Optimization of a public transport network with link prediction
Final Project Report

Ruben Bousbib
rbousbib@gmail.com

John-Evan Karcenty
john-evan.karcenty@student.ecp.fr

Figure 1: Construction of the Paris Metro in 1902-1910.

ABSTRACT

The construction of public transport networks is often gradual with for example almost a century for the construction of the Parisian metro. One may wonder whether these networks, which many people use every day, are adapted to the needs of the average traveler. If we could redo the entire infrastructure today, could we find a more efficient network? Less expensive to build? In this work, the objective is to build a subway network by performing link prediction given the geographical position of the stations. We place ourselves in a context where initially, no link is present between the nodes. Therefore, classical unsupervised techniques such as neighborhood-based methods or proximity-based methods are hardly applicable. Instead, we try to perform link prediction in a more combinatorial fashion where the notion of neighborhood is not directly used. Instead of calculating similarities between nodes to predict a potential link, we are more interested in the utility of a link with respect to a score that we want to maximize. We formulate this objective as an optimization problem on a graph space and propose a heuristic to predict relevant links. Finally we compare the obtained graph with the real Parisian subway network.

KEYWORDS

graph theory, transport networks, link prediction

ACM Reference Format:

Ruben Bousbib and John-Evan Karcenty. 2018. Optimization of a public transport network with link prediction Final Project Report. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Graph theory is a field widely studied by scientists because it allows the modeling of problems as varied as complex. It is of particular interest in the study of transportation networks because of the economic stakes involved. These networks are created with a view of improving the flows specific to globalization (trade flows for example), they must therefore be efficient.

Nevertheless, their construction requires extremely costly works that must be carefully studied beforehand; the search for an efficient network at the lowest cost is therefore a major issue in urban planning. For example, the Grand Paris Express project aims to build four new automated metro lines around the capital. In total, 200 kilometers of network and 68 stations will be built by 2030, at an estimated cost of 25 billions euros.

The starting point of this work is a simple question : Given the positions of several subway stations within a city, how should we connect them to result in an effective transportation network with a reasonable cost ? This question calls for a first comment : Are the positions of the subway stations not variables that should be also optimized instead of being given data ? In fact, this could be another problem. However, here we consider that they are given because they represent popular places where people want to go anyway (business districts for example). Therefore, subway stations

Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

represent passengers' demand that the network operator must satisfy. Another comment would be, what do we mean with "effective network" ? In this work we define the average travelling time of a network that encapsulates this idea of effectiveness and intuitively, one may observe that for the traveller, the most effective network is the complete graph over the stations.
Unfortunately, this network is also the most expansive to build. This tells us about the trade-off that should be found between reducing the building cost and reducing the average travelling time. Optimizing both would be a very complex multi-objective optimization problem. Instead, we set a constraint over the building cost and try to optimize the average travelling time. Our problem is almost identical to what is called in the scientific literature the Network Design Problem, which is NP-complete [1]. We propose a heuristic that involves to start from a minimum spanning tree. It will give us one of the less expansive admissible graphs. Then, the idea will be to improve the average travelling time by performing carefully designed link prediction to add good edges without exceeding the cost constraint.

The plan of this study is the following : first, we formalize the problem as an optimization problem over graphs and we describe our objective. Then, we present our heuristic and the results. Finally, we give some guidelines for further research and other heuristics.

2 RELATED WORK

The problem we study is part of the more general framework of network design problems. A network design problem (NDP) is to select a subset of links in a transport network that satisfy passengers or cargo transportation demands while minimizing the overall costs of the transportation. It is one of the most challenging transport problems and is defined as follows. In the past decades, various approaches have been presented to address this issue. The solutions can be divided into two categories: exact solutions [1], [2] and heuristic solutions [3], [4].

Exact solution methods can deal with NDP in a rigorous manner. However, they are inefficient when dealing with large-scale real-world networks. Heuristic approaches, emerged in the past decades and provide approximate yet efficient solutions. The heuristic approaches can tackle a real-world problems with a large number of design variables and therefore these approaches are more popular than exact solutions. Many heuristic approaches are biologically inspired as living species have optimized their transportation networks over millions of years with evolution: the vascular systems of plants and animals, the foraging patterns of social insects, the migration paths of birds and animals, the hunting routes of predators. It is therefore often very fruitful to apply natural solutions to the design of man-made objects.

As an example, [5] uses a bio-inspired heuristic inspired from the Physarum polycephalum to compute an efficient network. It is a unique creature which exhibits properties of internal and external living transport systems and which creates a protoplasmic network to cover the food areas it is interested in and in a surprisingly optimal way.

