

**Optimization of Feeder Bus Routes for Electric Train Service  
using Ant Colony Algorithm**

by

Dellon Tan Tzet Yung  
15107

Dissertation submitted in partial fulfilment of  
the requirements for the  
Bachelor of Engineering (Hons)  
(Electrical and Electronics  
Engineering)

MAY 2014

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

**Optimization of Feeder Bus Routes for Electric Train Service using  
Ant Colony Algorithm**

by

Dellon Tan Tzet Yung  
15107

A project dissertation submitted to the  
Electrical and Electronics Engineering  
Programme Universiti Teknologi

PETRONAS

in partial fulfilment of the requirement for the  
BACHELOR OF ENGINEERING (Hons)  
Electrical and Electronics Engineering)

Approved by,

---

(Name of Main Supervisor)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

MAY 2014

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

---

DELLON TAN TZET YUNG

## **ABSTRACT**

Nowadays, it has become more and more convenient for the citizens of Malaysia to move around the different states within the country as public transportation such as buses and trains offer affordable rates for their services. One of the most effective and efficient mode of transportation in Malaysia would be the Electric Train Service (ETS). It is a service that allows one to travel by train between cities of Ipoh and Kuala Lumpur inexpensively in a timely fashion. The ETS services may be excellent but the public transportation such as taxis and buses taking us to the station to board the trains may be inefficient and unreliable.

In order to address this issue, an optimised feeder bus system is proposed. Hence, to achieve this, various optimisation algorithm needs to be studied and researched. The techniques that would need to have the capability to do route planning or selection while considering other criteria like time, cost and efficiency in addition of the displacement travelled. A model would then be developed using a tool such as Matlab with inputs of maps from Google Maps. This will consequently enable the feeder bus system to the ETS station in Ipoh to be optimised.

## **Table of Contents**

ABSTRACT .....	2
LIST OF TABLES .....	4
LIST OF FIGURES .....	4
CHAPTER 1	
INTRODUCTION .....	5
Background of Study.....	5
Problem Statement .....	5
Objective and Scope of Study .....	6
Objective.....	6
Scope of Study.....	6
CHAPTER 2	
LITERATURE REVIEW.....	7
Electric Train Service .....	7
Travelling Salesman Problem .....	8
Algorithms.....	9
CHAPTER 3	
METHODOLOGY.....	12
Gant Chart and Key Milestone Progress .....	13
Project Methodology .....	14-15
Results .....	15-30
CHAPTER 4	
CONCLUSION.....	31
REFERENCES.....	32-33
APPENDIX.....	34-50

## **LIST OF TABLES**

Table 1	Comparison Between Dijkstra and Bellman Ford	9
Table 2	Final Year Project I Gant Chart	13
Table 3	Final Year Project II Gant Chart	13
Table 4	Average Travelling Distance Between Each Designated Feeder Station	26
Table 5	Average Travelling Time Between Each Designated Feeder Station	26
Table 6	Profit Earned Over Time	26
Table 7	Feeder Station Indicator	29
Table 8	Solution of Two Stations Using Brute Force Method	29

## **LIST OF FIGURES**

Figure 1	Process Flow Diagram	12
Figure 2	Map of Aman Jaya	16
Figure 3	Map of Menglembu	17
Figure 4	Map of Station 18	18
Figure 5	Map of Kampung Sengat	19
Figure 6	Map of Taman Jubilee	20
Figure 7	Map of Medan Gopeng	21
Figure 8	Map of Kampung Bercham	22
Figure 9	Map of Ulu Kinta	23
Figure 10	Map of Taman Silibin	24
Figure 11	Pie Chart or Roulette System Representing possibility of all choice on 1st Iteration	27
Figure 12	2nd Iteration	27
Figure 13	2nd Iteration with Much Lower Decay Rate	28
Figure 14	Nth Iteration for Best Possible Solution	28
Figure 15	Solution Obtained from Matlab	29

# CHAPTER 1

## INTRODUCTION

### **Background of Study**

With the introduction of the electric train service (ETS), citizens of Ipoh and Kuala Lumpur now have a much more convenient mode of transportation to take between the major cities. In this study, we would like to investigate and formulate a solution to identify the best route available for the Feeder Bus service by taking into consideration on real-life inputs such as displacement, time, cost and demand. Other factors to be included would be explored as the study progresses. The different type of algorithm for route planning optimisation will be explored and consequently leads to the implementation of several bus route optimization models. Finally, a software application that incorporates mapping capability such as Google Maps will be developed so that the optimal route can be determined. The effectiveness of this system will be evaluated in terms of its user-friendliness, responsiveness and features.

### **Problem Statement**

With a service as effective and efficient such as the ETS, one would question whether it is possible for one to be late for an appointment or meeting. Yes, it is possible as one would still need to take either a taxi or bus to the train station before being able to board the train. It would be undeniable that one would be able to take one's own vehicle such as cars or motorcycles to the station but we must understand that not everyone can afford to own one of these. Thus, we would like to focus more on public transportation which is cheaper making it more accessible to a larger fraction of the population. This brings us to question the reliability of the public transportation available to the train station. This lead us to find out that there is still a serious lack of connectivity between the commuter's home or office to the ETS station. Thus, we would like to propose introducing an effective Feeder Bus System (FBS) to overcome this predicament.

## **Objective and Scope of Study**

### **Objective**

The objective of this study is to optimise a feeder bus route around Ipoh city to the ETS station in terms of cost, time and efficiency.

### **Scope of Study**

The study will be centred around analysing a number of route planning algorithms available and selecting the best practice. The selected algorithm will be modified to suit the project's requirement. The model will then be formulated to maximise profitability while taking into consideration aspect such as cost, time efficiency and other aspects if we see fit. The tool which will be explored to achieve this model is primarily Matlab. The integration of this model which is to be developed with Google Maps will also be explored. The end product that we would hope to achieve is a software with mapping capability that is able to provide an optimised route for the Feeder Bus Service.



## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **Electric Train Service**

In Malaysia, the Electric Train Service (ETS) is a service that allows one to travel by train between cities of Ipoh and Kuala Lumpur. This service was introduced as part of the deliverables of the Eighth and Ninth Malaysia Plans in the western region of Peninsular Malaysia by Keretapi Tanah Melayu on the 12th of August 2010.

The train line that initially continued south of KL Sentral train station onwards to Seremban has been discontinued as of October 2012.

The designs of train sets which were utilised to provide ETS services were done by Marubeni Corporation. They were jointly manufactured by Mitsubishi Electric of Japan and Hyundai Rotem of Korea. The train set weighs about 231.8 tons and measures a length of 128m each. As for dimensional specifications, each carriage spans a length of 22.95m, a width of 2.75m and a height of 4m with the capacity to transport a maximum of 350 passengers at any one time.

These trains which run on electrified lines were designed to travel at an astounding velocity of 160km/h but it has been found that their optimal operating speed is at 140km/h. The services that these remarkable trains offer are provided and managed by a subsidiary of Keretapi Tanah Melayu Berhad called ETS Sendirian Berhad.

Originally, the services that were provided were categorised into three classes namely Silver, Gold and Platinum. The variation between the classes were the number of stop they would make before arriving at their designated destination. These stops would in turn affect the time required by the train to make the whole journey. However, the Platinum service has been withdrawn as of late from the selection due recent rescheduling in the trains timetable [12].

## **Travelling Salesman Problem**

The origins of the travelling salesman problem remains unclear to date. However, the problem is believed to have been around since 1832 as it was found to be written in the handbook of a travelling salesman during that period of time. Only the idea of the problem was defined but no mathematical approach was taken to solve it.

In the 1800s, British mathematician, Thomas Kirkman alongside Irish mathematician, W. R. Hamilton worked collaboratively to mathematically formulate the Travelling Salesman Problem. In the 1930s, the mathematicians from both Vienna and Harvard began to study and analyse the general form of the problem. A noteworthy individual who contributed to the development of the Travelling Salesman Problem was Karl Menger as he was responsible for defining the it. The name Travelling Salesman Problem was actually introduced by Hassler Whitney of Princeton University [12].

The problem probes us to determine the shortest available route that allows the person in question basically the salesman to visit each of the city in a given list of allocated cities once while covering the least amount of distance possible before returning to the city of origin. The solution of this problem can be used to formulate an algorithm that is capable of optimising route selection for a feeder bus service.

## Algorithms

More often than not, when we talk about finding the shortest possible route, we would often come across the Dijkstra's algorithm. According to Pepping [1], the Dijkstra's algorithm may be the most renowned routing algorithm but often it is not used as it is a demanding algorithm and it would require high processing power to compute. Pepping [1] also stated that Dijkstra's algorithm is obsolete when compared to newer methods which are capable of solving a route selection problem in a much reduced time frame period. In addition, Dijkstra's algorithm which is a single objective optimisation technique would be unable provide a solution should there be two optimally equal solution to the particular problem [2].

Another algorithm that is often used in determining the shortest path to take between two designated points would be is the A\* algorithm. Initially this algorithm was used as a graph traversal technique but due to its precision and efficiency, it was later adapted for pathfinding purposes. [3] In essence, A\* algorithm is just an extension of the Dijkstra's algorithm which is able to achieve results in a much reduced period of time due to the usage of heuristics.

Besides that, another algorithm that is similar to the Dijkstra's algorithm would be the Bellman-Ford algorithm [2]. Compared to Dijkstra's algorithm in aspect of performance, this algorithm actually loses out. However, it is still used due to its versatility as it is able to accept and compute negative values [8].

<b>Parameter</b>	<b>Dijkstra</b>	<b>Bellman ford</b>
<b>Overheads</b>	Less	More
<b>Scalability</b>	More	Less
<b>Quality of the best available route.</b>	More	Less
<b>Negative edges</b>	No	Yes
<b>Delay</b>	Less	More

**Table 1 : Comparison Between Dijkstra and Bellman Ford [7]**

The optimal solutions for the Travelling Salesman Problem vary depending on the number of cities which are taken into consideration. In order to obtain an

optimal result for the range of cities around 40 to 60, it is advisable to use the branch - and - bound algorithm. Any number of cities above that up to 200 cities should be computed using the progressive improvement algorithms to acquire optimal result. Currently, the highest record of solving the Travelling Salesman Problem at 85900 cities can be achieved through the implementation of the branch - and - bound together with the branch - and - cut approach.

The methods above are exact methods but for route optimisation it is recommended to use flexible methods which are approximate instead as it is more optimal and can be adapted to various situation [9]. Exact algorithms do not always provide the most optimal results. As such we explore other more flexible and advance algorithms such as Genetic algorithm and Ant Colony Optimisation.

Genetic algorithm was inspired by Darwin's theory of natural selection which is the survival of the fittest [4, 5].

The basic idea of Genetic algorithm would be as followed:

