An Approach to Determine Public Facilities Placement: a Case Study based on Canberra Bus Network

Yinheng,Chen
u6341895@anu.edu.au

Shidong,Pan
u6342277@anu.edu.au

June 2, 2019

1 Introduction

Public facilities placement is a significant consideration of city arrangement, especially for high technology establishments which are generally expensive, error-prone but greatly convenient, such as ticket vending machines for bus, sharing bike docks or electronic vehicle charging stations etc.

Canberra is a rapid developing capital city. With the increasing of population, the pressure of public transportation system greatly surges as well. In the Planning of future part of Transport for Canberra Policy[9], it mentions “A frequent public transportation network supported by services, planning, infrastructure, land supply and location of facilities.” Therefore, a practical distribution plan of new public establishments is necessary for the future development. Specifically, in this project, we will focus on arranging the Ticket Vending Machine (TVM) in Canberra bus network by our method and compare it with other common distribution plans. On the Canberra public transport website, TVM is a touch screen based and user-friendly machines. Passengers are able to conveniently recharge their MyWay card instantly and check the card balance on a TVM.

Our methods and evaluations are based on two main attributes of networks: communities and robustness. Reflecting to the reality, the study on communities can ensure our model will cover as many people as possible. As for the analysis of robustness, it would help the government or local companies to provide better and more stable public facilities to citizens by simulating some real conditions such as, machines randomly broken down, traffic jam and bus routes randomly outage.

A network can be divided into some groups of nodes with dense connections. In our case, people live in different communities, to help people access to public facilites in their own neighbourhood, it is necessary to consider the community structure when we set public facilities placements. [6] Radicchi et al.

In the real situation, because of some uncertain effect such as public transportation outage, bad weather (random failure) even stealing (attack), public establishments are likely to become unreachable or damaged. In consideration of this issue, an analysis of robustness of the distribution pattern appear to be particularly important.

2 Related Work

2.1 Public facilities distribution on large-scale network

Efficiently distributing public facilities has always been a heat topic. For the selecting electric vehicle public charging station locations problem, it can be modelled into an optimization model by maximizing electrified vehicle-miles-travelled for potential environmental benefits and public charging demand, based on large-scale vehicle trajectory data. [7]Shahraki et al.

Furthermore, according to [8] Sonneberg et al. , to address the conflicting goals of fulfilling demands and maximizing profit when deciding the sharing electric car stations locations, factors like distributed demand, revenue for renting, expected driving distance user leasing cost, average energy consumption etc. should be considered to achieving the optimization.

2.2 Relationship between population dense and distance of bus stops

In the past, some interesting works have been conducted on public transportation stations placement selecting. [4]Fitzpatrick et al., who developed a strategy for allocating bus stops, conducted a review on bus stops plan that considers bus patron's convenience, safety and access time and the efficiency of transit operations. Apart from that, [10]Ziari et al., introduced a mathematical model that is formulated to optimize the locations of bus stops to achieve the minimum total travel time in consideration of population density and passenger's behaviours data. He also indicated that the distance between bus stops or number of stops within the areas can reflect the population density of the areas.

2.3 Robustness on removal of edges

In [1]Barabasi's book Network Science, he mentioned that "Enhanced robustness is not limited to node removal, but emerges under link removal as well" and shows that the threshold f_c of link removal and node removal to destroy the giant component are about same.

In reality, even if a bus stop is out of service, those routes which include the bus stop are still on service. Therefore, removing nodes seems has slight impact on the robustness of bus network in reality, removing edges is more realistic to affect the robustness of bus network since a route will be cut off if any trip in the route is removed.

3 Dataset

3.1 Dataset acknowledgement

We use a adapted version of Canberra bus network. The original public transport schedule data has been provided by ACT Government, Transport Canberra and City Services (Canberra). The adapted versions of the schedule data have been been created by Rainer Kujala, Christoffer Weckström and Richard Darst. In the dataset, the nodes represnet the IDs of bus stops and the edges represnet trips between nodes.

3.2 Preprocess to the Dataset

3.2.1 Importance of merging nodes

The original dataset seperates opposite bus stops. However, opposite bus stops usually have same routes and features and same routes should not appear in the network multiple times. Besides, some stops are very closed, each of them can reach others by just a few of steps. We should aviod setting mutilple facilites in a very small ares. Therefore, we need to merge nodes. The another advantage of merging nodes is that we can decrease the size of our network and make our approach more efficiently.

