Preserving Structural Properties in Edge-Perturbing Anonymization Techniques for Social Networks

Amirreza Masoumzadeh, *Student Member*, *IEEE*, and James Joshi, *Member*, *IEEE*

Abstract—Social networks are attracting significant interest from researchers in different domains, especially with the advent of social networking systems which enable large-scale collection of network information. However, as much as analysis of such social networks can benefit researchers, it raises serious privacy concerns for the people involved in them. To address such privacy concerns, several techniques, such as k-anonymity-based approaches, have been proposed in the literature to provide user anonymity in published social networks. However, these methods usually introduce a large amount of distortion to the original social network graphs, thus, raising serious questions about their utility for useful social network analysis. Consequently, these techniques may never be applied in practice. We propose two methods to enhance edge-perturbing anonymization methods based on the concepts of structural roles and edge betweenness in social network theory. We experimentally show significant improvements in preserving structural properties in an anonymized social network achieved by our approach compared to the original algorithms over several data sets.

Index Terms—Privacy, social network, social network analysis, data anonymization, data perturbation

1 Introduction

The study of social networks is growing fast as a scientific area in different domains such as academia, business, and even government. A social network is commonly defined as a collection of agents and relationships among them, which are modeled, respectively, as nodes and links in a graph. The purpose of such a study is to investigate different structural properties and patterns both at agent level and network as a whole depending on the application of interest. In many social network data sets, nodes represent people. Thus, any information released in the data set including node attributes and even relationships between nodes may be subject to privacy implications for the involved users. The advent of social networking systems has raised even more concerns about privacy as large network data sets collected by these systems may include huge amounts of privacy-sensitive information about their users. Therefore, it becomes critical to preserve privacy of users while allowing benefits of analysis of such valuable corpora of information. A prevalent approach for preserving privacy of data sets is to anonymize them so that people cannot be linked to their real-world identities (i.e., reidentified). A naive anonymization for social networks may simply replace real node identifiers with random ones. However, researchers have shown that such an approach is not fool-proof to node reidentification if an adversary has certain background knowledge about network structure of a target victim [1], [2], [3]. In order to provide stronger anonymization for social networks, researchers have mainly taken two different approaches to structural anonymization. In the *graph generalization* approach [4], [5], [6], a social network is summarized in a higher level graph, hiding details of the relationships among agents while providing overall structural summaries of the graph. In the *edge perturbation* approach [2], [7], [8], [9], [10], [11], the edge structure of the social network is modified, i.e., some edges are removed and some are added, in order to satisfy an anonymity property (typically based on k-anonymity [12]). For instance, a graph can be modified to have degree k-anonymity where for each node there are at least $k-1$ other nodes with the same degree. Therefore, an adversary, that knows the degree of a victim node, will not be able to reidentify her by a probability larger than $1/k$.

An expected consequence of structural anonymization is that running same analysis on a social network and its anonymized version may lead to different results. In order to use a social network anonymized by a generalization method it needs to be reconstructed by randomly generating substructures in place of supernodes based on the reported supernode properties. Modifying links in the perturbation methods to fulfill the anonymization criteria also severely affects the network structure. One may hope that such differences are negligible, so that results are still usable. But observations show that if a social network is anonymized up to an acceptable degree, the resultant network becomes highly distorted, thus, severely affecting their utility for analysis purpose [3]. For instance, a node with a low centrality value may become one with a high centrality value because of the addition of many fake adjacent links. Such a change can reduce the accuracy of centrality analysis of the network nodes. The key issue is that anonymization methods usually focus on achieving the anonymization objectives and disregard the crucial need to

• *The authors are with the School of Information Sciences, University of Pittsburgh, IS Building, 135 N. Bellefield Ave., Pittsburgh, PA 15260. E-mail: amirreza@sis.pitt.edu, jjoshi@pitt.edu.*

Manuscript received 8 Apr. 2011; revised 16 Jan. 2012; accepted 23 July 2012; published online 30 July 2012.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSCSI-2011-04-0095.
Digital Object Identifier no. 10.1109/TDSC.2012.65.

preserve the original structural semantics of a social network; hence, the outcome is a significant decrease in the utility of the results.

In this paper, we investigate ways to preserve better structural properties of social networks in edge-perturbation anonymization schemes, based on our initial results [13]. We propose two heuristic methods based on social network theories and concepts [14], [15] that can be applied to a typical edge perturbation algorithm to achieve better preservation of the utility of its output. In the first method, we leverage the notion of structural roles and positions in social networks. We show that maintaining role structure of a social network in an anonymization process can significantly help in preserving its structural semantics. The second method uses the notion of edge betweenness in a social network in order to minimize changes in shortest paths in the edge perturbation process, and consequently help preserve better many social network analysis measures that are related to shortest paths. We empirically show that our proposed enhancement methods can preserve structural properties of social networks significantly better compared to what original edge perturbation algorithms offer. For this purpose, we present results of experiments on three different enhanced anonymization algorithms and multiple data sets, analyzing social network analysis measures such as betweenness centrality (BC), clustering coefficient (CCoef), etc. Our contributions in this paper can be summarized as follows:

- We formalize the notion of roles based on regular equivalence in social networks, and propose an algorithm to calculate role dissimilarity for undirected social networks.
- We propose a method to enhance edge-perturbing anonymization algorithms in order to preserve structural properties of social networks by preserving their role structures.
- We propose a method to enhance edge-perturbing anonymization algorithms by minimizing changes in shortest paths.
- We present extensive evaluation of the proposed enhancements tested on different edge perturbation algorithms using both real-world and synthetic stylized social networks.

The rest of the paper is organized as follows: In Section 2, we provide an abstract representation of the key existing edge-perturbing anonymization algorithms. In Section 3, we define the notions of structural roles in social networks and role dissimilarity, and present an algorithm to calculate them for undirected networks. We propose our approaches to preserve structural properties by using concepts of roles and edge betweenness in Sections 4 and 5, respectively. We evaluate the proposed approaches using multiple data sets and various evaluation metrics in Section 6. In Section 7, we discuss privacy implications of our approach and a practical way to select anonymization enhancement parameters. We review the related literature in Section 8, and subsequently conclude the paper in Section 9.

2 Edge-Perturbing Anonymization

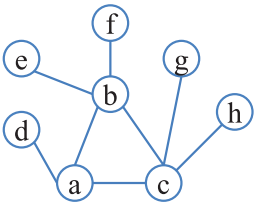
An undirected social network is defined as a graph, $G\langle V, E\rangle$, where the set of vertices V represents the agents in the network, and the set of undirected edges $E \subseteq \{(u,v)|u,v \in V\}$ represents the relationships between agents in V. Edge-perturbing anonymization techniques modify edges of a network to satisfy certain *anonymization criteria*. These techniques typically follow a greedy iterative approach, which can be abstractly expressed as in Algorithm 1.

Algorithm 1. Iterative Edge Perturbation
Input: $G\langle V, E\rangle$
Output: Anonymized version of $G\langle V, E\rangle$

```
repeat
    if an edge should be added then
        Choose non-existent edge (u, v) to be added
        E ← E ∪ {(u, v)}
    if an edge should be removed then
        Choose existing edge (u, v) to be removed
        E ← E\{(u, v)}
    if anonymization criteria is not achievable then
        return null
until anonymization criteria is achieved
return G⟨V, E⟩
```

In each iteration, Algorithm 1 selects an edge to be added/removed using a heuristic which depends on the specific technique. The iterations continue until the graph is considered anonymized according to the anonymization criteria. Different anonymization techniques have different anonymization criteria. In the *random perturbation* technique [2], the goal is to simply remove m edges randomly and then add m random edges. In the *k-anonymity-based* approaches (e.g., [8], [9], [10], [11]), the goal is, for instance, to achieve a graph with k-anonymous vertex degrees (e.g., `Supergraph` [8]). The algorithm aborts if the anonymization criteria cannot be achieved, which is also dependent on the actual technique.

The `Greedy-Swap` algorithm proposed in [8] includes an optimization phase to select a group of edge changes in the graph in each iteration, which results in a slightly different scheme (see Algorithm 2). The algorithm first creates an anonymized random graph based on a k-anonymous degree sequence of the original graph. In each iteration, every pair of edges in a subset of existing edges is examined for a *swap*. In a *swap* operation, a pair of edges are replaced with another pair with the same end nodes. Two swap options are considered for a pair of edges $\{(u,v),(u',v')\}$: either $\{(u,u'),(v,v')\}$, or $\{(u,v'),(u',v)\}$. Such swaps do not change vertex degrees, thus, ensuring the already-established degree k-anonymity. A *gain value* is calculated for each swap option, and the swap with maximum (positive) gain is selected. In [8], the authors calculate the *gain value* as the increment of edge overlap (intersection) between the interim and the original graph. Performing the swap with maximum gain at each iteration would greedily make the anonymized graph more structurally similar to the original one.



structural:
{{a},{b},{c},{d},{e, f},{g,h}}
automorphic:
{{a},{b,c},{d},{e,f,g,h}}
regular:
{{a,b,c},{d,e,f,g,h}}

Fig. 1. Sample equivalency classes for a network.

