Network Anonymization for Science: A Simulated Annealing Approach

AUTHORS
Student: Denisa Arsene; earsene@student.tudelft.nl
Supervisor & Responsible Professor: Dr. Anna L. D. Latour

AFFILIATIONS
Delft University of Technology

TU Delft Delft University of Technology

GITHUB
https://github.com/arsenedenisa/Simulated-Annealing-for-Network-Anonymization

BACKGROUND

Network Anonymization

- Social networks risk privacy leaks via structural re-identification.
- Network anonymization modifies graphs to make individuals indistinguishable.
- Existing methods of solving either lack quality of solution (Xie, 2023) or are too slow (ILP)

Preliminaries:

- **(n,m)-flavoured k-anonymity** (Latour, 2024): indistinguishability based on degree and triangle count.
- **d-k-anonymity** (de Jong et al., 2023): indistinguishability based on d-depth neighborhood structure.
- **Edge Deletion Budget:** % of edges that may be deleted to anonymize.
- **Goal**: Achieve maximum anonymization by deleting as few edges as possible within a given budget.

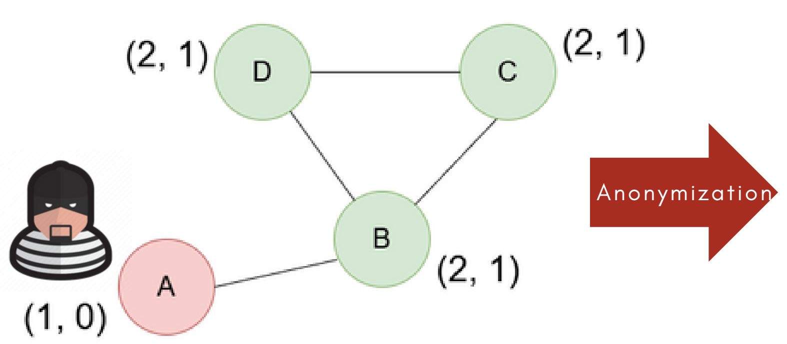

Fig. 1: Graph before (n,m)-anonymization

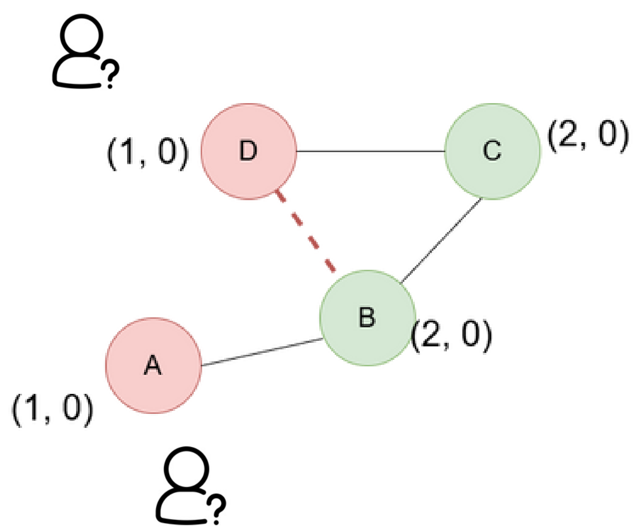

Fig. 2: Graph after (n,m)-anonymization

MOTIVATION - SIMULATED ANNEALING

- Structural privacy is hard to guarantee in real-world networks.
- Simulated Annealing (SA) offers a **compromise:** good quality of anonymization with acceptable run-time.
- **Goal:** Improve anonymity while staying within a **fixed** edge deletion **budget**.
- **SA advantages:** escapes local minima, flexible, easy to tune, has been previously used to solve other types of k-anonymity problems (Winkler, 2002).

REFERENCES

[1] de Jong, R. G., van der Loo, M. P. J., & Takes, F. W. (2023). Algorithms for Efficiently Computing Structural Anonymity in Complex Networks. ACM J. Exp. Algorithmics, 28, 1.7:1-1.7:22. T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks". In: IEEE Access (2019). DOI: https://doi.org/10.1109/ACCESS.2019.2909068.
[2] Latour, A. L. D. (2024). Research note—Anonymisation—ILP encoding. personal communication (unpublished).
[3] Xie, X. (2023). Anonymization algorithms for privacy-sensitive networks [LIACS, Leiden University]. https://theses.liacs.nl/2838
[4] Winkler, W. (2002). Using Simulated Annealing for k-Anonymity.

RESEARCH QUESTION

1. How does a Simulated Annealing-based anonymization approach compare to existing heuristic methods when achieving (n,m)- and d-k-anonymity in terms of:
   a. quality of solution
   b. running time
2. For which problem setting does SA perform best in terms of running time or anonymization quality?

