Network Anonymization for Science

A Simulated Annealing Approach

Elena-Denisa Arsene[1]

Supervisor & Responsible Professor: Dr. Anna L. D. Latour[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Elena-Denisa Arsene
Final project course: CSE3000 Research Project
Thesis committee: Dr. Anna L. D. Latour, Dr. Carolin Brandt

Abstract

An increasing volume of data is being collected for research purposes, often containing sensitive information. Leaving out unique identifiers is insufficient to ensure anonymity. One approach to mitigating this risk is to modify the graph structure by adding or deleting edges. Existing heuristic approaches offer speed but limited optimality, while exact methods can achieve better anonymization quality but are computationally expensive. This creates a need for strategies that balance solution quality and efficiency. We propose a method based on simulated annealing that removes edges from the original network structure to achieve anonymization under two formal privacy models: (n, m)-flavoured k-anonymity, which considers a node's degree and number of incident triangles, and d-k-anonymity, which is based on the structure up to depth d of a node. Our method aims to balance running time and solution quality while adhering to an edge deletion budget. Experimental results on several real-world social network datasets show that Simulated Annealing can outperform traditional methods in terms of anonymity quality, particularly when higher modification budgets are allowed. The running time is comparable to that of the baseline methods for smaller and medium-sized graphs, but higher for larger graphs.

1 Introduction

Maintaining user privacy has become more difficult as digital social networks keep growing in size. A particularly pressing concern is *structural privacy* [Romanini *et al.*, 2021], where individuals may be re-identified based on structural features such as the number of connections of a node in a graph [Zhou and Pei, 2008], or the local structure of a node [Cheng *et al.*, 2010]. An attacker with partial knowledge of the network graph could re-identify individuals by matching structural patterns, even after removing unique identifiers.

In order to solve this, *network anonymization* techniques alter the graph, usually by adding or removing edges, making individuals indistinguishable from one another according to formal anonymity models. A central principle in these techniques is *k-anonymity* [Wu *et al.*, 2010], which requires that each node is indistinguishable from at least $k-1$ others under some structural measure. Depending on the assumed attacker scenario, different types of measures are used: *$(n, m) - k$-anonymity* assumes an attacker knows the degree and number of incident triangles of a node, while *d-k-anonymity* assumes access to the full structure of a node's d-hop neighborhood [Xie, 2023]. These models capture different levels of attackers' knowledge.

There are mainly two kinds of anonymization techniques currently in use. Although heuristic-based techniques, like those developed by Xie [Xie, 2023], offer quick and scalable anonymization, they cannot guarantee optimality or consistency in maintaining graph properties [Campan *et al.*, 2015]. On the other hand, exact techniques that use Mixed Integer Programming (MIP) may give optimal results, but they are often too slow for networks in the real world. De Jong et al. [de Jong *et al.*, 2023] take a different approach, by providing evaluation tools instead of anonymization techniques by creating algorithms to compute anonymity efficiently without changing the network. This creates a gap: a method that finds a balance between high anonymization quality, computational efficiency, and minimal graph distortion.

In this work, we demonstrate how *Simulated Annealing* (SA) [Kirkpatrick *et al.*, 1983], a probabilistic metaheuristic inspired by thermodynamics, can fill that gap. SA has been successfully applied to other NP-hard graph problems, including *k-anonymity* [Winkler, 2002], and is known for its ability to escape local optima and find high-quality solutions. This makes it a promising candidate for network anonymization, under constraints on graph alteration. Moreover, Simulated Annealing provides flexibility, and could accommodate variations of the anonymization techniques by adjusting the features or weights of the features included in the cost function.

Specifically, we apply SA to remove edges from the graph, aiming to improve anonymity while minimizing information loss. These edge deletions are constrained by a fixed budget and guided by changes in the anonymity score at each step.

The rest of this paper is organized as follows. Section 2 defines the most important terms and notations used in this paper. Section 3 presents the approach: the implementation of the Simulated Annealing method and the baseline anonymity methods. The experimental setup, research questions, evaluation metrics, and results are described in Section 4. The paper is concluded in Section 5, which also provides guidance for future research.

2 Preliminaries

This section introduces the main concepts and notation used in the paper, which form the basis for understanding the anonymization process. It then presents a toy example to illustrate the ideas in context.

2.1 Definitions and Notation

This subsection introduces the terminology and notation used throughout the paper.

Let $G = (V, E)$ be an undirected, unweighted graph, where V denotes the set of nodes (individuals) and $E \subseteq V \times V$ denotes the set of edges (relationships). For a node $v \in V$, we let $\deg(v)$ denote its degree. The **distance** between two nodes $u, v \in V$, denoted $\mathrm{dist}(u, v)$, is the length of the shortest path between them in G. We define $T(v)$ as the number of incident triangles of node v , i.e., the number of distinct 3-cycles that include v.

Definition 2.1 ($d-$neighborhood)**.** *Let $N_d(v)$ denote the set of nodes within distance d from v, including v itself. This is called the $d-$neighborhood of node v.*

Definition 2.2 (Transitivity and Average Clustering Coefficient (ACC) [Xie, 2023])**.** *Transitivity and ACC measure the tendency of nodes to form triangles in a graph.*

*The **transitivity** of a graph G is defined as:*

$$\mathrm{Trans}(G) = \frac{3 \cdot |\{\{u,v,w\} \mid \{u,v\},\{v,w\},\{u,w\} \in E\}|}{|\{\{u,v,w\} \subseteq V \mid \{u,v\},\{v,w\} \in E\}|} \tag{1}$$

It measures the fraction of potential triangles that actually exist.

*The **average clustering coefficient** is defined as:*

$$\mathrm{ACC}(G) = \frac{1}{|V|} \sum_{v \in V} \mathrm{CC}(v) \tag{2}$$

where $\mathrm{CC}(v)$ is the local clustering coefficient of node v, given by:

$$\mathrm{CC}(v) = \frac{2 \cdot |\{\{u,w\} \in E \mid \{u,v\},\{v,w\} \in E\}|}{\deg(v) \cdot (\deg(v)-1)}$$

for $\deg(v) > 1$, and $\mathrm{CC}(v) = 0$ otherwise.

Definition 2.3 (Structural Signature)**.** *To assess anonymity, we assign each node a **signature** $\sigma_M(v)$ describing its local structural features under model M. Two nodes $u, v \in V$ are considered **equivalent** if $\sigma_M(u) = \sigma_M(v)$; we denote this $u \sim_M v$. The equivalence class [de Jong* et al.*, 2024] of a node v is denoted*

$$\mathrm{EC}_M(v) = \{u \in V \mid \sigma_M(u) = \sigma_M(v)\} \tag{3}$$

Each structural signature $\sigma_M(v)$ corresponds to a particular *attack model*, where $M \in \{(n,m), d\text{-}k\}$. The choice of model reflects assumptions about the adversary's background knowledge.

