP*-hard parameterized by average degree δ of the input graph.*

PROOF. We reduce any instance of ELIMINATING SIMILARITY to another instance of ELIMINATING SIMILARITY where, in the reduced instance, the average degree is constant. Let $\Pi = (\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, C, k)$ be an arbitrary instance of ELIMINATING SIMILARITY. We consider the following instance $\Pi' = (\mathcal{G}' = (\mathcal{V}', \mathcal{E}'), \mathcal{S}', C', k')$ of ELIMINATING SIMILARITY. Let $|\mathcal{V}| = n$ and v be an arbitrary vertex of $\mathcal{G}$.

$$\mathcal{V}' = \mathcal{V} \cup \{w_i : i \in [n^2]\}$$
$$\mathcal{E}' = \mathcal{E} \cup \{\{w_i, w_{i+1}\} : i \in [n^2 - 1]\} \cup \{\{v, w_1\}\}$$
$$\mathcal{S}' = \mathcal{S}, C' = C, k' = k$$

We observe that the average degree of $\mathcal{G}'$ is at most $\frac{2n^2}{n^2+n} \leqslant 2$. We now claim that the two instances are equivalent. In one direction, if $\mathcal{F} \subseteq \mathcal{E}$ is a solution for Π then $\mathcal{F}$ is a solution for Π' too. On the other hand, if $\mathcal{F}' \subseteq \mathcal{E}'$ is a solution for Π', then $\mathcal{F}' \cap \mathcal{E}$ is a solution for Π too. □

Due to Proposition 2.4, we immediately have the following from Theorem 4.8.

COROLLARY 4.9. *Both the* REDUCING TOTAL SIMILARITY *and* REDUCING MAXIMUM SIMILARITY *problems are para-*NP*-hard parameterized by average degree δ of the input graph.*

5 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our FPT algorithm in Theorem 3.2 using real and synthetic networks for the optimization version of the ELIMINATING SIMILARITY problem. We call our solution as FPTA. We implement our solutions in python and executed on a 3.30GHz Intel core with 30 GB RAM.

Dataset	Type	$\|V\|$	$\|E\|$
power	Power	662	1.5k
hamsterster	Social	2.4k	16.6k
euroroad	Road	1.1k	1.4k
web-edu	Web	3k	6.4k

Table 2: Description of Datasets

Datasets: We use synthetic graphs from two well-studied models: (a) Barabasi-Albert (BA) [3] and (b) Erdos-Renyi (ER) [13]. While the BA model has "small-world" property and scale free degree distribution, ER does not have any of these properties. We generate both the datasets of one thousand vertices (dataset of similar size as in [41]) and approximately two thousands edges. The real datasets are from different genres: Web, social, road and power networks. Table 2 shows the statistics. The datasets are available online[1].

Baselines: We compare our algorithm (FPTA) with two baselines. Note that our algorithm produces optimal results. **(1) Greedy:** Our first baseline is the greedy baseline. It selects an edge in each iteration which decreases a maximum number of common neighbors between the given pairs of nodes and removes it from the graph. **(2) High Jaccard Similarity (HJ):** It selects top edges based on the similarity of endpoints of the edge to delete until every given pair vertices have disjoint neighborhood. **(3) Random:** It iteratively selects random edges to delete until the total similarity of the target pairs becomes zero. The performance metric of these algorithms is the number of edges being deleted to remove similarity for all the given pairs. Hence, the quality is better when the number of edges is lower.

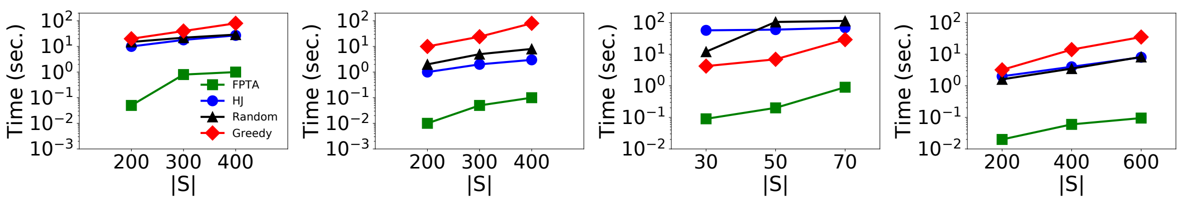
In the experiments, we choose the target pairs (S) randomly from all the pairs of vertices. The size of S is varied depending of the size and nature of the datasets.

