# Custom topics for SOS Exercise 1

This document lists a few custom topic suggestion where either some form of optimisation or simulation could be applied to. As these topics are less well-known than others you might pick, if you chose to work with those, it is sufficient to use just ONE problem, instead of the otherwise required 2-3.

The topics are described below, but we are happy to clarify aspects that are not clear also in further discussion.

## Time-constrained "Travelling Christmas market visitor"

Christmas is around the corner, and with that comes an optimisation problem! Imagine you are only one day in Vienna, but want to visit as many Christmas markets as possible (either for drinking Glühwein or shopping, whatever you prefer :-) ). Your task is thus to heuristically optimise a route that maximises the amount of markets you can visit[1]. Below you find a list of markets, but your dataset is still incomplete, as there are no timings between the markets, thus a first sub-task is to obtain those; ideally you use an API to query e.g. walking and public transport (Google maps: https://developers.google.com/maps/documentation/directions/overview?hl=de, Wiener Linien or similar)

Further, there are the following constraints:

- At each market, you stay a predefined amount of minutes (use 30 minutes as default; make this easily configurable)
- Each market has a different opening and closing time, which are constraints on whether a market can be visited
- The overall available time is limited by the earliest opening and latest closing time
- Markets should only be visited once!
- You can assume that you start at any given market, and end at any given market, you do not care to start from e.g. your hotel/apartment :-)
- As it is impossible to visit all markets in one day, let the number of days to spend on completing the route be open. E.g. if you set 2 days, then this basically means you shall solve the fastest route for the second day with just the remaining markets. You can treat this as a simple iterative processing, you do not need to optimise the route for all days at the same time. For spanning over multiple days, let the user provide a different duration of stay at the markets for each day (simulating being less fresh on the 2nd and subsequent days)

You shall describe your design, coding and solution, together with an analysis of the results, in a report.

1 So technically, this is not a travelling salesperson, as you might not be able to visit all locations, but the name for this still was chosen as very descriptive in illustrating the task to solve!

| Name | Map | Opens | Closes |
|---|---|---|---|
| MQ | https://g.page/weihnachtsquartier?share | 14:00 | 23:00 |
| Altes AKH | https://goo.gl/maps/vLFEPeaSVHzfksP77 | 11:00 | 23:00 |
| Am Hof | https://goo.gl/maps/2cMY3oeEZbqstDtP6 | 10:00 | 21:00 |
| Spittelberg | https://goo.gl/maps/M1cs1DnR1NJRnzPy5 | 10:00 | 21:30 |
| Stephansplatz | https://goo.gl/maps/Bg3JX4wNoKwmnpv5A | 11:00 | 21:00 |
| Opera | https://goo.gl/maps/zRGFceVt4vgKmicm7 | 11:00 | 21:00 |
| Türkenschanzpark | https://goo.gl/maps/SmpWwiEbWanCXuLUA | 12:00 | 22:00 |
| Maria-Theresien-Platz | https://goo.gl/maps/dLFGPAZB7E5UHQNt6 | 11:00 | 21:00 |
| Blumengärten Hirschstetten | https://goo.gl/maps/UodxLTKcLArtyG9i8 | 10:00 | 22:00 |
| Palais Liechtenstein | https://goo.gl/maps/qnUX4zmdazoCtY3x9 | 11:00 | 21:00 |
| Belvedere | https://g.page/belvederemuseum?share | 10:00 | 21:00 |
| Karlsplatz | https://goo.gl/maps/9UAJTtsxSRAmrPuU7 | 12:00 | 20:00 |
| Messe | https://goo.gl/maps/mgUHYBriHmwAFcK19 | 14:00 | 22:00 |
| Altwiener Christkindlmarkt | https://goo.gl/maps/QbTxRmduaxttA3Tz5 | 10:00 | 21:00 |
| Biobauern-Adventmarkt | https://goo.gl/maps/Fc8DGKbv3aMZjVec7 | 10:00 | 19:30 |
| Rathaus | https://goo.gl/maps/Xq7g7TsExSszrhsB6 | 10:00 | 21:30 |
| Schönbrunn | https://goo.gl/maps/KY6oixzsxM5PVjvb8 | 10:00 | 21:00 |
| Prater / Riesenrad | https://goo.gl/maps/bJ2obKMdq2nZWiQE6 | 11:00 | 22:00 |
| Mittelalterlicher Adventmarkt | https://goo.gl/maps/r2EJuj1KivV2HmYj9 | 09:00 | 22:00 |
| Weihnachtsmarkt am Michaelerplatz | https://goo.gl/maps/vyGZaJK2gnvRvje98 | 10:00 | 20:00 |
| Schloss Wilhelminenberg | https://goo.gl/maps/LYSYyQ6LfiwkFGXq8 | 11:00 | 20:00 |
| Advent in der Stallburg | https://goo.gl/maps/tiqojzWBSU2VVfTw9 | 11:00 | 20:00 |
| Adventmarkt Mariahilf | https://goo.gl/maps/KhmGod5UH9k7GFYp9 | 09:00 | 20:00 |
| IKEA Westbahnhof | https://goo.gl/maps/fV8e9wMTG611ZJTZ6 | 14:00 | 20:00 |
| Christkindlmarkt Floridsdorf (am Franz-Jonas-Platz) | https://goo.gl/maps/cRwood7CfKZVjyCe8 | 09:00 | 21:00 |
| Adventmark Verkehrsmuseum Remise | https://goo.gl/maps/6PJ6FzGiNYF7LQun7 | 14:00 | 21:00 |
| Weihnacht im Wald | https://goo.gl/maps/UaQ2izUcyZzfHr1GA | 16:00 | 21:00 |
| Adventmarkt Favoriten | https://goo.gl/maps/5a92CE7tc1PdRnVo6 | 10:00 | 20:30 |
| Winterzauber im Böhmischen Prater | https://goo.gl/maps/u6Smhr98B1iztBrg8 | 11:00 | 20:00 |

