Assignment 3: Logistics Network Design

Data-Driven Decision Making

José Pedro Pinto - up201603713

Introduction

Optimization problems arise every day in almost every aspect of life. They range in complexity from what would be considered intuitive to large scale intractable problems, for which no exact answer can be reasonably obtained. These can be seen, for example in personal finances, where we try to decide how, where and when to apply capital. In marketing, where we try to do the correct amount and type of marketing, to obtain the best profit. Or in personal life, where we decide how much of a product to buy at a time, to balance, convenience and risk of spoilage or waste.

In this project we are going to tackle a particular case of optimization, facility location problems. There is a large variety of these, which range in complexity. For our particular case, we have three main components, airports, distribution centers (DCs) and locations.
We have three airports, which are always open, and which supply our DCs. With the cost increasing, the further away the airport is from the DC.
We have DCs, whose placements coincide with the locations, supplying said locations, once more, with the cost increasing with distance. There is a limit to the amount each DC can distribute also a cost for each open DC (those that transport something).
Finally, we have the locations, for which the items will be transported. Each has its own demand, which must be fulfilled.
We as such, want to reduce the cost, characterized by sum of the cost of opening the facilities, the cost of air transportation and the cost of land transportation.

Implementation

Our approach was extremely focused on the optimization models, with only very simple and lightweight visualization and preprocessing. Several models were created, some being evolutions of previous ones, with only advantages, while others being variants, with different strengths and weaknesses.

Preprocessing

The original data we received was about cities and towns in Portugal, containing city name, total populations, longitude and latitude. This is not our intended format, as such a small amount of preprocessing was performed. First, the city names were changed into an unique numerical identifier, then, the population was transformed into the total demand, and finally, latitude and longitude were transformed to the distance in Km to an origin point (point of minimum latitude and longitude in the data).

Exact models

Base model

We started the project by doing a very **simple base model**, which is consistent with our demands and would in theory be capable of obtaining the best solution. The model only possesses two constraints. The capacity constraint, which forces the DCs not to distribute more than their capacity. And the demand constraint, which forces all demand from al cities to be supplied. The model also contains a set of parameters, the number of airports, number of facilities, number of locations, demand of each location, air transport costs and land transport costs.

From all the parameters, the air and land costs are the most difficult to obtain. The distance between pairs of locations is calculated as the Manhattan distance (the sum of the vertical and horizontal distance), as opposed to the more common Euclidean distance. These distances are calculated for the pairs of airports and DCs, creating an I by j matrix, with I being the number of airports and j the number of DCs. The land cost is calculated in a similar way, for all pairs of DCs and locations. As the placement of DCs coincides with the locations, it creates j by j symmetric matrix. Finally, the distances are multiplied by the cost per Km for air and land transport respectively.

This model was very **inefficient**, with very **low scalability**, not being able to obtain results for datasets with over 100 locations.

First optimized model

Given the limitations of this model, we started work on an **optimized one**. **The first optimization** came at the realization that, under no circumstance, would the best solution have air transport from one airport to a DC such that the distance was not the minimum for all airports. As such, we were able to combine both costs, adding the best air transport cost per unit to the land cost, creating a single matrix. **The second optimization** is a restriction on the number of open facilities. As the demand must be always be satisfied, we know that the number of open facilities must be at least enough to cover said demand. On the other hand, we can also know a transport cost that is necessary equal or worse to the best solution, by delivering using the costliest route. Given that amount, we can obtain an upper limit on the number of facilities, by restricting it such that the total facility cost, does not exceed this cost, for as long as the demand can be satisfied. This constraint is consistent with the optimal solution, as if the facility cost exceeds the value, then it means, as it is worse than the best obtainable value, that we can reduce the number of facilities without increasing the total cost.

The combination of the land and transport cost was also used for all the models presented after this one.

This model was **much better** than the base one, with much **better scalability**, being capable of obtaining results up to 250 facilities. However, was **still not good enough** to solve our problem.

Approximated models

Having tested a large number of possible optimizations, without any noticeable improvements to performance, we decided to try some approximated approaches, were the best results were not guaranteed, or in fact likely, but similar results were.

Greedy model

The first of these approximated approaches was a **greedy search** model. In this model, we select **one facility at a time**, forcing its total capacity to be used, getting at each time the facility and transports that minimize the cost. After obtaining each facility and transports, we **remove the facility** from the list of possibilities, **by greatly increasing its transport cost**, and reduce the demand for the supplied locations. Afterwards we repeat the search. When the demand in no longer greater than that of a facility capacity, we proceed to get the final facilities using the first optimized model.