Theorem 3.2 shows that an FPT algorithm for ELIMINATING SIMILARITY parameterized by the number of edges that we are allowed to delete. Thus our algorithm always outputs a minimum set of edges. However, we evaluate the efficiency of our method in terms of both quality and running time. Table 3 shows the results varying $|S|$ (the number of target pairs) on four different real datasets. The results in synthetic graphs also have similar trend. The optimal set of edges are quite low compared to the Random and HJ baselines. We also find that the greedy algorithm also produces nearly optimal results. However it is quite time consuming. We show the results regarding the time taken by different algorithms in Figure 1 (real graphs) and Figure 2 (synthetic graphs). The Y-axis is in logarithmic scale. In all six datasets, the time taken by our algorithm FPTA is at least two order faster. The most time consuming algorithm is the greedy algorithm. While Random and HJ are faster than Greedy in most of the cases, it produces much worse results compared to FPTA (Table 3).

6 RELATED WORK

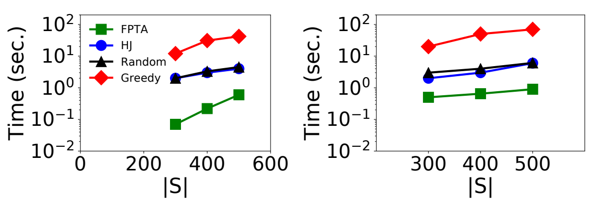
Zhou et al. initiated the study of the problem of attacking node based similarity measures via edge removal [41]. The authors proved that

[1] http://networkrepository.com



(a) Web (b) Power (c) Social (d) Road

Figure 1: Time taken by different algorithms: Our algorithm FPTA outperforms other baselines in terms of efficiency.



(a) ER (b) BA

Figure 2: Synthetic Graphs: Time taken by different algorithms: Our algorithm FPTA outperforms other baselines in terms of efficiency.

Graph	$\|S\|$	FPTA	HJ	Random
Social	30	9	3k	4k
	50	14	9k	10k
	70	20	13k	15k
Power	200	3	473	803
	300	7	682	1.3k
	400	11	721	1.5k
Web	200	3	1.9k	2.3k
	300	7	4k	4.3k
	400	12	5k	6.3k
Road	200	3	421	400
	400	4	800	623
	600	7	1.1k	1.2k

Table 3: Comparison for the number of edges produced by FPTA and Random.

the problems based on local and global structural similarities are NP-hard and provided some heuristics. However, we focus on the parameterized complexity of this problem and its variations. Our work is also related to studying vulnerability of social network analysis (SNA). Attacking against centrality measures via edge manipulation had been studied in the past [11, 37]. These papers show that it is computationally hard to hide for a leader inside a covert network. Waniek et al. [38] and Yu et al. [39] recently discussed the problem of hiding or anonymizing links on networks. Similarly, measuring network robustness via network modification is also a well studied problem [18, 25, 40, 43]. Laishram et al. [18] formulated network resilience in terms of the stability of k-cores against deletion of random nodes/edges. Zhang et al. [40] studied a related problem which was to find b vertices such that their deletion reduces the k-core maximally. The edge version of the same problem were discussed in [25, 43].

All the above problems are in the category of manipulating network structure to optimize for certain objective. There is another line of study that is related to network design. Paik et al. first initiated this line of work by studying several network design problems based on vertex upgradation to decrease the delays on adjacent edges [30]. Since then these problems have received a significant amount of research attention. Meyerson et al. designed algorithms for the minimization of shortest path distances [28]. A series of recent work [12, 23, 24, 27] studied the problem of minimizing the shortest path distance by improving edge or node weights in different context. Another related line of research work concern the problem of increasing the centrality of a node or a set of nodes by adding edges [9, 15, 26]. Boosting or containing diffusion processes in networks were investigated under different well-known diffusion models such as Linear Threshold [16] and Independent Cascade [6, 17]. Controlling opinion were studied recently by Amelkin et al. [2].