3.2.2 Method to merge nodes

Our method to merge nodes is very simple but efficient, we calculate distance of two nodes to determine whether these two nodes should be merged or not. The steps of our method are

1. We calculate the distance among nodes(bus stops) by their longitude and latitude, If the distance between two nodes is less than 100 meters(maybe others), we merge them. The new node incident to any edge that was incident to the original two nodes, and the location of new node is the weighted average longitude and latitude of original two nodes.

2. Repeat step 1 until we cannot merge any pair of nodes.

3.3 Properties of the network

The following table shows the some basic properties of the network:

Features	Original	After merging
NO. of nodes	2520	1523
NO. of edges	2908	1976
Averge degree	2.31	2.60
Max. degree	10	16
Min. degree	1	1
Diameter	51	44
Raidus	26	22

The Canberra bus network is like a scale-free network, many bus stops only in a single route and each of them only connects two other stops, while a few number of nodes are in mutilple routes. As we can see, the average degree of original network is just over 2 and the max degree is 10. Some nodes which are in the end of routes have 1 degree.

As we expected, after we merged the nodes of the original network, the size of network decrease approximately 40%. Since we merged some closed nodes, the averge degree and maximum degree of the network increases, while the minimum degree s unchanged. In the new network, many nodes still keep their degree of 2 (figure 1). Therefore, our method of merging nodes is sufficient.

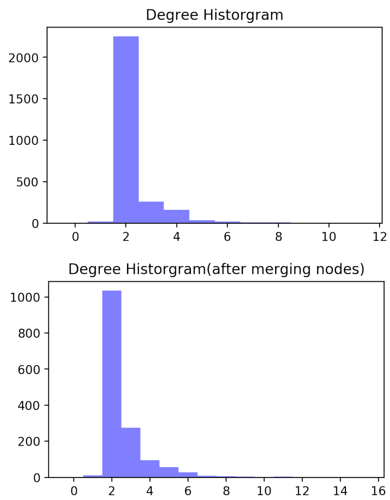

Figure 1: Historgram

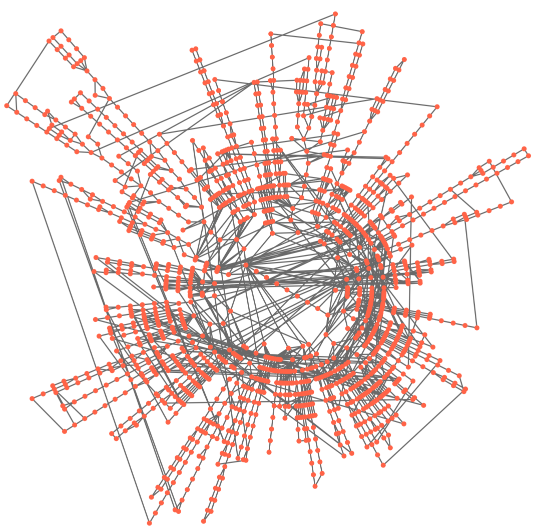

Figure 2: Network topology

4 Algorithms and methods

Our target is to find a distribution of facilites on stops which has the shortest average shortest distance between an arbitrary node and the nearest machines. We measure the number of stops as the unit of distance and we treat our networkas unweighted and undirected.

4.1 Summary of some methods

Initially, we have four methods to distribute stops:

1. Random distribution
2. Distribute facilites to those nodes have high degree
3. Optimized solution
4. Greedy

Obviously, the first method is not effcient because it is random. At the beginning, we expected the second method has a good result. However, the second method represnets its disadvantages when the number of facilites increases in our simulation. The main problem of the second method is that bus network show a good homophily with respect to degree. For many nodes with degree of 2, it is unlikely for them to directly connect nodes with high degree.

As for optimized solution, we assume the network has n number of nodes, m number of edges and we want to assign b number of facilites. The total number of selection will be $\binom{n}{b}$, for each selection, we apply Breath First Search(BFS) to find the shortest distance for all nodes to the nearest facilites and then we compute the average shortest distance, in the worst case, the time complexity is $O(n*(n+m))$. So, the total time complexity of optimized solution is

$$O(\binom{n}{b}*n*(n+m)) \tag{1}$$

It is a NP-hard problem, since we have more than 1000 nodes in the network, this method is discarded.

