# Comparing Random-based and $k$-Anonymity-based Algorithms for Graph Anonymization

Jordi Casas-Roma, Jordi Herrera-Joancomartí, and Vicenç Torra

Department of Computer Engineering, Multimedia and Telecomunications (EIMT), Universitat Oberta de Catalunya (UOC), Barcelona. E-mail: jcasasr@uoc.edu
Department of Information and Communications Engineering (DEIC), Universitat Autònoma de Barcelona (UAB), Bellaterra. E-mail: jherrera@deic.uab.cat
Artificial Intelligence Research Institute (IIIA), Spanish National Research Council (CSIC), Bellaterra. E-mail: vtorra@iiia.csic.es

**Abstract.** Recently, several anonymization algorithms have appeared for privacy preservation on graphs. Some of them are based on randomization techniques and on $k$-anonymity concepts. We can use both of them to obtain an anonymized graph with a given $k$-anonymity value. In this paper we compare algorithms based on both techniques in order to obtain an anonymized graph with a desired $k$-anonymity value. We want to analyze the complexity of these methods to generate anonymized graphs and the quality of the resulting graphs.

**Keywords:** Privacy, Anonimization, Social networks, Graphs, k-Anonymity.

## 1 Introduction

Currently, the data mining processes require large amounts of data, which often contain personal and private information of users and individuals. Although basic processes are performed on data anonymization, such as removing names or other key identifiers, remaining information can still be sensitive, and useful for an attacker to re-identify users and individuals. E.g., birthday and ZIP codes might be enough to re-identify individuals [1]. To solve this problem, methods that perform introduction of noise in the original data have been developed in order to hinder the subsequent processes of re-identification.

In this paper we will discuss anonymization techniques applied to graph formatted data. One of the most well known data that can be represented as graphs are social networks. Social networks are very interesting for their analysis by scientists and companies, nevertheless any release to a third party for their analysis requires the application of a protection procedure.

There are multiple methods for privacy preservation in graphs. One of the most used are random-based methods, which modify graphs at random to hinder re-identification processes. Other methods are based on the concept of $k$-anonymity [1]. These methods are more complex than random-based. In this

paper we ask ourselves if we can get the same results using a random algorithm and a algorithm to select $k$-anonymous graphs.

This paper is organized as follows. In Section 2, we review different anonymization methods for graphs' privacy preservation. Section 3 presents our experimental framework, including anonymization algorithms, graph and re-identification risk assessment and data sets used in our experiments. In Section 4, we show the experiments and discuss the results. Finally, in Section 5, we discuss conclusions and future work.

## 2 State of the Art

Anonymization methods depend on the type of data they are intended to work with. In this paper, we will work with simple, undirected and unlabelled graphs. Because these graphs have no attributes or labels in the edges, information is only in the structure of the graph itself and, due to this, the adversary can use information about the structure of the network to attack the privacy. However, since all of the information is contained in it, we want to preserve the structure of the graph.

### 2.1 Random-based Methods

One widely adopted strategy of graph modification approaches are randomization methods. Randomization methods are based on adding random noise in original data. There are two basic approaches to work with graph data: (1) *Rand Add/Del*: randomly add and delete the same number of edges from the original graph (this strategy keeps the number of edges) and (2) *Rand Switch*: exchange edges between pairs of nodes (this strategy keeps the number of edges and the degree of all nodes).

Hay et al. [2] proposed a method to anonymize unlabelled graphs. This method is called *Random Perturbation* and is based on two phases: first, $m$ edges are randomly removed from the graph and then false $m$ edges are randomly added. The set of vertices is not changed and the number of edges is preserved in the anonymized graph.

Ying et al. [3] proposed a variation of *Rand Add/Del* method, called *Rand Add/Del-B*. This method implements modifications (by adding and removing edges) on the nodes at high risk of re-identification, not at random over the entire set of nodes. The authors expect to introduce fewer perturbations (with better utility preservation) to achieve the same privacy protection.