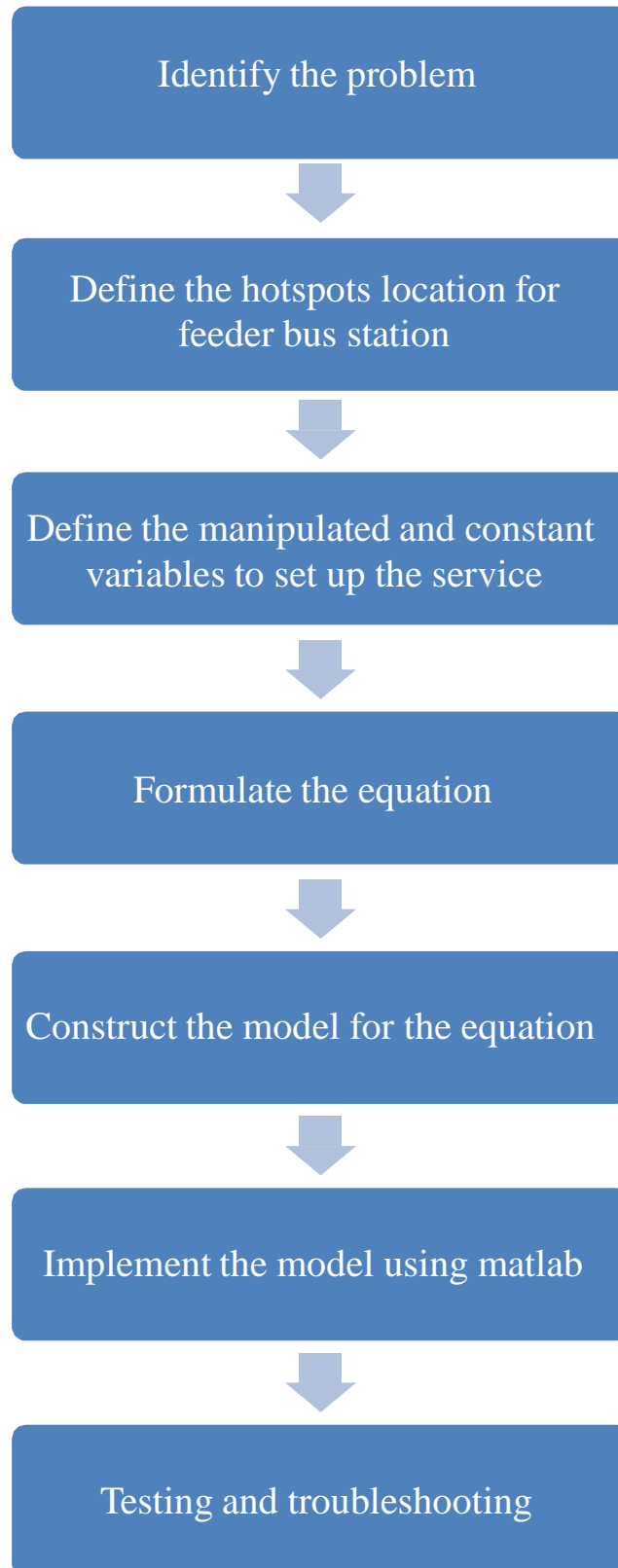
1. It begins with the intialisation phase where a random population is generated.
2. Next, it moves on to the evaluation phase where the 'fitness' of each individual member of the population is being inspected. The term 'fitness' here described how well a particular subject of the population satisfy the desired requirements. These requirements need to be specified and defined.
3. After that, it is followed by the selection phase where only the fittest of individuals are kept while the others are discarded.
4. It is then moves on to the crossover phase where 'fitter' offspring will be created by the combination of the selected individuals which are now classified as the parents.
5. Next, in the mutation phase, a little bit of randomness is added to the genetics of the population to create newer variation of individuals to make up the next population which is to be evaluated.

The cycle from steps 2 to 5 is repeated until the termination phase is achieved. The termination phase usually occur when an optimum result is obtained. Sometimes the process can arrive at the termination phase faster due to the constraint of time and resources.

The Ant Colony algorithm is inspired by the behaviour of ant colonies when searching for sources of food [6]. Initially in this process, the ants will wander aimlessly without purpose. However, when a food source is found by a particular ant, it will walk back to its colony leaving behind markers known as pheromones. These pheromones will act as an indicator for the path from its nest to location of the food source. There is a possibility that other ants which are wandering about to come into contact with these markers and follow it. If that happens, the strength of the marker will be enhanced as they too will be leaving behind pheromones of their own as they bring back the food to the nest. This will in turn bring about a chain reaction and a number of different paths will be formed. However, consequently the path with the shortest displacement would be the one that is most densely populated with traffic as the strength of the pheromone of that path will be the strongest as it is the one that is most travelled by the ants colony. The path most travelled by would then be the optimal solution to the particular problem.

Both Genetic algorithm and Ant Colony algorithm can be used to develop a route planning optimisation software. Nonetheless, Ant Colony algorithm has a slight advantage over Genetic algorithm as it has a feature called pheromone evaporation which would allow us to avoid being restricted to a locally optimal solution. If such a feature were non-existent in the Ant Colony algorithm, the initial selection of route will be the only solution causing other options to not be explored.

**CHAPTER 3**  
**METHODOLOGY**



**Figure 1 : Process Flow Diagram**

# Gant Chart and Key Milestone Progress

## FYP 1

No.	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Title selection														
2	Preliminary research work														
3	Identifying and understanding available algorithm														
4	In depth analysis and selection of optimal algorithm														
5	Variables and Hotspots Identification														
6	Research and formulation of objective function														

**Table 1**

## FYP 2

No.	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Research and formulation of objective function														
2	Research and construction of process model														
3	Implementation of model using Matlab														
4	Testing and troubleshooting														

**Table 2**

**Milestone**  
 **Progress**

## **Project Methodology**

The area of focus will be Ipoh city, thus the ETS station at Batu Gajah will not be taken into consideration. The hotspot locations for buses to the ETS station in Ipoh needs to be identified and the demand that each location would have need to be roughly estimated. A map containing all the roads that will subsequently lead to the ETS station in Ipoh needs to be obtained. The algorithm will then be designed using Matlab and modified to include parameter such as displacement, time, cost and demand for initial test. Google maps will be used to obtained an updated map to the roads available in Ipoh city. Initial testing will be done by determining the route for a single bus from a selected bus station to the ETS station using the algorithm and the result obtained will be compared with manual computation and research of the particular route option for the designated bus. Next the depending on the demand, another bus or two might be added to accommodate all the available bus stations that were selected and the software will be tested again. If the functionality of the software is not compromised, we will then move on to explore extra features which could be added to the software.

Matlab is chosen to implement the algorithm as its language is similar to fuzzy logic. Fuzzy logic is a type of logical system similar to Boolean logic where it has its own sets of rules that defines the logic. It is a form of many-valued logic which deals with reasoning that is approximate rather than fixed or exact. Boolean or binary logic usually gives us a value of 1 or 0 to specify for truth or false. However for fuzzy logic the value of truth can range from 0 to 1 giving a value of 0.7 true for example to describe the probability for an event being true.