4.2 Greedy Algorithm

In this section, we will introduce our first greedy algorithm to dertermine the distribution of facilites.

Before we introduce our algorithm, we would like to show our definition of the average shortest distance. We assume the shortest distance for an arbitrary stop i to the nearest machine is S_i, if there is no path for a stop to reach a machine, we assign S_i as the diameter of graph. So, for all stops $i \in N$ and all machines $m \in M$, $d(i,m)$ means distance between stop i and machine m, we have

$$S_i = \begin{cases} 0 & i \in M \\ min(d(i,m)) & (\exists m(d(i,m))) \\ diameter(G) & otherwise \end{cases} \tag{2}$$

so, the average shortest distance for all nodes to their nearest machine in the network is

$$A = \frac{\sum_i^n S_i}{n} \tag{3}$$

This algorithm is very simple and intuitive. The steps are:

1. Everytime we assign a machine to a stop which can minimize the current average shortest distance.
2. Repeat step 1 until all machines are assigned.

Analysis of time complexity:
Since we want to assign b machines, the total number of selections will be

$$n + (n-1) + (n-2) + \cdots + (n-b+1) \tag{4}$$

In each selection, the time complexity for an arbitrary node to find its nearest machine by using BFS is $O(m+n)$. So, the total time complexity for each selection is $O(mn+n^2)$. Then, the total time complexity by using greedy algorithm will be:

$$\begin{aligned} &O((n+(n-1)+\cdots+(n-b+1))*(mn+n^2)) \\ &\approx O(bmn^2+bn^3) \end{aligned} \tag{5}$$

A method to improve the time complexity of greedy algorithm is that we can first use BFS to find all shortest distance for all pairs of nodes and use an appropriate data structure store all paths instead of applying BFS mutilple times in each selection, this spends $O(mn+n^2)$. So, in each selection, the time complexity for an arbitrary node to find its nearest machine will be current number of machines, to simply our calcualtion, we assume it is the total number of machine $O(b)$, then the total time complexity for each selection will be $O(nb)$.

By using the method, the total time complexity of greedy algorithm is

$$\begin{aligned} &O(mn+n^2+bn*bn) \\ &\approx O(b^2n^2) \end{aligned} \tag{6}$$

4.3 A More Efficient Greedy

The time complexity of the first greedy that we introduced is $O(b^2n^2)$, compared to optimized solution, this algorithm has a much faster speed. However, one problem for this greedy is that when we need to assign a great number of facilites, or the network becomes very large, it still runs slowly. To have a better efficiency, we will introduce our second greedy algorithm, which is based on community detection.

There are many community detection methods, we have applied greedy modularity maximization which was introduced by Clauset, Newman and Moore in 2004 [3] and Louvain community detection algorithm which was introduced by Blondel et al in 2008 [2]. In our model, these two algorithms have similar performance. In this section, we will focus on greedy modularity maximization.

4.3.1 Greedy modularity maximization

In a network, a community is defined as nodes are more likely to link nodes in their communities rather than to nodes in other communities. We use modularity [3] to measure the strength of divison of a network into communities. The definition of modularity is

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \tag{7}$$

which was introduced by Clauset, Newman and Moore in 2004 [3]. In [3][5], the idea of Greedy modularity maximization is quite simple, we start out with assigning each vertex to a one-vertex group, and then we merge groups in pairs, choosing at each step the merging of pairs that can maximize the modularity. Repeat the above step until we cannot find any pair of group whose amalgamation can increase the modularity. In many cases, the algorithm run essentially linear time with time complexity $O(nlog^2n)$ by using appropriate data structure.

4.3.2 Applying greedy algorithm in communities

The conception of this greedy algorithm is very simple, we first devide graph into communities, and we treat each community as a sub-graph, then we apply greedy algorithm in each community. One advantage is that the size of each sub-graph is much samller than the whole graph. Assuming we have a distribution with a great number of facilites which can minimize the average shortest distance between am arbitrary stop to the nearest machines in the whole graph, the distribution can also extremely decrease the average shortest distance for am arbitrary stop to nearest machine within each community. Similarly, if some nodes can minimize the average shortest distance within their own communities, their distribution in the whole network can extremely decrease the average shortest distance in the whole network (if most communities are assigned machines).