On the other hand, the application of link prediction techniques to transportation networks is not new. Similar problems have already been studied with for example the optimization of an air transport network [6].

Finally, when we design a transportation network, we should ideally make it fault tolerant, capable of handling traffic accidents, terrorist attacks, and urgent road maintenance. Fault tolerance and high performance lead to higher construction and maintenance costs. However for simplicity purpose, in this study, we do not focus on making the network fault tolerant as in [5].

3 PROBLEM DEFINITION

The efficiency of a transport network is most often measured by the ability to move easily through the network, i.e. to get from point A to point B in a suitable time. In order to give a concrete meaning to the mathematical objects we manipulate, we will take the example of a subway network but this is also applicable to any type of transport network.

Let $G = (V, E)$ be a graph with V denoting the set of stations and E the set of tunnels connecting them. We define the global building cost of the graph G as :

$$C(G) = \sum_{(i,j)\,\in E} l_{ij}\alpha$$

where l_{ij} is the tunnel length between stations i and j and α is the cost for a one kilometer tunnel.

For each station i, we associate the traffic f_i that represents the number of travellers that come in the subway through i over a year. We suppose that the number of travellers coming out from i is the same. Then $\frac{f_i f_j}{F^2}$ is the probability for a traveller to go from station i to station j with F denoting the total amount of travellers using the network over a year :

$$F = \sum_{i \in V} f_i$$

Finally, we define the average travelling time of G as :

$$T(G) = \sum_{(i,j)\in V^2} \frac{\delta_{ij}}{S} \frac{f_i f_j}{F^2}$$

where S denotes the average speed of a subway wagon and δ_{ij} the distance of the shortest path between i and j in G. If G is not connected, then $T(G)$ is infinite.

The operator must satisfy travellers' demand at best i.e to supply the fastest possible network while being consistent with stations traffic. However, he doesn't have an unrestricted budget. Let us call R the maximum building cost that the operator allows himself.

Then, the problem comes down to :

$$\begin{aligned} \min_{G} \quad & T(G) \\ \text{s.t.} \quad & C(G) \leq R \\ & G \text{ connected} \end{aligned}$$

The notion of efficiency of a graph is thus quantitatively defined and measured by the value of $T(G)$. It is this one that we try to optimize and that we will use to perform the link prediction.

In this definition of the problem, we place ourselves in the case where no initial graph is given to us, only the set of stations. More precisely, we start from the initial graph $G_0 = (V, E_0)$ with $E_0 = \varnothing$ and try to add new tunnels to E_0 to achieve a good $T(G)$.

4 METHODOLOGY

4.1 Data collection

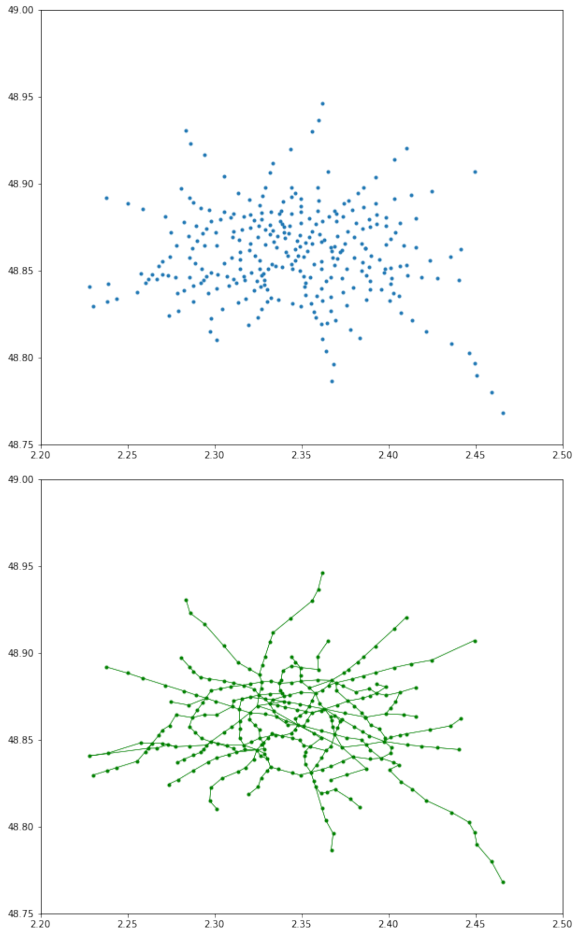


Figure 2: Paris today's network.