As we were obtaining one facility at a time, the constraints for the number of facilities present in the optimized model were replaced by a constraint forcing the number of facilities to be one.

This model presented very similar, however slightly worse solutions to those of the optimized, exact model, however with an extreme increase to performance, being capable of obtaining solutions for the totality of the data.

Multiple greedy model

In an attempt to obtain better results than those obtained with the greedy model, we modified it to create **a multiple facility greedy model**, meaning, that at each step, we obtain the best n facilities. This is done, at the level of the model, by replacing the constraint of the number of facilities, to the intended amount. Once again, we obtain the n facilities at a time, remove the selected ones from the data, remove the demand and run the model again on the new data. The most important change is that, when there is **not enough demand for the number of facilities**, we obtain the rest with the **first greedy model.** Once there in not enough for one facility, we once again use the first optimized model.

Transport model

To further explore the problem, we **modified it to a transport problem**. A transport problem is very similar to the facility one, where there too are DCs that supply and locations that have demand. The main difference between the location and the transport problems is that in the transport problems there is **no cost to open a facility**.

This model was as expected **very fast**, even for the **total amount of data**, the preprocessing taking longer. This solution gives us what the best possible cost would be for the transport. As there is no land cost for the transport of a DC to its own location, the obtained cost, represents the air costs.

Final model

With the insights and solutions obtained from the previous models and methods, we proceeded to create our final model. Much like the first two, an exact model, capable of always obtaining the best solution. This second optimized model introduces over the first one two new constraints and changes one. The first constraint is over price, with it having to be **between the lower bound**, obtained by adding to the **transport cost** the cost for the minimum number of facilities, and an **upper bound**, obtained by the **multiple greedy cost.** The second new constraint is in the amount transported, being that for each facility, the transport to a customer cannot exceed capacity. The changed constraint is the bounds on the number of facilities, the lower bound remains, however, the upper bound is now defined to be the amount of facilities that would exceed the greedy cost.

This new model managed to obtain results for the full dataset, being much faster than the first optimized model, however, still being much slower than the approximated models. It took about 2 hours to run on the machine used.

Results

Having managed to obtain the best solution using the new optimized model, we now present the results obtained and the solution.

The total cost obtained was of 1088021.668 while for the greedy models was of 1160875, not a large difference.

The selected DC locations were **Lisbon, Porto, Cacem, Barreiro, Monsanto, São Bras de Alportel, Joane e Sao Roque**.

Finally, the town with the largest delivery cost was Coimbra, with a cost of 41808.26.

There were some problems with rounding errors during optimization. As these are quite small, there is very little chance that they would incur in a change of the selected facilities. However, the costs and amounts transported might be slightly off the real value.

The runtimes for each of the models and dataset sizes are as follows. For the base model, **1 minute** for the first **75 locations**. For the first optimized model, **1 minute** for the first **100 locations**. For the greedy model, **3 minutes** for the **total dataset**. For the multiple greedy model **15 minutes** for the **full dataset**. For the transport model, less than **a second** for the **full dataset**. And finally, for the second optimized model, **2 hours** for the **full dataset**.

Conclusions

The presented problem is, at first look, surprisingly complicated. With such a simple optimization problem to define and a low amount of data, we would expect to have no difficulties. However, of further inspection, the true nature of the problem arises, as an exponential problem of high degree.

Finding the best solution for this problem is extremely difficult, and with a higher amount of data might not be feasible.
Three of the tried approaches have their usefulness, with different strengths and weaknesses.

The first of these, greedy search, is capable of obtaining results quite close to the optimum in a very low amount of time and great scalability. For problems of higher dimension, this one might be the best bet.

The second one, multiple greedy search, is placed in the middle of the three models, presenting overall better results, but also worse performance, taking substantially longer. It is also not as scalable as the first model.

The third one, always obtains the best possible answer, however, this comes with a high computational cost.

For this particular problem, in an industrial setting, two hours to compute, in order to increase the profit of margin even slightly is perfectly acceptable, with the last model being the best one. However, with an increase in data sizes, quickly would this model have to be changed to the second one, and later to the first one.
The first two models are of particular use, when the costs, demands, etc. are only estimates, and we would be interested in trying different scenarios, allowing a broader range of scenarios to be tested quickly, allowing overall for better decisions to be made over what little improvement in accuracy the final model would give.