GSTF Journal on Computing (JOC)
DOI: 10.5176/2251-3043_4.4.358
ISSN:2251 - 3043 ; Volume 5, Issue 1; 2016 pp. 91-100

The Effects of Ant Colony Optimization on Graph Anonymization

Gautam Srivastava
Brandon University
Dept. of Computer Science 270 18th Street
Brandon, Canada
srivastavag@brandonu.ca

Evan Citulsky
University of Waterloo
Dept. of Computer Science 200 University Avenue W Waterloo, Canada
evancitulsky@gmail.com

Kyle Tilbury
Dalhousie University Dept. of Computer Science 6299 South Street Halifax, Canada
ktilbury21@gmail.com

Ashraf Abdelbar
Brandon University
Dept. of Computer Science
270 18th Street
Brandon, Canada
ashrafa@brandonu.ca

Toshiyuki Amagasa
Tsukuba University Information and Systems
1-1-1 Tennodai
Ibakari, Japan , amagasa@cs.tsukuba.ac.jp

ABSTRACT

The growing need to address privacy concerns when social network data is released for mining purposes has recently led to considerable interest in various techniques for graph anonymization. These techniques and definitions, although robust are sometimes difficult to achieve for large social net-works. In this paper, we look at applying ant colony opti-mization (ACO) to two known versions of social network anonymization, namely k-label sequence anonymity, known to be NP-hard for k ≥ 3. We also apply it to the more recent work of [23] and Label Bag Anonymization. Ants of the artificial colony are able to generate successively shorter tours by using information accumulated in the form of pheromone trails deposited by the edge colonies ant. Computer simu-lations have indicated that ACO are capable of generating good solutions for known harder graph problems.

The contributions of this paper are two fold: we look to apply ACO to k-label sequence anonymity and k=label bag based anonymization, and attempt to show the power of ap-plying ACO techniques to social network privacy attempts. Furthermore, we look to build a new novel foundation of study, that although at its preliminary stages, can lead it ground breaking results down the road.

I. INTRODUCTION

The recent explosion of activity on the internet has given rise to huge amounts of social network data which is usefully viewed as a collection of entities and associations between them. One such example is the PatientsLikeMe social net-work. Here, members get the chance to connect with others dealing with similar health issues. This information could be vital in the study for disease research. However, can we ensure sensitive information, when studied, will still protect the members associated with it?

While significant amounts of useful information may be extracted from this kind of network data, there are many pri-vacy concerns that need to be addressed before the data is released. Particularly, the data may contain sensitive infor-mation about individuals that cannot be disclosed without compromising their privacy. Examples include Facebook, Twitter, LinkedIn, and many other online social networks that have become the social lifeline of many individuals. An-other example is the sensitive patient data of pharmaceuti-cals purchased by a customer of an on-line pharmacy. It has already been shown that naive attempts to hide this sensi-tive information do not work [15, 16]. They showed attacks that could check for the existence of edges between targeted nodes in the anonymized version of the network. These re-sults demonstrate the need for a rigorous approach to graph anonymization.

Liu and Terzi proposed a simple graph anonymization tech-nique in order to prevent identity attacks [24]. They assume that the adversary has prior knowledge of degrees of certain vertices in the network, and may use this basic structural in-formation to try and identify certain targeted nodes in the anonymized network. To fight such attacks, they defined the concept of k-degree anonymity, for a input parameter k. A graph G is said to be k-degree anonymous if it is the case that for every vertex v of G, there are at least k − 1 other vertices in G with the same degree as v. They study the problem of converting a given graph into a k-degree anony-mous graph with the *minimum* number of edge additions.

This work led to a basis of study for graph anonymization that led to a broad net of work. Within this net were two very popular and powerful graph anonymization techniques, namely k-label sequence anonymity and k-label bag based anonymity, where k is the size of the anonymous groups in terms on nodes. Clearly we see that increasing the size of k will in fact increase the level of privacy of a given node. The issue that follows is that the larger the size of k for a given network graph, the harder the procedure to anonymize the graph becomes.

A. Our Problem

In this paper, we study two natural generalizations of k-degree anonymity. Firstly, the graph representing a social network can have labelled edges. As the need to represent social networks as graphs grows, so will the amount of information that needs to be stored in these graphs. The label often gives auxiliary information associated with a relationship. In a purchasing example, an edge may represent the fact that a shopper has bought a certain product. Associated with such a relationship could be data such as dates of purchase, quantities, ratings, etc. In order for our graph model to support this way of associating auxiliary data with relationships, we will considered graphs whose edges are labelled by elements of some label set. For such graph, the degree of a vertex is replaced by its *label sequence* containing all the labels of the edges incident on it.

These considerations lead to the problem of k-label sequence anonymity in which we are given an edge labelled graph and we would like to ensure that a given subset of vertices of G is k-label sequence anonymous by adding a minimum number of edges. We will also study this problem for bipartite graphs, where the vertices to be anonymized are from one side of the bipartition. The bipartite model is useful in cases where vertices represent two types of entities, and edges exist only between entities of different types.

This problem was studied in depth in [6]. We look to build off this work by introducing the procedures of Ant Colony Optimization (ACO) to the main problem defined in [6].

Secondly, we apply similar ACO techniques to the algorithm shown in [23] dealing with Label Bag based anonymization. We try and show that our results using ACO are slightly better than the results given in the original paper, thus showing the power ACO can have on anonymization problems in general, and building a solid framework for further study.

B. Ant Colony Optimization

Ant colony optimization (ACO) is a population-based metaheuristic that can be used to find approximate solutions to difficult optimization problems.

In ACO, a set of software agents called artificial ants search for good solutions to a given optimization problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants (hereafter ants) incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model, that is, a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants.

The easiest way to understand how ant colony optimization works is by means of an example. We consider its application to the travelling salesman problem (TSP). In the TSP a set of locations (e.g. cities) and the distances between them are given. The problem consists of finding a closed tour of minimal length that visits each city once and only once.

