

Efficiency Simulation for Bus Traffic in Taipei City

Tsung-Hsien Wen
r00921033@ntu.edu.tw

Po-Chun Hsu
r00921034@ntu.edu.tw

ABSTRACT

For newcomers in Taipei, the bus system there is hard to figure out since there are too many bus lines. In addition, some of the lines are tortuous and make travellers spend much more time than expected. Consequently, travelling by bus might be inefficient for those who are not experienced in taking bus in Taipei. From another perspective, this kind of bus network is energy-consuming as well. A chessboard-like bus network has been thought to be a solution to these problems for many years, although it has not been promoted successfully in Taipei. We thus want to know whether a chessboard-like bus network works better in Taipei. We set up a simulation to observe what are the pros and cons of a chessboard-like bus network and the current bus network. We encoded a part of roads in Taipei into a graph. Two types of agents, buses and clients, are placed in the simulation environment. In order to simulate the behaviors of buses, we record the routes and intervals of buses from the Taipei e-bus website. We also set up different scenario types to simulate client behaviors at different time points of a day. The simulation results suggest that a chessboard-like bus network is more efficient in terms of the resource utilization. Even though the data are still limited so real-world clients is hard to model, this simulation is an initial attempt to model the bus activities in Taipei and gives us some insight on future policies of public transportation.

1. INTRODUCTION

In Taipei City, which is the city with its total area around 271.7997 square kilometers, there are about 300 bus routes in operation in which the density is extremely high. Several of them are long trajectory detours, which makes the bus management really hard to handle. Due to the overfull bus routes, people living in Taipei City could have observed that many routes overlap with others and that sometimes bus jam occurs at a bus stop. In addition, for those tortuous bus routes, buses travelers may spend more time hanging on the bus than the real value it really takes. On the other hand, in the resource utilization efficiency point of view, the long and wind bus routes make the resource utilization very inefficient and hard to optimize. The overlapped bus routes produce the redundancy of buses and thus make the average passenger volume too small, which doesn't make the best benefit of each bus in operation. Furthermore, since there're many long routes on the map which is based on the truth that the average bus run on the map in each particular time span is more than needed, these facts cost the government or companies need to spend more money on the investment

of bus numbers rather the quality that really affects the experience of transportation. These problems have existed and been noticed for many years. Many people have tried to find out a better working bus system that brings the best of benefits into play in Taipei City.

A chessboard-like bus network has been thought to be a solution to the above problems. In the late 1980s', an initial prototype of a chessboard-like bus network started to operate, but soon they became not that Most of the routes are therefore not in operation now. Nevertheless, in 2006, the the Commissioner of the Taipei City Department of Transportation tried to promote a new bus route network again. Due to opposition from some frequent bus takers who did not transfers between buses and, more importantly, from the bus companies which had vested interests (about 15 bus companies are in operation), the new policy was not executed. In 2010, Taipei City councilors hurried the government to promote the policy, but it is still the old system operated in Taipei City.

Despite that the problems of bus transportation is not only a engineering problem, we want to know whether a chessboard-like bus network is more efficient or more convenient than the current system in Taipei City theoretically. The purpose of this project is to set up a simulation of behaviors of buses and clients and to observe what influences the efficiency of the bus traffic system.

2. DATA COLLECTION

For our real-case simulation, we transform a map of part of Taipei City into a graph with roads as edges and intersections as nodes. We need distances between each two nodes, current traffic information, stops of the bus routes and time intervals of the buses. They are all available but are hard to fetch automatically. To get these data, we have used the most powerful parser in the world! We collect the road conditions from Google Map and the bus information from the website of Taipei Public Transportation Office manually.

We select an area of Taipei City as in Figure 1. We transform it into a graph as in Figure 2, where edges represent the roads in the area and nodes represent the intersection of the roads. On the graph, we add the information of roads as in Figures 3 and 4. We record the bus routes and intervals as in Figures 5.

3. AGENTS

In our scenario, only two kinds of agents are involved in the simulation framework to simply the over-complex world

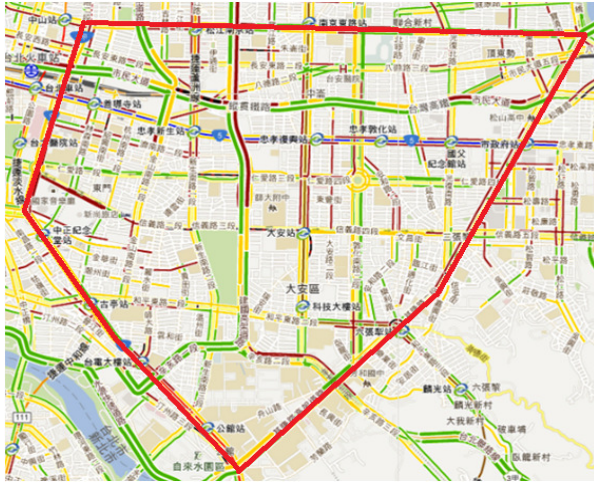


Figure 1: The framed area is transformed into the graph shown in Figure 2.