| | | | |
|---|---|---|---|
| Adventmarkt Meidling | https://goo.gl/maps/pVgbJNCYryCeyaRt8 | 10:00 | 20:30 |
| Ottakringer Weihnachtszauber | https://goo.gl/maps/X2yJEAZA8HF1c5Y86 | 11:00 | 22:00 |
| Adventmarkt im Schloss Neugebäude | https://goo.gl/maps/SVhtiYyTX69Fnqw97 | 10:00 | 20:00 |

## Seidel Rally Vienna

The “Seidel Rally” has the goal to drink one drink (typically a Seidel, i.e. a 0,3l beer) in each of Vienna’s 23 districts during one day. Besides being a challenge in terms of the amount of liquid to drink, it is also a problem of logistics. Thus, similar to the travelling salesperson and the travelling Christmas market visitor (above), you shall optimise the route for visitor.

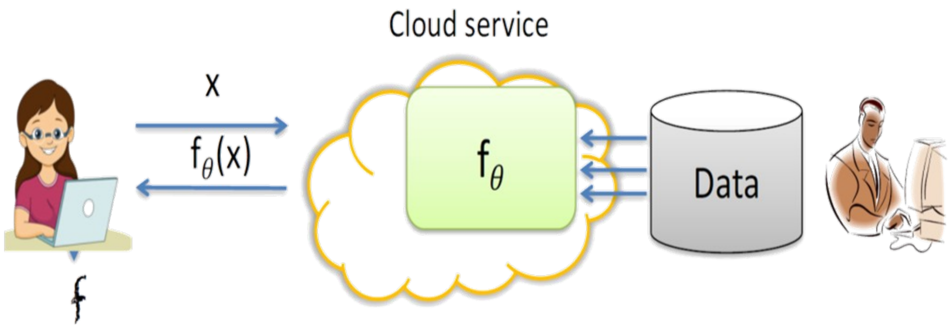
In this example, you shall pick 3 possible locations for each district (can be randomly) - so in total you will have 69 locations - and then optimise the route in a way that each district is covered exactly once. While an issue in real settings, you can ignore opening hours of the locations.

You can either follow a specific sequence of the districts (23 → 1), which means you primarily optimise the routes between the possible locations in adjacent districts, or arrange them by overall best time in any sequence. Just as a side constraint, in any case, the final destination should be in one of the locations in the first district.

For deciding on the route time/distance, you shall, similar to the Christmas market setting, use the API of a routing service; you must not drink & drive, so the only options are walking and public transport (taxis are too expensive :-) ).

You shall describe your design, coding and solution, together with an analysis of the results, in a report.

## Model Stealing/Extraction using evolutionary algorithms



Model "stealing" means to obtain a trained model from a source that generally only provides a "prediction API", i.e. a means to obtain a prediction from the model when providing a specific input, but does NOT disclose the actual model (i.e. "ML-as-a-service"). This might be because the model may be deemed confidential due to their sensitive training data, commercial value, or use in security applications. A user might train models on potentially sensitive data, and then charge others for access on a pay-per-query basis.