To apply ACO to the TSP, we consider the graph defined by associating the set of cities with the set of vertices of the graph. This graph is called construction graph. Since in the TSP it is possible to move from any given city to any other city, the construction graph is fully connected and the number of vertices is equal to the number of cities. We set the lengths of the edges between the vertices to be proportional to the distances between the cities represented by these vertices and we associate pheromone values and heuristic values with the edges of the graph. Pheromone values are modified at runtime and represent the cumulated experience of the ant colony, while heuristic values are problem dependent values that, in the case of the TSP, are set to be the inverse of the lengths of the edges.

The ants construct the solutions as follows. Each ant starts from a randomly selected city (vertex of the construction graph). Then, at each construction step it moves along the edges of the graph. Each ant keeps a memory of its path, and in subsequent steps it chooses among the edges that do not lead to vertices that it has already visited. An ant has constructed a solution once it has visited all the vertices of the graph. At each construction step, an ant probabilistically chooses the edge to follow among those that lead to yet unvisited vertices. The probabilistic rule is biased by pheromone values and heuristic information: the higher the pheromone and the heuristic value associated to an edge, the higher the probability an ant will choose that particular edge. Once all the ants have completed their tour, the pheromone on the edges is updated. Each of the pheromone values is initially decreased by a certain percentage. Each edge then receives an amount of additional pheromone proportional to the quality of the solutions to which it belongs (there is one solution per ant).

This procedure is repeatedly applied until a termination criterion is satisfied.

C. Related Work

In recent years, many interesting definitions for graph anonymization have been proposed and studied. Each of them starts by modelling the background information that an adversary will use to attack the data. Once that is done, a notion of anonymity is defined and studied.

Liu and Terzi proposed a simple graph anonymization technique to prevent identity disclosure attacks [24]. They assume that the adversary has prior knowledge of degrees of certain vertices in the network, and may use this information to try and identify certain nodes in the network. To fight such attacks, they defined the concept of k-degree anonymity. For an input parameter k, a graph G is said to be k-degree anonymous if for every vertex v in G, there are at least $k-1$ other vertices in G with equal degree as v.

Hay *et al.* [16] model the information available to the adversary using two types of queries–vertex refinement queries and subgraph knowledge queries–and study the vulnerability of various datasets under such an attack. They propose an anonymization technique based on random perturbations against such adversaries.

Zheleva and Getoor [37] study the problem of protecting certain sensitive edges in an edge-labeled graph under link re-identification attacks. They propose anonymization techniques using edge-removal and node-merging to prevent such attacks.

Zhou and Pei [14] focus on neighbourhood attacks, which was expanded by Tripathy and Panda [35]. In their model, an adversary uses information about a node's neighbours to target them. To prevent such attacks, they define a notion of k-anonymity on graphs so that nodes in an anonymized group will have isomorphic neighbourhoods. They show that anonymizing a graph under their definition using a minimal number of edge additions is NP-hard and they develop a method that well works in practice.

Thompson and Yao [34] study i-hop degree-based attacks. In their model an adversary's prior knowledge includes the degree of the target and the degree of its neighbours within i hops. They develop a inter-cluster matching method for anonymizing graphs against 1-hop attacks through edge addition and deletion. Thomson and Yao use bipartite graphs, namely the Netflix Prize Data, to help motivate their work.

Wu et al. [36] recently proposed the k-symmetry model. They state for any vertex v in the network, there exists at least $k-1$ structurally equivalent counterparts. The authors also proposed sampling methods to extract approximate versions of the original network from the anonymized network so that statistical properties of the original network could be evaluated. Cormode et al [9] consider a new family of anonymizations, for bipartite graph data, called (k, l)-groupings. These groupings were used to preserve the underlying graph structure perfectly, and instead anonymize the mapping from entities to nodes of the graph. They created "‘safe"’ groupings that were able to withstand a set of known attacks.

More recently, Salas in [30, 29] studied conditions to approximate a given graph by a regular one. Obtaining optimal conditions for a few metrics such as the edge rotation distance for graphs, the rectilinear and the Euclidean distance over degree sequences. Then, requiring the approximation to have at least k copies of each value in the degree sequence, this is a property proceeding from data privacy that is called k-degree anonymity.

Motivated by a strongly growing interest in graph anonymization as recently as 2016, [20] studied the NP-hard Degree Anonymity problem asking whether a graph can be made k-anonymous by adding at most a given number of edges. Herein, a graph is k-anonymous if for every vertex in the graph there are at least $k1$ other vertices of the same degree.

Finally, motivated strongly by our work in [6], we see clearly the strong urge to work in k-anonymization in [5, 25], more specifically in k-degree anonymization. Degree anonymization by vertex addition is computationally intractable in general. Posing structural restrictions on the edges connected to the new vertices seems to make the problem even harder. There are some tractable special cases, for example, when the number of new edges is small, which is where our work leads.

D. Our Results

We introduce a new procedure for finding an anonymous graph in k-anonymity scenarios for labelled graphs. We consider k-anonymization with respect to the collection of labels of incident edges, in two forms. Namely the k-label sequence anonymity proposed in [6] and the k-Label Bag Sequence anonymity proposed in [23]. In §**II** we lay out all the background information needed to comprehend Ant Colony Optimization, k-label sequence anonymity, and k-label bag based anonymity.

In §**III** we deal with implementing ACO techniques on k-label bag based anonymization. Here we see the power that ACO can have on a well laid out procedure.

In §**IV** we consider k-label sequence anonymization. For $k = 3$, we present a polynomial time procedure, based on recent work in [6].

In §**V** we present some interesting future work that and proposed problems that will further the scope of this paper down the road.

II. PRELIMINARIES

In this section, we define the concept of k-anonymity for tables, unlabelled graphs and labelled graphs to help show the progression of the definitions. We also introduce the definitions of Label Bag based Anonymity and give some basic background of Ant Colony Optimization.

A. Tables and k-Anonymity

Table Anonymization has been extensively studied [2, 4, 13, 19, 26]. Suppose we want to publish a table of data containing potentially sensitive information. To help protect the data, we have the ability to suppress the data entries in the table with *'s. To achieve k-anonymization by suppressing the entries, we require that after suppression, for any given row in the table, there are $k-1$ other rows that look identical.