Why Fuzzy Logic is chosen:

- It is easy to understand

The mathematical concepts makes up the reasoning behind fuzzy logic is simple. It is not a sophisticated or complex logic system.

- It is a flexible logic system

Fuzzy logic is capable of applying more layer with additional functionality on top of any particular system without having the need to start from scratch.



- It is able to process imprecise data.

All data inputs from our daily lives are analog data which are more often than not imprecise and prone to continuous slight variation if the sensitivity of the equipment permits it to be measured. Fuzzy logic has the capability to process these types of data inputs.

## **Results**

### **Hotspots Identification**

Hotspots are area in which we consider strategically suitable for the establishment of a feeder bus station to the ETS station in Ipoh. We also believe that with the introduction of this service at these particular areas, the citizens around the area would profit and benefit from it in term of convenience and ease of travel. We have considered many aspect in specifying the requirement for an area to be considered a suitable hotspot. First of all, we consider the demand or the need for a feeder bus transport system from that particular area to the Ipoh ETS station. Most of the areas which are selected are that are densely populated are housing estates and those which we believe are highly frequented would be area of interest such as shopping malls and resorts. Areas in which institution of higher learning are present are also targeted as students often travel back home during the weekends or short holidays. Last of all, we also consider areas with high number of government buildings to be an area that would benefit from the introduction of the feeder bus service as government officials would need to travel quite often for business purposes. Of course, we also consider whether there is a suitable waiting area around the particular location for the passengers to wait for the feeder bus to arrive such as fast food outlets like Mc Donalds and KFC or any petrol kiosk station nearby such as Shell, Petronas or Mobil so that there is a safe location to wait at if one is boarding the bus during the wee hours of the morning or the late hours of the night.

All diagram of maps are obtained from google maps [11].

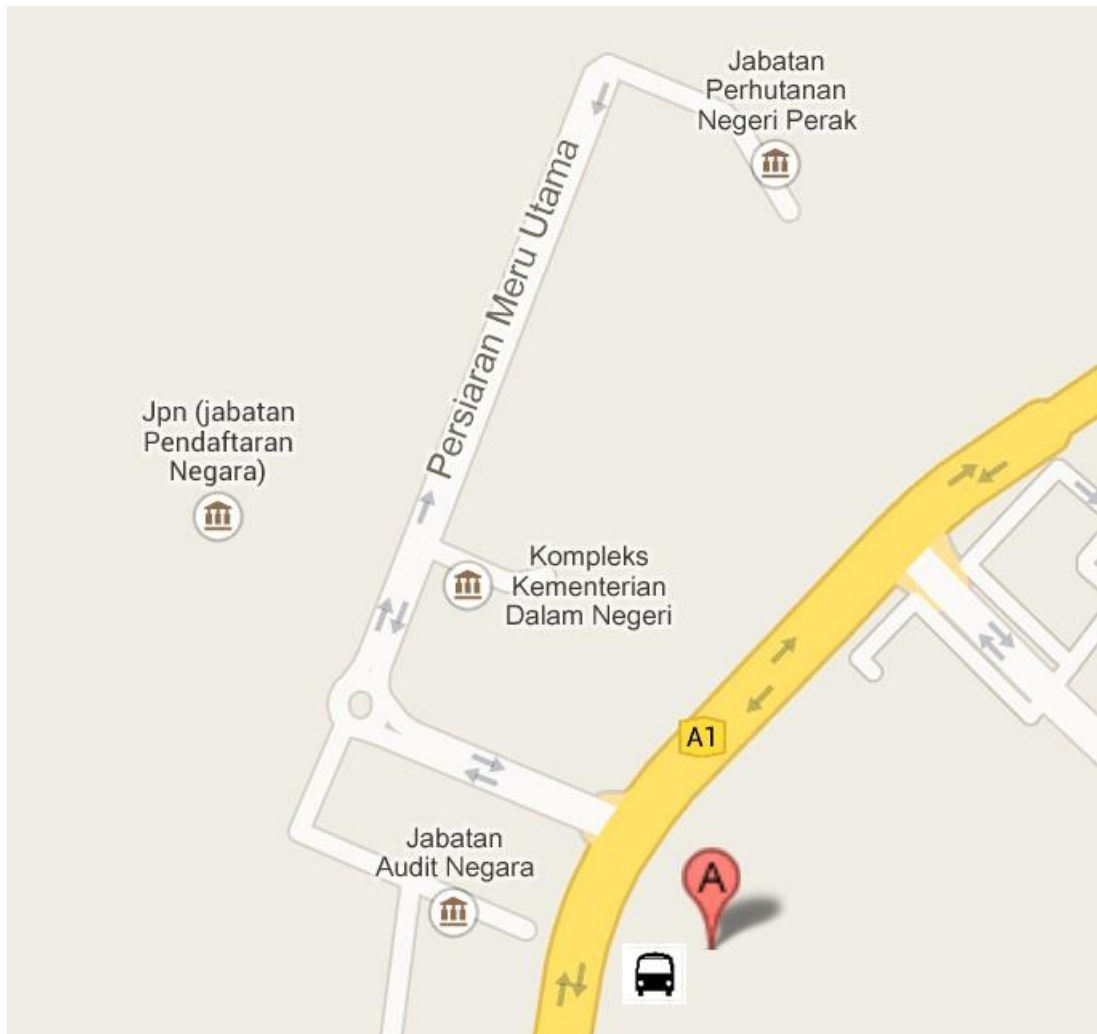


Figure 2

A is the main bus terminal in Aman Jaya where buses from other states will arrive at. It is a suitable spot to put a feeder bus station to cater for the locals and the tourist alike. Besides, there is quite a number of government building nearby and we believe that most of the staffs would be travelling often due to business purposes.