Algorithm 2. `Greedy-Swap`
Input: $G\langle V, E\rangle$
Output: Anonymized version of $G\langle V, E\rangle$
1: Create anonymized random social network $G'\langle V, E'\rangle$, where $|E'| = |E|$
2: **repeat**
3: Select randomly $\log(|E'|)$ edges from E'
4: **for all** pairs of selected edges (u, v) and (u', v') **do**
5: Calculate the ***gain value*** considering swapping of the pair either with (u, u') and (v, v'), or (u, v') and (u', v)
6: Perform the swap with maximum gain (if any)
7: **until** no edge swap is performed
8: **return** $G'\langle V, E'\rangle$

3 Structural Roles

There are three major approaches to classify agents in a network based on relationships among them into their social positions: *structural*, *automorphic*, and *regular equivalence* [14], [16]. Two agents are *structurally equivalent* if they have identical ties with the same other agents. In *automorphic equivalence*, agents in the same position must have identical ties with different sets of agents that play the same role in relation to that position. Finally, two agents are *regularly equivalent* if they have same kind of relationships with agents that are also regularly equivalent. Fig. 1 shows a small example of equivalency classes based on each of these concepts. In this work, we rely on regular equivalence as it captures the concept of roles very well, and is the least restrictive among the three concepts.

In this section, we formally define the notion of roles in the context of undirected social networks, adopting some definitions from [17]. We also define the extent of regular equivalence between roles and introduce an algorithm for identifying roles and calculating such a measure.

3.1 Roles Based on Regular Equivalence

Definition 1 (Role Assignment). *A role assignment for network $G\langle V, E\rangle$ is a function $\Phi : V \to R$, defined for every member of V, where R is a set of roles.*

A role assignment partitions agents into equivalency classes. Two agents are considered *equivalent* ($\equiv_\Phi$) if they are assigned the same role: $\forall u, v \in V; u \equiv_\Phi v \Leftrightarrow \Phi(u) = \Phi(v)$. In other words, a role assignment is a projection of an equivalence relation. Of our particular interest is the *regular equivalence*. The following definition captures the relationships between agents.

Definition 2 (Neighbor Role Set). *$\Gamma_\Phi : V \to 2^R$ is a function that maps an agent in network $G\langle V, E\rangle$ to the roles of its neighbors according to role assignment Φ, i.e., $\Gamma_\Phi(u) = \{\Phi(v)|(u, v) \in E\}$.*

We recall that regularly equivalent agents must have the same kind of relationships with other regularly equivalent agents. Therefore, we define a role assignment that projects a regular equivalence relation as follows:

Definition 3 (Regular Equivalence Role Assignment). *A role assignment $\Phi : V \to R$ projects a regular equivalence for agents in network $G\langle V, E\rangle$ iff*

$$\forall u, v \in V, \Phi(u) = \Phi(v) \Rightarrow \Gamma_\Phi(u) = \Gamma_\Phi(v).$$

Two agents are regularly equivalent if and only if they have neighbors with the same roles. We refer to this as *RE-role assignment* in the paper.

3.2 Extent of Role Equivalence

In the case two agents are not regularly equivalent, it is sometimes desirable to know to what extent they are playing similar roles. That is particularly useful in our enhancement approach, which relies highly on the existence of large classes of equivalent agents. As per our experiments, algorithms for computing regular equivalence usually result in low-populated equivalency classes. Therefore, as we discuss in later sections, agents playing similar roles can be considered as alternatives to the nonexisting regularly equivalent agents. We abstractly define a dissimilarity measure for roles as follows:

Definition 4 (Regular Equivalence Role Dissimilarity). *$\Delta_\Phi : V \times V \to [0, 1]$ is a role dissimilarity function for agents of network $G\langle V, E\rangle$ corresponding to role assignment Φ, where, at the two extremes, $\Delta_\Phi(u, v) = 0$ implies agents u and v have the same role ($\Phi(u) = \Phi(v)$), and $\Delta_\Phi(u, v) = 1$ implies agents u and v have completely dissimilar roles.*

The role dissimilarity measure is usually dependent on the regular equivalence computation scheme in use. We provide the details in Section 3.3.

Subsequently, we are interested in a dissimilarity measure between two sets of roles, given the dissimilarity measure for individual pairs of roles. The rationale behind computing such a dissimilarity measure is to see how similar a modified neighbor role set of an agent is to its original one if some of its ties are changed. We define this measure as follows:

Definition 5 (Regular Equivalence Role Set Dissimilarity). *Let $S, S' \subseteq R$ be two subsets of roles. The regular equivalence dissimilarity between S and S', written as $\Lambda(S, S')$, is equal to:*

$$\frac{\sum_{x \in S} \sqrt[|S'|]{\prod_{y \in S'} \Delta(x,y)}}{|S|} + \frac{\sum_{y \in S'} \sqrt[|S|]{\prod_{x \in S} \Delta(x,y)}}{|S'|} \Big/ 2.$$

The above formula essentially calculates the (asymmetric) dissimilarities of S to S', and S' to S, and then takes the average to compute an overall (symmetric) dissimilarity between S and S'. The dissimilarity of S to S' (the first

expression in the numerator) is calculated as follows: For every role x in S, the product of its dissimilarities with all roles in S' is calculated, and its $|S'|^{\text{th}}$ root is taken. This gives us an overall dissimilarity value between x and roles in S'. If one of the roles in S' is the same as x, the result would be zero; otherwise, the dissimilarity values for each will be effective in the result. The average of all such dissimilarities for all the roles in S is considered as the dissimilarity of S to S'. The dissimilarity of S' to S is calculated in a similar fashion.

3.3 An Algorithm to Identify Roles

REGE is a simple algorithm to partition agents based on regular equivalence. However, as pointed out by Borgatti [18], there are some inconsistencies with the algorithm in recognizing regular equivalence partitions. Also, there are issues related to the similarity measure it generates such as being affected by the degree of nodes (which theoretically should not occur because of the nature of regular equivalence). CATREGE [18] is an alternative solution for finding regular equivalence in categorical network data, i.e., networks with different types of edges. It also works for noncategorical data that are the concern of this paper. Moreover, the similarity measure computed by CATREGE avoids the above-mentioned issues.

In this paper, we deal with noncategorical (single-type edge), undirected social networks. Employing CATREGE for such social networks results into an uninteresting regular equivalence: all agents will be classified in a same equivalency class. We modify the CATREGE algorithm to tackle this issue, as shown in Algorithm 3. We initialize our algorithm with two partitions of equivalent agents: agents with minimum degree (degree one or larger, while disregarding isolates) form one partition, and the rest of the agents form the other partition. In each iteration, the algorithm checks if pairs of nodes that were equivalent in the previous iteration are connected to other agents that were equivalent themselves. If not, they are marked as nonequivalent. The procedure is repeated until there is no change in the equivalencies compared to the previous iteration. The extent of regular equivalence between two agents can be obtained by counting the number of iterations it takes them to split into different partitions. The algorithm obtains a normalized dissimilarity by subtracting this value from and dividing it by the total number of iterations. A naive implementation of Algorithm 3 has time complexity $O(dn^3)$, where n and d are node count and maximum node degree of the input network. However, in practice, the algorithm converges much sooner than the worst case complexity indicates.