Table 1: Table Data before Anonymization

Fname	LName	Age	Grad Year
Harry	Potter	30	2012
John	Connor	45	2013
Harry	Houdini	30	2010
Sarah	Connor	32	2013

If we want to 2-anonymize the above data, then using the fewest suppressions to acheive 2-anonymity would be:

Table 2: 2-Anonymous Table

Fname	LName	Age	Grad Year
Harry	*	30	*
*	Connor	*	2013
Harry	*	30	*
*	Connor	*	2013

DEFINITION 1. *A table consisting of a multiset V of* rows, *that is sequences of length m over a set Σ of* entry values. *Let $t : V \longrightarrow (\Sigma \bigcup \{*\})^m$. If for all $v \in V$ and $j = 1, \ldots, m$ it is the case that $t(v)_j \in \{v_j, *\}$, we call t a* suppressor. *The table $t(V)$ resulting from a suppressor t is defined to be k-anonymous iff for all $v \in V$ there exist at least $k-1$ distinct rows $v_1, \ldots, v_{k-1}$ such that $t(v) = t(v_1) = \ldots = t(v_{k-1})$. In other words, after applying t, each row is identical to at least $k-1$ other rows.*

B. Anonymizing entries is hard

In [26], the problem of finding the minimum number of suppressions to anonymize a table was proven NP-hard for $k \geq 3$ and $|\Sigma| \geq n$ From this, [2] lowered the alphabet size to $|\Sigma| = 3$. Finally, it was shown in [4] that the problem remains hard for $|\Sigma| = 2$ and $k \geq 3$.

C. Unlabeled Graphs and k-Anonymity

Let $G = (V, E)$ be a simple graph where V, $|V| = n$, denotes the set of vertices and E denoted the set of edges. We denote the degree of a vertex v by $d(v)$.

DEFINITION 2 (DEGREE SEQUENCE). *Let $X = \{x_1, x_2, \ldots, x_n\}$, $X \subseteq V$, be a subset of vertices of G. The* degree sequence *of X is $(d_1, d_2, \ldots, d_n)$ where $d_i = d(x_i)$ is the degree of the vertex x_i.*

DEFINITION 3 (k-ANONYMITY). *A sequence of values $S = (s_1, s_2, \ldots, s_n)$ is said to be k-anonymous if every distinct value in S occurs at at least k times. A subset of vertices X in a graph G is k-anonymous if its degree sequence is k-anonymous.*

Degree-Based subset Anonymization Problem (D-SAP): Given a graph $G = (V, E)$, $X \subseteq V$ and an integer k, find a graph $G' = (V, E \cup E')$ such that X is k-anonymous in G' and the number of new edges added, $|E'|$, is minimized.

Note: We state our anonymization problems in the optimization version of [2, 4, 26], and indeed the algorithms we give are naturally viewed in this way. On the other hand, for hardness we in fact deal with the decision version of these problems. That is, we have another input $t \in \mathbf{N}$, and we ask whether there is a set E' of edges such that G' is k-anonymous and $|E'| \leq t$.

Example 1: Here we present a small example of **D-SAP**. Consider the graph G in **Figure 1.** Suppose we want 2-anonymity for the subset of vertices $\{v_1, v_2, v_5, v_6\}$, which has degree sequence $(2, 4, 2, 2)$. Adding the dotted edges of **Figure 1(b)** will result in the degree sequence $(2, 4, 2, 4)$, which 2-anonymous. Since, for 2-anonymity, we require at least 2 vertices of degree 4 in the sequence, the number of edges added is the minimum.

D. Labeled Graphs and k-Anonymity

Edge-labelled graphs are a natural model for the representation of social networks and related forms of data. The Netflix movie database [34], can be represented with nodes for movies and users and labeled edges to represent how users rank these movies.

DEFINITION 4 (EDGE-LABELLED GRAPH). *An* edge-labeled graph *is a tuple $G = (V, E, \Sigma)$ where V is the set of* vertices*, Σ is the* label set *and $E \subseteq \mathcal{P}_2(V) \times \Sigma$, is the set of* (labelled) edges*. Here $\mathcal{P}_2(V)$ denotes the 2-element subsets of V. E must satisfy the property that there is at most one $\ell \in \Sigma$ such that $(\{u, v\}, \ell) \in E$. If $(\{u, v\}, \ell) \in E$ is a labelled edge, we say that ℓ* is the label of edge *$\{u, v\}$.*

DEFINITION 5 (LABEL SEQUENCE). *For $v \in V$, we say that $S_v = (\ell_1, \ell_2, \ldots, \ell_m)$ is a* label sequence *of v if it corresponds to some ordering of the labels of the edges incident on v. We consider label sequences to be equivalent up to permutations.*[1]

DEFINITION 6 (LABEL SEQUENCE ANONYMITY). *Given an edge-labelled graph $G = (V, E, \Sigma)$, a subset $X \subseteq V$ of vertices is* k-anonymous in *G if for every vertex v in X, there are at least $k-1$ vertices in X whose label sequence is equivalent to the label sequence of v. If v and v' are vertices with equivalent label sequences we say that they are* similar *and write $v \equiv v'$.*

Clearly $\equiv$ is an equivalence relation and so induces a partition $X/\equiv$ of X. We now define the anonymization problem for subsets of labelled graphs.

Label Sequence-Based Subset Anonymization Problem (LS-SAP):
Given an edge-labelled graph $G = (V, E, \Sigma)$, $X \subseteq V$, and an integer k, find an edge-labelled graph $G' = (V, E \cup E', \Sigma \cup \Sigma')$ such that X is k-anonymous in G' and the number edges added, $|E'|$, is minimized.

In other words, we would like to k-anonymize X by adding the minimum number of new labelled edges to G. Note that the added edges may have labels from an expanded set $\Sigma \cup \Sigma'$.
Note: we call E' an *anonymizing set of edges* for X.