In model stealing/model extraction, an adversary with black-box access (but no knowledge of the model parameters or training data), aims to duplicate the functionality of the model. Important aspects for stealing is the data used to query (in general: the closer to the original / the more represantive, the better), and the number of queries, where we want to reduce the queries to be less likely to be detected or to lower the costs. Several strategies thus look at optimising the selection of which input data to send to the model, to lower the required amount of queries to obtain an eventually equivalently well working model,

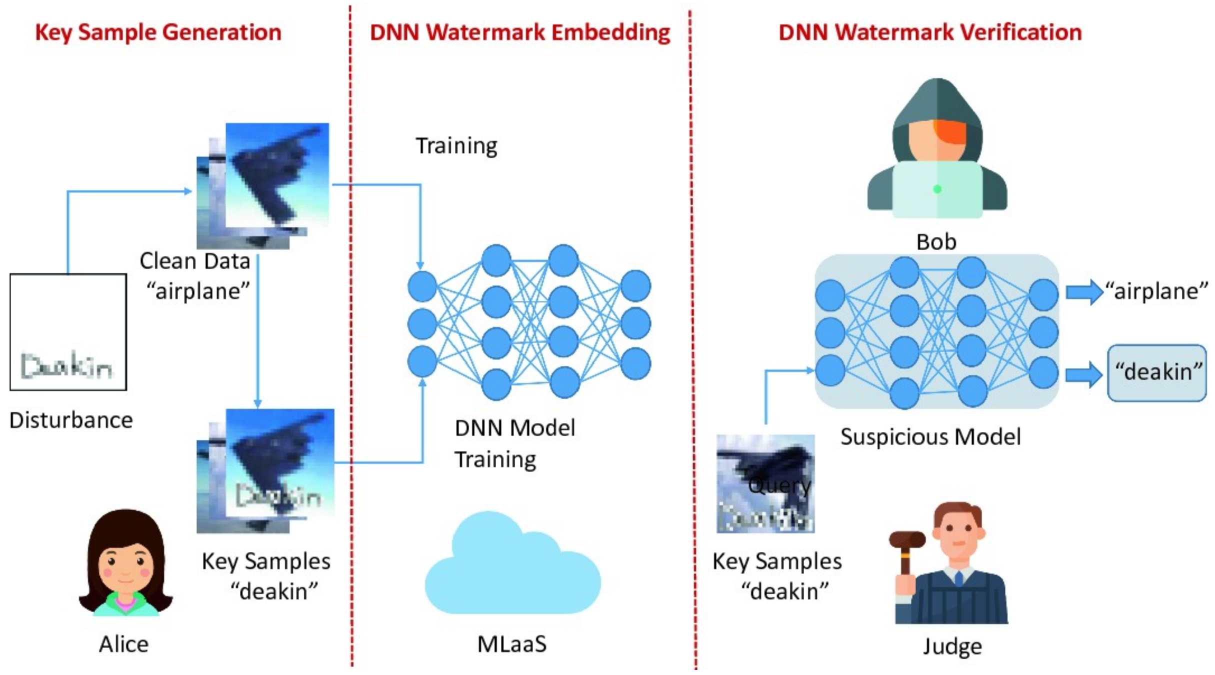
In this task, you shall recreate experiments from the paper „Black-Box Ripper: Copying black-box models using generative evolutionary algorithms" (https://dl.acm.org/doi/abs/10.5555/3495724.3497413). This paper utilizes an evolutionary algorithm for picking the most promising samples for performing a model stealing attack. These samples are used to query a target model (the model one wants to steal), and using samples with returned outputs to train a substitute model (an illegal copy model). The main goal is to train the substitute model to reach similar performance as the target model. Task details:

- Use CIFAR10 dataset
- Repeat the procedure described in Algorithm 1 from Section 3 to create a dataset. You can pick one generator model (either VAE or GAN)
- Train a substitute model using this dataset
- Train another substitute model using randomly selected samples (the dataset should be the same size as the one created using the evolutionary algorithm)
- Compare performance of substitute models obtained using the evolutionary algorithm and random sample selection strategy