We consider two types of signatures:

Definition 2.4 ((n,m)-Signature)**.**

$$\sigma_{(n,m)}(v) = (\deg(v), T(v)), \tag{4}$$

where $T(v)$ is the triangle count of node v. Two nodes u, v are equivalent under this model if $\sigma_{(n,m)}(u) = \sigma_{(n,m)}(v)$, denoted $u \simeq_{(n,m)} v$.

Definition 2.5 (d-k Signature)**.**

$$\sigma_{d\text{-}k}(v) = [G[N_d(v)]], \tag{5}$$

where $[G[N_d(v)]]$ denotes the isomorphism class of the d-neighborhood subgraph. Two nodes u, v are equivalent, denoted $u \simeq_{d\text{-}k} v$, if there exists an isomorphism $\phi : G[N_d(u)] \to G[N_d(v)]$ with $\phi(u) = v$.

Definition 2.6 (Uniqueness and k-Anonymity [de Jong *et al.*, 2024])**.** *A node $v \in V$ is **unique** if $|\mathrm{EC}_M(v)| = 1$. Conversely, it is k-anonymous if $|\mathrm{EC}_M(v)| \geq k$. The **graph uniqueness** score is defined as:*

$$\mathrm{U}_M(G) = \frac{|\{v \in V \mid |\mathrm{EC}_M(v)| < k\}|}{|V|}. \tag{6}$$

We denote the original edge set by E_0, and use $G_t = (V, E_t)$ to represent the graph at iteration t of the anonymization process. The best graph encountered so far is $G^* = (V, E^*)$, and the final anonymized graph is denoted G'.

2.2 Anonymity Method and Structural Measures

We adopt the *k-anonymity* framework for structural anonymization in networks. A node $v \in V$ is considered k-anonymous if it belongs to an equivalence class $\mathrm{EC}_M(v)$ of size at least k, i.e., it is structurally indistinguishable from at least $k-1$ other nodes under model M.

Definition 2.7 ((n,m)-flavoured k-anonymity [Latour, 2024])**.** *A graph satisfies (n,m)-flavoured k-anonymity if all nodes have at least $k-1$ others with the same degree and number of incident triangles. That is, for each $v \in V$, there exist at least $k-1$ nodes $u \in V \setminus \{v\}$ such that $u \simeq_{(n,m)} v$.*

In this paper, we use $k = 2$ as default, and for ease, we refer to (n,m)-k-anonymity as (n,m)-anonymity.

Definition 2.8 (d-k-anonymity [de Jong *et al.*, 2023])**.** *A graph satisfies d-k-anonymity if every node has at least $k-1$ others with structurally isomorphic d-neighborhoods. For all $v \in V$, there exist $u_1, \ldots, u_{k-1} \in V \setminus \{v\}$ such that $u_i \simeq_{d\text{-}k} v$. Here, d represents the adversary's knowledge. In this paper, we assume $d = 1$.*

Definition 2.9 (Edge Deletion Budget)**.** *To achieve k-anonymity, we modify the graph by deleting edges. The number of deletions is constrained by a predefined **edge deletion budget**, expressed as a percentage of total edges in $G = (V, E_0)$.*

Definition 2.10 (Anonymization Problem Setting)**.** *An anonymization problem instance is defined by a tuple $\mathcal{I} = (G_0, m, B)$, where:*

- *$G_0 = (V, E_0)$ is the original graph extracted from a dataset.*
- *M is a structural measure, where $M \in \{(n,m), d\text{-}k\}$.*
- *Let $B \in \mathbb{N}$ be the edge deletion budget, with $B \in [0, 100]$.*

The goal is to construct an anonymized graph $G' = (V, E')$, such that:

$$|E_0 \setminus E'| \leq B \cdot |E_0|/100$$

and the number of non-k-anonymous nodes is minimized:

$$\min_{E' \subseteq E_0} U_M(G') \quad \textit{subject to} \quad |E_0 \setminus E'| \leq B \cdot |E_0|/100.$$

2.3 Toy Example

Consider a small social network with 4 nodes (Figure 1). Node A has degree 1 and participates in 0 triangles. Even after pseudonymization, such a signature makes A easily re-identifiable. An anonymization process might reduce $\deg(D)$ or triangle count by deleting edges, making A less unique.

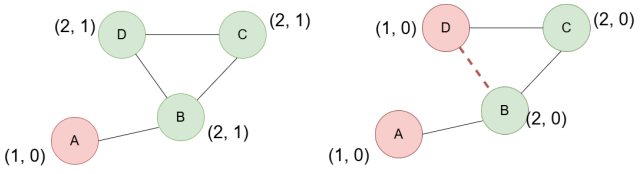

Figure 1: A toy graph before and after anonymization under (n,m)-k-anonymity. Node A becomes less unique after deleting one edge, reducing the degree and number of incident triangles of node D.

3 Approach

In this section, we describe our approach of anonymizing the graph in detail, which includes the simulated annealing method. Moreover, we will briefly explain the 3 baseline methods that are used to compare the SA algorithm against.

3.1 Baseline methods

In this subsection, we describe three of the five methods presented in the work of Xie [Xie, 2023], which we use as baselines for comparison with our Simulated Annealing approach. We focus on the three methods that demonstrated the best performance in terms of solution quality in the original study. The remaining two methods—Degree Deletion and Random Edge Deletion—did not yield comparably strong results in prior work, so we omit them from our evaluation.

3.1.1 UA-Based Deletion

The UA-based method assigns weights to edges based on the number of affected nodes that would become anonymous after their removal. It intuitively prioritizes edges whose deletion would affect as few already anonymous nodes as possible while also reducing the number of unique nodes (i.e., nodes with $|\mathrm{EC}_M(v)| = 1$) while avoiding affecting already anonymous nodes ($|\mathrm{EC}_M(v)| \geq k$). For an edge $\{u, v\}$, the set of affected nodes includes the endpoints u, v, and any nodes that share triangles with the edge.

3.1.2 Logistic Regression-Based Deletion

Edge prioritization for network anonymization can be formulated as a binary classification problem to decide whether removing an edge will decrease node uniqueness. Features like node degrees, the quantity of incident triangles, and uniqueness labels are used to train a logistic regression model on preprocessed graph data. After training, each edge is given a probability, indicating how advantageous it would be to remove it in order to anonymize the network. Edges with the highest predicted scores are deleted as long as there is available budget left, with the aim of obtaining the smallest final uniqueness.

3.1.3 (n, m)-Greedy Deletion

Deleting an edge $\{u, w\}$ affects the signatures of:

- nodes u and w (as their degree changes),
- and nodes forming triangles with $\{u, w\}$ (as their local triangle count changes).

The algorithm evaluates each edge by computing how many unique nodes it would make anonymous (or vice versa), without actually deleting it. The edge with the greatest positive effect is then deleted, and the process repeats until the budget is exhausted.