The goal of this work is to build a transport network from the geographical position of the stations to be served. In order to evaluate our link prediction method we want to compare the obtained graph with a real network. To do so, we have collected the geographical position of the Parisian metro stations as well as their associated traffic for the year 2014 on the website `https://data.ratp.fr`. Fig 2 shows the position of the stations and the actual Parisian network that connects them.

4.2 Initial graph

In our problem there is no initial structure given to us to perform link prediction. In our opinion, this is more consistent with the real situation where the transport company wants to build from scratch the best network possible to meet passenger demand. This is different from the assumption used in standard link prediction problems that use the neighborhood of nodes to predict the existence of new links as it is the case in neighborhood-based methods for example. The main idea of these techniques is to assign a similarity score to a pair of nodes based on the number of common neighbors. The higher this score is, the more likely there is a link between the two. Here, the goal is not to predict links using a pre-existing structure, but to find an optimal global structure from scratch.

Thus, we need to find an initial graph from which we could predict the most interesting links. We could create a k-NN graph using the position of the stations, only we would then have to proceed to the deletion of irrelevant tunnels which is not straight forward. To find such a graph, we must take into account the two optimization constraints of our problem. Namely, the connectivity constraint as well as the cost constraint. It happens that we can compute a connected graph that is optimal from the builder's point of view, i.e. that solves the following problem :

$$\begin{array}{ll} \min\limits_{G} & C(G) \\ \text{subject to} & G \text{ connected} \end{array}$$

This graph is a minimal spanning tree over the set of stations and we can obtain it via Kruskal's algorithm [7] which sorts the set of possible links between each station according to length and add them in increasing order if they don't create a cycle.

Algorithm 1 Kruskal's algorithm

```
Input: $L$ is a sorted list of all possible edges between stations.
Ouput: $A$ is a minimum spanning over the stations.
  $A \leftarrow \emptyset$
  $C \leftarrow \emptyset$ //Connected components, C[x] denotes the components of the node x.
  for $(u, v) \in L$ do
    if $u \notin A$ and $v \notin A$ then
      Add $(x, y)$ to $A$
      Add $[x, y]$ to $C$
    else if $x \in A$ and $y \notin A$ then
      Add $(x, y)$ to $A$
      Add $y$ to $C[x]$
    else if $x \notin A$ and $y \in A$ then
      Add $(x, y)$ to $A$
      Add $x$ to $C[y]$
    else
      if $y \notin C[x]$ then
        Add $(x, y)$ to $A$
        Concatenate $C[y]$ and $C[x]$
      end if
    end if
  end for
  return A
```

4.3 Link prediction heuristic

Once the initial graph is obtained, we have a structure that ensures connectivity and an optimal building cost. However, this graph is one of the worst acceptable graphs from the traveler's point of view because there are no cycles, thus forcing unnecessary detours. Indeed, as can be seen in Figure 3, the tree structure means that some stations that are very close are not easily connected. Adding a tunnel between them, especially if they are highly popular places,

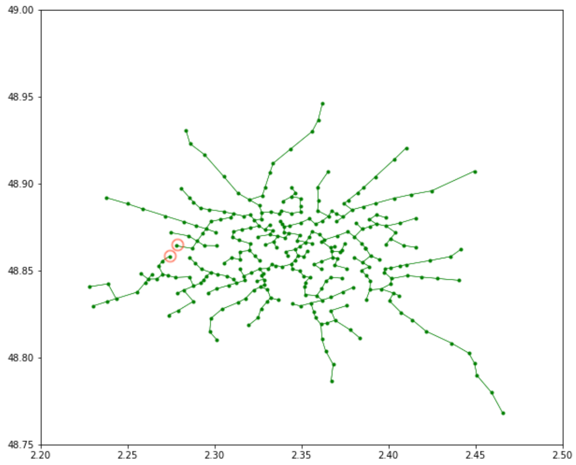


Figure 3: Minimum spanning tree.
Red stations are very close but the path that connect them is very long. We might want to predict a link here.

would be very useful and it is this type of connection that we are trying to predict.

Moreover, two subway stations having many neighboring stations in common have no reason to be connected. Indeed, this would mean that there is already a very short path between the two. Neighborhood-based and proximity-based methods are then not relevant here.