Algorithm 3. Calculate Agents' Role Dissimilarities

1: $p_1 \leftarrow \{u \in V | degree(v) = Min(degree(v_i \in V))\}$
2: $p_2 \leftarrow V \setminus p_1$
3: $P \leftarrow \{p_1, p_2\}$
4: Initialize Φ according to partition set P:
   $\forall u \in V[u \in p_i \rightarrow \Phi(u) = r_i]$
5: $j \leftarrow 1$
6: **repeat**
7: **for** each partition $p \in P$ **do**
8: Split p into independent partitions $\{p_i\}$ such that
   $\forall u, v \in p\ [u, v \in p_i \leftrightarrow \Gamma_\Phi(u) = \Gamma_\Phi(v)]$
9: **if** $\Gamma_\Phi(u) \neq \Gamma_\Phi(v)$ **then**
10: $Similarity(u, v) \leftarrow j$
11: Substitute p in P with partitions $\{p_i\}$
12: Update Φ according to partition set P
13: $j \leftarrow j + 1$
14: **until** no changes in partitions set P
15: **for** every pair of agents u and v **do**
16: $\Delta_\Phi(u, v) \leftarrow (j - Similarity(u, v))/j$

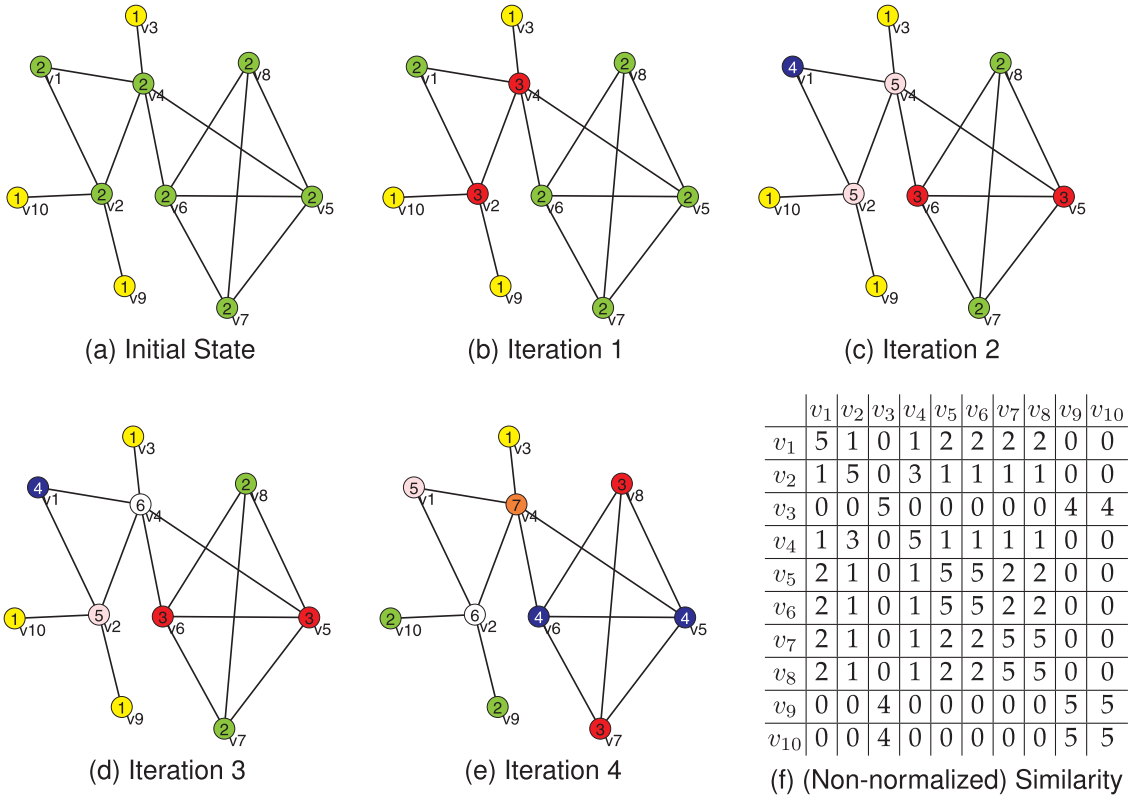
The above algorithm is different from the original CATREGE in two aspects. First, it does not deal with multiplex matrix required for categorical data. Second, it begins with a specific initial partitioning, as opposed to all agents being in the same partition in the original CATREGE. Our initial partitioning essentially indicates that peripheral agents in a network are more regularly equivalent to each other, and less so with the other agents that fall inside the network.

Fig. 2 illustrates the execution of Algorithm 3 on a small network. In each iteration, nodes within a same partition are marked with a same color (number). Note that partition colors (numbers) only indicate equivalent agents in one iteration and do not carry any other semantics. In the initial state (Fig. 2a), vertices v_3, v_9, and v_{10} are colored yellow, and all the others are colored green. Fig. 2b illustrates the resultant partitions after the first iteration. Since in the previous step, the yellow vertices were all connected to the green vertices, they will not separate in this iteration. However, the previously green vertices are divided into two partitions: the ones that were only connected to greens, and the ones that were connected to both yellows and greens. If we continue the procedure, the final result is obtained after iteration 4 (Fig. 2e); further iterations will not change the partitions. Fig. 2f shows the (nonnormalized) extent of regular equivalence between agents. For instance, the similarity value between v_1 and v_2 is 1, because they were separated after the first iteration. Analogously, the similarity value between v_5 and v_7 is 2, because they were separated after the second iteration. If two vertices eventually remain equivalent, their similarity value will be the maximum number of steps (e.g., 5 for v_5 and v_6). These values are converted to dissimilarity measure in the last statement of Algorithm 3.

4 Preserving Structural Properties Using Roles

Our intuition is that preserving the role structure of a social network in the anonymization process would preserve to some extent the network structural properties such as centralities. Therefore, the anonymized network will be more suitable to be used for typical network analysis. To this end, we need to ensure that an RE-role assignment in the original network is applicable to its edge-perturbed version as well. In this section, we present our proposed approach that preserves role structure, and apply it on the edge perturbation algorithms presented in Section 2.

Preserving an RE-role assignment while perturbing a social network is not straightforward. Because modifications to the edge structure of a network during perturbation and changing neighborhoods of agents can easily invalidate



	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_1	5	1	0	1	2	2	2	2	0	0
v_2	1	5	0	3	1	1	1	1	0	0
v_3	0	0	5	0	0	0	0	0	4	4
v_4	1	3	0	5	1	1	1	1	0	0
v_5	2	1	0	1	5	5	2	2	0	0
v_6	2	1	0	1	5	5	2	2	0	0
v_7	2	1	0	1	2	2	5	5	0	0
v_8	2	1	0	1	2	2	5	5	0	0
v_9	0	0	4	0	0	0	0	0	5	5
v_{10}	0	0	4	0	0	0	0	0	5	5

Partition Color Codes: 1=Yellow, 2=Green, 3=Red, 4=Purple, 5=Pink, 6=White, 7=Orange.

Fig. 2. A sample execution of Algorithm 3.

an RE-role assignment for the edge-perturbed version of a network. The following theorem captures a sufficient condition to ensure that.

Theorem 1. *Let $G'\langle V, E'\rangle$ be an edge-perturbed version of network $G\langle V, E\rangle$. An RE-role assignment Φ for G is also an RE-role assignment for G' if*

$$\forall u \in V \ \left[\Gamma^{G'}_{\Phi}(u) = \Gamma^{G}_{\Phi}(u)\right].$$

Proof. The proof follows from the above condition and Definition 3. Given Φ is an RE-role assignment for G, for a pair of agents u and v where $\Phi(u) = \Phi(v)$, by Definition 3, we have $\Gamma^{G}_{\Phi}(u) = \Gamma^{G}_{\Phi}(v)$. Now, if the condition in the theorem is true, we also have $\Gamma^{G'}_{\Phi}(u) = \Gamma^{G}_{\Phi}(u)$ and $\Gamma^{G'}_{\Phi}(v) = \Gamma^{G}_{\Phi}(v)$. Based on these three equalities, we have $\Phi(u) = \Phi(v) \Rightarrow \Gamma^{G'}_{\Phi}(u) = \Gamma^{G'}_{\Phi}(v)$, which is a necessary and sufficient condition for Φ to be an RE-role assignment for G', according to Definition 3. □

Theorem 1 states that keeping the neighbor role sets of agents in a network intact in the anonymization process will preserve an RE-role assignment. As an edge perturbation algorithm involves a series of edge additions/removals, the above condition can be further elaborated with regards to the set of added or removed edges as in the following theorem.

Theorem 2. *Let $G'\langle V, E'\rangle$ be an edge-perturbed version of network $G\langle V, E\rangle$. An RE-role assignment Φ for G is also an RE-role assignment for G' if the following conditions are met:*

$$\forall (u, v) \in E_i \ \exists (u, v') \in E \ [\Phi(v) = \Phi(v')] \tag{1}$$

$$\forall (u, v) \in E_d \ \exists (u, v') \in E' \ [\Phi(v) = \Phi(v')], \tag{2}$$

where sets $E_i = E'\backslash E$ and $E_d = E\backslash E'$ represent added and removed edges, respectively.