3.2 Simulated Annealing for Graph Anonymization

Simulated Annealing (SA) is a probabilistic optimization method inspired by the physical process of annealing in metallurgy, where controlled cooling reduces structural defects [Kirkpatrick *et al.*, 1983]. The SA algorithm explores the solution space by allowing probabilistic transitions to worse states at high "temperatures", enabling escape from local optima. As the temperature decreases, the algorithm gradually favors moves that improve the objective, balancing **exploration** and **exploitation**. We apply SA to the problem of minimizing network uniqueness under an edge-deletion budget and a chosen anonymity measure. We are using an iterative improvement method inspired by the Metropolis algorithm.

3.3 Applicability to Uniqueness Reduction

In the (n, m)-anonymity measure, each node is assigned a signature based on its neighborhood size (n) and number of incident triangles (m). Removing an edge (u, v) may change the equivalence class of nodes u, v, and any node involved in a triangle containing that edge, and make them unique.

We conducted preliminary experiments to study the impact of deleting one edge on uniqueness. Our aim was to check whether removing a single edge can anonymize a large number of nodes. The results showed that removing a single edge rarely produces a large reduction in uniqueness. This motivates the use of Simulated Annealing, which is required only to be applied to problems with small jumps in the cost function between iterations [Davidson and Harel, 1996].

3.4 Algorithm Description

This subsection outlines the Simulated Annealing procedure we use to reduce network uniqueness. Algorithm 1 provides the corresponding pseudocode. We describe the algorithm's structure and the function of its main parameters.

Initialization and Temperature Setting. Algorithm 1 starts from an initial graph G_0 (Line 1) and proceeds with a predefined initial temperature T_0 (Line 2), which serves to control the acceptance of uphill moves during the early phase of the optimization.

Candidate Generation. At each iteration (Lines 4–13), a candidate graph G' is created by flipping a randomly chosen edge $(u, v) \in E_0$ (Lines 5–6), either deleting or adding it back to the set of edges, depending on its current state. Budget constraints are checked during implementation.

Evaluation of the Objective Function. The cost function $\mathrm{U}_M(G)$ measures the fraction of nodes that fail the k-anonymity condition under the signature model M. This is evaluated after generating the candidate graph G' to compute the change in cost $\Delta u = \mathrm{U}_M(G') - \mathrm{U}_M(G)$ (Line 7). Recomputing the equivalence classes and the uniqueness is computationally expensive. Thus, we use an *incremental evaluation strategy* for the (n, m)-anonymity measure. Specifically, we keep track of each node's equivalence class and at every deletion, recompute classes only for affected nodes.

The affected nodes include:

- the endpoints u and v, whose degrees decrease and whose triangle counts are updated;
- and any common neighbors w such that $\{u, w\} \in E$ and $\{v, w\} \in E$, since the number of their incident triangles decreases by one.

For each affected node x, we incrementally update its (n, m)-signature by adjusting the degree and triangle count accordingly. The pseudocode for the incremental update can be

found in Appendix A.1. This update mechanism avoids full recomputation of all signatures and equivalence classes.

Acceptance Criterion. Candidates that reduce the cost are always accepted (Line 12). Otherwise, acceptance is probabilistic, based on $\exp\left(-\frac{\Delta u+\eta}{T\cdot s}\right)$, where η is zero-mean Gaussian noise (Lines 10–13). This strategy follows Franzke and Kosko [Franzke and Kosko, 2019] to escape local optima.

State Update and Temperature Cooling. When a candidate is accepted, the graph and best-seen uniqueness are updated (Lines 12 and 14). The temperature decreases according to a geometric schedule (Line 16) with a cooling rate α and an iteration counter t.

Termination Condition. Algorithm 1 terminates after a fixed number of iterations or when no improvement has been observed within a predefined patience threshold (Lines 3 and 17). This avoids unnecessary computation once the uniqueness has finished its convergence, reducing the running time.

Algorithm 1 Simulated Annealing for Network Anonymization

1: Initialize $G \leftarrow G_0$, $u^* \leftarrow \mathrm{U}_M(G_0)$
2: Initialize T_0
3: Set iteration limit I, cooling rate α, and `patience` threshold
4: **for** $t = 1$ to I **do**
5: Sample edge (u, v) uniformly at random
6: Generate candidate graph G' by flipping (u, v)
7: Compute $\Delta u = \mathrm{U}_M(G') - \mathrm{U}_M(G)$
8: ▷ For $\mathrm{U}_{(n,m)}G'$: use incremental update as described in Subsection 3.4.
9: ▷ For $\mathrm{U}_{(d\text{-}k)}G'$: recompute uniqueness on full graph
10: Sample $\eta \sim \mathcal{N}(0, \sigma^2)$
11: **if** $\Delta u < 0$ **then**
12: Accept: $G \leftarrow G'$, $u^* \leftarrow \min(u^*, \mathrm{U}_M(G'))$
13: **else if** random() $< \exp\left(-\frac{\Delta u+\eta}{T\cdot s}\right)$ **then**
14: Accept: $G \leftarrow G'$
15: **end if**
16: Update temperature: $T \leftarrow T_0 \cdot \alpha^t$
17: **if** no improvement in last `patience` steps **then**
18: **break**
19: **end if**
20: **end for**
21: **return** G^* with minimal u^* found

4 Experiments

This section describes the experiments and finding of this paper. Subsection 4.1 describes the empirical datasets used, including structural statistics and uniqueness scores. Subsection 4.2 provides details of the experimental setup of the Simulated Annealing method. Subsection 4.3 present the main research question and three subquestions that concern anonymization quality, running time and problem setting performance conditions. Finally, in Subsection 4.4 we present our experimental results with comparisons of the methods, figures, and key insights.

4.1 Datasets

In this study, we use datasets that represent empirical social networks. We gather networks of various sizes, sparsities, and initial uniqueness. For example, the `CollegeMsg` dataset contains fewer nodes than the `ca-GrQc` dataset but contains approximately the same number of edges, which makes it denser. Likewise, the `ego Facebook` network has fewer nodes than `ca-GrQc` but contains over 5 times more edges, which makes it much denser. Table 1 shows an overview of each dataset. Short descriptions of each dataset are provided below:

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	ACC	T	U_{nm}	U_{dk}
Copnet SMS	568	697	0.139	0.154	0.026	0.044
fb-pages-food	620	2 102	0.330	0.222	0.191	0.404
Copnet FB	800	6.429	0.315	0.244	0.472	0.817
CollegeMsg	1.899	13.838	0.109	0.057	0.239	0.401
ca-GrQc	5.242	14.496	0.530	0.630	0.055	0.135
hamsterster	2.426	16.630	0.537	0.231	0.248	0.399
ego Facebook	4.039	88.234	0.606	0.519	0.587	0.812

Table 1: Statistics of datasets used in the experiments. T is short for transitivity. Transitivity and ACC measure the tendency of nodes to form triangles in a graph.