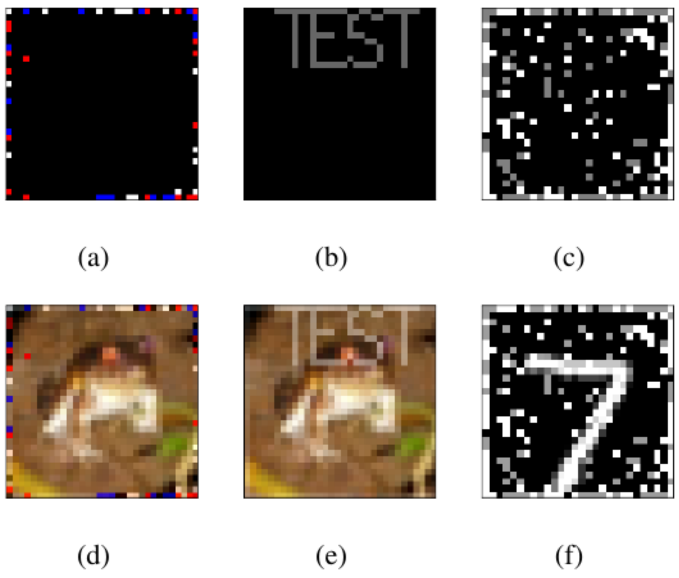
You can use the code published by authors (https://github.com/antoniobarbalau/black-box-ripper), as well as target and generator models they trained (but the substitute should be trained by you)

## Evolutionary Trigger Set Generation for DNN Black-Box Watermarking

Watermarking machine learning models is mostly done by creating a so-called backdoor on a specific model, i.e. having a specific pattern that will trigger unexpected responses from the model, as shown in the figure below.



One recent approach uses evolutionary algorithms to find an optimal pattern resp. an optimal position to superimpose such a pattern on the image for these triggers, with the goal to reduce the number of false positives, i.e. images without these patterns triggering the unexpected response, such as the pixel-patterns below in (a) and (c) and the logo pattern in (b) in the image below.



Your task is to re-implement the approach from https://arxiv.org/abs/1906.04411 to find positions that will be optimal. We will provide you with a pre-trained model (alternatively, you can use any pre-trained model yourself).

## Optimal subset of Anonymisation solutions

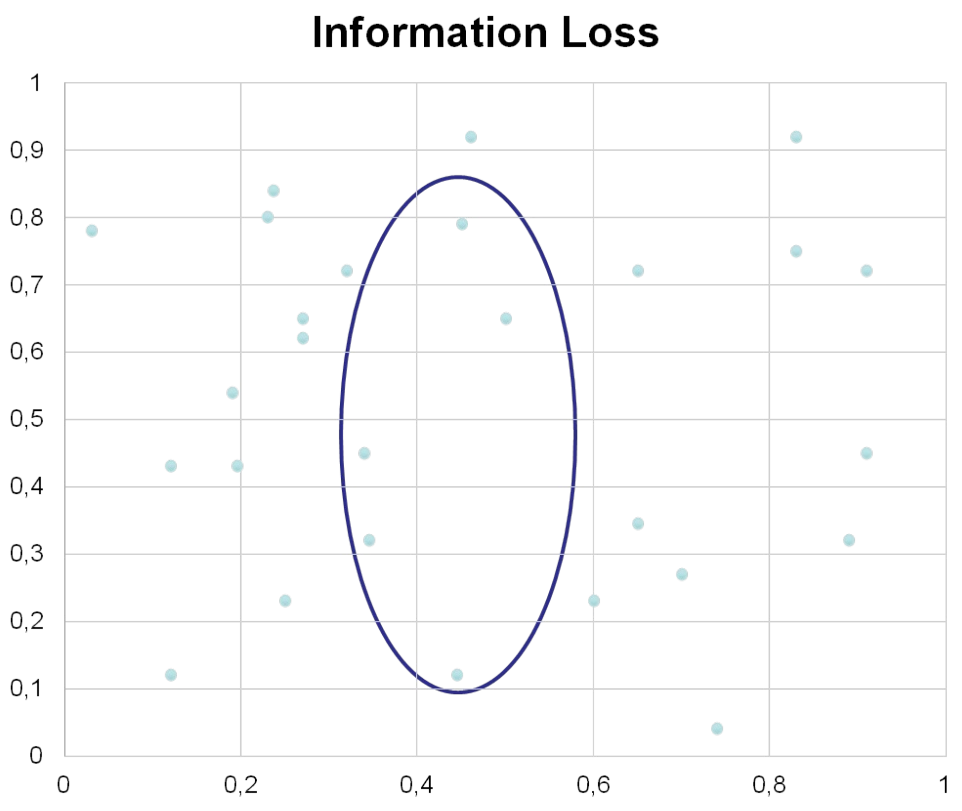
k-anonymity is a property of a dataset where the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appear in the release. For the same level of anonymity (k), the original dataset can be anonymized in multiple ways. The anonymous datasets are obtained by generalization or suppression of data values, therefore the utility of the anonymized dataset is expected to decrease - this may be measured via information loss metrics. When searching for the optimal anonymous solution, one would need to minimize a chosen information loss metric.