Proof. Since the same role assignment Φ is considered for both G and G', any difference between $\Gamma^{G}_{\Phi}(u)$ and $\Gamma^{G'}_{\Phi}(u)$, for any agent u, can only be the result of either addition or removal of an edge adjacent to u. For an added edge (u, v), by condition (1), we have $\exists (u, v') \in E[\Phi(v) = \Phi(v')]$ and therefore $\Phi(v) = \Phi(v') \in \Gamma^{G}_{\Phi}(u)$, i.e., an added edge would not affect the neighbor role set of an agent. For a removed edge $\langle u, v\rangle$, by condition (2), we have $\exists (u, v') \in E'[\Phi(v) = \Phi(v')]$ and therefore, $\Phi(v) = \Phi(v') \in \Gamma^{G'}_{\Phi}(u)$, i.e., a removed edge would not affect the neighbor role set of an agent. These suggest

$$\forall u \in V \ [\Gamma^{G'}_{\Phi}(u) = \Gamma^{G}_{\Phi}(u)],$$

which is a sufficient condition for Φ to be an RE-role assignment for G', according to Theorem 1. □

4.1 Role-Enhanced Iterative Edge Perturbation

Based on Theorem 2, we extend and enhance the iterative edge perturbation techniques represented by Algorithm 1 as follows: After selecting an edge for addition, it is added only if it conforms to condition (1). For this purpose, line 4 of the algorithm should be replaced with the following:

if $\exists(u,v'),(u',v) \in E\ [\Phi(v) = \Phi(v') \wedge \Phi(u) = \Phi(u')]$
then
$E \leftarrow E \cup \{(u,v)\}$

This checks if there exists vertex v' in u's neighborhood with the same role as v's, and if there exists vertex u' in v's neighborhood with the same role as u's. If either of the checks fails the edge addition is discarded. Analogously, an edge removal should be allowed only if it conforms to condition (2). As per Theorem 2, such a modified version of Algorithm 1 will preserve an RE-role assignment for the graph in each iteration. Therefore, an RE-role assignment for the original social network graph will be valid for its final edge-perturbed version.

Although theoretically sound, the above-mentioned strategy may not perform well in practice. Based on our experiments on network data sets, algorithms such as the one presented in Section 3.3 identify very small number of agents with the same role. Therefore, when adding/removing an edge, e.g., (u,v), the chance of finding an agent with the same role as v's in u's neighborhood is very low, and vice versa. The above-mentioned strategy is hard to be applied in such a situation as it would reject changes to the network, because of low population of equivalent agents in every class. In order to overcome this limitation, we use a relaxed version of the conditions in Theorem 2. Instead of an exact role match as proposed in the conditions, we propose a partial match by using a threshold on RE-role dissimilarity between agents. Algorithm 4 provides the pseudocode for the enhanced version of the iterative edge perturbation approach. As the input arguments, it requires role dissimilarity values for agents (Δ_Φ) and a threshold, $\delta \in [0,1]$, which indicates the extent of nonperfect role matching to be allowed. The time complexity of Algorithm 4 is clearly dependent on the actual iterative edge perturbation method. However, it will be bounded by $O(n^2)$ since in the worst case all the candidate edges for addition would be tested.

Algorithm 4. Role-Enhanced Iterative Edge Perturbation
Input: $G\langle V,E\rangle$, Δ_Φ, and δ
Output: Anonymized version of $G\langle V,E\rangle$

```
repeat
  if an edge should be added then
    Choose non-existent edge (u,v) to be added
    if ∃(u,v'),(u',v) ∈ E [Δ_Φ(v,v') < δ ∧ Δ_Φ(u,u') < δ] then
      E ← E ∪ {(u,v)}
  if an edge should be removed then
    Choose existing edge (u,v) to be removed
    E' ← E\{(u,v)}
    if ∃(u,v'),(u',v) ∈ E' [Δ_Φ(v,v') < δ ∧ Δ_Φ(u,u') < δ] then
      E ← E'
  if anonymization criteria is not achievable then
    return null
until anonymization criteria is achieved
return G⟨V,E⟩
```

The following example demonstrates how maintaining the neighbor role set of an agent can help in preserving structural properties of a network. In Fig. 2e, assume we need to remove one of the adjacent edges to v_4. Based on our proposed scheme, an agent should exist in v_4's neighborhood that has low role dissimilarity with the agent that is removed from that neighborhood, and vice versa. Note that we can instead consider high similarity values in Fig. 2f. The following list shows the most similar agent in v_4's neighborhood after removing each candidate from it.

- v_1: v_5/v_6 with similarity value 2.
- v_2: $v_1/v_5/v_6$ with similarity value 1.
- v_3: No similar agent exists.
- v_5: v_6 with similarity value 5.
- v_6: v_5 with similarity value 5.

The following list shows the most similar agent in each of the above candidate's neighborhood that may replace v_4's role.

- v_1: v_2 with similarity value 3.
- v_2: v_1 with similarity value 1.
- v_3: No other neighbor agent exists.
- v_5: $v_6/v_7/v_8$ with similarity value 1.
- v_6: $v_5/v_7/v_8$ with similarity value 1.

As suggested by the above similarity values, edge (v_1,v_4) seems to be the best option to remove. Because $v_1(v_4)$ has an agent in its neighborhood with moderate similarity to $v_4(v_1)$. To ensure this is indeed the best choice, we calculate two network measures, i.e, mean betweenness and closeness centralities, for the result of each case. The mean betweenness (closeness) centrality for the original network is $5.1(0.487)$, and updates as the following after removing each of the candidates:

- v_1 : $5.7\ (0.458)$
- v_2 : $6.9\ (0.411)$
- v_3 : $3.9\ (0.318)$
- v_5 : $5.7\ (0.456)$
- v_6 : $5.7\ (0.456)$

The above centrality values confirm our choice of removing (v_1,v_4), since the corresponding result network has closer centrality values to the original network compared to the other choices.

4.2 Role-Enhanced `Greedy-Swap`

As mentioned in Section 2, the `Greedy-Swap` algorithm follows a different overall procedure than most of the other perturbation approaches. Hence, we need a different approach to enhance it for preserving role structure. We do so by proposing a new *gain function* in Algorithm 2. Recall that the `Greedy-Swap` technique [8] starts with an anonymized version of the network but with randomized edges, and performs edge swaps to make the anonymized network as similar as possible to the original graph. To this end, the authors define the gain measure as the increase in the edge overlap between the original and the anonymized networks. We propose to substitute the *gain function* in Algorithm 2 with a *role similarity gain* measure which is calculated based on regular equivalence role structure. The role similarity gain function measures how much each of the involved vertices in an edge swap gets closer (more similar) to its corresponding original state in terms of the role structure. Recall from Theorem 1 that the neighbor role

set of an agent acts as an important factor in preserving its role. Hence, we consider it as the main clue for calculating such a role similarity gain.

Let u be an agent involved in an edge swap, and $\Gamma_\Phi^G(u)$ be its neighbor role set in the original network. Also, in the ith iteration of Algorithm 2, let $\Gamma_\Phi^{G_i}(u)$ be its neighbor role set in the interim network, and $\Gamma_\Phi^{G_{i+1}}(u)$ be its neighbor role set in the next state of the interim network if the swap is performed. The objective of optimization based on role similarity gain is to obtain better similarity between $\Gamma_\Phi^{G_{i+1}}(u)$ and $\Gamma_\Phi^G(u)$ compared to $\Gamma_\Phi^{G_i}(u)$ and $\Gamma_\Phi^G(u)$. In other words, using dissimilarity measures defined in Section 3.2, we need to have

$$\Lambda\big(\Gamma_\Phi^G(u), \Gamma_\Phi^{G_i}(u)\big) \geq \Lambda\big(\Gamma_\Phi^G(u), \Gamma_\Phi^{G_{i+1}}(u)\big).$$

Hence, the role similarity gain can be measured by the *decrease in dissimilarity* between the neighbor role sets in the original and interim networks. The bigger the gap between the two sides of the above inequality, the larger the role similarity gain will be. As there are four vertices involved in a swap of a pair of edges (u, v) and (u', v'), we calculate the total role similarity gain of a swap as follows:

$$\frac{\sum_{x\in\{u,v,u',v'\}}\big[\Lambda\big(\Gamma_\Phi^G(x), \Gamma_\Phi^{G_i}(x)\big) - \Lambda\big(\Gamma_\Phi^G(x), \Gamma_\Phi^{G_{i+1}}(x)\big)\big]}{4}.$$

The time complexity is $O(\log^2 n)$ for Algorithm 2, and $O(d^2)$ for the above gain function, where n and d are node count and maximum degree of the input network, respectively. Therefore, the time complexity for role-enhanced `Greedy-Swap` is $O(d^2 \log^2 n)$.

5 Preserving Structural Properties Based on Edge Betweenness

Computing shortest paths between pairs of nodes in a network is an underlying factor for social network analysis measures, ranging from simple graph-level measures such as characteristic path length (average path length (APL) between node pairs) and diameter (maximum shortest path in the network) to node level centrality measures such as betweenness (proportion of shortest paths that pass through a node) and closeness (average distance of a node to all the other nodes). In this section, we propose an algorithm to maintain structural properties in a perturbed network by limiting the amount of changes to the shortest paths in the network.

We leverage the notion of *edge betweenness* to control the shortest paths, which was introduced in the Newman-Girvan community detection algorithm [19]. Edge betweenness is defined for an edge as the number of shortest paths between any pair of nodes that pass through that edge. If there are more than one shortest paths for a pair of nodes, they are counted proportionally so that they sum up to unity. Intuitively, based on the definition, if an edge has a low edge betweenness centrality removing/adding it from/to the network will have less effect on the shortest paths compared to removing/adding an edge with higher betweenness. A lesser number of shortest path changes in the network due to such edge addition/removal would help to have less change in SNA measures such as closeness. However, note that although this strategy limits the number of shortest path changes, it cannot control the amount of change in the shortest paths. This observation is central to our proposed approach to perturb a network with limited changes to shortest paths.