Copnet SMS [Sapiezynski *et al.*, 2019]: Consists of SMS interaction metadata between university students for a period of several weeks. Students are represented as nodes, and edges indicate the exchange of text messages.

Copnet FB [Sapiezynski *et al.*, 2019]: Derived from data representing Facebook friendship among the same group of students as Copnet SMS. Nodes represent users and edges represent Facebook friendships.

CollegeMsg [Leskovec and Krevl, 2014]: Based on message exchanges from an online campus social platform. Nodes are users, and an edge exists if at least one message was sent between two users, forming a sparse communication network.

ca-GrQc [Leskovec and Krevl, 2014]: A co-authorship network from the General Relativity and Quantum Cosmology category on arXiv. Nodes are authors, and an edge between two users represent a relation of co-authorship.

ego Facebook [Leskovec and Krevl, 2014]: An aggregated network of ego-centric Facebook data. Nodes are users, and edges correspond to Facebook friendships, resulting in a dense social graph.

fb-pages-food [Rossi and Ahmed, 2015a]: A network of Facebook pages related to food. Nodes are pages, and edges represent mutual likes, illustrating thematic connections between interest-based communities.

hamsterster [Rossi and Ahmed, 2015b]: Extracted from the Hamsterster online community. Nodes represent users, and edges represent reciprocal friendships, providing an example of user interactions on a niche platform.

4.2 Experimental Setup

Solving Methods: We evaluate our Simulated Annealing (SA) anonymization method against three baseline methods from Xie [Xie, 2023]—UA-Based Deletion, Logis-

tic Regression-Based Deletion, and (n, m)-Greedy Deletion (Section 3.1). All four methods are tested on seven real-world datasets and evaluated using final uniqueness scores under both (n, m)-anonymity and d-k-anonymity.

We conduct experiments at four budget levels: 1%, 3%, 5%, and 10% (see Definition 2.9). For `Copnet SMS`, `fb-pages-food`, and `Copnet FB`, results for UA and SA are averaged over 10 runs to account for randomness; the remaining datasets are evaluated once due to computational constraints.

Uniqueness values are reported with three decimal digits to reflect fine-grained differences (e.g., 0.001), and running times with two digits, as greater precision adds little value. Due to the high cost of computing d-k-anonymity, it is evaluated only on the three smaller datasets. These results should be interpreted as exploratory.

Experimental parameters: We tune the parameters for Simulated Annealing empirically. We test the effect of three parameters of SA on the uniqueness reduction for different graphs: initial temperature T_0, number of iterations, and cooling rate α. Based on a comparison of the effect of different temperatures for different types of networks, we select $T0 = 0.1$. For the cooling rate, we choose the value $\alpha = 0.6$ for small networks, $\alpha = 0.75$ for medium networks and $\alpha = 0.995$ for large networks, in line with prior work [Davidson and Harel, 1996]. The number of iterations is set to $B \cdot |E| \cdot 3$, where B is the edge deletion budget, and the algorithm terminates early if no improvement is seen over $\texttt{patienceRatio} \cdot \texttt{iterations}$ steps. We use $\texttt{patienceRatio} = min(8000, 0.3 \cdot iterations)$ unless stated otherwise. More parameter comparisons and figures are included in Appendix A.2, Figures 5-9.

Software: We implement all methods in Python 3.12.3. For graph processing, we use NetworkX to load and initialize graphs, and then convert them to igraph for efficient graph operations. We use the baseline methods from Xie [Xie, 2023] without modification, running them from the original repository and Jupyter Notebook provided by the author. We integrate our Simulated Annealing (SA) implementation into the same notebook environment to allow direct comparison under identical setup and evaluation conditions. For d-k-anonymity, we incorporate the implementation by Rachel de Jong [de Jong, 2025]. The source code for our Simulated Annealing implementation is publicly available.[1]

We also use NumPy, pandas, and scikit-learn (for the logistic regression baseline) and rely on the `subprocess` library to interface with a C# tool for d-k-anonymity.

Hardware: We conducted all experiments using the $(n.m)$ measure on the DelftBlue supercomputer cluster. For each experiment, we allocate one CPU core and 2 GB of RAM per task, and run seven tasks in parallel. For the d-k measure, we run all experiments locally on a machine with an Intel Core i7-1165G7 processor (4 cores, 8 threads, 2.80 GHz) and 16 GB of RAM, running a 64-bit Linux system under Microsoft Hyper-V virtualization.

[1] GitHub repository: arsenedenisa/Simulated-Annealing-for-Network-Anonymization

4.3 Research Questions

The main research question of this thesis is as follows:

> RQ. *How does a Simulated Annealing-based anonymization approach compare to the existing heuristic methods (as described in Section 3) in terms of running time and anonymization quality when achieving d-k-anonymity and (n, m)-anonymity in social network graphs?*

These are the main *subquestions* we explore in this research paper, along with their evaluation criteria:

1. **Anonymization Quality:** How does the anonymization quality of Simulated Annealing compare to established heuristic methods (e.g., (n, m)-Greedy, UA, Logistic Regression) on standard network anonymization benchmarks [Xie, 2023]?

   Evaluation criterion: Lower or comparable final uniqueness ($\mathrm{U}_M(G^*)$) under both d-k-anonymity and (n, m)-anonymity, given the same edge deletion budget.

2. **Running Time Efficiency:** How does the running time of Simulated Annealing compare to the aforementioned heuristic methods across graphs of varying sizes and densities?

   Evaluation criterion: Comparable running time (e.g., within 2x–3x), particularly in cases where Simulated Annealing yields a better anonymization quality than the other three methods.

3. **Problem Setting Performance Conditions:** For which problem setting (e.g., type of anonymity measure used, number of edges in the graph, number of nodes in the graph, initial uniqueness, graph density) does Simulated Annealing obtain a better solution quality or running time than the baseline methods from Subsection 3.1.

   Evaluation criterion: Identification of conditions where the Simulated Annealing method yields a better anonymization quality, running time, or both (compared to the other three methods).

4.4 Experimental Results

In this subsection, we present our results and answer our research questions. The results are obtained by running the methods described in Subsection 3.4 on the datasets from Subsection 4.1.

Q1: Anonymization Quality. Figure 2 illustrates that in a setting of a small dataset (`sms`) with a small budget, the Greedy method performs better due to its efficient local choices. However, for larger budgets (e.g., 10%), SA leverages its global cost optimization to achieve better anonymity. SA also maintains better performance on bigger networks such as `ca-GrQc` and `hamsterster`, where its thorough search compensates for high structural complexity.