## Menglembu

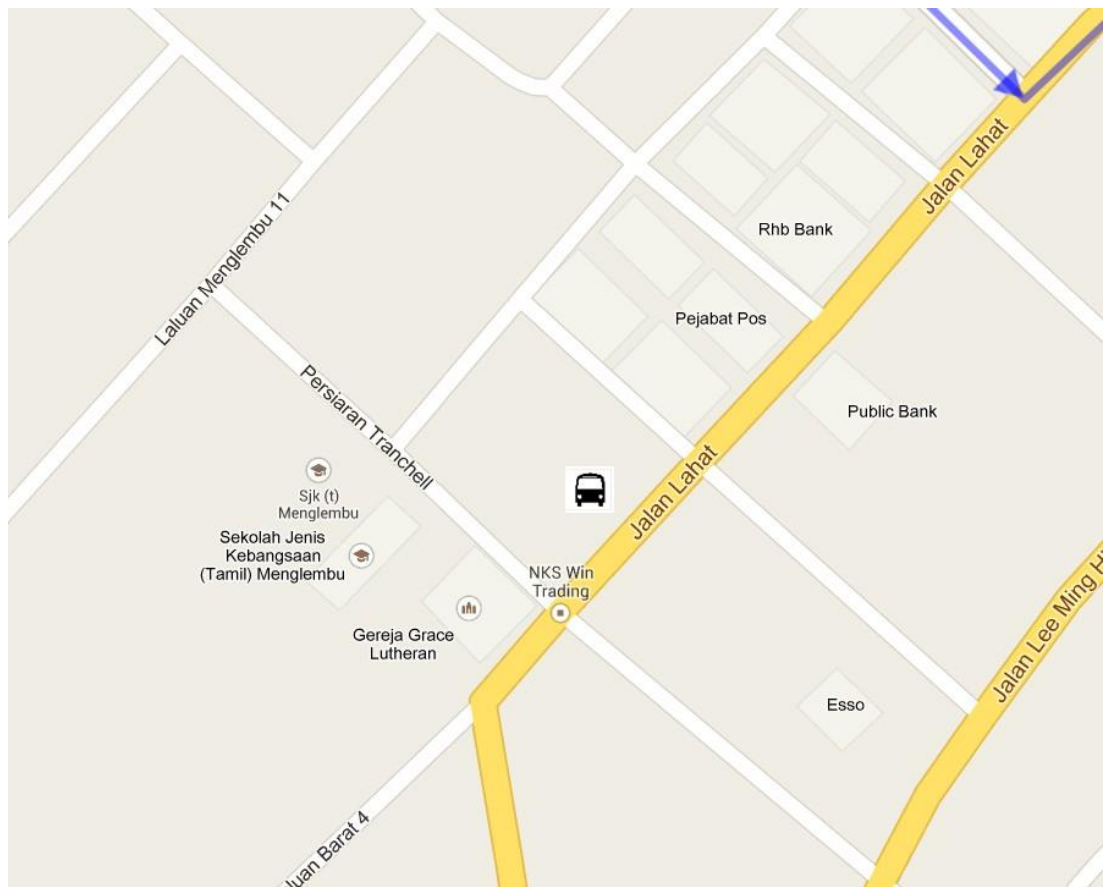


Figure 3

In the figure above, the feeder bus station serves to cater to the residents of the housing estates around Menglembu area. There is a post office and a number of banks such as RHB Bank, Public Bank and even a May Bank nearby. Business trips for the employees would also be more convenient.



## Kampung Sengat

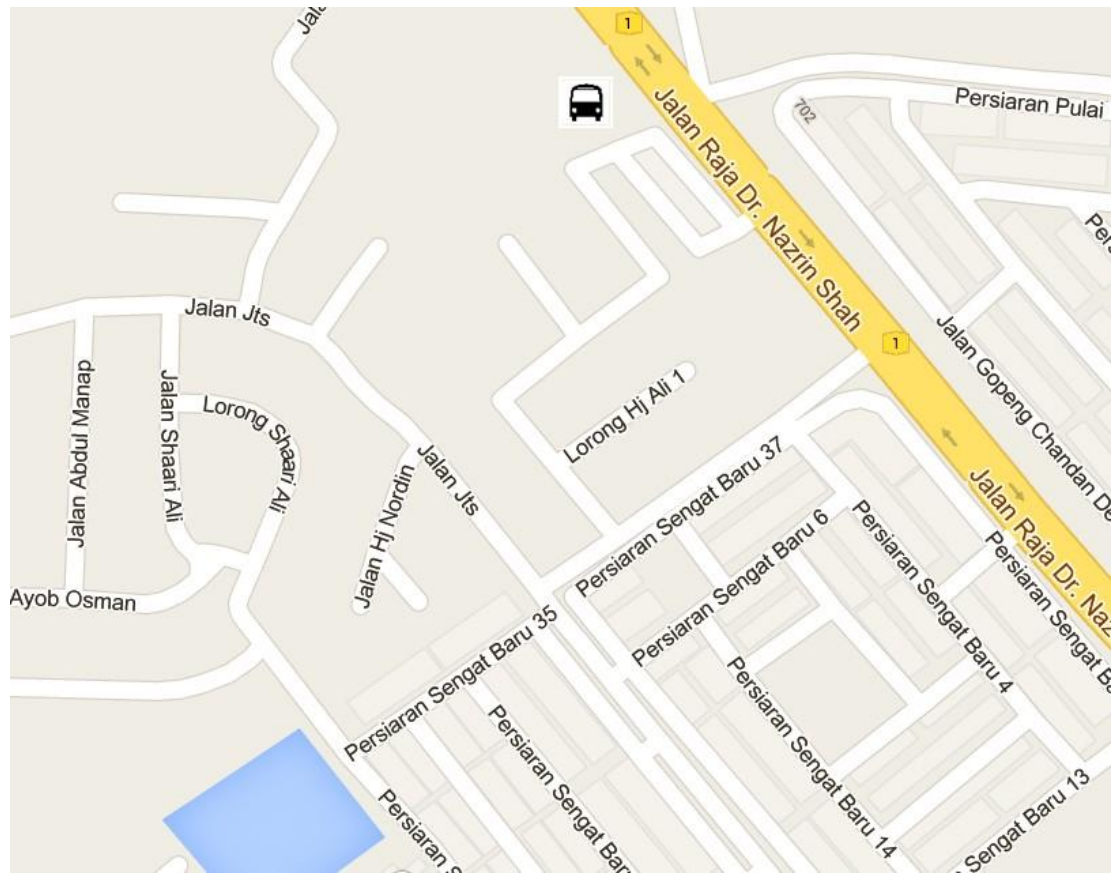


Figure 5

The feeder bus station above aims to cater for the residents staying in the housing estate in that area.