Algorithm 5 provides the pseudocode for the enhanced version of the iterative edge perturbation anonymization using edge-betweenness. Here, function *Normalized-Edge-Betw* calculates the betweenness of an edge and normalizes it based on the maximum edge betweenness value of the edges in the graph. $\beta \in [0, 1]$ is an input argument that limits the potential set of edges to be added/removed, based on their normalized edge betweenness centrality value. Edge-betweenness calculation has time complexity $O(ne)$, where n and e are node and edge count, respectively. Therefore, a naive implementation of the algorithm will have time complexity $O(n^3 e)$.

Algorithm 5. Edge-Betweenness-Enhanced Iterative Edge Perturbation
Input: $G\langle V, E\rangle$ and β
Output: Anonymized version of $G\langle V, E\rangle$

repeat
  if an edge should be added **then**
    Choose non-existent edge (u, v) to be added
    $E' \leftarrow E \cup \{(u, v)\}$
    if $Normalized{-}Edge{-}Betw_{\langle V,E'\rangle}((u, v)) < \beta$ **then**
      $E \leftarrow E'$
  if an edge should be removed **then**
    Choose existing edge (u, v) to be removed
    if $Normalized{-}Edge{-}Betw_{\langle V,E\rangle}((u, v)) < \beta$ **then**
      $E \leftarrow E \backslash \{(u, v)\}$
  if anonymization criteria is not achievable **then**
    return null
until anonymization criteria is achieved

6 Experimental Results

We conducted experiments to evaluate the performance of our proposed enhancement methods by extending three major edge-perturbing anonymization algorithms and testing them on five different social network data sets.

6.1 Data Sets

Since our approach relies on structural properties of the input network, we expected it to perform differently for different data sets. Therefore, we chose two real-world social networks, and generated three synthetic networks with different topologies, as follows:

- **`PolBooks`:** A network of books about US politics sold by Amazon.com around the 2004 presidential election (compiled by V. Krebs, www.orgnet.com). Edges between books represent their frequent purchase by the same buyers.
- **`Jazz`:** A collaboration network of jazz musicians [20], where nodes represent bands and edges indicate that the corresponding bands share a common musician.

TABLE 1
Structural Properties of Social Network Data Sets

Dataset	# Nodes	# Edges	Density	Diameter	APL	C.Coef.
PolBooks	105	441	0.081	7	3.079	0.348
Jazz	198	5484	0.281	6	2.235	0.520
ER	100	1000	0.202	3	1.813	0.203
BA	100	990	0.200	2	1.927	0.118
SW	100	1014	0.205	8	3.576	0.646

- **ER:** A synthetic random network based on Erdos-Renyi model.
- **BA:** A synthetic scale-free network based on Barabasi's model.
- **SW:** A synthetic small-world network.

Table 1 lists some of their structural properties.

6.2 Evaluation Measures

In order to evaluate the effectiveness of the proposed approaches, we calculate a number of social network analysis measures on the outputs of both original anonymization algorithms and their enhanced versions, and compare them with the corresponding measurements on the original network. An anonymized output that provides closer measurements to those of the original network is intuitively more useful for analysis. We have considered the following measures: *Average path length* is the average distance of all the node pairs in the network. *Clustering Coefficient* measures the tendency of nodes in a network to cluster together, by counting the ratio of closed triplets to connected triplets in the network. *Betweenness Centrality* is the number of times a node falls on the shortest paths between node pairs in the network. *Closeness Centrality (CC)* is the average distance of a node to all the nodes in the network. We note that it is important in social network analysis to identify the most central nodes in a network. Therefore, in addition to the mean centrality values we investigate the accuracy of preserving of individual nodes' centrality rankings. For this purpose, we consider the function of an anonymization algorithm as a classifier that should classify the top-k central nodes in the original network as the most central nodes. We evaluate the performance of such a classifier by measuring the area under the ROC curve (AUC). The accuracy measures *Top3AUC* and *Top10pAUC*, respectively, indicate such performance metric for classifying the top three and the top decile of central nodes in the original network as the most central in the anonymized network. Finally, R^2 measures the square of Pearson correlation between centralities of nodes in the original and perturbed networks, as an alternative metric.

6.3 Implemented Algorithms

We implemented the original, role-enhanced, and edge-betweenness-enhanced versions of random perturbation [2], Supergraph [8], and Greedy-Swap [8]. The first two algorithms are variations of the iterative edge perturbation approach, as described in Section 2. The random perturbation algorithm first removes m edges from a network and then adds m other edges at random to the network. In our experiments, we set m equal to 10 percent of the number of edges in each network to provide adequate anonymization, as suggested by Hay et al. [2]. Also, for the role-enhanced version, we vary the role equivalence threshold (δ) between 0.3 and 1, and average the measurements over 500 runs.

Both the Greedy-Swap and Supergraph algorithms ensure degree k-anonymity for network nodes. They begin by constructing a k-anonymous *degree sequence*. The Supergraph algorithm adds edges to the network until the network meets the anonymized degree sequence. The Greedy-Swap algorithm builds an anonymized random graph based on the degree sequence and swaps its edges to obtain a network close to the original network. More details on these algorithms can be found in Section 8. In our experiments, we vary anonymization value k between 2 and 10 for role-enhanced Greedy-Swap, and evaluate edge-betweenness-enhanced Supergraph for $k = 10$ and by varying threshold β between 0.1 and 1. The reported measures are averaged based on 50 runs. Since our implementation of Supergraph only considers adding edges to the graph, i.e., increasing node degrees, it will not perform well for networks which have very few nodes with very high degrees, such as scale-free networks. Therefore, we do not perform the experiments on the Jazz and BA data sets in the case of the edge-betweenness-enhanced Supergraph algorithm.

We found the above-mentioned number of runs good enough to represent performance results while accounting for existing randomness in the algorithms. Our efforts to include other edge-perturbation algorithms and asses our proposed scheme on them were not successful, because of either possible flaws with the techniques (e.g., clustering-based methods proposed in [9] as described in Section 8), or lack of available algorithmic details to implement them (e.g., [10]).

6.4 Results

Figs. 3, 4, and 5, respectively, demonstrate the performance of role-enhanced random perturbation, role-enhanced Greedy-Swap, and edge-betweenness-enhanced Seupergraph algorithms, in terms of preserving structural properties of original networks using the measures described in Section 6.2.

In all the figures, lines with symbols represent measurements corresponding to the enhanced algorithms while lines without symbols represent results of the corresponding original algorithms. The measurements are normalized based on the original networks' measurements. For instance, in the APL plots, we divide measurements for the PolBooks network by 3.079 which is the APL of the original network. Given that the measurements are normalized, the closer a measurement is to unity, the better the network structural properties has been preserved. Improvements over the original anonymization algorithms can be easily evaluated by subtracting the corresponding measurement differences to unity. For instance, if for a specific measure the original algorithm achieves 0.7 and the enhanced algorithm achieves 1.1, the improvement is calculated as $|1 - 0.7| - |1 - 1.1| = 0.2$. Due to space limitations, we only provide accuracy measurements for betweenness centrality. Closeness centrality results were very similar in nature to those of betweenness.