Example 2: Here we present a small example of subset label sequence anonymization. Consider graph H in **Figure 1(c)**. Here, if we have $X = \{v_1, v_2, v_5, v_6\}$, with $k = 2$ similar to **Example 1**, adding the dotted edges in **Figure 1(d)** with the given edge labels gives us 2-anonymity. In this case it is not sufficient just to have a 2-anonymous degree sequence; we must also consider the labels of incident edges for each vertex.

E. Label Bag Based Graph Anonymity

Here, we give the formal definition of label-bag (LB) based graph anonymization problem. In this work, we assume that a graph is undirected and simple, i.e., there is no self-loop and no multiple edges between two nodes. A graph G is defined as a quadruple (V, E, L, λ), where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, L is the set of edge labels, and $\lambda : E \rightarrow L$ is the mapping from an edge to a label. Note that each node $v_i \in V$ has its identity (i). Then, the label bag is defined as follows:

DEFINITION 7 (LABEL BAG; LB). *For an edge-labelled graph G, the label bag LB_i of a node v_i in G is the multi-set of edge labels, such that $LB_i = \{\lambda(e) \mid e \in E$ and e has v_i in either of the connected nodes.$\}$*

Let us consider the the example in Figure 2. Figure 2 (a) is the original graph, and Figure 2 (b) is obtained by replacing the node names with identifiers. $LB_1 = \{a, b\}$ and $LB_2 = \{a, b, b\}$. Hereafter, we abbreviate the label bag as the concatenation of labels, such as $LB_1 = ab$ and $LB_2 = abb$.

Next, we define the concept of label-bag based k-anonymity.

DEFINITION 8 (LABEL-BAG BASED (LB) K-ANONYMITY). *Given an edge-labelled graph G and an integer k, G is said to be k-anonymized, if there exist at least k nodes with the same label-bag LB_i for any node $v_i \in V$.*

For example, Figure 2 (b) is 2-anonymized, because v_1 and v_3 (v_2 and v_4) have the same LB ab (abb, resp.). Then, the label-bag based anonymization problem is defined as follows.

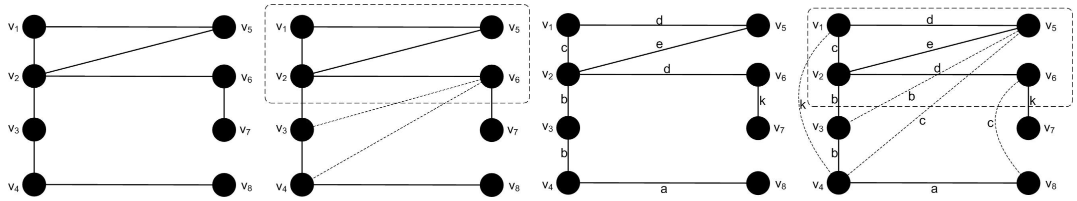
DEFINITION 9 (LB K-ANONYMITY PROBLEM [21]). *Given an edge-labelled graph G and an integer $k(\geq 2)$, the LB k-anonymity problem of G is to construct a graph $G' = (V, E \cup \Delta E, L, \lambda)$, such that G' is LB k-anonymized.*

As can be seen from the definition, in this problem, we only allow edge addition as the graph modification operation. Notice that introduction of new labels is not allowed, either. In [21], Kapron proved that the computational complexity of this problem is NP-hard when $k > 2$.

Let us take a look at Figure 2, Figure 2 (b) can further be anonymized by adding an edge $(1, 3)$ with label b. As a result, the graph is 4-anonymized, because all four nodes have the same LB abb.

From a practical point of view, it is important to make G' as similar to G as possible to minimize information loss

[1] We use permutation-invariant sequences rather than multisets to avoid the need to deal explicitly with multiplicities.



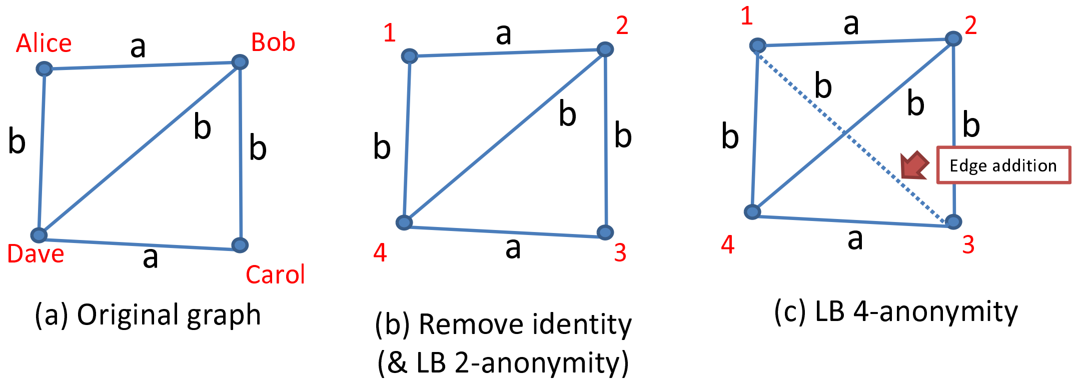
(a) Graph G

(b) 2-anonymized subset Anonymity

(c) Graph H

(d) 2-label sequence Anonymization of X in H

Figure 1: Example 1: D-SAP and Example 2: Subset Label Sequence Anonymization



(a) Original graph

(b) Remove identity (& LB 2-anonymity)

(c) LB 4-anonymity

Figure 2: LB k-anonymity example.

from the original graph. To this end, $|\Delta E|$ should be minimized and the edges to add (ΔE) should be chosen carefully in such a way that the *utility* of the graph is maintained as much as possible. In this work, we exploit several graph utility metrics, and incorporate them in the anonymization algorithm.

F. Ant Colony Optimization

Ant colony optimization (ACO) [7, 11, 12, 33] is a general-purpose, biologically-motivated, population-based, discrete optimization paradigm that can be applied to a wide variety of problems.