Therefore instead of computing a similarity score between two nodes we rather compute the utility score of a pair with respect to an objective function to optimize, in this case $T(G)$. More specifically, we define the *utility score* of the pair (x, y) as

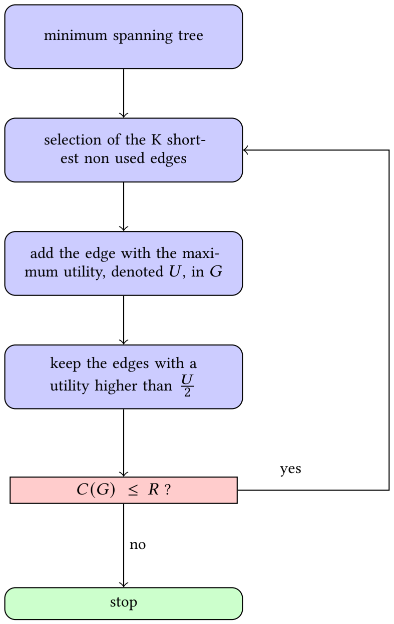
$$c(x,y) = \frac{T(G) - T(G \cup (x,y))}{l_{xy}\alpha}$$

This value denotes the ratio between the time saved by adding (x, y) to G and the corresponding additional building cost. As we want an edge to increase $T(G)$ and to be short, we should add edges with the highest utility.

To compute $c(x, y)$ we need to compute the average travelling time $T(G)$ of a graph $G = (V, E)$ and therefore we must quickly access to the length of the shortest path δ_{xy} for each pair of stations $(x, y) \in V$. An idea could be to use the dynamic programming Floyd-Warshall algorithm [8]. However, the complexity of this algorithm is $O(|V|^3)$ and it takes it takes about 10 minutes to compute the distances between all pairs of vertices on our computer. Therefore, we will not be able to quickly iterate our heuristic because at each step, computing $T(G)$ would be too long.

To get around this problem, we use a property of the graphs we are dealing with : they are sparse which means that they have very few edges. Then using Dijkstra's algorithm [9] for each source node is more effective because the complexity is $O(|E|.|V|.\log(|V|))$. On the same computer, we are now able to compute all the distances in few seconds.

However, there are more than 45.000 possible edges over the 302 stations ($\frac{|V|(|V|-1)}{2}$). So we can't just compute $T(G)$ for each of them because it would be too long and we want a heuristic that works for higher $|V|$ than 302. To address this issue, we propose the following simple heuristic :


minimum spanning tree
selection of the K shortest non used edges
add the edge with the maximum utility, denoted U, in G
keep the edges with a utility higher than $\frac{U}{2}$
$C(G) \leq R$?
yes
no
stop

The idea of this meta-heuristic is to select among the K shortest possible edges, the one with the best utility score and to eliminate the ones that are short but not very useful i.e. that don't increase much $T(G)$.

By iterating this process until reaching the desire maximum building cost R, we are able to transform the minimum spanning tree into an effective network that takes into account both the operator's and the travellers' objectives. Of course, the more we increase R, the more effective is the network at the end. Thus, the choice of R is up to the operator who will choose considering his budget.

5 EVALUATION

In order to evaluate our link prediction approach, we compare the computed graph on Paris subway stations to the real network in terms of average travelling time.

We run the heuristic with $K = 10$ and $R = 25$ billions € which is the cost of the real Parisian network.

After some research on subway networks, we choose α =120 million € per kilometer and $S = 25$ km/h (taking the stop into account by averaging).

6 RESULTS

Fig 4 show the graph obtained after adding 75 tunnels to the initial graph. Table 1 show the comparative results on the initial, computed and real network.

By observing the predicted link in red, one can see that some shortcuts were added between stations that were geographically close but not well connected. By performing our heuristic on top of a minimal spanning tree, we are able to compute a network almost as efficient as the real Parisian subway and at the same cost.