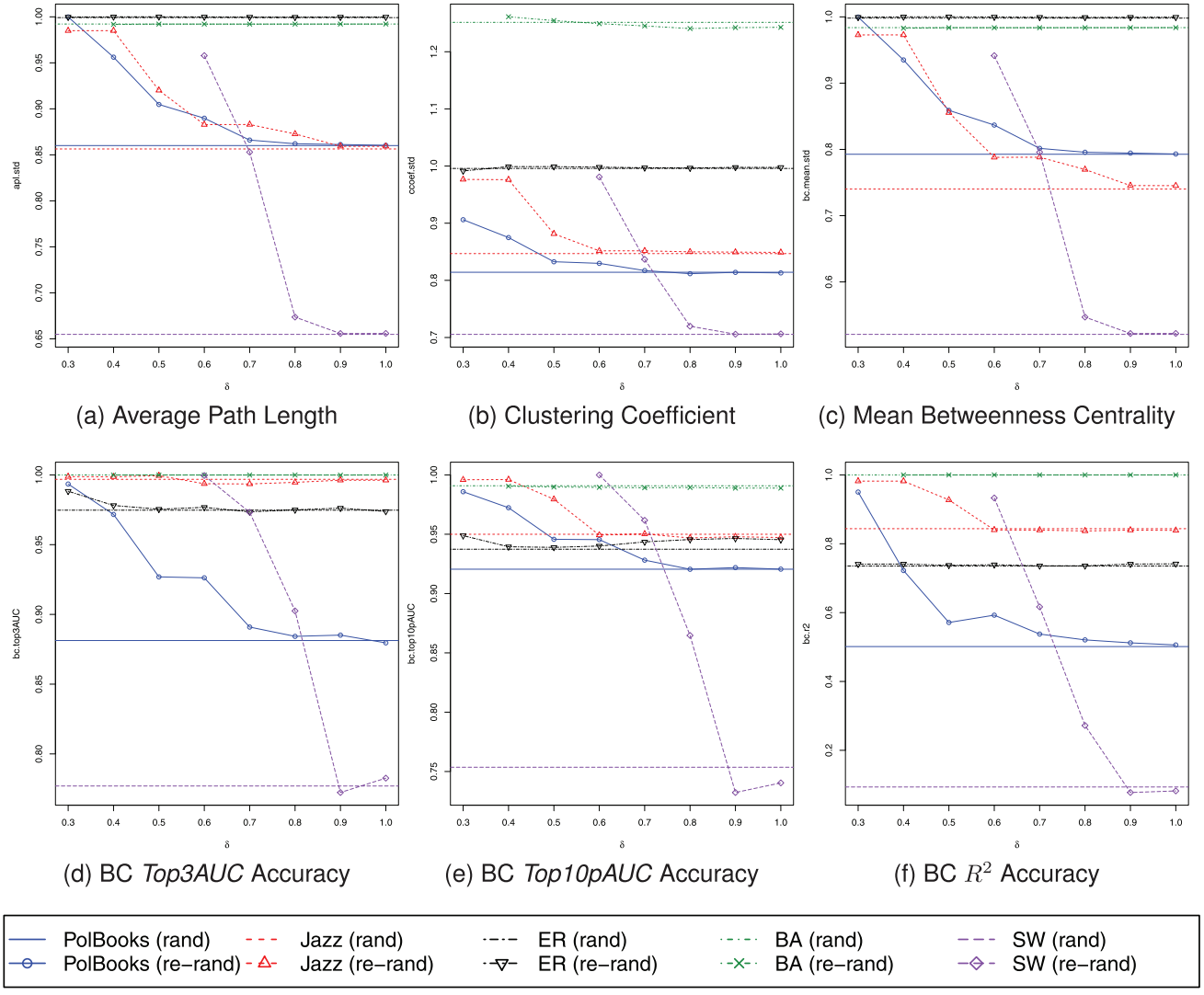


Fig. 3. Evaluation measurements for Role-Enhanced Random Perturbation ($m = 10\%|E|, 0.3 \leq \delta \leq 1$).

Our findings based on the evaluation measurements reported in Figs. 3, 4, and 5 are as follows:

Dependency on data set: As expected, the amount of distortion to the measures and effectiveness of our enhancement approach is not the same for different data sets. While the enhanced algorithms show better performances than their original counterparts in most cases, there are also cases with neutral and negligible negative effects. The role-enhanced random perturbation (Fig. 3) has almost no effect on the random and scale-free data sets, while it has significant improvement for the other three networks, i.e., `PolBooks`, `Jazz`, and `SW`. The improvement can be attributed to their similar topological characteristics as in small-world networks. In the case of role-enhanced `Greedy-Swap` (Fig. 4), results show moderate improvements on all measures, with the exceptions of scale-fee data set for the first three measures and small-world data set for the accuracy measures. The edge-betweenness-enhanced `Supergraph` (Fig. 5) also results in better performance in almost all the measures.

It is worth mentioning that the random graph (`ER`) is in fact very structurally robust to all the perturbation algorithms. Since the edges are randomly distributed in such a topology anyway, their replacements would not have much impact on the structural properties.

Effects of parameter variations: Improvements in the role-enhanced random perturbation (Fig. 3) are negatively correlated with threshold δ. That is expected since increasing δ allows for less perfect role matching, and subsequently less structural preservation. Interestingly, *improvements on structural preservation is almost independent of anonymization parameter k.* This is suggested by the almost parallel lines in Fig. 4 that correspond to the enhanced and original algorithms for each data set. As for β, the edge-betweenness enhancement parameter, it does not show monotonic improvement as seen for δ. Unexpectedly, the lower range values of β perform worse than middle-range values in preserving the measures.

Measure correlations: We notice that the APL and mean BC plots follow similar overall pattern for the same networks, for all the experiments. Also, the mean CC plots follow the same pattern in a vertically mirrored fashion, which are omitted due to space limitations. However, note that the pattern similarity is not always perfect and the improvements are in different scales. Nevertheless, this

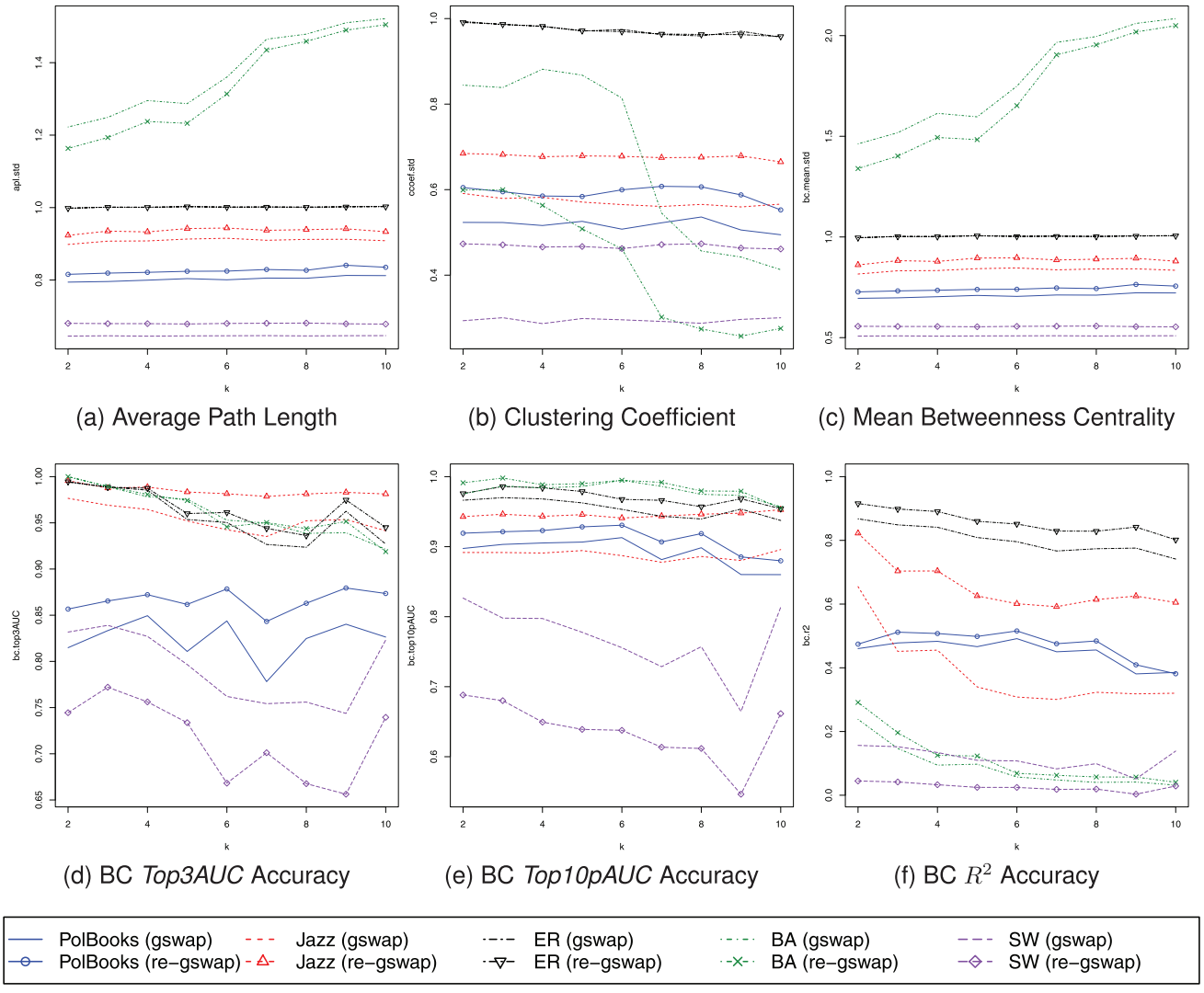


Fig. 4. Evaluation measurements for Role-Enhanced `Greedy-Swap` ($2 \leq k \leq 10$).

observation suggests that APL, mean BC, and mean CC measures are highly correlated, independent of the change in anonymization parameters and enhancement thresholds. Such correlation can significantly help in choosing an appropriate anonymization enhancement method for a data set since there will be less number of measures to be analyzed for preservation.

7 Discussions

7.1 Privacy Analysis

In this section, we analyze if the proposed enhancement approaches have any negative effect on the anonymity of the edge-perturbation schemes.