7 CONCLUSION

We have proposed three graph theoretic problems and argued that they capture the main computational challenge of attacking local similarity measures. We have studied these problems in parameterized complexity framework. We have considered the number of edges that we are allowed to delete and the number of targeted pairs of nodes as our parameters and shown either existence of FPT algorithm or parameterized intractability for them. We have also exhibited polynomial time algorithm for the Eliminating Similarity problem in an important special case. We finally establish effectiveness of our FPT algorithm for the Eliminating Similarity problem in real and synthetic data sets. An important future direction of research is to study kernelization techniques for these problems.

ACKNOWLEDGEMENT

Palash Dey thanks DST INSPIRE Faculty Fellowship grant DST/INSPIRE/04/2016/001479 and ISIRD grant by IIT Kharagpur for supporting this work.

REFERENCES

[1] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. 2006. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security.*

[2] Victor Amelkin and Ambuj K Singh. 2019. Fighting opinion control in social networks via link recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 677–685.

[3] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

[4] Mattia G Campana and Franca Delmastro. 2017. Recommender systems for online and mobile social networks: A survey. *Online Social Networks and Media* 3 (2017), 75–97.

[5] Peter J Carrington, John Scott, and Stanley Wasserman. 2005. *Models and methods in social network analysis*. Vol. 28. Cambridge university press.

[6] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. 2012. Recommendations to boost content spread in social networks. In *WWW*. 529–538.

[7] Hsinchun Chen, Xin Li, and Zan Huang. 2005. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*. IEEE, 141–142.

[8] Jianer Chen, Iyad A. Kanj, and Ge Xia. 2006. Improved Parameterized Upper Bounds for Vertex Cover. In *Proc. 31st Mathematical Foundations of Computer Science 2006, 31st International Symposium, MFCS*. 238–249.

[9] Pierluigi Crescenzi, Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. 2015. Greedily Improving Our Own Centrality in A Network. In *SEA*. Springer International Publishing, 43–55.

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

[11] Palash Dey and Sourav Medya. 2019. Covert Networks: How Hard is It to Hide?. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 628–637.

[12] Bistra Dilkina, Katherine J. Lai, and Carla P. Gomes. 2011. Upgrading shortest paths in networks. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer, 76–91.

[13] Paul Erdős and Alfréd Rényi. [n. d.]. On the evolution of random graphs. ([n. d.]).

[14] Michael R Garey, David S Johnson, and Larry Stockmeyer. 1974. Some simplified NP-complete problems. In *Proc. sixth annual ACM symposium on Theory of computing*. ACM, 47–63.

[15] Vatche Ishakian, Dóra Erdos, Evimaria Terzi, and Azer Bestavros. 2012. A Framework for the Evaluation and Management of Network Centrality. In *Proc. SIAM International Conference on Data Mining*. 427–438.

[16] Elias Boutros Khalil, Bistra Dilkina, and Le Song. 2014. Scalable diffusion-aware optimization of network topology. In *SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1226–1235.

[17] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2008. Minimizing the Spread of Contamination by Blocking Links in a Network.. In *AAAI*.

[18] Ricky Laishram, Ahmet Erdem Sariyüce, Tina Eliassi-Rad, Ali Pinar, and Sucheta Soundarajan. 2018. Measuring and Improving the Core Resilience of Networks. In *Proceedings of the 2018 World Wide Web Conference*. 609–618.

[19] Hendrik W Lenstra Jr. 1983. Integer programming with a fixed number of variables. *Mathematics of operations research* 8, 4 (1983), 538–548.

[20] Michael Levandowsky and David Winter. 1971. Distance between sets. *Nature* 234, 5323 (1971), 34–35.

[21] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58, 7 (2007), 1019–1031.