## Taman Jubilee

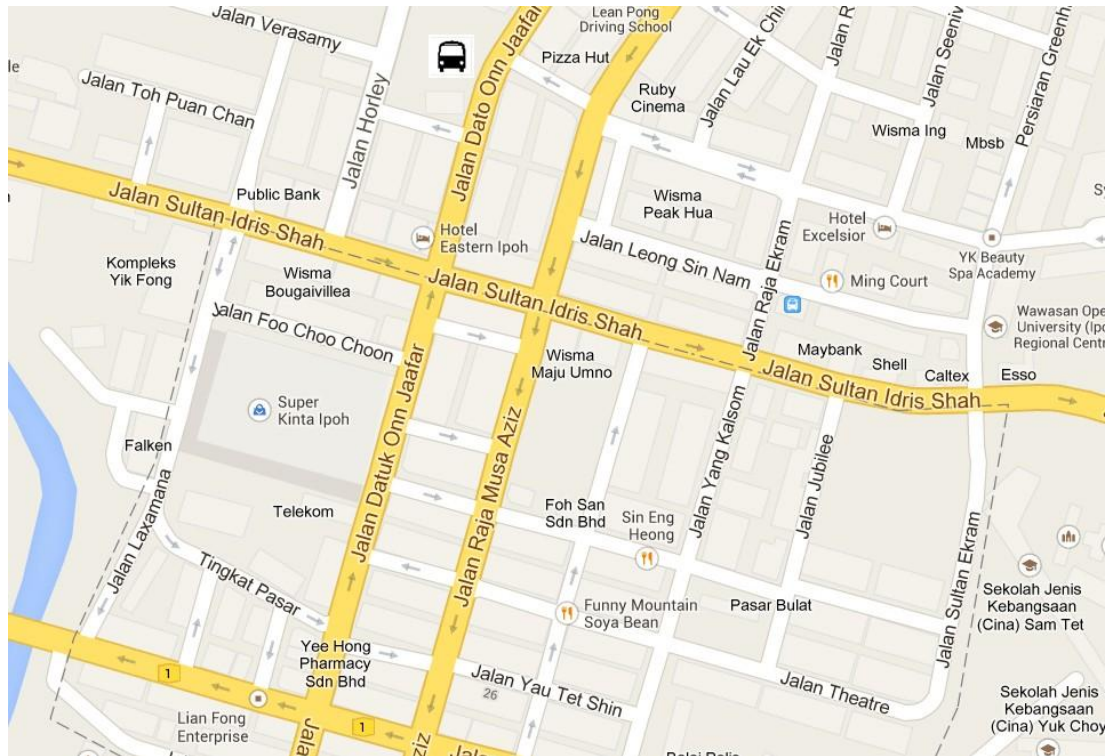


Figure 6

The Super Kinta Ipoh mall is nearby making it a place of interest. The Yoyo Bus Office is also nearby. On the far right end, we can see a YK Beauty Spa Academy and Wawasan Open University Ipoh Regional Centre which the feeder bus services can cater for. Quite a number of hotels and government building are also available in the further upper region of this area which is not found in the Figure above.