We have similar definitions of some variables for this greedy algorithm. Assuming we have a q number of communities $c_1, c_2, \cdots c_q$ and their corresponding sub-graphs are $sg_1, sg_2 \cdots sg_q$. For a stop i in a community and for all machines $m \in M$, we have

$$S_i^{c_j}(i \in c_j) = \begin{cases} 0 & i \in M \\ min(d(i,m)) & (\exists m(d(i,m) \wedge \\ & m \in c_j)) \\ diameter(sg_j) & otherwise \end{cases} \tag{8}$$

Then, the average shortest distance for an arbitrary stop to its nearest machine within the community becomes

$$A_{c_j} = \frac{\sum_i^{n_j} S_i^{c_j}}{n_j} \tag{9}$$

The steps of the algorithm are:

1. Using a community detection method to divide network into communities.
2. First we assume that the number of machines is greater than the number of communities of the best partition.
   (a) We first assign one machine to each community.
   (b) In each community (c_j), we assign a machine to a stop i ($i \in c_j$) which can minimize A_{cj}.
   (c) Once step (a) finished, then we assign a machine to a community which has the least dense of machines by step (b), in our model, we define it as $\frac{NO.(m)}{NO.(n)}$, where $NO.(m)$ is the current number of machines in the community and $NO.(n)$ is the number of nodes in the community.
   (d) Repeat step (c) until all machines are assigned.
3. If the number of machines is less than the number of communities (although it is unlikely to occur for most real networks), we just assign machines to those communities which have the largest size.

Analysis of time complexity:
Assuming the best partition for the network has q number of communities, and all communities have same size. We have

$$\frac{n}{q} + \frac{n}{q} + \cdots + (\frac{n}{q} - 1) + (\frac{n}{q} - 1) + \cdots \approx \frac{bn}{q} \tag{10}$$

number of selections. In each community, we have $\frac{n}{q}$ number of nodes, than the time complexity for an arbitrary node to find its nearest machine by using BFS becomes $O(\frac{m+n}{q})$. Then, for each selection, the time complexity becomes $O(\frac{n}{q} * \frac{m+n}{q})$.

So, in the best case, the total time complexity of this greedy is:

$$\begin{aligned} & O(\frac{bn}{q} * (\frac{n}{q} * \frac{m+n}{q})) \\ & \approx O(\frac{bmn^2 + bn^3}{q^3}) \end{aligned} \tag{11}$$

We can use the same method in previous section to find all shortest paths for all pair of nodes and use an appropriate data structure to store the length of all paths. So, in each selection, the time complexity for a node to find its nearest machine within the communities becomes $O(\frac{b}{q})$, the total time complexity for each selection becomes $O(\frac{b}{q} * \frac{n}{q})$.

Then, in the best case, the total time complexity is

$$O(n*(m+n)+\frac{bn}{q}*(\frac{b}{q}*\frac{n}{q})) \approx O(\frac{b^2n^2}{q^3}) \quad (12)$$

5 Evaluation and Experiment

Our simulations running environment is: macOS, version:10.14.3, CPU 2.3GHz Intel Core i5, Memory 8GB 2133 MHz LPDDR3. All tests are implemented on the Canberra bus network (after preprocessing). For four methods that are reachable, their running time are:

Methods	O	Runtime
Random	$O(b)$	< 1s
High degree first	O(n lg$n$)	<1s
Greedy	$O(b^2n^2)$	48 mins
Greedy on communities	O($\frac{b^2n^2}{q^3}$)	3s

The table clearly shows the distinct difference on computational time complexity and real runtime of those algorithm. As we can see, by implementing greedy on communities, the algorithm runs approximately 1000 times faster than previous greedy, and it does not run much slowly than random and high degree first.

In the Algotirhm section, we discussed that the average distance to the nearest machine can be regarded as the measurement of the quality of distribution patterns. In Figure 3, the orange line denotes the average result of 100 times simulation of Random distribution. The rest results are only based on one simulation because they are decidable. It clearly demonstrates that if we arrange a number of TVMs in this network, Greedy and Local Greedy (Greedy on communities) have a much lower average distance to the nearest machine, which means those two distribution patterns are much more better than the others. Thus, Random and Degree first distributions will be discarded in the following analysis of robustness because of their poor performance.

In reality, it is not rational to only arrange a small number of TVMs on stops, since it is unlikely to meet citizens' needs. As we can see, when we arrange more than 40 TVMs, the gap between the results of Greedy and Local greedy is very slight, which means these two greedy algorithms have similar results if we arrange rational number of TVMs.