As shown in Figure 3, the performance gap between SA and Greedy narrows as the budget increases in large networks. This indicates that SA might prematurely converge after a number of iterations, despite having remaining budget available for edge deletions, likely due to the acceptance probability's rapid decay. Although this is not explored in this paper,

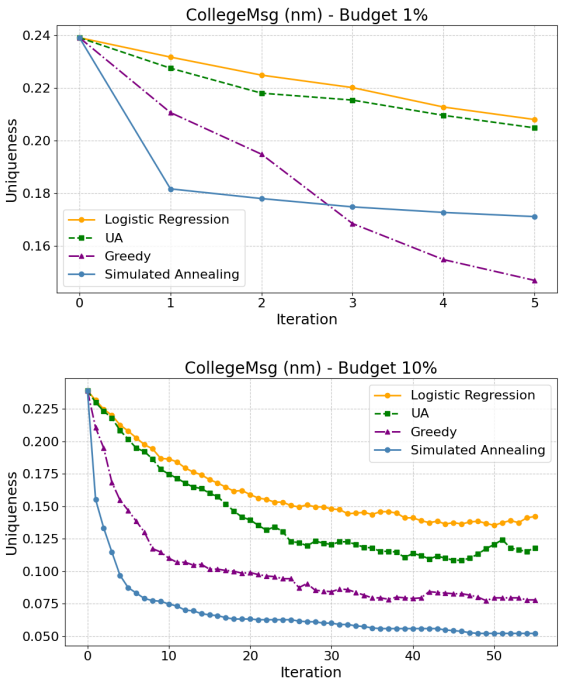

Figure 2: Methods Comparison ((n, m)-anonymity). Small network with small budget (1%) vs. small network with a large budget (10%).

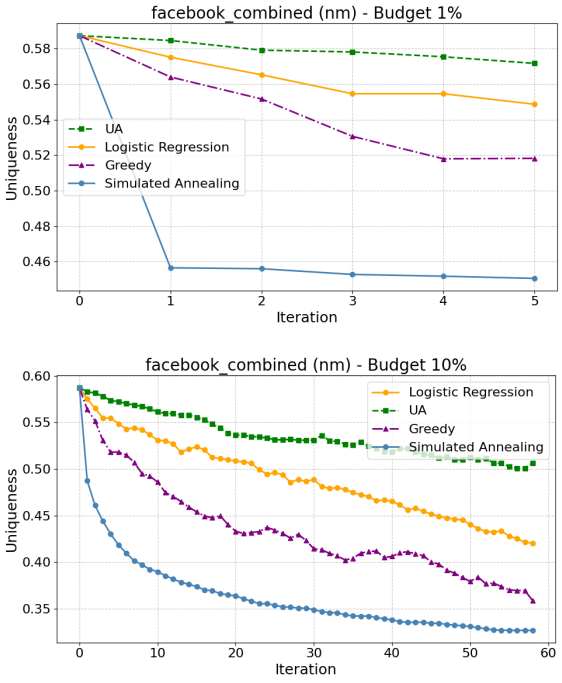

Figure 3: Methods Comparison ((n, m)-anonymity). Large network with small budget (1%) vs. large network with large budget (10%).

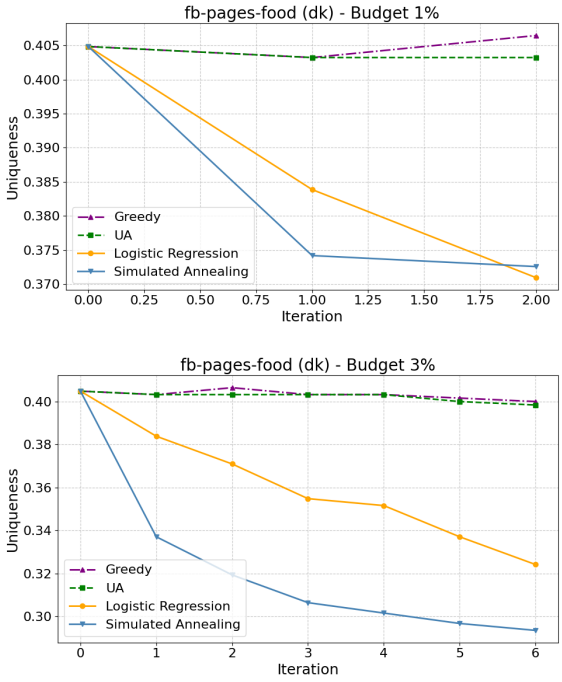

Figure 4: Methods Comparison (d-k-anonymity). Medium network with small budget (1%) vs. medium network with medium budget (10%).

a Simulated Annealing with a Restart Strategy (SARS) could potentially address this issue [Yu *et al.*, 2021].

Figure 4 demonstrates that results of the anonymization quality under the d-k-anonymity measure are mostly consistent with those for (n, m)-anonymity. For the `fb-pages-food` network, SA yields the best solution with a medium budget but ranks second under a small budget. Notably, under d-k-anonymity, the best-performing competitor to SA varies—being either Logistic Regression, Greedy, or UA—depending on the scenario, as reflected in Table 2.

Finally, Tables 2 and 3 confirm that SA consistently achieves better final uniqueness compared to Logistic Regression (LR) and UA. When compared to the Greedy method, SA tends to perform worse at small budgets (e.g., 1%) but surpasses it as the budget increases. For instance, in the `CollegeMsg` dataset, SA achieves the lowest uniqueness among all four methods at the 10% budget level.

Q2: Running Time Efficiency. Running time comparisons (Tables 2, 3, 4) show that SA has a significantly higher running time than the Greedy and UA methods.

Moreover, experiments that use the d-k-anonymity measure are ≈ 10 to 30 times slower than those that use (n, m)-anonymity. The high running time of $d - k$ is caused by repeated computations of uniqueness, which is slow for large graphs.

The UA method is the fastest method for all problem settings. However, UA also consistently gives the worst (highest) uniqueness of all four methods (except for one problem setting). Tables 2 and 3 show that for medium-sized networks (`CollegeMsg`) and when using the n, m-anonymity

Dataset	Method	1%				3%			
		$T_{n,m}$	$U_{(n,m)}$	$T_{d\text{-}k}$	U_{d-k}	$T_{n,m}$	$U_{(n,m)}$	$T_{d\text{-}k}$	U_{d-k}
Copnet SMS	LR	0.34	0.012	0.48	0.028	0.197	0.008	0.37	0.008
	Greedy	0.14	**0.008**	0.16	**0.024**	0.085	0.007	0.12	0.021
	UA	0.007	0.014	0.07	0.030	0.023	0.007	0.20	**0.005**
	SA	0.04	0.020	2.12	0.035	0.223	**0.006**	20.56	0.010
fb-pages-food	LR	1.38	0.162	1.79	**0.370**	4.01	0.141	4.26	0.324
	Greedy	0.51	**0.140**	0.73	0.406	1.49	0.106	2.32	0.400
	UA	0.039	0.175	0.28	0.401	0.11	0.159	1.41	0.387
	SA	0.75	0.153	32.30	0.372	2.69	**0.099**	163.29	**0.293**
Copnet FB	LR	16.94	0.443	24.35	**0.775**	51.96	0.365	69.70	0.706
	Greedy	5.61	**0.273**	5.54	0.818	17.55	0.201	26.70	0.806
	UA	0.38	0.417	1.23	0.817	1.22	0.342	3.81	0.808
	SA	7.03	0.353	71.38	0.777	32.06	**0.202**	696.03	**0.695**

Table 2: Uniqueness scores and running times (in seconds) of anonymization methods for budgets of **1%** and , for **small datasets**, across both (n, m) and d-k anonymity, for **small datasets**. Gray cells represent the settings for which SA gives the best final uniqueness.