### 2.2 $k$-Anonymity-based Methods

Another strategy widely adopted for privacy-preserving is based on the concept of $k$-anonymity. This concept was introduced by Sweeney [1] for the privacy preservation on relational data. Formally, the $k$-anonymity model is defined as: let $RT(A_1, \ldots, A_n)$ be a table and $QI_{RT}$ be the quasi-identifier associated with

it. $RT$ is said to satisfy $k$-anonymity if and only if each sequence of values in $RT[QI_{RT}]$ appears with at least $k$ occurrences in $RT[QI_{RT}]$. The $k$-anonymity model indicates that an attacker can not distinguish between different $k$ records although he manages to find a group of quasi-identifiers. Therefore, the attacker can not re-identify an individual with a probability greater than $\frac{1}{k}$.

Different concepts can be used to apply the $k$-anonymity model on graphs. A widely option is to use the node degree as a quasi-identifier [4]. It is called $k$-degree anonymity. We assume that the attacker knows the degree of some nodes. If the attacker identifies a single node with the same degree in the anonymized graph, then he has re-identified this node. $K$-anonymity methods are based on modifying the graph structure (by adding and removing edges) to ensure that all nodes satisfy the $k$-anonymity properties for the degrees of all the nodes. In other words, the main objective is that all nodes have at least $k-1$ other nodes sharing the same degree.

### 2.3 Graph Assessment

Several measures and metrics have been used to quantify network structure in graph formatted data. Usually, the authors compare the values obtained by the original data and the anonymized data in order to quantify the noise introduced by the anonymization process.

Hay et al. [2] proposed five structural properties from graph theory for quantifying network structures. For each node, the authors evaluate closeness centrality (average shortest path from the node to every other node), betweenness centrality (proportion of all shortest paths through the node) and path length distribution (computed from the shortest path between each pair of nodes). For the graph as a whole, they evaluate the degree distribution and the diameter (the maximum shortest path between any two nodes). The objective is to keep these five steps closer to their original values, assuming that it involves little distortion in the anonymized data.

Zou et al. [6] defined a simple method for evaluating information loss on undirected and unlabelled graphs. The method is based on the difference between the original and the anonymized graph edges. Formally, $Cost(G, \widetilde{G}) = (E \cup \widetilde{E}) - (E \cap \widetilde{E})$ where $G(V, E)$ is the original graph, $V$ is the node set, $E$ is the edge set, and $\widetilde{G}(\widetilde{V}, \widetilde{E})$ is the anonymized graph.

### 2.4 Risk Assessment

Re-identification risk in anonymized graph is important to evaluate the quality of any anonymization process. Determining the knowledge of the adversary is the main problem. From the knowledge of the adversary, different methods for assessing the re-identification risk have been developed.

Zhou et al. [13] model the background knowledge of adversaries in various ways: Identifying attributes of nodes, nodes degrees, link relationship, neighbourhoods, embedded sub-graphs and graph metrics. We focus on a knowledge of the adversary based on degree nodes.

Hay et al. [2] [5] proposed a method, called *Vertex Refinement Queries*, to model the knowledge of the adversary. This class of queries, with increasing attack power, models the local neighbourhood structure of a node in the network. The weakest knowledge query, $\mathcal{H}_0(v_j)$, simply returns the label of the node $v_j$. The queries are successively more descriptive: $\mathcal{H}_1(v_j)$ returns the degree of $v_j$, $\mathcal{H}_2(v_j)$ returns the list of each neighbours' degree, and so on. The queries can be defined iteratively, where $\mathcal{H}_i(v_j)$ returns the multi-set of values which are the result of evaluating $\mathcal{H}_{i-1}$ on the set of nodes adjacent to $v_j$:

$$\mathcal{H}_i(v_j) = \{\mathcal{H}_{i-1}(v_1), \mathcal{H}_{i-1}(v_2), \ldots, \mathcal{H}_{i-1}(v_m)\} \tag{1}$$

where $v_1, v_2, \ldots, v_m$ are the nodes adjacent to $v_j$.

A candidate set for a query $\mathcal{H}_i$ is a set of all nodes with the same value of $\mathcal{H}_i$. Therefore, the cardinality of a candidate set for $\mathcal{H}_i$ is the number of indistinguishable nodes in $G$ under $\mathcal{H}_i$. Note that if the cardinality of the smallest candidate set under $\mathcal{H}_1$ is $k$, the probability of re-identification is $\frac{1}{k}$. Hence, the $k$-degree anonymity value for $G$ is $k$.