## Medan Gopeng

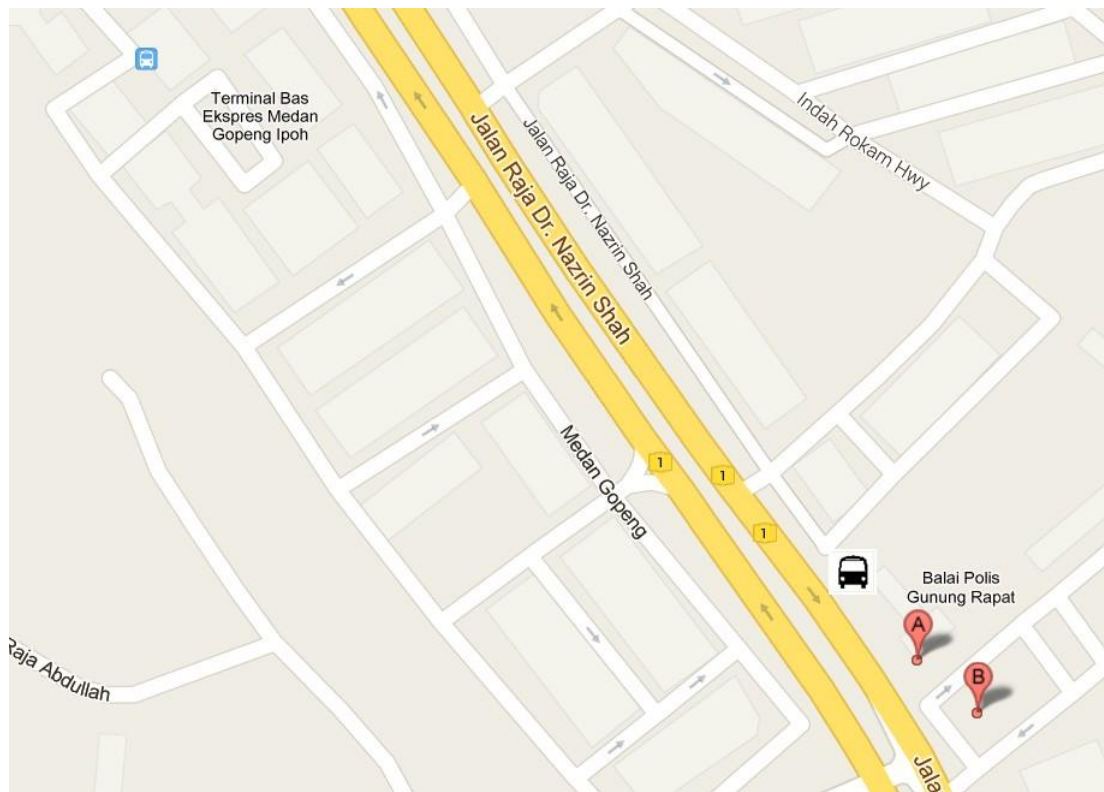
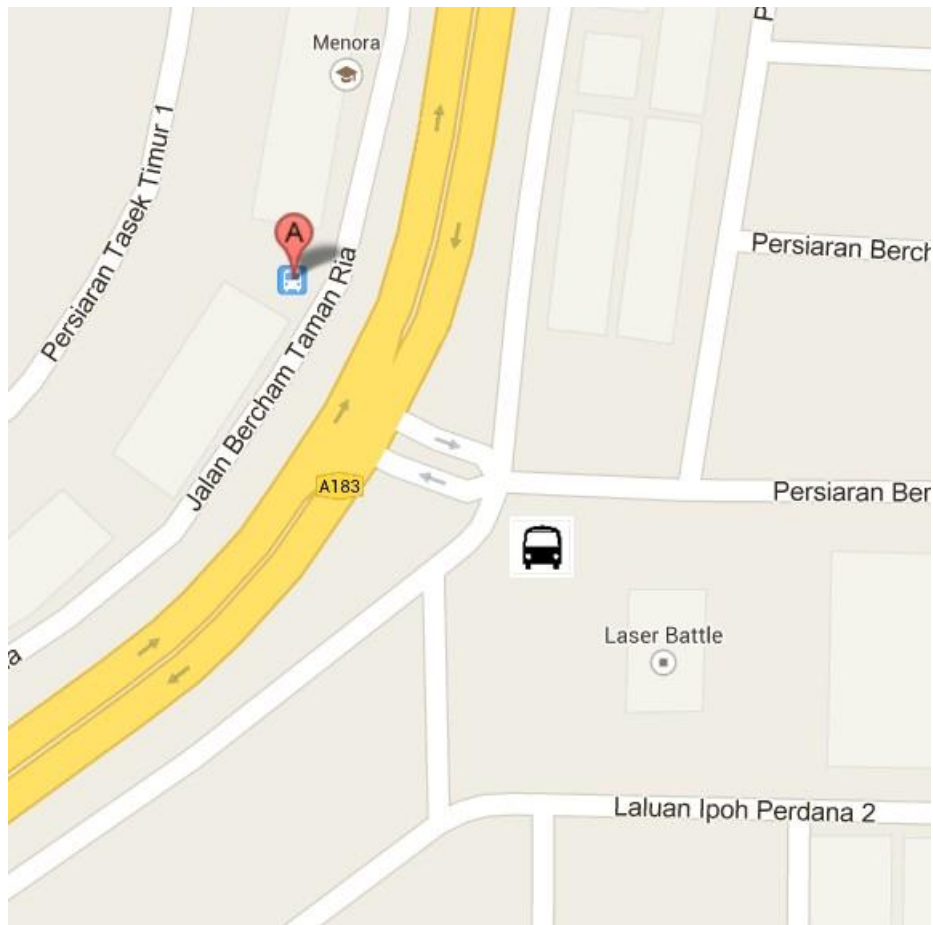


Figure 7

The old Medan Gopeng Express Bus Terminal can be found in the figure above. At the bottom part of the figure the marker B is a 24/7 Mc Donalds and there is Burger King outlet just next to it. Beside, them is the Gunung Rapat Police Station thus the area can be considered to be quite safe.

## Kampung Bercham



**Figure 8**

The feeder bus station above aims to cater for the Yoyo bus station which is in the area. There are a number of shop lots around the Yoyo bus station where passengers can have their meal while waiting to board their respective buses.