ACO is based on a number of primitive processing elements, each operating in parallel with little centralized control. In ACO, the processing elements are called *ants*, and the collection of processing elements is called a *colony*. In each iteration, each ant i generates a candidate solution x_i, and the set of solutions generated by all ants is used to update a central data structure, conventionally called τ, that can be thought of as representing the collective wisdom of the group. In generating its solution in a given iteration, each ant makes use of the τ data structure, as well as making use of a problem-dependent heuristic function η.

A number of different algorithms [1, 3, 8, 10] have been introduced within the ACO paradigm. The abstract framework presented in Figure 1 describes most of these algorithms for a static combinatorial optimization problem such as TSP. The different ACO algorithms that have been studied are generally similar in the **SolutionConstruction** step, but different in the **PheromoneUpdate** step. One of the earliest ACO algorithms was Ant System (AS) [7, 11], and one of the currently best-performing ACO algorithms is $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [32, 31, 33].

In applying ACO to a given problem, the τ data structure would include an entry for every potential solution component for that problem. For the graph anonymization problem described earlier, the set of solution components would include every potential labelled edge that can be added to the graph. The greedy heuristic described earlier consists of two stages: in the first stage, the graph is partitioned into subgraphs; in the second graph, edges are between different subgraphs. We would apply ACO in the second stage of the algorithm. Thus, the solution components would consist of every potential edge between two nodes in two different subgraphs.

Consider two such nodes i and j. The pheromone amount τ_{ij} varies over time and represents the extent to which the collective wisdom of the colony is inclined to add the edge from i to j. The heuristic information η_{ij} represents a static problem-dependent heuristic function that represents the "goodness" of adding an edge from i to j. In the present work, the η function would be based on the greedy heuristic described earlier.

To construct a candidate solution, each ant starts with an empty solution and adds solution components, one by one. In adding each solution component, an ant selects among the still-available feasible solution components, and chooses among them according to the roulette-wheel equation:

$$\Pr(\text{select }(i,j)) = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{(a,b)\in\mathcal{D}}[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta} \qquad (1)$$

Table 3: Experimental dataset.

Category	Name
Synthetic	Small World Graph [27]
Real 1	Speed Dating Data [17]
Real 2	arXiv E-print Archive [28]
Real 3	Enron Email Data [22]

where $\mathcal{D}$ represents the set of available feasible edges, and α and β are external parameters used to adjust the relative emphases of the two terms.

Once all ants construct candidate solutions, the pheromone structure is updated in some way based on the constructed solutions. Different ACO algorithms differ in the specifics of the pheromone update stage. In the approach we follow in this paper, the ant with the best solution constructed in the current iteration, called the iteration-best ant, deposits pheromone on the solution components that make up its constructed solution. Suppose the iteration-best ant is ant k. Then, all edges (i, j) included within ant k's solution would have their pheromone τ_{ij} values increased by an amount that is proportional to the quality of the constructed candidate solution.

III. EXPERIMENTAL RESULTS FOR LABEL BAG BASED ANONYMIZATION

A. Experimental environment

We conducted a series of experiments to evaluate the efficiency and effectiveness and viability of the ACO framework on anonymization problems. The experimental environment is a PC (2-Intel Xeon L5520 2.27 GHz CPU Quad Core, 24 GB memory), and the program is written in Java compiled by Java JDK 1.8.

B. Experimental dataset

The experimental datasets are shown in Table 3. We try and keep our experiments for label bag based anonymization as close to those in [23] to show easy comparison. We use synthetic data for testing the proposed algorithm in controlled situations, while it is also tested using several real datasets.

C. ACO Parameters

For these experiments, we used the parameters for ACO shown in Table 4. We vary k values identically to the original paper. The ACO parameters were chosen based on results garnered and comparison to the findings in the original paper. These parameters gave us the best results. To note here is that some chosen values for α and β led to failure of the procedure.

```
Initialization
while (termination criteria not reached) do
    SolutionConstruction
    LocalSearch            // optional
    PheromoneUpdate
```

Figure 3: Algorithmic framework for ACO algorithms.

Table 4: Ant Colony Optimization Parameters

Parameter	Value
Number of Ants	10
α	1
β	0
ρ	0.05
Number of Iterations	300
k	varying

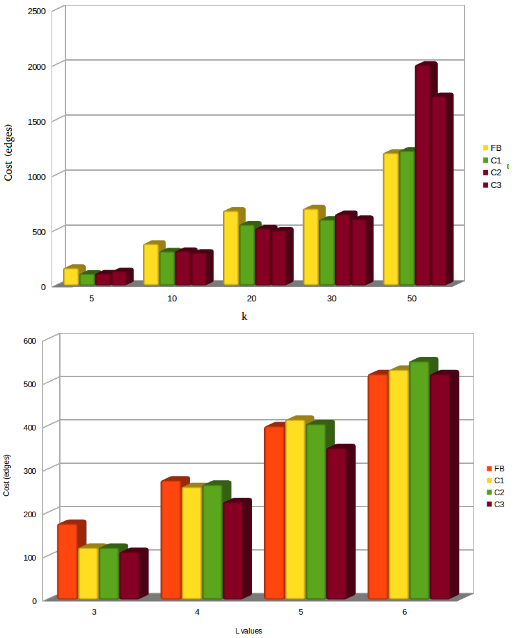


Figure 4: Experimental results: Synthetic varying k (top) and varying L (right).

D. Experimental results

Due to the fact that social networks are well-modelled by the small world graph, we use a Small World Graph with 550 nodes as the synthetic data. Figure 4 shows the result when varying k (top) and varying the number of labels L (bottom). The vertical axis (cost) is the number of edges added to anonymize the graph, which is typical metric for this type of procedure. We compare the feature-based grouping (FB) and the clustering-based grouping using different distance metrics (C1 to C3) similar to the original paper.

We can observe that both k and L have positive effects to the total cost. In Figure 4 (top) the results compare closely with the original algorithm, however we can clearly see better performance using ACO, compared in Table 6.