## 3 Experimental Set Up

Our main objective is to compare random-based and $k$-anonymity-based algorithms for privacy preservation on graphs. If we want to anonymize a graph to a specific value of $k$-anonymity, then we should ask ourselves, what kind of method is the best to achieve this purpose. I.e., we want to compare random-based and $k$-anonymity-based methods to anonymize graphs to a specific value of $k$-anonymity.

Random-based methods modify the structure of the graph, so they can modify the value of $k$-degree anonymity. But we can not specifically control the desired value. Therefore, if we want to get an anonymized graph with a specific value of $k$-anonymity, we must generate multiple anonymized graphs until we find one with the desired $k$-anonymity value.

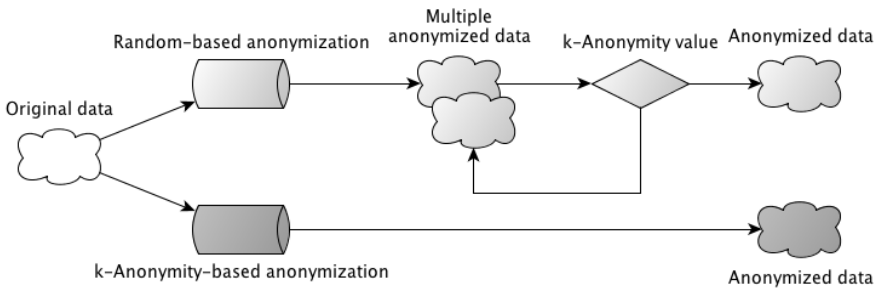



Fig. 1: Experimental framework.

To conduct this experiment, we choose three graph formatted datasets, two anonymization algorithms and several quality measures. Figure 1 shows our experimental framework. First, we anonymize the graphs data sets (details are shown in Section 3.1) using two anonymization algorithms (Section 3.2). Then, we evaluate original and anonymized data using measures for quantifying network structures (Section 3.3). And finally, we use risk assessment measures (Section 3.4) to assess the improvement in privacy-preserving on anonymized data.

### 3.1 Data Sets

Three different data sets are used in our experiments. Table 1 shows a summary of the data sets' main features. The data sets considered are the following ones:

- Zachary's Karate Club [7] is a graph widely used in the literature. The graph shows the relationships among 34 members of a karate club.
- American College Football [8] is a graph of American football games between Division IA colleges during regular season Fall 2000.
- Jazz Musicians [9] is a graph of jazz musicians and their relationships.

Table 1: Data sets properties.

| *Data set* | *Nodes* | *Edges* | *Average degree* | *Average distance* | *Diameter* |
|---|---|---|---|---|---|
| Zachary's Karate Club | 34 | 78 | 4.588 | 2.408 | 5 |
| American College Football | 115 | 613 | 10.661 | 2.508 | 4 |
| Jazz Musicians | 198 | 2,742 | 27.697 | 2.235 | 6 |

### 3.2 Anonymization Methods

We choose the following random-based and $k$-anonymity-based anonymization algorithms for our experiments.

#### Random-based Algorithm

Among all existing random-based anonymization algorithms, we use **Random Perturbation** (RP) [2]. This algorithm removes and adds the same number of edges from the original graph, by keeping the total number of edges in the graph.

As Figure 1 describes, we perform multiple anonymizations using RP algorithm. The total number of anonymized graphs depends on each data set and will be specified in each experiment. For each $k$-anonymity value we want to achieve, we execute the RP algorithm iteratively, until we get a graph with the $k$-anonymity value. The process starts with an anonymization percentage of 1%. If a graph with the desired $k$-anonymity value is generated, it is the solution

and anonymization process finishes. After 100 iterations, and if a graph with the desired $k$-anonymity value is not found, the anonymization percentage is increased at 1%. This process is repeated until the anonymization percentage reaches a limit of 50%, when the process stops without a solution. Therefore, 5,000 randomly anonymized graphs are generated before RP finishes without solution.

### *k*-Anonymity-based Algorithm

We use the **Genetic Graph Anonymization** (GGA) [10] for obtaining an anonymized graph which preserves $k$-anonymity on the degree. The approach, based on genetic algorithms, can be described in terms of the following two steps.