[22] Marcus Lim, Azween Abdullah, NZ Jhanjhi, and Mahadevan Supramaniam. 2019. Hidden Link Prediction in Criminal Networks Using the Deep Reinforcement Learning Technique. *Computers* 8, 1 (2019), 8.

[23] Yimin Lin and Kyriakos Mouratidis. 2015. Best upgrade plans for single and multiple source-destination pairs. *GeoInformatica* 19, 2 (2015), 365–404.

[24] Sourav Medya, Petko Bogdanov, and Ambuj Singh. 2018. Making a small world smaller: Path optimization in networks. *IEEE Transactions on Knowledge and Data Engineering* 30, 8 (2018), 1533–1546.

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

[26] Sourav Medya, Arlei Silva, Ambuj Singh, Prithwish Basu, and Ananthram Swami. 2018. Group centrality maximization via network design. In *Proc. 24th SIAM International Conference on Data Mining*. SIAM, 126–134.

[27] Sourav Medya, Jithin Vachery, Sayan Ranu, and Ambuj Singh. 2018. Noticeable network delay minimization via node upgrades. *Proceedings of the VLDB Endowment* 11, 9 (2018), 988–1001.

[28] Adam Meyerson and Brian Tagiku. 2009. Minimizing average shortest path distances via shortcut edge addition. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX-RANDOM)*. Springer, 272–285.

[29] Evelien Otte and Ronald Rousseau. 2002. Social network analysis: a powerful strategy, also for the information sciences. *J. Inf. Sci.* 28, 6 (2002), 441–453.

[30] D. Paik and S. Sahni. 1995. Network upgrading problems. *Networks* (1995).

[31] David J Rogers and Taffee T Tanimoto. 1960. A computer program for classifying plants. *Science* 132, 3434 (1960), 1115–1118.

[32] Jordi Sabater and Carles Sierra. 2002. Reputation and social network analysis in multi-agent systems. In *Proc. 1st International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*. 475–482.

[33] Nitish Talasu, Annapurna Jonnalagadda, S Sai Akshaya Pillai, and Jampani Rahul. 2017. A link prediction based approach for recommendation systems. In *2017 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2059–2062.

[34] Vijay V. Vazirani. 2001. *Approximation algorithms*. Springer. http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-65367-7

[35] Fei-Yue Wang, Kathleen M Carley, Daniel Zeng, and Wenji Mao. 2007. Social computing: From social informatics to social intelligence. *IEEE Intelligent systems* 22, 2 (2007).

[36] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. 2015. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences* 58, 1 (2015), 1–38.

[37] Marcin Waniek, Tomasz P Michalak, Talal Rahwan, and Michael Wooldridge. 2017. On the construction of covert networks. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 1341–1349.

[38] Marcin Waniek, Kai Zhou, Yevgeniy Vorobeychik, Esteban Moro, Tomasz P Michalak, and Talal Rahwan. 2018. Attack Tolerance of Link Prediction Algorithms: How to Hide Your Relations in a Social Network. *arXiv preprint arXiv:1809.00152* (2018).

[39] Shanqing Yu, Minghao Zhao, Chenbo Fu, Huimin Huang, Xincheng Shu, Qi Xuan, and Guanrong Chen. 2018. Target defense against link-prediction-based attacks via evolutionary perturbations. *arXiv preprint arXiv:1809.05912* (2018).

[40] Fan Zhang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. 2017. Finding Critical Users for Social Network Engagement: The Collapsed k-Core Problem. In *Thirty-First AAAI Conference on Artificial Intelligence*. 245–251.

[41] Kai Zhou, Tomasz P. Michalak, Marcin Waniek, Talal Rahwan, and Yevgeniy Vorobeychik. 2019. Attacking Similarity-Based Link Prediction in Social Networks. In *Proc. 18th International Conference on Autonomous Agents and MultiAgent Systems*. 305–313.

[42] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *The European Physical Journal B* 71, 4 (2009), 623–630.

[43] Weijie Zhu, Chen Chen, Xiaoyang Wang, and Xuemin Lin. 2018. K-core Minimization: An Edge Manipulation Approach. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1667–1670.