5.1 Simulations

Generally speaking, robustness is a remarkable ability of a structure to sustain basic functions even when some of its components fail. Specifically, network robustness refers to the impact of after random deletion and selective deletion of nodes or edges [1] in the network. To test our model by considering the robustness of the network as well as the robustness of our algorithms, we have simulated some rational situations by implmenting failure of machines or edges in the network. Each of foliowing failure situation reflects to one situation in the reality:

1. Randomly remove machines from nodes can be regarded as TVMs randomly broke down

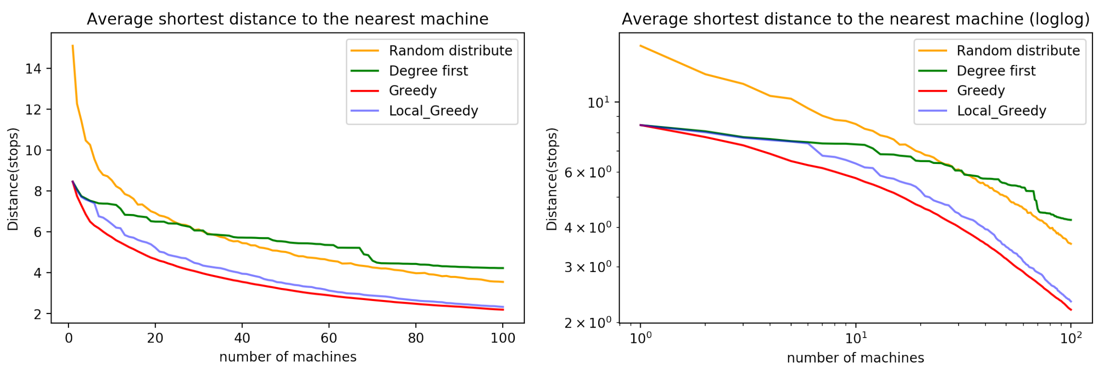

Figure 3: Benchmark

or in maintenance

2. TVMs at stops with high degree are more likely to be unavailable because of a long queue or tending to have problem due to high using frequency;
3. Random edge failure represents randomly road work or road close by reason of accident
4. High betweenness edge failure corresponds the traffic congestion at rush hour

We notice that when 60 TVMs are arranged in the network, the average shortest distance for the Greedy and Local Greedy is about 3 stops which is quite rational in reality; therefore, the analysis of simulations will stand on this situation.

5.1.1 Failure of machiens

Figure 4 illustrates the results of simulation of removing machines from nodes. The failure probability of machines is from 0 to 0.25, higher failure probailities of machines is unlikely to occur in reality, each point in each line has 500 times simulations. The average shortest distance is over 3.8 for both blue line (Local Greedy with faliure situation 1) and green line (Local Greedy with failure situation 2) when the failure probability comes to 0.25, showing the decent resilience toward Random node failure. However, we see a different trend emerging, with the increasing of failure probability, the gap between oragne line and greent line narrows, which indicating that Local Greddy algorithm is more robuts with respect to faliure situation 2.

5.1.2 Removal of edges

In this section, we will test and compare performance of two greedy algorithms under the failure situation 3 and situation 4.

We first test two greedy algorithms under the failure situation 3, random edge failure represents randomly road work or road close by reason of accident. In reality, the situation of most roads closed or most trips of buses outage is unlikely to occur, so we will randomly remove edges with low probability of 1%,3% and 5% in our