1. Given the degree sequence of the nodes of $G(V, E)$, $d = \{d_1, \cdots, d_n\}$, we construct a new sequence $\widetilde{d}$ that is $k$-degree anonymous and minimizes the distance between the two sequences.
2. We construct a new graph $\widetilde{G}(\widetilde{V}, \widetilde{E})$ with degree sequence $\widetilde{d}$ in which $\widetilde{V} = V$ and $\widetilde{E} \cap E \approx E$.

Our proposal for the first step of the anonymization algorithm uses genetic algorithms. These algorithms use the mutation process and the fitness function defined as follows.

**Mutation process.** Add one to an element of the degree sequence and subtract one from another element. This transaction represents a change in one of the nodes of an edge. For example, if we modify node $v_1$ to $v_2$ on the edge $e_{0,1} = (v_0, v_1)$ we get the edge $e_{0,2} = (v_0, v_2)$. This node change is represented in the degree sequence as subtracting one to the degree value of node $v_1$ and adding one to the degree value of node $v_2$. Note that our genetic algorithm does not use the recombination of pairs of parents, since this process systematically breach the rule that preserves the number of edges in the graph, and therefore generate no valid candidates.

**Fitness function.** The fitness function, which evaluates candidates, is computed from three parameters: (1) the current $k$-anonymity value, where the objective is to achieve a $k$-anonymity value greater than or equal to the desired value. (2) The distance between the anonymized and the original degree sequence, computed by Equation 2, where the objective is to minimize this value. And (3), the number of nodes that do not meet the desired value of $k$-anonymity, which will decrease until it reaches 0, when we get the desired $k$-anonymity value.

$$D(d, \widetilde{d}) = \sum_{i=0}^{n} \mid \widetilde{d_i} - d_i \mid \tag{2}$$

where $n = \mid V \mid$.

In the second step we make the necessary changes in the original graph to obtain the anonymized graph. The changes that have occurred in the degree sequence indicate the nodes that should change its degree. I.e., indicate the edges to be modified.

### 3.3 Graph Assessment

We use different measures for quantifying network structure. These measures and metrics are used to compare both the original and the anonymized data in order to quantify the level of perturbation introduced in the anonymized data by the anonymization process. These measures and metrics evaluate some key graph properties.

In the rest of this section we review the measures used. In the definitions we use $G(V, E)$ and $\widetilde{G}(V, \widetilde{E})$ to indicate the original and the anonymized graphs, $n =\mid V \mid$ to denote the number of nodes, and $d_{ij}$ to denote the length of the shortest geodesic path from node $v_i$ to $v_j$.

The first one is **average distance**. It is defined as the average of the distances between each pair of nodes in the graph. It measures the minimum average number of edges between any pair of nodes. Formally, it is defined as:

$$AD(G) = \frac{\sum_{i,j} d_{ij}}{\binom{n}{2}} \tag{3}$$

The second is **edge intersection**. It is defined as the intersection of the edges set. Formally:

$$EI(G, \widetilde{G}) = \frac{E \cap \widetilde{E}}{\mid E \cup \widetilde{E} \mid} \tag{4}$$

The third is **betweenness centrality**, which measures the fraction of the number of shortest paths that go through each vertex. This measure indicates the centrality of a node based on the flow between other nodes in the graph. A node with a high value indicates that this node is part of many shortest paths in the graph, which will be a key node in the graph structure. This measure is normalized to be in the range [0,1]. Formally, we define the betweenness centrality of a node $v_i$ as:

$$BC(v_i) = \frac{1}{n^2} \sum_{st} \frac{g^i_{st}}{g_{st}} \tag{5}$$

where $g^i_{st}$ is the number of geodesic paths from $s$ to $t$ that pass through $v_i$, and $g_{st}$ is the total number of geodesic paths from $s$ to $t$.

The fourth one is **closeness centrality**, which is defined as the inverse of the average distance to all accessible nodes. It is normalized in the range [0, 1]. Closeness is an inverse measure of centrality in that a larger value indicates a less central node while a smaller value indicates a more central node. Formally, we define the closeness centrality of a node $v_i$ as:

$$CC(v_i) = \frac{n}{\sum_j d_{ij}} \tag{6}$$

The betweenness and closeness centrality lead to different results for the same graph as they focus on different aspects of centrality. As shown above, both compute a value for each node. To compare the original and the protected graph, it is convenient to aggregate these values in a single one. For each of the two measures, we compute an average difference using the root mean square (other average functions [11] could be used here as well) as follows:

$$Diff(G, \hat{G}) = \sqrt{\frac{1}{n}((g_1 - \hat{g}_1)^2 + \ldots + (g_n - \hat{g}_n)^2)} \tag{7}$$

where $g_i$ is either the betweenness centrality or the closeness centrality of node $v_i$.

