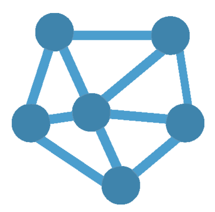
A Pseudo-Boolean Approach to Full Graph Anonymisation

Emke de Groot, supervised by dr. Anna Latour
E.A.deGroot-1@student.tudelft.nl

Motivation

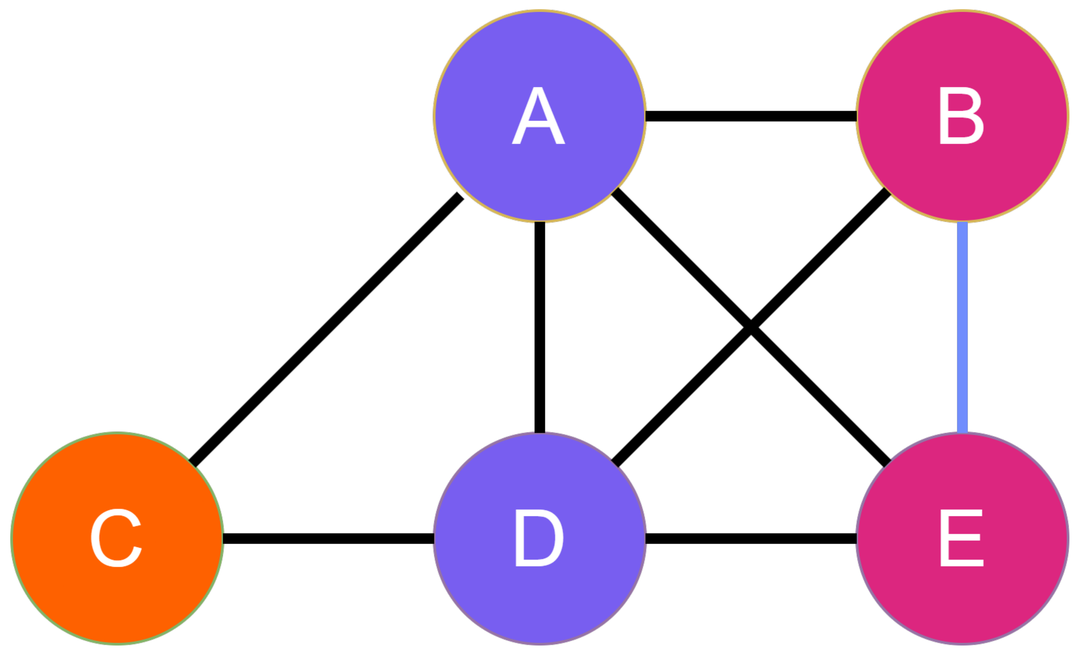


Before making a network available for social scientists, we must:

- Completely **anonymise** the network
- Change **as little** as possible
  → Give a **guarantee** that we made the **smallest possible** change to the network

Problem Statement

We want to **completely anonymise** a graph $G := (V, E)$, using the (n, m)-k-anonymity measure [4], by removing as **few edges** as possible.



Current Approaches

- **No state of the art** for complete methods with the (n, m)-k-anonymity measure
- No method that allows for the creation of **certificates of optimality**
- Unpublished **Integer Linear Program** (ILP) [2]
  - Uses both **Boolean** and **integer** variables
  - Introduces a Boolean variable $\ell_{u,v}$ for every **edge** $(u, v) \in E$ in the input graph
  - **Maximises** sum of all ℓ

Research Question

How does a **Pseudo-Boolean** approach to network anonymisation compare to the existing methods in terms of **solving time**, **memory consumption** and **quality of the solution** on networks with differing characteristics?

Pseudo-Boolean Constraints

Linear pseudo-Boolean (PB) constraint [1]:

$$\sum_j a_j l_j \bowtie b$$

where a_j and b are integer constants, l_j are literals and $\bowtie$ is one of $(=, >, \geq, <$ or $\leq)$.

Constraint Conversion

We have one **original constraint** for each triangle (u, v, w) in the input graph:

$$t_{u,v,w} \Leftrightarrow (\ell_{u,v} + \ell_{v,w} + \ell_{u,w} \geq 3)$$

We convert this constraint to **two PB constraints**:

$$3 \cdot \overline{t_{u,v,w}} + \ell_{u,v} + \ell_{v,w} + \ell_{u,w} \geq 3$$
$$t_{u,v,w} + \overline{\ell_{u,v}} + \overline{\ell_{v,w}} + \overline{\ell_{u,w}} \geq 1$$