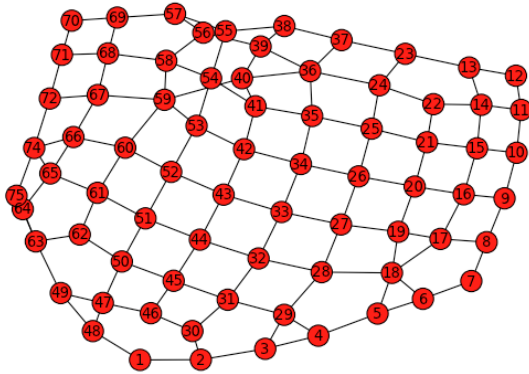


Figure 2: In this graph, edges represent the roads and nodes represent the intersections.

ID	NAME	#Neighbor	Verge	Nodes
1	基隆羅斯	2	Y	2;48
2	基隆辛亥	2	Y	3;30
3	基隆敦化	3	Y	2;4;29
4	基隆和平	3	Y	3;5;29
5	基隆光復	3	Y	4;6;18
6	基隆信義	3	Y	5;7;18
7	基隆松壽	2	Y	6;8
8	基隆仁愛	3	Y	7;9;17
9	基隆忠孝	3	Y	8;10;16
10	基隆市民	3	Y	9;11;15
11	基隆八德	3	Y	10;12;14
12	基隆南京	2	Y	11;13
13	光復南京	3	Y	12;14;23
14	光復八德	4	N	11;13;15;22
15	光復市民	4	N	10;14;16;21
16	光復忠孝	4	N	9;15;17;20

Figure 3: Part of the node information (totally 74 nodes). The entries for each node are its ID number, its name, its number of adjacent nodes, whether this node is on the verge, and its adjacent nodes.

1 2 基隆羅斯 基隆辛亥 1300	1 2 基隆羅斯 基隆辛亥 40
2 3 基隆辛亥 基隆敦化 300	2 3 基隆辛亥 基隆敦化 40
2 30 基隆辛亥 復興辛亥 450	2 30 基隆辛亥 復興辛亥 30
3 4 基隆敦化 基隆和平 450	3 4 基隆敦化 基隆和平 40
3 29 基隆敦化 敦化和平 300	3 29 基隆敦化 敦化和平 30
4 5 基隆和平 基隆光復 750	4 5 基隆和平 基隆光復 30
4 29 基隆和平 敦化和平 350	4 29 基隆和平 敦化和平 20
5 18 基隆光復 光復信義 350	5 18 基隆光復 光復信義 40
5 6 基隆光復 基隆信義 400	5 6 基隆光復 基隆信義 30
6 18 基隆信義 光復信義 210	6 18 基隆信義 光復信義 30
6 7 基隆信義 基隆松壽 350	6 7 基隆信義 基隆松壽 30
7 8 基隆松壽 基隆仁愛 220	7 8 基隆松壽 基隆仁愛 30
8 9 基隆仁愛 基隆忠孝 450	8 9 基隆仁愛 基隆忠孝 30
8 17 基隆仁愛 光復仁愛 450	8 17 基隆仁愛 光復仁愛 50
9 16 基隆忠孝 光復忠孝 650	9 16 基隆忠孝 光復忠孝 30

(a)

(b)

Figure 4: Figure 4(a) is part of the edge information (totally 134 edges), where the entries for each edge are its two ends, the two ends' names, and the distance between the two ends; Figure 4(b) is part of the traffic conditions, where the number in the last entry is the speed buses can drive at on the corresponding edge.

1 63;49;48;1;2;3;4;5	1 10
2 71;72;74	2 40
3 63;62;61;51;44;45;31;29;4	3 80
5 55;57;69;70;71;72	5 15
12 70;69;57;55	12 15
14 71;72	14 15
15 72;67;66;65;64;63;62;50;45;31;29;4	15 10
18 72;74;75;64;63;62;50;45;31;29;4	18 15
20 74;65;61;51;44;32;28;18;6;7	20 10
22 72;67;66;65;61;51;44;32;28;18;6	22 12
26 70;69;57;55	26 80
28 8;9;10;11	28 60
32 7;8;9	32 120
33 23;24;25;26;27;28;18;6	33 20
37 74;66;60;52;43;33;27;19;17	37 60
38 74;65;61;51;44;32;28;18;6	38 30

(a)

(b)