| Name | Birthdate | ZIP code | Sex | Disease |
|---|---|---|---|---|
| * | 01/1976 | 53 7** | Male | Flu |
| * | 04/1986 | 53 7** | Female | Hepatitis |
| * | 02/1976 | 53 7** | Male | Bronchitis |
| * | 01/1976 | 53 7** | Male | Broken Arm |
| * | 04/1986 | 53 7** | Female | Sprained Ankle |
| * | 02/1976 | 53 7** | Female | Hang Nail |

| Name | Birthdate | ZIP code | Sex | Disease |
|---|---|---|---|---|
| * | 01/21/1976 | 53 7** | * | Flu |
| * | 04/13/1986 | 53 7** | * | Hepatitis |
| * | 02/28/1976 | 53 7** | * | Bronchitis |
| * | 01/21/1976 | 53 7** | * | Broken Arm |
| * | 04/13/1986 | 53 7** | * | Sprained Ankle |
| * | 02/28/1976 | 53 7** | * | Hang Nail |

Assume now a scenario where the data owner wants to anonymize their data and distribute the anonymized releases to different parties such that every party receives a different release. Besides the general goal of minimizing information loss, the goal is that the parties receive “similarly good/useful” datasets, so that the distribution is fair. The number of the recipient is not known, however, the more recipients we are able to distribute the data, the better.

The solution space may look as in the figure below:



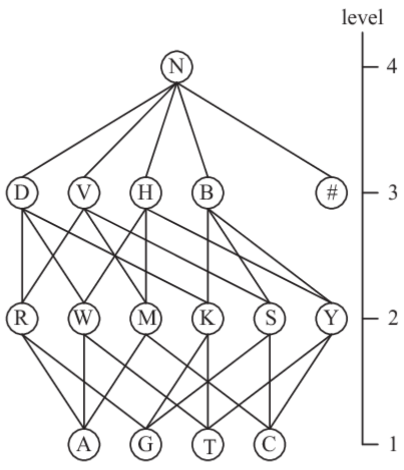
Two information loss metrics are presented on x- and y-axis, and each dot presents one k-anonymous dataset. The task is to find the set of solutions that would satisfy the mentioned criteria via Genetic Algorithm/Agent-based system/…. You may think of the variables to include in the cost function that would fit this problem. These may, for example, include:

- Average information loss of the subset
- Variance of information loss within a subset
- Size of the subset
- …

## Sequence alignment for Genetic Data (DNA Lattice) Anonymization

The approach of DNA Lattice Anonymization (DNALA) has first been introduced in [1] and is applied to databases of DNA sequences. DNALA provides the possibility to prevent identification of DNA. For each DNA sequence, the technique finds **the most similar sequence** in the database and generalizes both to a common sequence. The distance between the sequences is computed according to the DNA generalization lattice in the figure below.

| | |
|---|---|
| A = **A**denine | C = **C**ytosine |
| G = **G**uanine | T = **T**hymine |
| R = pu**R**ine | Y =p**Y**rimadine |
| S = **S**trong hydrogen | W = **W**eak hydrogen |
| M = a**M**ino group | K = **K**eto group |
| B = not A | D = not C |
| H = not G | V = not T |
| # = gap | N = i**N**determinate |



For example, Adenine (A) and Cytosine (C) generalize to Amino Group (M), as can be seen by following the only arrows from A and C pointing to the same letter in the next level of the lattice. The function *gen(x,y)* returns the distance in the lattice between *x* and *y* and is defined by

$$gen(x,y) = 2level(z) - level(x) - level(y),$$

where *z* is the value *x* and *y* generalize to. E.g., we have *gen(A,C) = 2*2 – 1 – 1 = 2.* Subsequently, this function is used to define distances between whole DNA sequences.

As the simple algorithm has a complexity of $O(n^3)$, using a heuristic approach for solving the sequence alignment problem is desired. More efficient methods work on solving a form of the maximum-weight-matching problem on graphs (https://en.wikipedia.org/wiki/Maximum_weight_matching).