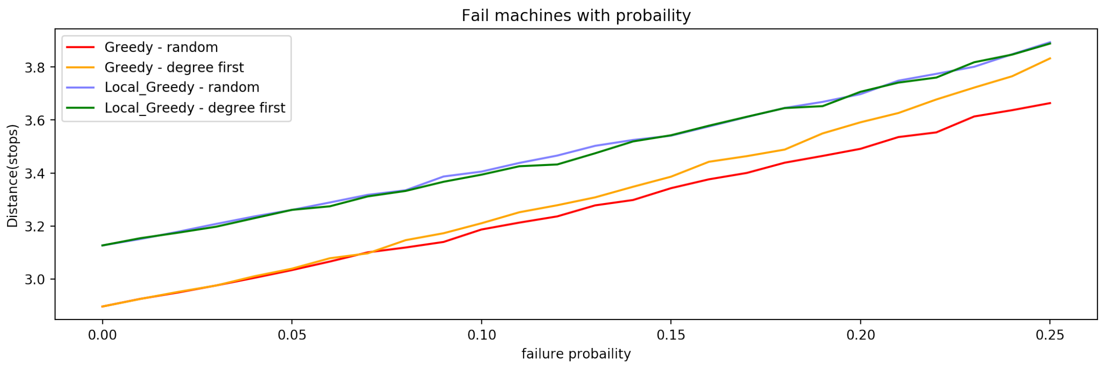

Figure 4: Failure of machines

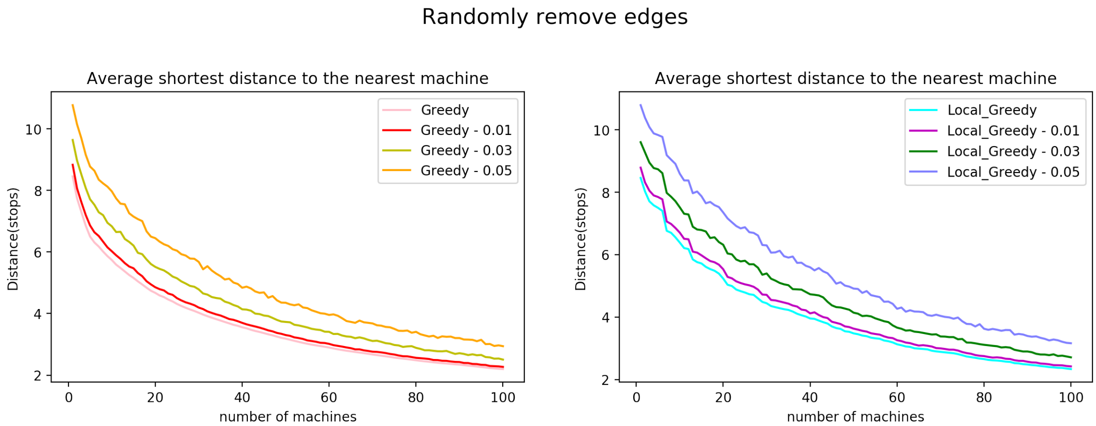

Figure 5: Randomly remove edges.

simulations.

The result of figure 5 are the average number of 300 times simulation. In the figure, it demonstrates the Greedy and Local greedy both have same shape and similar results with same probability of removing edge, therefore their distribution pattern robustness after random remove edges are basically same. When we randomly remove 1% edges in the network, which is rational in reality, the average shortest distance of two algorithms almost keep unchanged.

Edges with high betweenness can be considered to be more important than other edges. These edges can represent busy roads or trips of buses with high population flow, these roads and trips are likely to have outage at rush hour, we will test and compare two algorithms under this situation.

Figure 6 also shows the average number of 300 times simulation. We only focus on the part where the number of machines is greater than 40 because analysing a few number of machines is impractical and meaningless. It exhibits both algorithms generally have a same trend toward attacking on high betweenness edges. Both two algorithms show their robustness when we have rational number of TVMs and remove 1% edges with highest betweenness, and the gap between lines of Local greedy is slightly narrower, which means that it has a slightly better robustness.

We also simulated an extreme situation of this scenario: What if the giant component vanishes? The fraction of nodes or edges to remove before the giant component vanishes is:

$$f_c = 1 - \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle^2} \tag{13}$$

Based on the equation and the definition in [1] Barabasi's book Network Science, we calculate that we need to remove approximately 54% of edges to destroy the giant component in Canberra bus network.