Figure 5: Figure 5(a) is part of the bus route information (totally 124 routes), where the entries for each bus are its route number and the stops (in the selected area) on its route; Figure 5(b) is part of the bus interval information, where the number in the second entry is the departure interval of the corresponding route.

we want to model. The following sections show how we assume them and model them:

3.1 Buses

In our task, the buses' behavior is simplify to listen to the orders from a centralized bus manager, whose work is scheduling the bus intervals and manage the routes. This bus manager plays the pivot role in our simulation. Since that all buses are issued directly by the bus manager, the bus manager must to maintain a policy (or a kind of configuration) to specify what task should be done now or how a task is been done. Rather than running by its own internal will but by the guide from the centralized bus manager, the buses are sort of single agent system rather an multi-agent one in which usually the agents involved is more than two. However, even though the buses are run by a single agent system, it proves to be reasonable assumption because of the nature of buses is obeying the order (or a predefined route and time interval) rather than act freely even under the bus driver's control. More specifically, the bus will have several parameter resources to refer before it starts to move to the next iteration. They are:

- The route information that needs to follow from the bus manager.
- The start command sends from the bus manager. If a start command is sent to the bus, the bus will start its journey.
- The bus interval sent also from the manager, which specifies how many minutes should send a bus out.
- The traffic information encoded in the edges of the map graph.

The bus behavior will then be:

- It will look up the traffic information in the graph to decide its current speed at each iteration.
- After the run of each move, if the bus passes a node, the bus will stop in the node if the node is one of the schedule stop handed from the bus manager.
- If the bus reaches the end of the scheduled route, the bus will turn around and go back to the beginning of the route.
- After the round trip route is finished, the bus will eliminate from the simulated world. All the necessary statistics will be collect to compute some indicator digits that help us to judge the performance.

3.2 Clients

At each iteration, client agents are generated on a given node and a given destination node. The number of clients generated on a particular node is decided by the type of scenario we want to model, which will be explained more detailed in the following sections. Furthermore, in order to reduce the computational cost, the clients' behaviors are assumed to always greedy in a global fashion. A detailed explanation is show as follows:

- If a client is at a bus stop, it first compares all the buses stopping at this bus stop.

- The client will then choose a bus whose one of the next-N stops will take him to the nearest node to his target destination. N is a positive integer.
- Then he will get on the chosen bus until the next stop is reached and repeat the process again.
- Otherwise, the client will wait for the chosen bus until it comes into the stop.

At first glance, although greedy algorithm might be not that intuitive to us (or as least can only apply to partial group of people but not at all), we believe it can still reflect some interesting facts we want by the simplified simulation.

4. SIMULATION

In this section, we first introduce how we assume our environments how we run our simulations on the predefined environments. Three scenarios to mimic the real world behavior under different time situations and three different policies of bus agents are taken into consideration. Last but not least, the experiment results, analysis, and discussion regarding to the results are presented at the end of this section.

4.1 Environment Settings

In our simulations, each iteration in the simulated world is one minute in the real world. We use a small time span for each iteration to mimic the continuous real environment dynamics as like as possible. Three bus policies are take into consideration:

Original: The current bus routes in operation in which the data is collected manually from internet.

Chessboard: The main policy that we want to compared with. A chessboard-like bus route configuration.

All-pair: A funny example we do in our experiments. We have a bus for each of the edges in the map. Each bus is only run on the give edge.

The virtual buses and clients take actions alternately at every minute/iteration. We run 10000 minutes/iterations for every three real world scenarios and three different bus policy configurations. We use three types of scenarios, they are:

Off-peak time: Under this scenario, the flow of people are assumed to be under the assumption of random walk theorem, the average flow of people from node to node is zero. In this case, the number of clients generating from a particular node i is draw from a Gaussian distribution $N_i^{(o)} \sim Gauss(\mu^{(o)}, \sigma^{(o)})$ with mean $\mu^{(o)}$ equal to 1.2 and variance $\sigma^{(o)}$ equal to 0.5.

Morning rush hours: This scenario tries to model the rush hours during each morning where people comes from the outskirts of the city. The only difference between it and the off-peak time is that the number of client generated from the verge node v is draw from a Gaussian distribution $N_v^{(m)} \sim Gauss(\mu^{(m)}, \sigma^{(m)})$ with mean $\mu^{(m)}$ equal to three times of other nodes 3.6.