The number of nodes, edges and average degree are not considered to assess the anonymization process because the methods analysed in this work keep these values constant.

### 3.4 Risk Assessment

As we have discussed above, it is necessary to define the adversary's knowledge to define a method for assessing the re-identification risk. In this paper we assume a knowledge of the adversary based on the degree of the nodes and we use Vertex Refinement Queries of level 1 ($\mathcal{H}_1$) as a re-identification risk measures.

The $\mathcal{H}_1(v_i)$ indicates the degree of node $v_i$ and the candidate set of $\mathcal{H}_1$, $cand_{\mathcal{H}_i}$, is the set of all nodes grouped by their degree. That is, one subset corresponds to all nodes of degree value equal to 1, another to all nodes of degree value equal to 2, and so on. Therefore, the minimum cardinality of the subsets corresponds to the value of $k$-degree anonymity. But $cand_{\mathcal{H}_i}$ also shows interesting information about how re-identification risk is distributed on all nodes of the graph.

$$cand_{\mathcal{H}_1} = \{v_j \in V \mid \mathcal{H}_1(v_i) = \mathcal{H}_1(v_j)\} \tag{8}$$

In our experiments we analyse how the candidate set evolves, so this allows us to see how the graph evolves in terms of re-identification's risk.

## 4 Experimental Results

In this section, we show the results of our experiments. We compare RP and GGA algorithms to anonymize a graph with a specific $k$-anonymity value.

### 4.1 Zachary's Karate Club

The original graph has a $k$-anonymity value equal to 1. RP algorithm achieves anonymized graphs with a $k$-anonymity values equal 2,3 and 4, while GGA achieves graphs with a $k$-anonymity values equal to 2, 4 and 5. So, GGA algorithm gets a higher $k$-anonymity value than RP algorithm. Table 2 shows that RP algorithm is much faster than GGA algorithm.

Table 2: Zachary's Karate Club generation time.

| *Algorithm* | *k=2* | *k=3* | *k=4* | *k=5* |
|---|---|---|---|---|
| RP | 00:01 sec | 00:06 sec | 00:53 sec | - |
| GGA | 00:33 sec | - | 01:38 sec | 02:34 sec |

Average distance is shown in Figure 2a. GGA algorithm achieves better results than RP algorithm for all values of $k$. Note that the $k = 1$ values correspond to the original graph. Figure 2b shows edge intersection between original and anonymized graphs. Also, GGA algorithm achieves better results for all values of $k$. In addition, RP algorithm obtains a very bad result for $k$=4, where edge intersection measure falls to 20%.

The RMS error of the betweenness centrality, Figure 2c, and the RMS error of the closeness centrality, Figure 2d, show similar results on both measures, where GGA introduces less perturbation than RP.

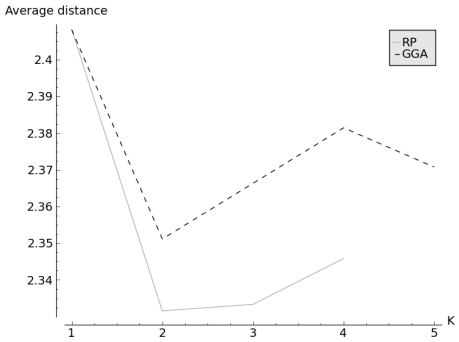
Figures 2e and 2f show the details of the $cand_{\mathcal{H}_1}$ results for RP and GGA algorithms. Nodes with a candidate set of size 1 have been uniquely re-identified (6 nodes, 17.64%, on the original graph). Nodes with a candidate set of size between 2 and 4 are in high risk of re-identification (5 nodes, 14.70%, on the original graph). However, nodes with candidates set between 5 and 10 and greater than 10 are well-protected (23 nodes, 67.64%, on the original graph).