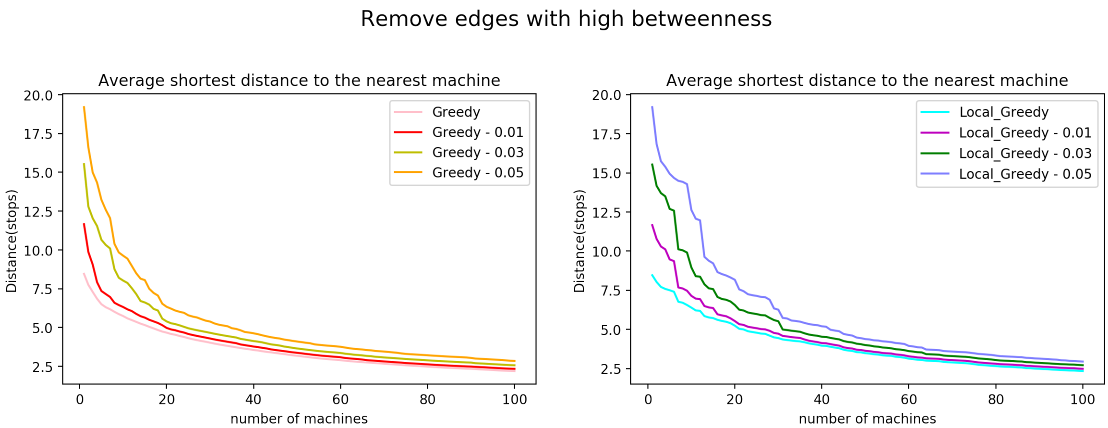

Figure 6: Remove edges with high betweenness

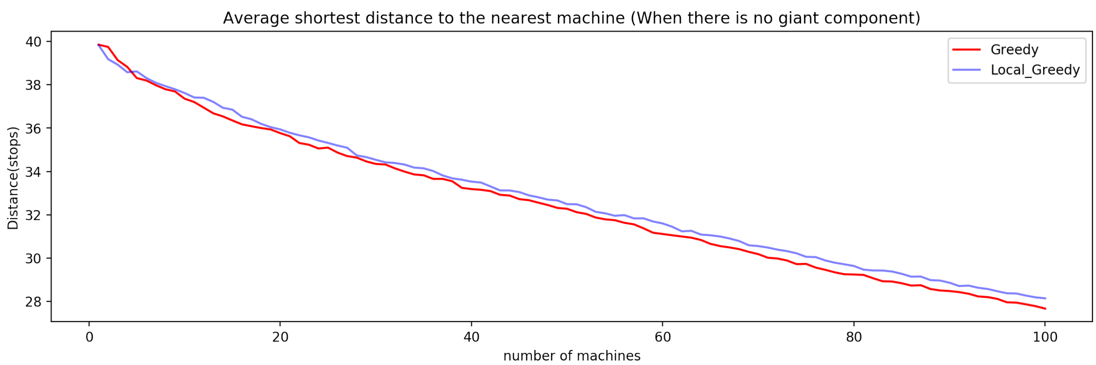

Figure 7: No giant component

In Figure 7, the performance of Greedy and Local greedy is illustrated. We can clearly see that basically they have similar results under this extreme situation. In addition, even we have a great number of TVMS, most people still cannot reach a TVM in the bus routes when the giant component vanished because Canberra bus network has a low robustness.

5.2 Solution

The end goal of this research is to provide a robust TVMs distribution plan in Canberra bus network in an efficient way for future placements plan making.

In Figure 8, we demonstrate one sample solution of distributing 60 TVMs in Canberra bus network by using Local greedy algorithm.

Intuitively, this solution seems rational enough because machines cover almost all suburbs of Canberra, no matter where the people live in Canberra, it is convenient for them to find a TVM nearby, even some remote suburbs are placed machines. We can also see many suburbs center has at least one machine, which is realistic. In the solution, some suburbs/areas of Canberra has no machines, this is mainly because there are few bus stops in this suburbs/areas, people live in these suburbs/areas rarely take bus, so we do not need to arrange TVMs in these areas/suburbs.

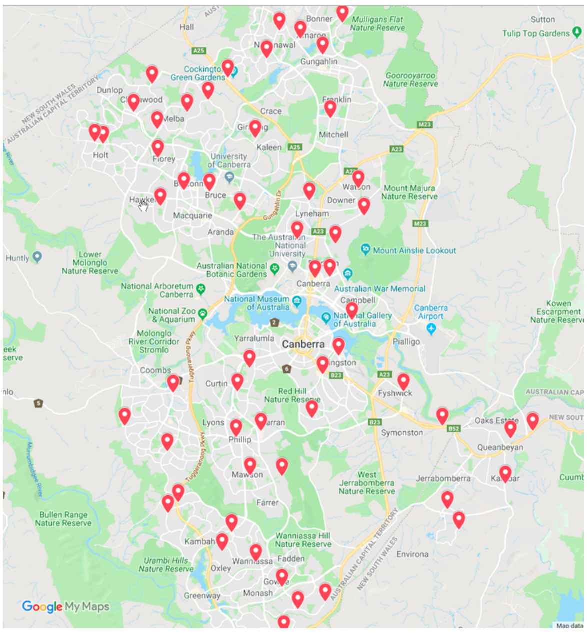

Figure 8: Solution

6 Conclusion

Public facilities placement is very important for city plan, Canberra is now developing more light rail and bus routes, finding a distribution of TVMs and appropriate number of TVMs is necessary. There are many more similar problems such as distribution of sharing bicyles, public service station, mobile power bank and etc. Although these problems have different domains, they can all be considered as facility distribution problems and problem for finding target by the average short path in the network.