**Goal of this assignment:** Implement the core method on DNALA in [1] and its improvements. In particular, focus on understanding the original method described in Section 3 and the improvement of the DNA sequence clustering described in Section 6. The problem is similar to an Assignment problem, and a special form of maximum weight matching on graphs. Thus, finally, implement an algorithm for the solution of the maximum-weight-matching problem in a programming language of your choice. Use the melanocortin.fasta file provided in TUWEL as a test dataset

**Primary reading:**

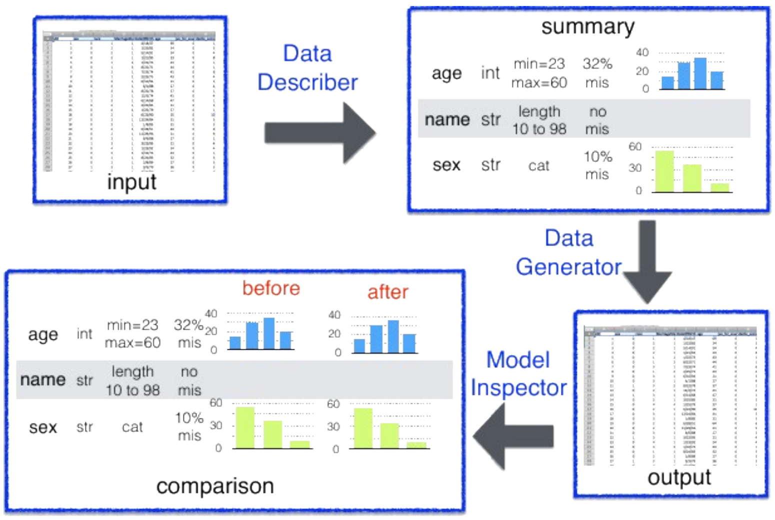
[1] B. A. Malin. *Protecting genomic sequence anonymity with generalization lattices.* Methods Inf Med., vol. 44, no. 5, pp. 687-692, 2005. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.2227&rep=rep1&type=pdf

**Further readings (optional):**

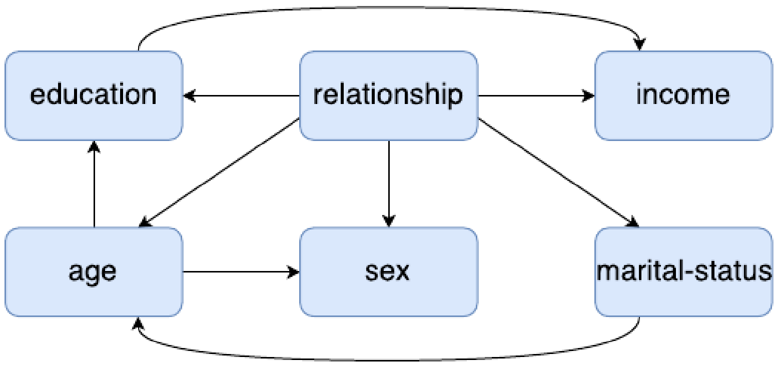
[2] G. Li, Y. Wang, X. Su. *Improvements on a privacy-protection algorithm for DNA sequences with generalization lattices*. Computer Methods and Programs in Biomedicine, vol. 108, no.1, pp. 1-9, 2012. https://www.sciencedirect.com/science/article/abs/pii/S0169260711000459?via%3Dihub
[3] S. Wan, M. Mak, S. Kung, *Protecting Genomic Privacy by a Sequence-Similarity based Obfuscation Method*. 2017. https://arxiv.org/abs/1708.02629

## Implement a heuristic for Bayesian Network search

Synthetic data is an approach to publish a dataset that keeps the global characteristics of an original dataset (such as distributions of attributes, correlation, …), but does not contain any actual data. This is beneficial when publishing when the data is sensitive, such as data on individuals (medical records, …)



A effective method for generating synthetic data is provided by the DataSynthesizer (https://github.com/DataResponsibly/DataSynthesizer), which learns a Bayesian network to represent data and generate from.



Unfortunately, the method for learning a Bayesian network does not scale well to a larger number of attributes.

Thus, your task is to adapt the existing solution (https://github.com/DataResponsibly/DataSynthesizer/blob/master/DataSynthesizer/DataDescriber.py) with a heuristic (except Genetic Algorithms, unless you implement a different approach than already existing) for finding an optimal network structure.