7.1.1 Effect on Anonymity Property

In the k-anonymity-based schemes, such as `Greedy-Swap` and `Supergraph`, our enhanced algorithms still fulfill the anonymity property. Therefore, there is no degradation in provision of the original anonymity property. For instance, using role-enhanced `Greedy-Swap`, for any node in the perturbed graph, there will be at least $k-1$ other nodes with the same degree. So an adversary cannot reidentify nodes based on their degree. In the case of random perturbation, our approach slightly reduces the random space based on the chosen threshold parameter, compared to the original algorithm. We believe that choosing a combination of large enough number of edge additions/removals and not-too-small threshold parameter can provide an acceptable anonymity. We plan to approach this issue more formally as a future work.

7.1.2 Role Structure as Background Knowledge

One plausible attack against role-enhanced perturbation could be to use role structure in the original and perturbed networks to reidentify nodes. Assume an adversary knows, for a target node t, role dissimilarities with every other node in the original network. We show them as a dissimilarity vector d_t of size $n = |V|$. The adversary can calculate role dissimilarities on the perturbed network, and then calculate correlation of d_t with that of every node, in the perturbed network, say d'_x. The nodes that have a high correlation in terms of role dissimilarities can be a potential match. However, such an attack is not computationally feasible. Since the attacker does not know the node identifiers, simply correlating the two vectors d_t and d'_x is not

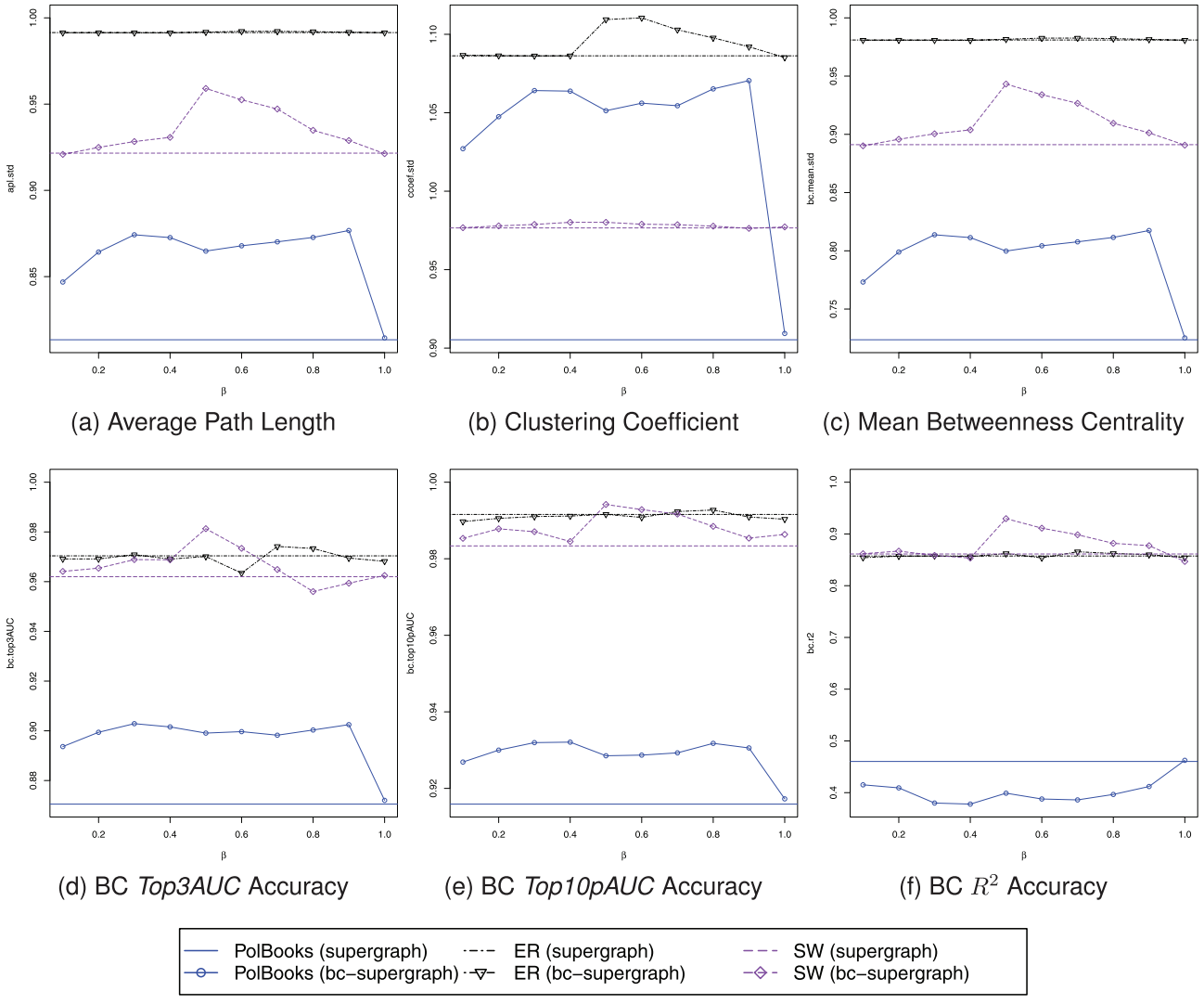


Fig. 5. Evaluation measurements for Edge-Betweenness-Enhanced `Supergraph` ($k = 10$, $0.1 \leq \beta \leq 1$).

meaningful: a dissimilarity value at a certain index in d_t may not correspond to the dissimilarity value at the same index in d'_x. In fact, d_t should be correlated against all permutations of values in d'_x. So the computation itself is of complexity $O((n+1)!)$. Even if the attacker can perform all the computations, due to nonperfect role matching in our enhanced algorithm and high sensitivity of role dissimilarity calculation algorithm to small changes in the network structure, there is no guarantee that high correlation in role dissimilarities can help in node reidentification.

7.1.3 Preserved Measures as Background Knowledge

One may argue that if certain node measures such as betweenness centrality are preserved better using our enhancement approaches, they might be misused by an attacker for node reidentification. For instance, if an enhanced algorithm provides perfect *Top1AUC* BC accuracy, an attacker that knows about the most central node will be able to easily reidentify that node in the perturbed network. Although it seems infeasible as a practical attack, one can adjust anonymization parameters so as to introduce more distortion of the centrality measures for the anonymized network (e.g., choosing a larger δ value in the case of role-enhanced random perturbation). Note that assuming such information as attacker's background knowledge contradicts the goal of the enhanced algorithms, i.e., preserving the measures.

7.2 Selecting Appropriate Thresholds

As expected and also suggested by the experimental results in Section 6, the improvement provided by our enhanced algorithms for iterative edge perturbation can be tuned based on threshold parameters. The role-enhanced and edge-betweenness-enhanced algorithms rely, respectively, on δ, the role dissimilarity threshold of considering nodes semiequivalent, and β, the edge betweenness threshold to consider an edge for addition/removal. Intuitively, we should use a threshold that results in the best preservation of structural properties of a social network. We can evaluate such preservation using different measures as suggested in our experiments. Based on our experiments, the amount of preservation depends on both the social network data set and the threshold, and does not always follow a monotonic growth/decrease with threshold values. For instance, in the role-enhanced random perturbation results, lower δ values dominantly show better performances. However, choosing

$\delta = 0.5$ over $\delta = 0.7$ for the PolBooks network provides less *Top1* BC centrality accuracy while the other BC accuracies are almost the same. Considering no significant improvement in the first case in other measures as well, one may prefer $\delta = 0.7$ over $\delta = 0.5$. This simple example shows that selecting a proper threshold value is probably not possible before actually performing enhanced anonymization on the input network for different threshold values. Note that the anonymization is an offline process before publishing a social network data set, and it seems feasible to invest some computation by trying out different thresholds to fine-tune anonymization which both guarantees anonymity criteria and provides usable output for analysis purpose.

8 Related Work

Backstrom et al. present a family of active/passive attacks that work based on uniqueness of some small random subgraphs embedded in a network [1]. Hay et al. study the extent of node reidentification based on structural information [2], [4]. They experiment using three types of structural queries as adversary background knowledge on real, naively anonymized social networks and show significantly low k-anonymity for such background knowledge queries. Narayanan and Shmatikov propose a different attack approach that relies on input of an auxiliary, overlapping, probably publicly available social network without any assumption about structural background knowledge of an adversary [3]. Empirical evaluation of their approach shows that a third of users who have accounts both on Twitter and Flickr can be reidentified in the anonymous Twitter graph with a low error rate. The nonnaive social network anonymization approaches in the literature can be categorized into two groups: graph generalization and graph perturbation.