Finding a optimized solution for facility distribution problem is impossible, since it is a NP-hard problem and there are many features should be considered. Thus, we can use greedy algorithm to find a reliable solution if the network has a few of features. However, for some larger network or larger number of simulations, greedy algorithm is also costly.

Since greedy algorithm is costly in some cases, we need to find a more effcient algorithm to solve the problem. One method is that we can relax the problem, for example, we can find a solution in local area of the network first. Using communities structure is powerful technique to improve intuitive greedy algorithm. By using greedy algorithm on communities, we have a similar result compare to result of intuitive greedy algorithm if the number of facilites is rational. However, greedy algorithm on communities is much faster than intuitive greedy algorithm, it also shows some slight advantages based on analysis on robustness of the network and robustness of solution.

As we can see, local greedy algorithm shows us a rational solution for distribution of TVMs in Canberra network, which is expected.

7 Future Work

Currently, we simply preprocess our dataset by merging nodes based on the distance among nodes. Some better preprocess methods are expected. We can have a better threshold of distance to merge nodes. Beside, although distance of many trips between stops are similar, there does exist some longer or shorter trips between stops, a better method is that we can give edges weight.

More TVMs are expected to be set in important stops to resolve the overloading problem, which means we can grant weights to nodes depending on their importance.

Although the density of bus routes and stops indirectly reflects the population density, our method can be improved by combining real and accurate people flow data to get more rational distribution plan, which can improve our method of determining which community is expected to be assign machine first in each step. For example, a community which contains bus stops in airport should have high priority.

8 Contribution

This project is conducted by a two-member group. The topic initially is a little bit ambuguous, and then we narrowed the domain down to a case study on Canberra bus network.

The personal contribution of our project is:

Yinheng Chen: Problem proposal; collecting and preprocessing dataset; algorithms and methods design and implmentation; evaluation; conclusion; future work; formatting the report.

Shidong Pan: Problem proposal; presentation; introduction; related work; evaluation; future work; checking the gramma error and polishing the report.

We would like to thank the COMP4880 teaching staff for instruction during lectures and tutorials and helpful feedback to our project propsoal and presentation.

The code of our project is on:
`https://gitlab.cecs.anu.edu.au/u6341895/comp4880-project---public-facilities-placement./tree/master`

References

[1] Barabási, A.L., 2016. Network science. Cambridge university press.

[2] Blondel, V.D., Guillaume, J.L., Lambiotte, R. and Lefebvre, E., 2008. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10), p.P10008.

[3] Clauset, A., Newman, M.E. and Moore, C., 2004. Finding community structure in very large networks. Physical review E, 70(6), p.066111.

[4] Fitzpatrick, K., Perkinson, D. and Hall, K., 1997. Findings from a survey on bus stop design. Journal of Public Transportation, 1(3), p.2.

[5] Newman, M., 2010. Networks: an introduction. Oxford university press.

[6] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. and Parisi, D., 2004. Defining and identifying communities in networks. Proceedings of the National Academy of Sciences, 101(9), pp.2658-2663.

[7] Shahraki, N., Cai, H., Turkay, M. and Xu, M., 2015. Optimal locations of electric public charging stations using real world vehicle travel patterns. Transportation Research Part D: Transport and Environment, 41, pp.165-176.

[8] Sonneberg, M., Kühne, K. and Breitner, M., 2015. A decision support system for the optimization of electric car sharing stations.

[9] Transport.act.gov.au.(2019).[online].Avaliable at: `https://www.transport.act.gov.au/__data/assets/pdf_file/0008/887255/Public_Transport_EDS_ACT_Transport_Policy_FA_final_web.pdf` [Accessed 31 May. 2019].

[10] Ziari, H., Keymanesh, M.R. and Khabiri, M.M., 2007. Locating stations of public transportation vehicles for improving transit accessibility. Transport, 22(2), pp.99-104.