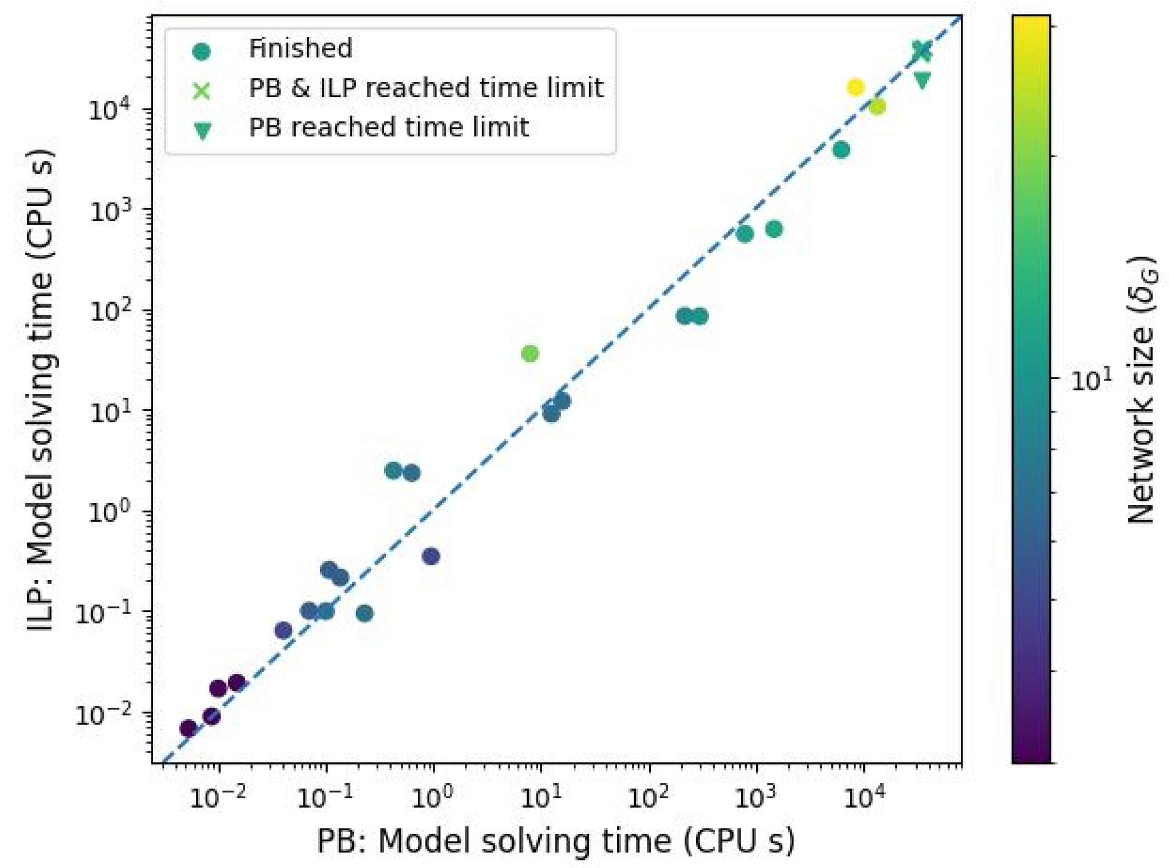
Our Approach

- Implementation of PB model in **Python**
- Solving the model with **Gurobi** and **SCIP**
- Using publicly available **animal social networks** as problem instances [3]
  → largest network: 171 nodes, 378 edges, 3276 triangles, maximum degree 31, maximum incident triangles 351
- Running experiments on **DelftBlue**
- Comparing encoding time, solving **time** and maximum **memory** consumption

Experimental Results

- PB is **slower** on average for most networks
  → expectation: by a **constant** factor
  → faster for smallest networks
- Encoding the PB model is **twice** as slow
  → but: encoding time **fraction** of total time
- Memory usage of both models is comparable
- **Gurobi** is faster for all networks
- **SCIP** does not find the **optimum** solution for some PB models
- Number of **edges** and **triangles** predict performance the best
  → expectation: **dense** networks perform worse