Table 5 shows the cost for anonymization using Real 1, which contains 550 nodes, $\approx$ 8,400 edges, and 2 kinds of labels. The max degree of this graph is 22 and the average degree is approximately 15. The results are similar to the original paper in the sense that the clustering-based algorithm basically outperforms the feature-based grouping when k is small, whereas the feature-based algorithm performs best when k is large. However, there are clear improvements to the cost, namely number of edges added for most cases. We also show a direct comparison of anony-

Table 5: Experimental result: Real 1.

k	Feature	Clustering-1	Clustering-2	Clustering-3
5	160	108	110	131
10	380	312	315	300
20	680	555	520	500
30	700	600	650	606
50	1205	1225	2004	1720

Table 6: Comparison result for Real 1.

k	Feature	Clustering-1	Clustering-2	Clustering-3
5	160(169)	108(112)	110(117)	131(131)
10	380(398)	312(309)	315(312)	300(291)
20	680(664)	555(567)	520(526)	500(540)
30	700(NA)	600(NA)	650(NA)	606(NA)
50	1205(1259)	1225(1325)	2004(2239)	1720(1820)

mization cost from the original paper and our methods in Table 6. We can clearly see in most instances our procedure including ACO did better..

Figure 5 (top) compares three real datasets (Real 1, 2, and 3) with different k using feature-based grouping. More precisely, Real 2 consists of ≈16,800 nodes and ≈ 48,000 edges with 3 kinds of labels. Average and max degree are 5.00 and 110, respectively. Real 3 contains ≈ 37,000 nodes and ≈ 368,000 edges. Notice that the labels are randomly generated. We can observe that the total cost is quite different depending on the dataset, because the graph size is different.

Figure 5 (bottom) shows the time breakdown for each algorithm applied to Real 1. We see clearly here that the ACO procedure is a time hog compared to the other components. However, we still see good results in a relatively small amount of time.

Table 7 compares the results of utility-based methods using Real 1 dataset. The row labelled "Original" shows the utilities for the k-anonymized data generated by the original baseline method where utility is not taken into account. The rows below show the respective utility values computed from the anonymized graphs considering the corresponding utility metrics. The result shows that, by the proposed method using ACO, the utility metrics changed (3.28 to 2.777 for ASPD and 0.45 to 0.19 for ACC), which are more close to the values in the non-anonymized graph. Unfortunately, EMD-based (Earth Movers Distance) methods (EMDD and EMDL) did not work well, and the values did not change much, which is as expected. This is due to the fact that the number of labels were small. We plan to evaluate these metrics using larger datasets in the future. For ASPD (average shortest-path distance) and ACC (average clustering coefficient), the results are very good.

IV. EXPERIMENTAL RESULTS FOR K-LABEL SEQUENCE ANONYMIZATION

Table 7: Experimental results: utility-based methods (Real 1).

		Utility of anonymized graph			
Utility	**Cost**	**EMDD**	**EMDL**	**ACC**	**ASPD**
Original	160	0.0198	0.0074	0.0455	3.2840
EMDD	160	0.0198	0.0074	0.0431	3.2954
EMDL	160	0.0198	0.0074	0.0431	3.2954
ACC	160	0.0198	0.0074	0.0051	2.5269
ASPD	160	0.0198	0.0074	0.0192	2.7787

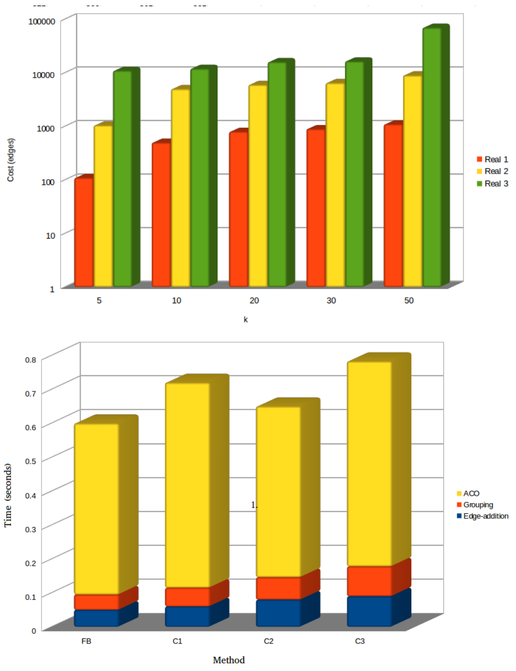


Figure 5: Experimental results: cost for different datasets (top) and time breakdown (bottom).

Table 8: Experimental environment.

Component	Description
CPU	2 - Intel Xeon L5520 2.27 GHz (4 cores)
Memory	24 GB
Language	Java
Compiler	Java JDK 1.8

A. Experimental environment

We keep the same environment for the work on k-label sequence anonymity. Knowing ahead of time that the problem is proven to be NP-hard in [6], we were hopeful of getting some interesting results in the positive or direction.The experimental environment was

B. Algorithm Description

The Code for this experiment is split into two different parts, the graph generator (small world graphs), and the main anonymization algorithm. First the graph generator generates a graph by the user defining how large of graph (number of nodes), how many label types, and how many edges to initially add to the graph. This is outputted to a file and there is no way of knowing if the graph is realistic or not. Next the graph is read into the main anonymization algorithm.

The main anonymization algorithm is based on an ACO System. It first starts by generating all possible edges that can be added to the graph. During each iteration a set number of ants are sent out to construct a solution to our problem, in this case anonymization of a graph. Each ant adds one by one a possible viable edge to the graph, checking each time whether the graph is anonymized. Once all the ants find a solution, pheromones are placed one the edges in which the ant had travelled. The update is done in the following manner:

Table 9: Ant Colony Optimization Parameters

Parameter	Value
Number of Ants	8
α	1
β	0
ρ	0.05
Number of Iterations	500
k	3

$$\tau(r,s) \qquad (1-\alpha)\cdot\tau(r,s)+\sum_{k=1}^{m}\Delta\tau_k(r,s) \tag{2}$$

where,

$$\Delta\tau_k(r,s)=\begin{cases}1/L_k & ,if\ (r,s)\in tour\ done\ by\ ant\ k\\ 0 & otherwise\end{cases}$$

where L_k is the number of edges added by the ant, $\tau(r,s)$ is the pheromone on an edge, and α is the decay rate of the pheromones (set by the user)[18].