If we compare the results of anonymized graphs with a value of $k$=2, we can see that the GGA algorithm achieves a smaller set of nodes at high risk of re-identification (41.17% RP and 35.29% GGA). If we compare the results with a value of $k$=4, we can see that the GGA algorithm achieves a smaller set of nodes at high risk of re-identification and a bigger set of well-protected nodes.

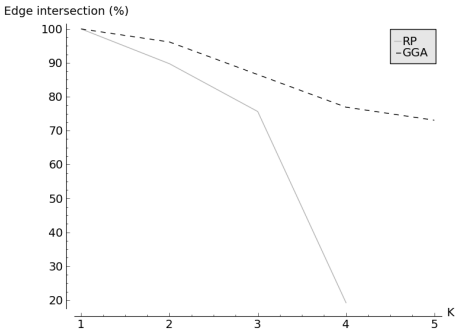
### 4.2 American College Football

The original graph has a $k$-anonymity value equal to 1. RP algorithm get values of $k$-anonymity of 2, 3, 4, 5 and 6, and GGA algorithm get values of $k$-anonymity of 4 and 10. Table 3 shows generation time for both algorithms. Like in the previous experiment, RP algorithm is faster than GGA algorithm, but GGA algorithm achieves higher values of $k$-anonymity than RP algorithm.

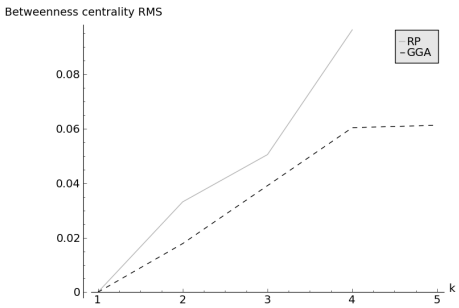
In Figure 3a we can see that GGA algorithm gets much better results on average distance than RP algorithm, especially for $k$-anonymity values greater than 5. Figure 2b shows the same behaviour for edge intersection.



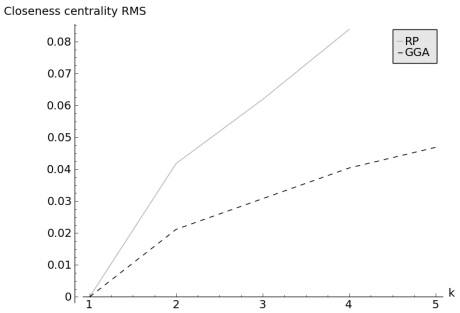

(a) Average distance



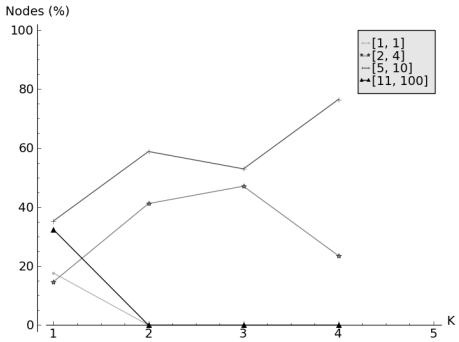

(b) Edge intersection



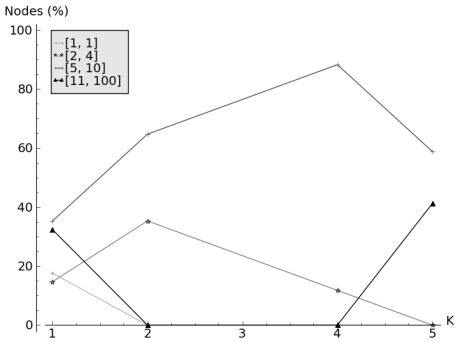

(c) RMS error of betweenness centrality




(d) RMS error of closeness centrality




(e) $cand_{\mathcal{H}_1}$ evolution for RP algorithm

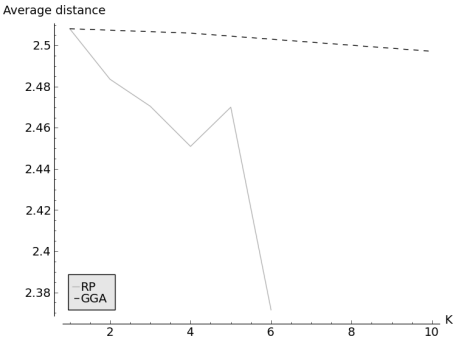



(f) $cand_{\mathcal{H}_1}$ evolution for GGA algorithm

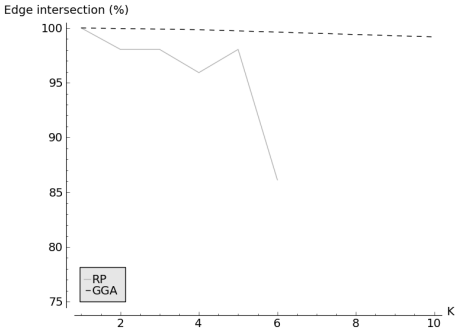
Fig. 2: Zachary's Karate Club

Table 3: American College Football generation time.