Solving Time



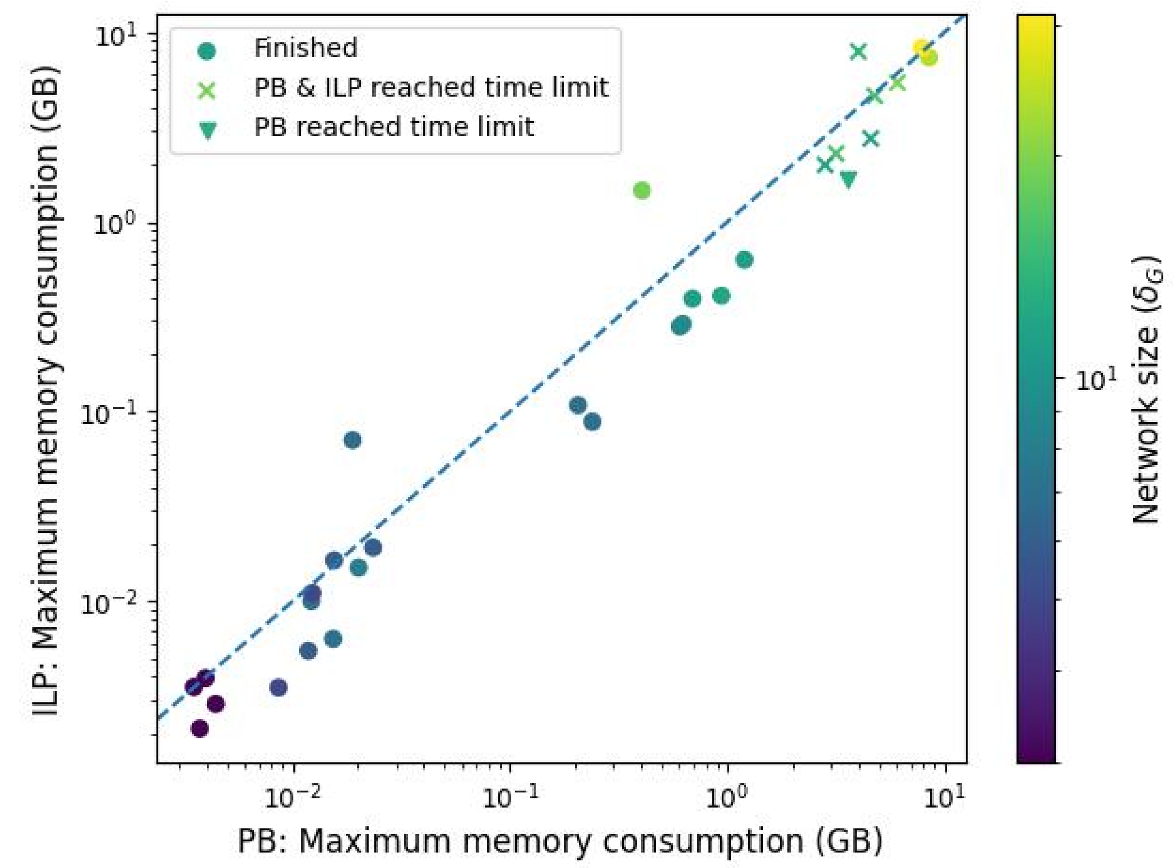
Conclusion

- Complete method → slow
- Comparable to the ILP approach
- Both methods perform worse for **dense networks**

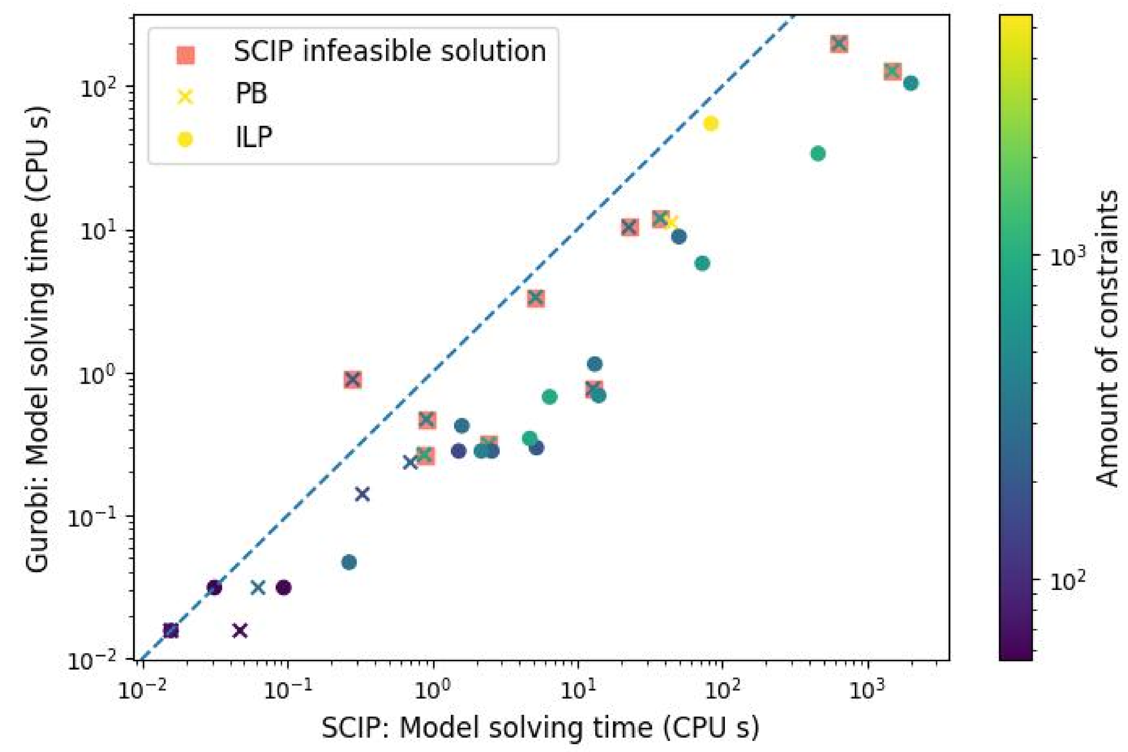
Future work:

- Budgeted version

Memory Usage



Different Solvers



References

[1] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of Satisfiability*. Number v. 336 in Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, second edition, 2021.

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

[3] Ryan A. Rossi and Nesreen K. Ahmed. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*, 2015.

[4] Xinyue Xie. *Anonymization Algorithms for Privacy-Sensitive Networks*. PhD thesis, LIACS, Leiden University, 2023.