Evening rush hours: This scenario tries to model the rush hours during each evening where people goes back to

		Original	Chessboard	All-pair
Client Statistics	Client Generated	506,146	506,402	505,904
	Client Left on Map	149	170	1841
	Completion Rate	99.97%	99.97%	99.64%
	Average Time Cost	14.63 min	14.35 min	39.76 min
	Expected Travel Dist.	2663.35 m	2645.99 m	2646.91 m
	Avg. Stop Transfer	1.97	1.33	4.59
Bus Statistic	Edge Cover Rate	99.52%	99.25%	100.00%
	Edge Repeat Rate	649.25%	11.19%	0.00%
	Bus Generated	81,650	19020	127,434
	Avg. Bus on Map	184	36	22
	Total Travel Dist.	1,062,990 km	216,069 km	196,107 km
	Average Capacity	2.18	9.69	5.79

Figure 6: Simulation under off-line scenario.

		Original	Chessboard	All-pair
Client Statistics	Client Generated	1,141,862	1,142,108	1,141,338
	Client Left on Map	403	449	4479
	Completion Rate	99.96%	99.96%	99.61%
	Average Time Cost	14.71 min	15.24 min	43.16 min
	Expected Travel Dist.	2885.72 m	2872.47 m	2865.24 m
	Avg. Stop Transfer	2.14	1.42	5.08
Bus Statistic	Edge Cover Rate	99.52%	99.25%	100.00%
	Edge Repeat Rate	649.25%	11.19%	0.00%
	Bus Generated	81,650	19020	127,434
	Avg. Bus on Map	184	36	22
	Total Travel Dist.	1,062,990 km	216,069 km	196,107 km
	Average Capacity	5.29	24.37	14.26

Figure 7: Simulation under morning rush hour scenario.

the outskirts of the city from inside. The only difference between it and the off-peak time is that the number of client generated from the non-verge node $n - v$ is draw from a Gaussian distribution $N_{n-v}^{(m)} \sim Gauss(\mu^{(m)}, \sigma^{(m)})$ with mean $\mu^{(m)}$ equal to three times of other nodes 3.6.

4.2 Results

Fig. 6, Fig. 7, and Fig. 8 shows our simulation result under three different bus policies when the scenario is off-peak, morning rush hour, evening rush hour respectively.

The client generated in three different scenarios are different since the gaussian mean is three times the number of Off-peak in verge nodes of Morning rush hour and Evening rush hour. If we only look into the off-peak result in Fig. 6, you will discover that in client statistics, it is important to note that the average time cost is better for Original and Chessboard policy compared to All-pair policy but the Original and Chessboard is not so different. However, in average stop transfer, the Chessboard policy exhibits a better performance than the Original policy. In bus statistics, the

		Original	Chessboard	All-pair
Client Statistics	Client Generated	1,141,024	1,141,725	1,142,253
	Client Left on Map	385	359	4268
	Completion Rate	99.97%	99.97%	99.63%
	Average Time Cost	14.53 min	14.51 min	40.20 min
	Expected Travel Dist.	2697.61 m	2681.17 m	2673.47 m
	Avg. Stop Transfer	2.00	1.34	4.65
Bus Statistic	Edge Cover Rate	99.52%	99.25%	100.00%
	Edge Repeat Rate	649.25%	11.19%	0.00%
	Bus Generated	81,650	19020	127,434
	Avg. Bus on Map	184	36	22
	Total Travel Dist.	1,062,990 km	216,069 km	196,107 km
	Average Capacity	4.94	22.08	13.17

Figure 8: Simulation under evening rush hour scenario.

benefit of chessboard is very clear. The edge repeat rate is only 11.19% compare to the extremely overlapped Original policy's 649.25%. The average bus left on the map is 36 for chessboard compared to 184 of original policy. For the total travel distance, the chessboard is less thus save more energy compared to the original one. And finally, for the average client capacity, the sparse capacity of original policy suggested that its overlapped nature makes the routes compete together and thus lower the overall efficiency of the system. For Fig. 7 and Fig. 8, the results shown also confirm to the Fig. 6 result.

5. CONCLUSION

We have discussed a problem about inefficient resource utilization of bus routes in Taipei city. Traditionally, the bus routes in Taipei city is full of detours, which makes the expected arrived time is far greater for a particular bus route. It also contains a lot of overlapped edges, which not only increase the number of buses need to be invested by the government and the bus companies but also decrease the utilization rate (average number of passengers) of a bus. These are all indicators of the inefficiency resource utilization of real world.

With our model which cast the real world situation into a simpler world which only includes two agents: buses and clients. The buses is then handled by a centralized bus manager whose responsibility is to schedule the bus under some predefined policies. Clients are assumed greedy and thus also simply the computational cost. The simulation is run under three different bus policies and three different scenarios. The result suggested that chessboard, no matter in terms of the view of the client side but also the overall resource utilization side is far more efficient than the bus policy in operation now, as well as the all-pair policy. Although in engineering's perspective, the chessboard policy does better than the current policy, the problem involves too many political and social issues underlines in the Taiwan local culture, the problem is not expected to be solved immediately in the near future but will progress toward a better direction gradually.