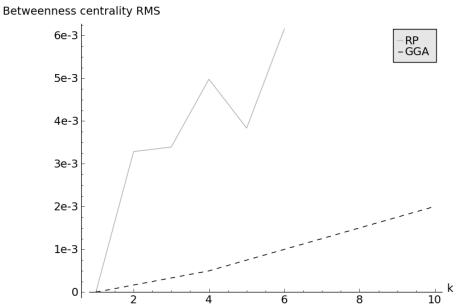
| *Algorithm* | *k=2* | *k=3* | *k=4* | *k=5* | *k=6* | *k=7* | *k=8* | *k=9* | *k=10* |
|---|---|---|---|---|---|---|---|---|---|
| RP | 00:01 sec | 00:01 sec | 00:04 sec | 00:06 sec | 01:21 sec | - | - | - | - |
| GGA | - | - | 00:51 sec | - | - | - | - | - | 02:01 sec |



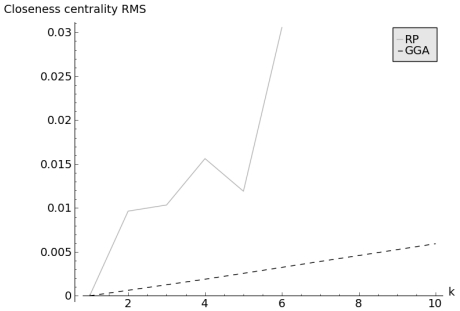

(a) Average distance



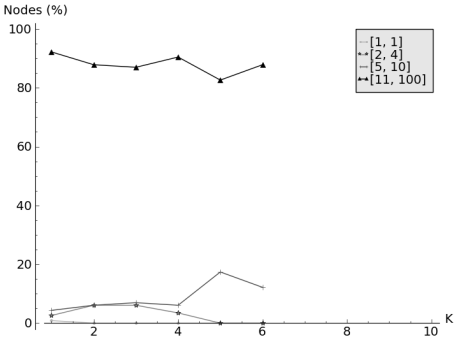

(b) Edge intersection



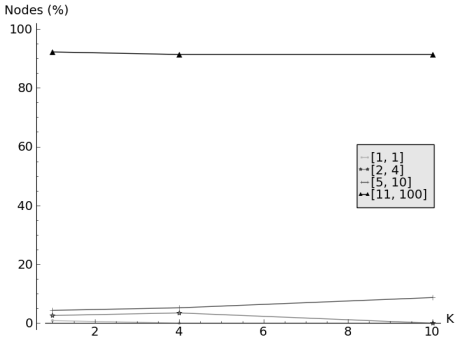

(c) RMS error of betweenness centrality




(d) RMS error of closeness centrality




(e) $cand_{\mathcal{H}_1}$ evolution for RP algorithm




(f) $cand_{\mathcal{H}_1}$ evolution for GGA algorithm

Fig. 3: American College Football

The RMS error of the betweenness centrality, Figure 3c, and the RMS error of the closeness centrality, Figure 3d, show that GGA algorithm introduces less perturbation in both measures.

Figures 3e and 3f show the details of the $cand_{\mathcal{H}_1}$ results for RP and GGA. GGA algorithm achieves excellent results at all the range of anonymization. RP algorithm achieves good results too, but fall short of those achieved by GGA.

### 4.3 Jazz Musicians

The original graph has a $k$-anonymity value equal to 1. Both algorithms only achieve graphs with a $k$-anonymity value equal to 2. Table 4 shows generation time for these processes.