Dataset	Method	5%				10%			
		$T_{n,m}$	$U_{(n,m)}$	$T_{d\text{-}k}$	U_{d-k}	$T_{n,m}$	$U_{(n,m)}$	$T_{d\text{-}k}$	U_{d-k}
Copnet SMS	LR	0.54	0.005	1.61	0.003	1.06	0.0	2.56	0.0
	Greedy	0.23	**0.0**	0.81	0.017	0.44	0.0	1.54	0.017
	UA	0.03	0.003	0.95	0.005	0.47	0.003	0.69	0.001
	SA	0.53	0.002	67.30	**0.001**	1.79	**0.0**	281.72	**0.0**
fb-pages-food	LR	6.42	0.119	12.89	0.296	13.02	0.095	20.77	0.211
	Greedy	2.45	0.083	4.68	0.396	5.03	0.071	8.40	0.353
	UA	0.19	0.143	1.77	0.364	0.40	0.130	3.15	0.356
	SA	6.50	**0.067**	445.04	**0.262**	17.65	**0.040**	1 276.03	**0.179**
Copnet FB	LR	85.60	0.325	175.45	0.675	165.21	0.272	218.95	0.566
	Greedy	28.89	0.170	39.677	0.768	56.80	0.145	79.860	0.733
	UA	2.04	0.299	8.131	0.788	3.96	0.261	13.177	0.753
	SA	73.26	**0.142**	1 502.84	**0.648**	661.06	**0.095**	5 534.08	**0.555**

Table 3: Uniqueness scores and running times (in seconds) of anonymization methods for larger budgets of **5%** and **10%**, across both (n, m) and d-k anonymity, for **small datasets**. Gray cells represent the settings for which SA gives the best final uniqueness.

Dataset	Method	1%		3%		5%		10%	
		$T_{n,m}$	$U_{(n,m)}$	$T_{n,m}$	$U_{(n,m)}$	$T_{n,m}$	$U_{(n,m)}$	$T_{n,m}$	$U_{(n,m)}$
CollegeMsg	LR	43.02	0.208	134.94	0.168	226.10	0.151	439.27	0.142
	Greedy	21.50	**0.147**	68.73	0.102	114.74	0.091	230.98	0.077
	UA	0.90	0.205	2.78	0.169	28.37	0.125	9.31	0.118
	SA	30.51	0.171	99.35	**0.096**	166.49	**0.075**	306.13	**0.052**
CA-GrQc	LR	45.61	0.040	153.77	0.028	249.12	0.024	490.44	0.018
	Greedy	80.05	**0.035**	266.67	0.028	433.07	0.023	853.22	0.021
	UA	0.85	0.042	2.76	0.044	4.67	0.041	8.98	0.033
	SA	31.24	0.036	55.46	**0.025**	69.17	**0.023**	139.39	**0.017**
hamsterster	LR	67.53	0.221	210.70	0.183	365.67	0.163	690.29	0.135
	Greedy	40.15	**0.180**	129.14	0.139	223.23	0.111	431.46	0.100
	UA	1.267	0.231	3.89	0.198	6.75	0.182	12.86	0.152
	SA	48.55	0.184	113.88	**0.126**	169.52	**0.110**	926.83	**0.073**
ego Facebook	LR	1 329.68	0.548	4 026.02	0.512	6 758.49	0.486	12 320.98	0.420
	Greedy	383.21	0.518	1 277.18	0.447	2 141.84	0.423	4 031.05	0.358
	UA	15.74	0.571	56.50	0.539	85.79	0.531	154.98	0.506
	SA	1 110.49	**0.45**	20 655.17	**0.332**	14 536.00	**0.331**	17 042	**0.320**

Table 4: Final uniqueness scores and running times for four anonymization methods across four **larger** datasets, evaluated under the (n, m)-anonymity model. Results are reported for different edge deletion budgets (1%, 3%, 5%, and 10%).

measure, SA is usually faster than LR but 2-3 times slower than the Greedy method. For bigger networks and when using (n, m)-anonymity (see Table 4), there are cases where SA is faster than both Greedy and LR. For example, for budgets of 1%, 3%, and 5% on the `CA-GrQc` dataset, which represents a sparser network, SA has the best running time (except for UA) while still obtaining the best anonymization quality. Notably, the SA running time increases substantially with dataset size and budget, as seen in `facebook_combined` and `fb_friends`.

However, the quality-to-time tradeoff can still be justified in cases where the search space is larger (e.g., `ego Facebook`). For a 1% budget, the running time of SA is approximately 3 times higher than that of Greedy, which achieves the second-best uniqueness. Despite this, SA reduces uniqueness by roughly a factor of 2 more than Greedy.

Although the running time of SA is not the best for datasets like `ego Facebook`, its flexibility allows for adjusting the number of iterations, potentially aligning the running time with the other methods while still achieving the best solution of all methods.

Q3: Performance Conditions. From the tables' results and the figures, we identify specific conditions where SA performs the best either in terms of running time or quality of solution:

- SA achieves the best running time for the (n, m) measure, as it uses an incremental evaluation strategy. In contrast, the d-k measure uses a version of SA that recalculates the uniqueness at every iteration, which is more costly in terms of running time. This is illustrated in Table 2.
- SA performs particularly well (using (n, m)-anonymity) in terms of balance between running time and solutions of quality on sparser or mid-sized graphs (e.g., `CA-GrQc`, `hamstersyer`)
- For d-k anonymity, SA achieves the lowest uniqueness on `Copnet SMS` for larger budgets (Table 3). On denser networks like `fb-pages-food` and `Copnet FB`, it outperforms all other methods across all budgets except for 1%.
- It outperforms other methods in terms of anonymization quality when the initial uniqueness is high (e.g., `Copnet FB`, `CA-GrQc`) and when the edge deletion budget is sufficient to enable global exploration.
- For (n, m)-anonymity, the performance gap in terms of anonymization quality, between SA and Greedy increases with the budget (5% and 10% for smaller networks and $> 3\%$ for larger networks), which shows that after SA passes the exploration phase, it converges to a better solution than the other methods.

In large, sparse networks (e.g., `ego Facebook`), SA may not finish within a reasonable time for large budgets. However, even under these constraints, SA manages to outperform others at smaller budgets.