The more ants that select a given edge in the graph each time increases its probability of being selected again next time. The probability that an edge is selected by an ant is

$$p_k(r,s)=\frac{[\tau(r,s)]\cdot[\eta(r,s)]^{\beta}}{\sum_{u\in J_k(r)}[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}} \tag{3}$$

where $J_k(r)$ is the set of currently viable edges that can be added to the graph, and β is a parameter set by the user that determines the importance of the pheromone. After an edge is selected and added, all other edges in $J_k(r)$ that are parallel to that edge are removed from $J_k(r)$.

The idea is that the ants will create *hot spots* in pheromones for certain edges which in turn will make them likelier to be picked. The hot spot edges that are created are then hopefully part of a somewhat optimal solution.

Future study will need to include optimization. Currently every time an edge is added it takes $O(n)$ time to check for anonymity. This will be reduced to $O(1)$ which will dramatically increase the efficiency of the algorithm.

C. ACO Parameters

For these experiments, we used the parameters for ACO shown in Table 9. We fix k to 3 matching the lower bound to the hardness result in [6] for k-label sequence anonymization. The ACO parameters were chosen to give optimal results for graph anonymity. To be noted here is that certain values for α and β led to unattainable results. It was through trial and error that these values for the ACO portion of the procedure were found to be optimal.

D. Experimental dataset

The experimental datasets are shown in Table 10. We use synthetic data for testing using a graph generator to use small world problems here. We found with our current implementation of ACO on k-label sequence anonymity, that the procedure would take large amounts of time, too long for proper experimentation. To note here is the procedure never failed to anonymize a graph for large sizes, it would however take long periods of time to do so and lead to close to complete graphs in doing so. However, for small graphs as in Table 10, the procedure performed well.

Table 10: Experimental Synthetic dataset.

Nodes	Edges	Edges Added
10	60	6
15	80	16
20	95	22
25	113	26
30	130	31
35	147	40
40	168	41
45	182	60
50	200	50
55	217	52
60	234	55

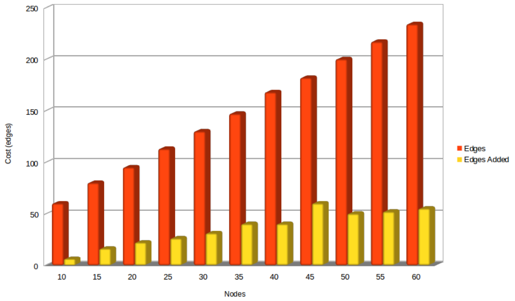


Figure 6: Experimental results: Synthetic graphs varying Nodes

E. Experimental results

Using small world graphs, we generate Small World Graphs varying the number of nodes between 10 and 60 as the synthetic data. Figure 6 shows the result. The vertical axis (cost) is the number of edges added to anonymize the graph, a typical metric for this type of procedure.We keep k constant at 3 for all experiments to adhere to the lower bound for the Hardness result shown in [6].

We can observe that for small graphs, ACO performs admirably. It is promising to see that even for small graphs, a known NP-Hard problem given the ability to traverse the graph with ACO, we can have good results. The maximum $\%$ of nodes added for any graph is less than 25%.

V. FUTURE WORK

The future work is bountiful, considering the exploration of ACO techniques on known graph anonymization is novel and new. We initially wish to pursue a systematic study of all major graph anonymization techniques and try to determine which instances provide the best chance of successful improvements to known procedures. Gaining the knowledge and insight from the work in this paper will help pave the way for this type of endeavour.

We also wish to see if we can improve the current results in k-label sequence anonymity. Although in [6], a problem was defined for k-label sequence anonymity, there was no clear guidance as to the means by which such a graph should be made anonymous. It was by the power of ACO alone, that solutions were able to be found quite well for small graphs.

However, the code and implementation on k-label sequence anonymity can still be optimized and improved. Currently every time an edge is added it takes $O(n)$ time to check for anonymity. This will be reduced to $O(1)$ which will dramatically increase the efficiency of the algorithm. Giving some insight to the ants prior to their departure often helps them returns with a solution. This can clearly be seen for label bag based anonymization, where a clear algorithm and method of anonymization was presented in [23], and using these techniques was clearly shown to provide slightly better results when ACO is added in.

There may also be area of study initiated in creating graph anonymization techniques specifically tailored to the strengths of ACO. This area has not been explored to date and may result in efficient algorithms for improving or solving graph anonymization techniques that are known hard problems.

VI. CONCLUSION

We study the problem of anonymizing graphs and looked at the application of Ant Colony Optimization to known procedures of graph anonymization. We make the following contributions to the field. For k-label bag based anonymity, based on the procedures laid out in [23], we mirror their experimental work and show improved cost factors in terms of edges added. We do however see that the added ACO component adds an expense to the Time cost of the procedure, however we find this addition to be negligible towards the overall gain in optimal results. We also take an initial approach using ACO to try and conquer the known NP-hard problem of k-label sequence anonymity. Without much guidance in terms of procedural steps for this task, our initial approach has been shown to be effective for small world graphs under size $V = 60$ nodes. There is still much room for improvement to this aspect of our paper to make the procedure viable for large graphs.

The next natural step of our research is to see if we can further optimize our work in §**IV**. Looking into the the way the graphs are actually anonymized using ACO may see an overall space savings of the procedure which seems to be the main hurdle at this point. Furthermore, it would also be vital to look into which other graph anonymization procedures can fit nicely into the framework of ACO.

It is important to note that research is ongoing to find applicable relaxed versions of these theoretical based problems that adhere to certain applications. Taking such theoretical problems as the base may lead to efficient solutions to anonymity constraints on such large scale networks. There may be other approaches to anonymizing graphs that may prove just as effective and with better efficiency. It has been shown that certain types of anonymization are not effective (naive anonymization for example), but there may still be effective approaches that have not been looked at. The whole field of graph anonymization is fairly new, there are many avenues within it still to be developed. Newer notions of l-diversity and t-closeness may have some reasonable solution spaces that could be shared with relaxed versions of the anonymity constraints.