METHODOLOGY

Compare Simulated Annealing to methods from Xie[1]:
1. Logistic Regression Deletion
2. UA Deletion
3. Greedy Deletion

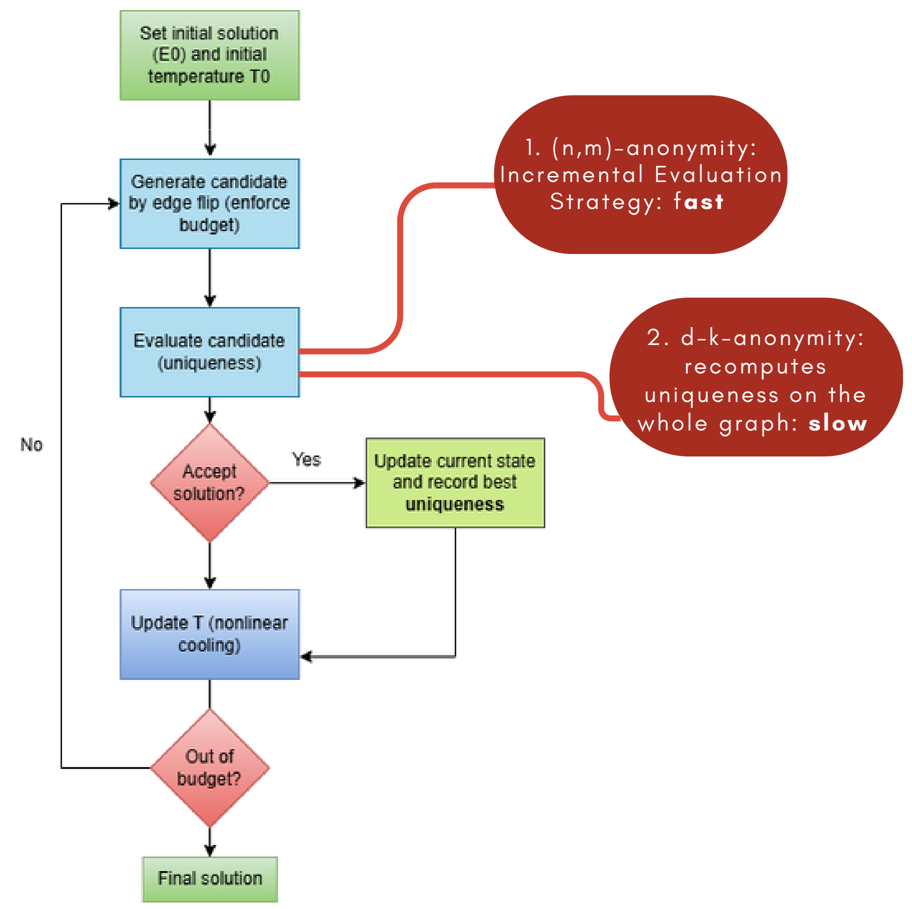

Fig. 3: Simulated Annealing Implementation for Network Anonymization

EXPERIMENTAL SETUP

Setup
- 7 **datasets** that represent empirical social networks
- 4 **budgets** for available edge deletion: {1%, 3%, 5%, 10%}
- 2 **metrics**: (n,m)-anonymity, d-k-anonymity
- 4 **methods**: SA & heuristics

RESULTS

1) QUALITY OF SOLUTION & RUNNING TIME

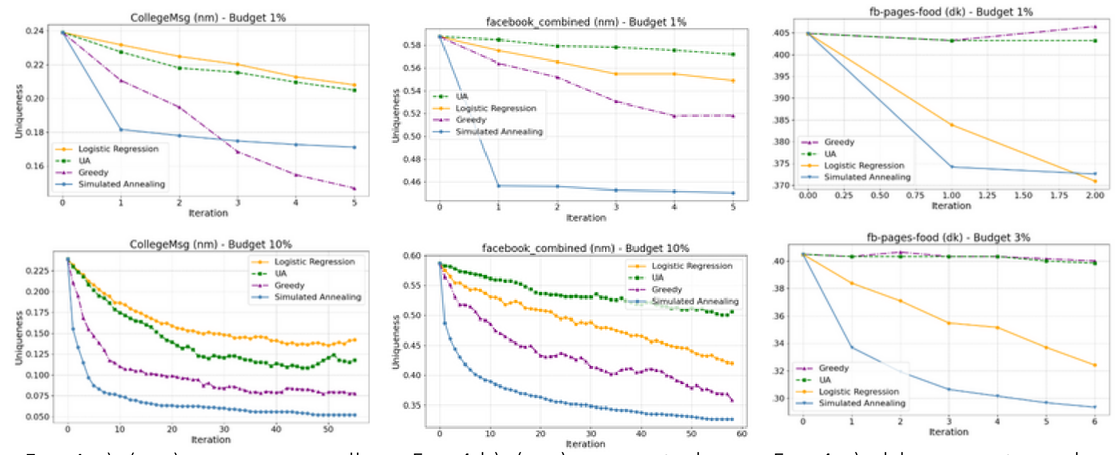

Fig. 4.a): (n,m)-anonymity, small network
Fig. 4.b): (n,m)-anonymity, big network
Fig. 4.c): d-k-anonymity, medium network