5 Conclusion and Future Work

In this paper, we explored how Simulated Annealing (SA), a probabilistic optimization technique, could be used to improve the anonymization of social networks. The main goal was to reduce the risk of re-identification in graphs by making users less unique under formal anonymity measures like (n, m)-anonymity and d-k-anonymity.

We compared our approach with three existing solving methods and tested them on seven real-world networks. The results showed that while SA takes longer to run, it often produces better anonymization, especially when there's a higher available edge deletion budget. In particular, SA achieves the best anonymity in situations where the network is dense or when there's a high initial uniqueness. These conditions all have in common that they would result in a larger search space.

Overall, Simulated Annealing turned out to be a strong alternative to traditional methods, especially when the focus is on anonymization quality over running time. The incremental evaluation strategy tailored for the (n, m) model also proved to be a significant advantage in reducing the computational burden.

For the future, it would be interesting to explore methods that use a restart strategy,such as Simulated Annealing with Restarts (SARS), to escape local optima. Another direction would be adapting SA with other existing anonymity measures, beyond (n, m)-flavoured and d-k-anonymity. Lastly, to investigate the trade-off between anonymity and structural metrics such as transitivity and average clustering coefficient (ACC), which measure the tendency of nodes to form triangles, it would be interesting to try modeling the cost function of Simulated Annealing to also include changes in ACC and Transitivity.

Acknowledgments

This research was supported by my supervisor, Anna Latour, within the Research Project course. I would like to thank R. G. de Jong and F. W. Takes from Leiden University for their valuable assistance during the research. I also gratefully acknowledge the support of TU Delft for providing the computational resources used in this study.

Responsible Research

This research complies with the TU Delft Code of Conduct [Roeser and Copeland, 2020] and its DIRECT values (Diversity, Integrity, Respect, Engagement, Courage, and Trust).

This work satisfies the integrity value from the code of conduct by referencing the repository of the implementation and results, the parameter values used, and all the datasets. This also ensures the reproducibility of the paper and allows others to replicate, validate, or extend our findings.

In the context of network anonymization for research, ethical considerations are a main concern of this study.

A primary ethical consideration is privacy, as the use of networks for research can expose individuals' identities, when the data is not anonymized. In this research, there were no human subjects involved, as all the networks used were publicly available networks. Moreover, no sensitive data was collected, and all results will be made public via this project's repository, following the F.A.I.R. principles [Wilkinson *et al.*, 2016].

We acknowledge that this work could be misused by malicious parties. It provides anonymization methods for networks, which could be used to create false anonymity or mask structural patterns to deceive researchers. We encourage transparency and responsible use and recommend that people consider the ethical aspects of it.

Some potential biases in this research might be the use of specific measures (n, m)-anonymity and d-k-anonymity. For datasets where attackers have other types of knowledge, we cannot evaluate the usability of the anonymization method described in this paper. The datasets we used in this research might also cause bias. They might not represent all populations or types of networks.

In this paper, no generative AI tools (e.g., ChatGPT) were used to produce or interpret results. All algorithms were implemented, run, and analyzed by the authors. AI tools (`Grammarly`) were only used for checking and refining the grammar of the paper.

References

[Campan *et al.*, 2015] Alina Campan, Yasmeen Alufaisan, and T.M. Truta. Preserving communities in anonymized

social networks. *Transactions on Data Privacy*, 8:55–87, 04 2015.

[Cheng *et al.*, 2010] James Cheng, Ada Wai-Chee Fu, and Jia Liu. K-isomorphism: Privacy preserving network publication against structural attacks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 459–470, 2010.

[Davidson and Harel, 1996] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996.

[de Jong *et al.*, 2023] R. G. de Jong, M. P. J. van der Loo, and F. W. Takes. Algorithms for efficiently computing structural anonymity in complex networks. *ACM Journal of Experimental Algorithmics*, 28:1.7:1–1.7:22, 2023.

[de Jong *et al.*, 2024] Rachel G. de Jong, Mark P. J. van der Loo, and Frank W. Takes. A systematic comparison of measures for k-anonymity in networks, 2024. arXiv preprint arXiv:2407.02290.

[de Jong, 2025] Rachel de Jong. dkAnonymity, January 2025.

[Franzke and Kosko, 2019] Brandon Franzke and Bart Kosko. Noise can speed markov chain monte carlo estimation and quantum annealing. *Phys. Rev. E*, 100:053309, Nov 2019.

[Kirkpatrick *et al.*, 1983] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[Latour, 2024] Anna L.D. Latour. Research note - Anonymisation - ILP encoding. 2024.

[Leskovec and Krevl, 2014] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data, 2014. Accessed: 2023-11-22.

[Roeser and Copeland, 2020] Sabine Roeser and Samantha Copeland. Tu delft code of conduct, 2020. In collaboration with Pavol Bauer, Saskia de Been, Martijn Blaauw, Rinze Benedictus, Irene van den Brink, Jenny Dankelman, Marja Elsinga, Katharina Ertman, Tim van der Hagen, Han Heijmans, Merle de Kreuk, Lotte Melenhorst, Adriaan van Noord, Jan Nuijten, Danko Roozemond, Hans Suijkerbuijk, Marthe Uitterhoeve, Anke Versteeg. Design: Paul van Elk. Licensed under Creative Commons Attribution 4.0 International License.

[Romanini *et al.*, 2021] Daniele Romanini, Sune Lehmann, and Mikko Kivelä. Privacy and uniqueness of neighborhoods in social networks. *Scientific Reports*, 11(1):20104, 2021.

[Rossi and Ahmed, 2015a] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[Rossi and Ahmed, 2015b] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[Sapiezynski *et al.*, 2019] Piotr Sapiezynski, Arkadiusz Stopczynski, David Dreyer Lassen, and Sune Lehmann. Interaction data from the copenhagen networks study. *Scientific Data*, 6(1):315, 2019.

[Wilkinson *et al.*, 2016] Mark D. Wilkinson, Michel Dumontier, and I. J. et al. Aalbersberg. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016.

[Winkler, 2002] William E. Winkler. Using simulated annealing for k-anonymity. Technical report, U.S. Census Bureau, 2002.

[Wu *et al.*, 2010] Wentian Wu, Yabo Xiao, Wei Wang, Zhiguo He, and Zhenying Wang. K-symmetry model for identity anonymization in social networks. In *Proceedings of the 13th International Conference on Extending Database Technology (EDBT)*, pages 111–122. Association for Computing Machinery, 2010.

[Xie, 2023] X. Xie. *Anonymization Algorithms for Privacy-Sensitive Networks*. PhD thesis, LIACS, Leiden University, 2023. Accessed: Feb. 13, 2025.

[Yu *et al.*, 2021] Vincent F. Yu, Winarno, Achmad Maulidin, A. A. N. Perwira Redi, Shih-Wei Lin, and Chao-Lung Yang. Simulated annealing with restart strategy for the path cover problem with time windows. *Mathematics*, 9(14), 2021.