Table 4: Jazz Musicians generation time.

| *Algorithm* | *k=2* |
|---|---|
| RP | 3:14:27 sec |
| GGA | 5:26:51 sec |

Using this data set, average distance decreases smoothly on GGA, like in previous data sets. Edge intersection gets a value of 99.53% on GGA $k = 2$ anonymized graph, while RP gets a value of 33.04% for a graph with the same $k$-anonymity value. This value indicates that RP affects the quality and the usefulness of the anonymized data. The RMS error of the betweenness centrality and the RMS error of the closeness centrality show that GGA algorithm introduces less perturbation in both measures. The details of the $cand_{\mathcal{H}_1}$ results for RP and GGA algorithms sows that RP algorithm increases the well-protected nodes until a value of 52%, while GGA algorithm maintains the data very similar to the initial values.

## 5 Conclusions

In this paper we have reported an experimental study of two anonymization algorithms. One of them is random-based, while the other is based on $k$-anonymity model. We have applied these anonymization algorithms on three real world social networks that have well-documented structures: Zachary's Karate Club network, American College Football teams' network and Jazz Musicians' network.

After seeing the results of the experiments, we can clearly see that $k$-anonymity-based algorithm gets the best results on all data sets. This algorithm, called *GeneticGraphAnonymization* (GGA), achieves a greater degree of anonymity and produces less perturbation on graphs. So, it produces a more useful data and a more protected data. However, GGA algorithm is slower than RP on all data sets.

Many interesting directions for future research have been uncovered by this work. Other graph anonymization methods should be evaluated and compared with our $k$-anonymity-based algorithm. Also, another interesting area is to evaluate different measures for the re-identification risk. There are several measures and it is interesting to compare all of them. Finally, another graph types will be considered, such as weighted [12] or directed graphs.

**Acknowledgments.** This work was partially supported by the Spanish MCYT and the FEDER funds under grants TSI2007-65406-C03 "E-AEGIS", TIN2010-15764 "N-KHRONOUS", CONSOLIDER CSD2007-00004 "ARES", and TIN2011-27076-C03 "CO-PRIVACY".

## References

1. Sweeney, L. (2002). k-anonymity: a model for protecting privacy. of Uncertainty Fuzziness and Knowledge Based, 10(5), 557-570. River Edge, NJ, USA: World Scientific Publishing Co., Inc.
2. Hay, M., Miklau, G., Jensen, D., Weis, P., Srivastava, S. (2007). Anonymizing Social Networks. Science (pp. 1-17).
3. Ying, X., Pan, K., Wu, X., Guo, L. (2009). Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. of the 3rd Workshop on Social Network (p. 10:1–10:10). New York, NY, USA: ACM.
4. Liu, K. (2008). Towards identity anonymization on graphs. Proceedings of the 2008 ACM SIGMOD international, 93-106. New York, NY, USA: ACM.
5. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P. (2008). Resisting structural re-identification in anonymized social networks. Proc. VLDB Endow., 1(1), 102-114. VLDB Endowment.
6. Zou, L., Chen, L., Ozsu, M. T., zsu, M. T. (2009). K-Automorphism: A General Framework For Privacy Preserving Network Publication. VLDB 09: Proceedings of the Thirtieth international conference on Very large data bases (Vol. 2, pp. 946-957). Lyon, France: VLDB Endowment.
7. Zachary, W. (1977). An information flow model for conflict and fission in small groups. Journal of anthropological research, 33, 452-473.
8. Girvan, M., Newman, M. E. J. (2002). Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America, 99(12), 7821-7826. Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA.
9. P.Gleiser and L. Danon (2003). Adv. complex syst.6, 565.
10. Casas-Roma, J., Herrera-Joancomartí, J., Torra, V. (2012). Algoritmos genéticos para la anonimización de grafos. In XII Spanish Meeting on Cryptology and Information Security (RECSI 2012), Donostia-San Sebastián, Spain.
11. Torra, V., Narukawa, Y. (2007) Modeling decisions: information fusion and aggregation operators, Springer.
12. Das, S., Egecioglu, A., Abbadi, A. E. (2010). Anonymizing weighted social network graphs. ICDE (pp. 904-907). IEEE.
13. Zhou, B., Pei, J., Luk, W. (2008). A brief survey on anonymization techniques for privacy preserving publishing of social network data. ACM SIGKDD Explorations Newsletter, 10(2), 12-22. New York, NY, USA: ACM.