In ***generalization techniques***, the network is first partitioned into subgraphs. Then, each subgraph is replaced by a supernode, and only some structural properties of the subgraph alongside linkage between clusters are reported. Hay et al. propose a k-anonymity-based generalization approach, where supernodes contain at least k nodes, which optimizes fitness to the original network via a maximum likelihood approach [4]. Zheleva and Getoor propose a generalization approach to avoid disclosure of exact nonsensitive edge structure, which could be used for predicting sensitive edges [5]. Campan and Truta also follow a similar approach [6] but propose a greedy optimization solution that can be tuned to control information loss. In order to use a generalized social network for analysis purpose, one should sample a random graph in accordance with the reported generalized properties. Although such a network may maintain some local structural properties of the original network, much of high-level graph structure is lost [9], which impacts negatively the utility of results.

In ***perturbation techniques***, the network is modified to meet desired privacy requirements. This is usually carried out by adding and/or removing graph edges. Although, theoretically, perturbation can be introduced to graph nodes as well, it is not considered plausible because of adverse effects on the data set.

Hay et al. propose a random perturbation approach, in which a sequence of m edge removals followed by m edge additions [2]. Assuming an adversary needs to consider the set of possible worlds implied by m removals/additions, the authors reason that it could be intractable for an attacker to achieve exact identification. However, this cannot guarantee that the adversary will not succeed in (sufficiently accurate) identification of selected individuals. Ying et al. formulate the confidence of an adversary in identifying a node in a randomly perturbed network based on the degree of the target as background knowledge [7].

Liu and Terzi propose an edge perturbation approach that provides k-anonymity for vertices based on their degrees [8]. Initially, a k-anonymous degree sequence for the graph is constructed, in which there exist at least k nodes of each degree and the total degree difference between the anonymized and the original degree sequence is minimum. Then, the problem reduces to realizing a graph with the anonymized degree sequence from the original graph. They propose two different algorithms to solve it. The Supergraph algorithm greedily perturbs the original graph until it reaches the target anonymized degree sequence. Since such a greedy algorithm cannot guarantee an answer, a probing scheme is proposed by the authors that retries the procedure with slight modification of the degree sequence, until an anonymized graph is realized. The Greedy-Swap algorithm starts by constructing a random graph based on the anonymized degree sequence. It then modifies the graph to maximize its overlap with the original graph, while preserving the anonymized degree sequence.

Thompson and Yao propose a k-anonymity-based two-phase clustering and perturbation approach [9]. Vertices are first clustered into groups of size of at least k, and then edges are greedily added/removed so that each vertex is anonymous to the vertices in its cluster. Anonymity is either based on the vertex degree or the vertex and its neighbor degrees, which is also used as similarity measure for forming clusters. They propose two alternative clustering algorithms for this purpose: Bounded t-Means, and Union-Split. Although their clustering approach seems promising, the proposed greedy perturbation algorithm based on clusters does not guarantee an answer. An approach such as probing in [8], that takes into account realizability of the graph based on formed clusters, seems necessary.

Zhou and Pei propose a scheme to k-anonymize vertex neighborhoods [10], by constructing isomorphic neighborhoods. The method seems to be inefficient and highly distorting, since it requires recurring anonymization of node neighborhoods. He et al. propose a different neighborhood anonymization scheme [11], by making isomorphic groups of k-graph partitions. Furthermore, we believe that making every k-grouped partitions isomorphic and adding back interpartition edges in an isomorphic-preserving manner as adopted in [11] will create a very symmetric structure; considering the need for addition of about k^2 edges per original interpartition edge, the result does not seem to maintain well its original structural properties in general.

We take a generic enhancement approach in this paper, rather than offering alternatives to the perturbation algorithms in the literature. Applied to such algorithms, our approach can significantly improve the utility of their results, as shown in our experiments in Section 6.

9 Conclusion

Distortion introduced to structure of social networks by network anonymization schemes can significantly reduce the utility of their outputs. In this paper, we proposed two heuristic approaches to improve preserving utility of results in the edge-perturbing anonymization algorithms. These methods help preserving structural properties of social networks by either maintaining roles based on the notion of regular equivalence in a social network, or limiting changes in the shortest paths based on the notion of edge betweenness. Our empirical results show very promising improvements in utility, without sacrificing privacy. As a future work, we plan to investigate theoretically the relation between structural roles and social network measures such as centralities. Moreover, we plan to investigate a nongreedy optimization approach for employing our second heuristic method.

Acknowledgments

This research has been supported by the US National Science Foundation (NSF) award IIS-0545912. The authors thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore Art Thou r3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography," *Proc. 16th Int'l Conf. World Wide Web,* pp. 181-190, 2007.

[2] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing Social Networks," Technical Report 07-19, Univ. of Massachusetts Amherst, 2007.

[3] A. Narayanan and V. Shmatikov, "De-Anonymizing Social Networks," *Proc. IEEE 30th Symp. Security and Privacy,* pp. 173-187, Aug. 2009.

[4] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting Structural Re-Identification in Anonymized Social Networks," *Proc. VLDB Endowment,* vol. 1, no. 1, pp. 102-114, Aug. 2008.

[5] E. Zheleva and L. Getoor, "Preserving the Privacy of Sensitive Relationships in Graph Data," *Proc. First ACM SIGKDD Int'l Conf. Privacy, Security, and Trust in KDD,* F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, eds., pp. 153-171, 2008.

[6] A. Campan and T.M. Truta, "A Clustering Approach for Data and Structural Anonymity in Social Networks," *Proc. Second ACM SIGKDD Int'l Workshop Privacy, Security, and Trust in KDD (PinKDD '08), in Conjunction with KDD '08,* 2008.

[7] X. Ying, K. Pan, X. Wu, and L. Guo, "Comparisons of Randomization and K-Degree Anonymization Schemes for Privacy Preserving Social Network Publishing," *SNA-KDD '09: Proc. Third Workshop Social Network Mining and Analysis,* pp. 1-10, 2009.

[8] K. Liu and E. Terzi, "Towards Identity Anonymization on Graphs," *SIGMOD '08: Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 93-106, 2008.

[9] B. Thompson and D. Yao, "The Union-Split Algorithm and Cluster-Based Anonymization of Social Networks," *ASIACCS '09: Proc. Fourth Int'l Symp. Information, Computer, and Comm. Security,* pp. 218-227, 2009.

[10] B. Zhou and J. Pei, "Preserving Privacy in Social Networks Against Neighborhood Attacks," *Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE),* pp. 506-515, 2008.

[11] X. He, J. Vaidya, B. Shafiq, N. Adam, and V. Atluri, "Preserving Privacy in Social Networks: A Structure-Aware Approach," *WI-IAT '09: Proc. IEEE/WIC/ACM Int'l Joint Conf. Web Intelligence and Intelligent Agent Technology,* vol. 1, pp. 647-654, 2009.

[12] L. Sweeney, "K-Anonymity: A Model for Protecting Privacy," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, pp. 557-570, 2002.

[13] A. Masoumzadeh and J. Joshi, "Preserving Structural Properties in Anonymization of Social Networks," *Proc. Sixth Int'l Conf. Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '10),* Oct. 2010.

[14] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications.* Cambridge Univ. Press, 1994.

[15] M. Newman, *Networks: An Introduction,* first ed. Oxford Univ. Press, May 2010.

[16] D. Knoke and S. Yang, *Social Network Analysis (Quantitative Applications in the Social Sciences),* second ed. Sage Publications, Inc., Nov. 2008.

[17] J. Lerner, "Role Assignments," *Proc. Network Analysis, LNCS,* U. Brandes and T. Erlebach, Eds. Springer Berlin/Heidelberg, vol. 3418, pp. 216-252, 2005.

[18] S. Borgatti, "Two Algorithms for Computing Regular Equivalence," *Social Networks,* vol. 15, no. 4, pp. 361-376, 1993.

[19] M.E.J. Newman and M. Girvan, "Finding and Evaluating Community Structure in Networks," *Physical Rev. E,* vol. 69, no.2, p. 026113, Feb. 2004.

[20] P. Gleiser and L. Danon, "Community Structure in Jazz," *Advances in Complex Systems,* vol. 6, no. 4, pp. 565-573, 2003.

Amirreza Masoumzadeh received the BS and MS degrees in computer engineering (Software) from Ferdowsi University, Iran and Sharif University of Technology, Iran in 2004 and 2007, respectively. He is currently working towards the PhD degree at the School of Information Sciences, University of Pittsburgh and is a member of the Laboratory of Education and Research on Security Assured Information Systems (LERSAIS). His research interests include information security, privacy, and trust in modern information systems. He is a student member of the IEEE and ACM.

James Joshi received the MS degree in computer science and the PhD degree in computer engineering from Purdue University in 1998 and 2003, respectively. He is an associate professor and the director of the Laboratory for Education and Research on Security Assured Information Systems (LERSAIS) in the School of Information Sciences at the University of Pittsburgh. His research interests include role-based access control, trust management, and secure interoperability. He is a member of the IEEE and ACM.