[Zhou and Pei, 2008] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. *2008 IEEE 24th International Conference on Data Engineering*, pages 506–515, 2008.

A Appendix

A.1 Incremental Uniqueness Update Pseudocode

Algorithm 2 Incremental Uniqueness Computation for (n, m)-anonymity

Require: Graph G, previous equivalence class mapping `prev_classes` (optional), affected nodes A (optional), anonymity threshold k

Ensure: Updated uniqueness score u, set of unique nodes, updated node-to-class and class-to-node mappings

if `prev_classes` is provided **then**
    `node_to_class` $\leftarrow$ copy of `prev_classes`
else
    `node_to_class` $\leftarrow$ empty map
end if
if $A = \emptyset$ **then**
    $A \leftarrow$ all nodes in G
end if
for each node $v \in A$ **do**
    Compute the (n, m) equivalence class of v
    Update `node_to_class`$[v]$
end for
Initialize `class_to_nodes` as empty map
for each node v and its class c in `node_to_class` **do**
    Add v to `class_to_nodes`$[c]$
end for
`unique_nodes` $\leftarrow \{v \mid$ `class_to_nodes`$[c] < k$ for $c =$ `node_to_class`$[v]\}$
$u \leftarrow \frac{|\text{unique\_nodes}|}{|V(G)|}$
return u, `unique_nodes`, `node_to_class`, `class_to_nodes`

A.2 Parameter tuning

A.2.1 Initial temperature T_0

We tested values in the set $\{0.01, 0.1, 1.0, 10.0, 100.0\}$. We concluded that the value 0.1 yields the best results for all types of networks. Figure 5 shows an example of running SA on a small network with multiple initial temperature values. Appendix A.2 contains comparison of temperatures on more networks.

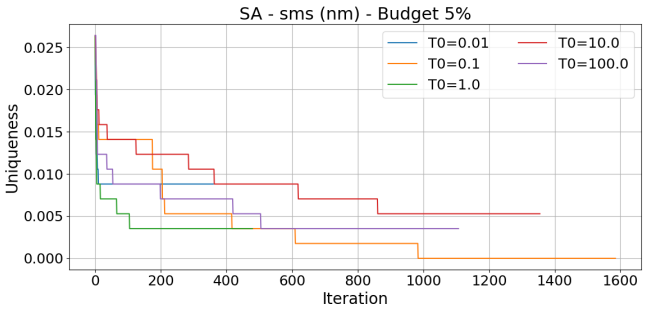

Figure 5: Results of SA run on `Copnet SMS` datasets with different initial temperature T_0. The only value for T_0 that obtains a 0 uniqueness is $T_0 = 0.1$.

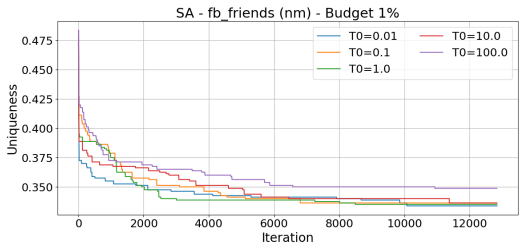

Figure 6: Uniqueness variation for different temperature values - `Copnet FB` network. For a medium network and smaller budget, the smallest temperature (0.01 value achieves the smallest final uniqueness.

A.2.2 Cooling rate α

We tested values in the set $\{0.5, 0.6, 0.75, 0.85, 0.9, 0.95\}$. Our experiments showed that $\alpha = 0.75$ offers a good trade-off between exploration and convergence speed for a medium number of deletions available, in line with prior work [Davidson and Harel, 1996]. For networks with a smaller number of edges, a smaller alpha value (0.5 or 0.6) yields the best results (Figure 7). However, for larger graphs where the solution space is more complex, a slower cooling rate of $\alpha = 0.995$ consistently achieved lower uniqueness, consistent with observations in [Kirkpatrick *et al.*, 1983].

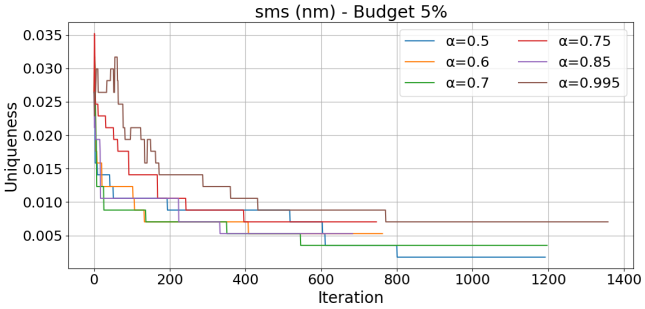

Figure 7: Uniqueness variation for different alpha values - `Copnet SMS` network. For this small network, $\alpha = 0.5$ achieves the best uniqueness.

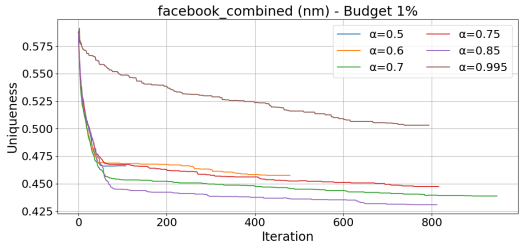

Figure 8: Uniqueness variation for different alpha values - `ego Facebook` network. For a large network and smaller budget, a medium to higher α value achieves the smallest final uniqueness.

A.2.3 Iterations and patience

The number of iterations was set to $B \cdot |E| \cdot 3$, where B is the edge deletion budget, and the algorithm terminated

early if no improvement was seen over $\texttt{patienceRatio} \cdot \texttt{iterations}$ steps. We used $\texttt{patienceRatio} = min(8000, 0.3 \cdot iterations)$ unless stated otherwise. In Figure 9, we observe that after iteration 250, the uniqueness is constant. In this case, it is not worth exploring new solutions.

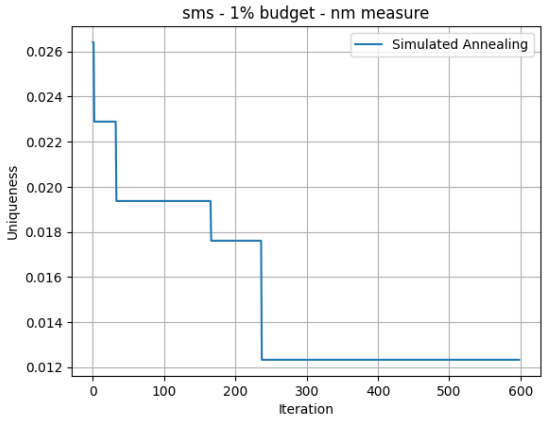

Figure 9: Progress of Simulated Annealing's uniqueness with a 1% budget over time. After about 250 iterations, improvement stops.