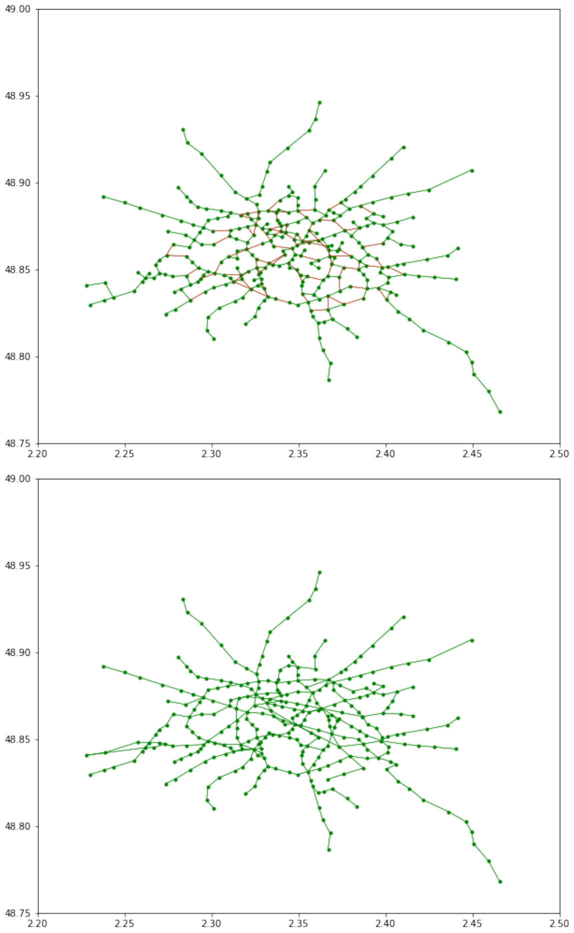


Figure 4: On top : computed network with $K = 10$ and $R = 25$ bil. € Red links are the one predicted by our heuristic. Below : the real Parisian metro as a comparison.

This shows that the modeling of our problem, the definition of the average travelling time as well as the utility score we introduced are relevant to create congestion resilient transportation networks.

	Spanning tree	Heuristic network	Paris network
Total length	154 km	208 km	209 km
$C(G)$	18,518 md	25,018 md	25,234 md
$T(G)$	23,00 min	15,44 min	15,27 min

Table 1: Building cost and average travelling time of the different graphs.

7 CONCLUSION

In this work we have described the problem of constructing a transportation network as a constrained optimization problem on a graph space. This approach is part of the Network Design Problems. Starting from the minimum spanning tree, we succeeded in founding a network almost identical to the real Parisian subway. We could iterate the heuristic with a higher maximum building cost to find an even more effective network but it would take more time to run. Our heuristic could be used for other graph optimization problems where there is a notion of average travelling time. For example, one can imagine a computer network problem where the operator wants to connect servers in an effective but not too expansive manner.

A first difficulty of the problem is that we have placed ourselves in the case where no initial structure is provided; the operator must build the network from scratch from the geographical positions of the stations and the associated traffics. This required us to find an initial structure that can respect the connectivity and cost constraints, here a minimum spanning tree.

A second difficulty is that we cannot treat this problem in an exact way nor use classical link prediction methods based on neighborhood or proximity information which were not relevant for our problem. Thus we propose a link prediction heuristic based on the spanning tree. To do so, we define a utility score for a given pair of stations and with respect to our objective function.

The results obtained and the comparison with the real Parisian metro show the relevance of our approach.

8 FURTHER IMPROVEMENTS

Several improvements are to be considered. First, the computation time is still relatively long because of the need to obtain all the shortest paths of the graph on a regular basis. Secondly, the minimum spanning tree structure as the initial graph may not be the best one because we directly try to minimize the cost of the operator. A multi-objective and more complex approach could surely be considered by seeking to optimize the average travelling time from the start. This tree structure also makes it difficult to use stochastic methods such as random walks because of the non-existence of cycles in the graph. Finally, a last avenue of improvement would be to perhaps consider that the operator is trying to improve an already existing transportation network. Supervised link prediction techniques could certainly be interesting for this purpose.

REFERENCES

[1] R. Dionne and M. Florian. Exact and approximate algorithms for optimal network design. *Networks*, 9(1):37–59, 1979.

[2] V. Gabrel, A. Knippel, and M. Minoux. Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letters*, 25(1):15–23, 1999. ISSN 0167-6377.

[3] Falko Dressler and Ozgur B. Akan. Bio-inspired networking: from theory to practice. *IEEE Communications Magazine*, 48(11):176–183, 2010. doi: 10.1109/MCOM.2010.5621985.

[4] Chia-Feng Juang. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2):997–1006, 2004. doi: 10.1109/TSMCB.2003.818557.

[5] Xiaoge Zhang, Andrew Adamatzky, Felix Chan, Yong Deng, Hai Yang, Xin-She Yang, Michail-Antisthenis Tsompanas, Georgios Sirakoulis, and Sankaran Mahadevan. A biologically inspired network design model. *Scientific Reports*, 5, 05 2015. doi: 10.1038/srep10794.

[6] Yue Zheng, Wenquan Li, Xi Cao, and Chunyang Sun. *Optimization of Air Transportation Network Using Link Prediction Methods*, pages 991–1000.

[7] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[8] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, June 1962. ISSN 0001-0782.

[9] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271, December 1959. ISSN 0029-599X.