Key Findings (Solutin Quality):
- **(n,m)-anonymity**: SA outperforms all other methods in terms of solution quality for budgets >3% (Fig. 4.a). In the other cases, Greedy is faster.
- **d-k-anonymity**: SA outperforms all other methods for budgets >3% (Fig. 4.c).
- For the largest network, the gap between SA and Greedy becomes smaller for a bigger budget (Fig. 4.b)

budget →
network size ↓

Dataset	Method	1% $T_{n,m}$	1% $U_{(n,m)}$	3% $T_{n,m}$	3% $U_{(n,m)}$	5% $T_{n,m}$	5% $U_{(n,m)}$	10% $T_{n,m}$	10% $U_{(n,m)}$
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 1: Running time and final Uniqueness for large datasets, using (n,m)-anonymity

Key Findings (Running time):
- SA is usually slower than the Greedy method
- UA is the fastest method but yields the worst solution.
- For d-k-anonymity, the running time increases quickly with the size of the graph.

2) CONDITIONS FOR SA OPTIMALITY

budget →
network size ↓

Dataset	Method	1% $T_{n,m}$	1% $U_{(n,m)}$	1% $T_{d\text{-}k}$	1% U_{d-k}	3% $T_{n,m}$	3% $U_{(n,m)}$	3% $T_{d\text{-}k}$	3% U_{d-k}	5% $T_{n,m}$	5% $U_{(n,m)}$	5% $T_{d\text{-}k}$	5% U_{d-k}	10% $T_{n,m}$	10% $U_{(n,m)}$	10% $T_{d\text{-}k}$	10% U_{d-k}
Copnet SMS	LR	0.34	0.012	0.48	0.028	0.197	0.008	0.37	0.008	0.54	0.005	1.61	0.003	1.06	0.0	2.56	0.0
	Greedy	0.14	**0.008**	0.16	**0.024**	0.085	0.007	0.12	0.021	0.23	**0.0**	0.81	0.017	0.44	0.0	1.54	0.017
	UA	0.007	0.014	0.07	0.030	0.023	0.007	0.20	**0.005**	0.03	0.003	0.95	0.005	0.47	0.003	0.69	0.001
	SA	0.04	0.020	2.12	0.035	0.223	**0.006**	20.56	0.010	0.53	0.002	67.30	**0.001**	1.79	**0.0**	281.72	**0.0**
fb-pages-food	LR	1.38	0.162	1.79	**0.370**	4.01	0.141	4.26	0.324	6.42	0.119	12.89	0.296	13.02	0.095	20.77	0.211
	Greedy	0.51	**0.140**	0.73	0.406	1.49	0.106	2.32	0.400	2.45	0.083	4.68	0.396	5.03	0.071	8.40	0.353
	UA	0.039	0.175	0.28	0.401	0.11	0.159	1.41	0.387	0.19	0.143	1.77	0.364	0.40	0.130	3.15	0.356
	SA	0.75	0.153	32.30	0.372	2.69	**0.099**	163.29	**0.293**	6.50	**0.067**	445.04	**0.262**	17.65	**0.040**	1 276.03	**0.179**
Copnet FB	LR	16.94	0.443	24.35	**0.775**	51.96	0.365	69.70	0.706	85.60	0.325	175.45	0.675	165.21	0.272	218.95	0.566
	Greedy	5.61	**0.273**	5.54	0.818	17.55	0.201	26.70	0.806	28.89	0.170	39.677	0.768	56.80	0.145	79.860	0.733
	UA	0.38	0.417	1.23	0.817	1.22	0.342	3.81	0.808	2.04	0.299	8.131	0.788	3.96	0.261	13.177	0.753
	SA	7.03	0.353	71.38	0.777	32.06	**0.202**	696.03	**0.695**	73.26	**0.142**	1 502.84	**0.648**	661.06	**0.095**	5 534.08	**0.555**

Table 2: Running time and final Uniqueness for small datasets, using (n,m) and d-k-anonymity

Key Findings:
- SA performs better than baselines in final uniqueness for dense graphs like ego Facebook.
- SA performs best when:
  - Graph is **dense** or large,
  - Initial uniqueness is **high**,
  - Deletion budget is ⩾5%.

CONCLUSION

Key Findings:
- SA excels on dense networks and with high initial uniqueness, especially under larger edge deletion budgets.
- While usually slower than Greedy, SA achieves consistently lower uniqueness for (n,m)-anonymity.
- For d-k-anonymity, SA is the slowest but yields a better solution for more complex networks.
- **Overall, SA provides a strong middle ground between heuristic speed and optimality of solution.**

Future Work:
- Use restart strategies (e.g., SARS) to escape local minima.
- Extend to other anonymity models beyond (n,m) and d-k.
- Compare SA method with optimal solutions