The delicate balance between user privacy and requirements for analysis is something that needs to be considered when data of a private nature is released to third parties. This is what motivates our results in this paper. With a strong grasp now on the underlying complexity of how ACO can be applied to graph anonymization, we can now move forward and examine efficient solutions to the many anonymity problems at hand. This inaugural work in ACO techniques applied to graph anonymization will pave the way for future research and development in this rapidly expanding and crucial field.

VII. REFERENCES

[1] A. M. Abdelbar. Is there a computational advantage to representing evaporation rate in ant colony optimization as a gaussian random variable? In *Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012*, pages 1–8, 2012.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT 2005*, pages 246–258, 2005.

[3] R. H. B. Bullnheimer and C. Strauss. An improved ant system algorithm for the vehicle routing problem. In *Annals of Operations Research*, pages 319–328, 1999.

[4] P. Bonizzoni, G. D. Vedova, and R. Dondi. The - anonymity problem is hard. In *FCT 2009*, pages 26–37, 2009.

[5] R. Bredereck, V. Froese, S. Hartung, A. Nichterlein, R. Niedermeier, and N. Talmon. The complexity of degree anonymization by vertex addition. *Theoretical Computer Science*, 607, Part 1:16 – 34, 2015. Algorithmic Aspects in Information and Management.

[6] S. Chester, B. M. Kapron, G. Srivastava, and S. Venkatesh. Complexity of social network anonymization. *Social Netw. Analys. Mining*, 3(2):151–166, 2013.

[7] A. Colorni, M. Dorigo, and V. Maniezzo. An investigation of some properties of an ant algorithm". In *Parallel Problem Solving from Nature 2, PPSN-II, Brussels, Belgium, September 28-30, 1992*, pages 515–526, 1992.

[8] O. Cordón, I. F. de Viana, and F. Herrera. Analysis of the best-worst ant system and its variants on the QAP. In *Ant Algorithms, Third International Workshop, ANTS 2002, Brussels, Belgium, September 12-14, 2002, Proceedings*, pages 228–234, 2002.

[9] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. *VLDB J.*, 19(1):115–139, 2010.

[10] M. Dorigo and L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evolutionary Computation*, 1(1):53–66, 1997.

[11] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 26(1):29–41, 1996.

[12] M. Dorigo and T. Stützle. *Ant colony optimization*. MIT Press, 2004.

[13] B. C. M. F. et al. Anonymity for continuous data publishing. In *EDBT 2008*, pages 264–275, 2008.

[14] B. Z. et al. Preserving privacy in social networks against neighborhood attacks. In *ICDE 2008*, pages 506–515, 2008.

[15] L. B. et al. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW 2007*, pages 181–190, 2007.

[16] M. H. et al. Resisting structural re-identification in anonymized social networks. *PVLDB*, 1(1):102–114, 2008.

[17] FlowingData. http://flowingdata.com.

[18] J. Fourmis. http://www-igm.univ-mlv.fr/~lombardy/ens/JavaTTT0708/fourmis.pdf.

[19] A. Gionis and T. Tassa. k-anonymization with minimal loss of information. In *ESA 2007*, pages 439–450, 2007.

[20] S. Hartung, A. Nichterlein, R. Niedermeier, and O. SuchÃ¡. A refined complexity analysis of degree anonymization in graphs. *Information and Computation*, 243:249 – 262, 2015. 40th International Colloquium on Automata, Languages and Programming (ICALP 2013).

[21] B. M. Kapron, G. Srivastava, and S. Venkatesh. Social network anonymization via edge addition. In *Proc. ASONAM 2011*, pages 155–162, 2011.

[22] J. Leskovec. Enron email network. http://snap.stanford.edu/data/email-Enron.html.

[23] C. Li, T. Amagasa, H. Kitagawa, and G. Srivastava. Label-bag based graph anonymization via edge addition. In *International C* Conference on Computer Science & Software Engineering, C3S2E '14, Montreal, QC, Canada - August 03 - 05, 2014*, pages 1:1–1:9, 2014.

[24] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD 2008*, pages 93–106, 2008.

[25] T. Ma, Y. Zhang, J. Cao, J. Shen, M. Tang, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan. $$*kdvem*$$kdvem: A $$k$$k-degree anonymity with vertex and edge modification algorithm. *Computing*, 97(12):1165–1184, Dec. 2015.

[26] A. Meyerson and R. Williams. General $k$-anonymization is hard. In *Principles of Database Systems*, 2004.

[27] NetworkX. http://networkx.lanl.gov/index.html.

[28] M. E. J. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Phys. Rev. E*, 64:016132, Jun 2001.

[29] J. Salas and V. Torra. Graphic sequences, distances and -degree anonymity. *Discrete Applied Mathematics*, 188:25 – 31, 2015.

[30] J. Salas and V. Torra. Improving the characterization of p-stability for applications in network privacy. *Discrete Applied Mathematics*, pages –, 2016.

[31] T. Stützle. Lokale suchverfahren für constrain satisfaction probleme: die *min conflicts* heuristik und tabu search. *KI*, 11(1):14–20, 1997.

[32] T. Stützle. *Local search algorithms for combinatorial problems - analysis, improvements, and new applications*, volume 220 of *DISKI*. Infix, 1999.

[33] T. Stützle and H. H. Hoos. MAX-MIN ant system. *Future Generation Comp. Syst.*, 16(8):889–914, 2000.

[34] B. Thompson and D. Yao. The union-split algorithm and cluster-based anonymization of social networks. In *ASIACCS 2009*, pages 218–227, 2009.

[35] B. K. Tripathy and G. K. Panda. A new approach to manage security against neighborhood attacks in social networks. In *ASONAM*, pages 264–269, 2010.

[36] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang. k- symmetry model for identity anonymization in social networks. In *EDBT 2010*, pages 111–122, 2010.

[37] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *PinKDD 2007*, pages 153–171, 2007.