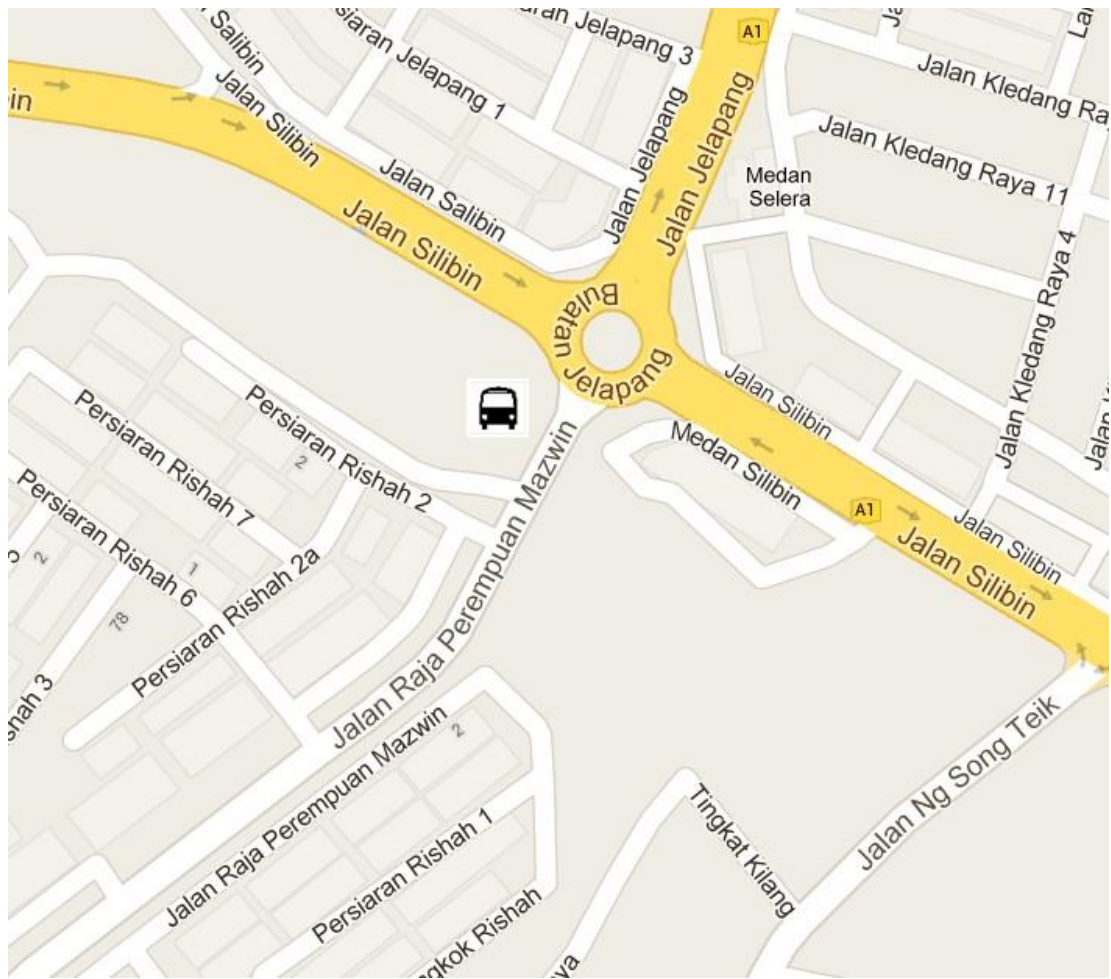
## Ulu Kinta



**Figure 9**

The marker A is Sunway College which means that the feeder bus can cater to them for their short trip back home during the weekend or the festive season. In addition Lost World of Tambun which is a tourist attraction is also nearby making this a strategic location to place a feeder bus station. There are a number of hotels and resorts nearby due to the presence of the Lost World of Tambun.

## Taman Silibin



**Figure 10**

The feeder bus station in the figure above aims to cater for the resident of the housing estate around it namely Taman Silibin, Taman Lim and Taman Pertama.

## Initial objective function formulation

Bus fee (ticket price per person) - BF

Demand (number of passenger) - DM

Fuel price (burned per km) - FP

Average travelling distance between feeder station - DT

Shortest displacement from feeder station to the Ipoh KTM station - SD

Average travelling time between station - T

Fare rate (increase in rate of profit) - FR

Bus capacity (number of seats) - C

$$Profit = \frac{(BF \times DM) - (FP \times DT)}{T} \dots \dots (1)$$

$$where; BF = FR \times \frac{FP \times SD}{C} \dots \dots (2)$$

From analyzing equation (1) , we can understand that a number of variables are fixed variable such as the fuel cost which is burned per kilometers and the average distance travelled from one station to another. I am also trying to consider the average travelling time as a constant variable for each trip between the different routes to simplify the computation. Two of the variables which can be considered as manipulated variable are the ticket price which is influenced by the fare rate and the demand of the service at that particular station.

The price for the fuel price per km is given by :

Averagely a bus used about 0.5 litres per km and the current price of diesel is RM 2 per litres

$$Fuel\ price\ per\ km = 2 \times 0.5 = RM\ 1$$

Demand is set at 10 per station while bus capacity is assumed to be 40 and the fare rate is 16 so as to avoid losses.

		Source location								
Average Travelling Distance / km		Aman Jaya	Menglembu	Station 18	Kampung Sengat	Taman Jubilee	Medan Gopeng	Kampung Bercham	Ulu Kinta	Taman Silibin
D e s t i n a t i o n	Aman Jaya		15	22	19	10.7	16.6	10.4	15.6	7.2
	Menglembu	15.3		6.1	13.9	7.4	9.9	14.6	16.8	7.6
	Station 18	22.4	6.1		12.8	7.7	8.6	14.7	16.8	14.7
	Kampung Sengat	25.5	17.3	11.8		8.8	4.4	12.3	17.7	12.8
	Taman Jubilee	12.7	7.1	7.5	8.8		4.9	7.9	10.8	4.3
	Medan Gopeng	17.6	9.9	8.6	4.8	4.8		8.2	10.4	8.5
	Kampung Bercham	12.8	16.8	14.8	12.3	7.2	8.4		5.6	12.9
	Ulu Kinta	16.4	16.8	16.3	13.8	10.8	9.9	5.6		14.1
	Taman Silibin	7.6	7.6	14.7	12.6	5.5	8.7	14.5	19.7	
	Ipoh KTM	12.6	5.6	7.8	10.5	2.1	5.4	10.4	11.6	3.6

**Table 4 : Average travelling distance between each designated feeder station**

		Source location								
Average Travelling Time / mins		Aman Jaya	Menglembu	Station 18	Kampung Sengat	Taman Jubilee	Medan Gopeng	Kampung Bercham	Ulu Kinta	Taman Silibin
D e s t i n a t i o n	Aman Jaya		20	29	27	15	20	18	20	12
	Menglembu	23		10	24	14	18	30	31	11
	Station 18	31	10		18	21	17	31	31	19
	Kampung Sengat	24	15	16		16	8	24	22	25
	Taman Jubilee	18	15	22	16		9	17	19	10
	Medan Gopeng	23	21	19	10	8		16	18	16
	Kampung Bercham	15	28	30	26	14	17		11	21
	Ulu Kinta	18	31	31	26	20	17	12		25
	Taman Silibin	15	11	20	26	10	16	16	26	
	Ipoh KTM	18	9	17	18	7	10	15	21	7

**Table 5 : Average travelling time between each designated feeder station**

The information above were obtained from google maps with. The average distance travelled between each station obtained from google maps are the shortest possible route to take to reach each destination. The time taken is also the average time taken between locations. The data obtained has the assumption of average to little traffic occurring during each travel.

		Source location								
Profit earned over time / (RM/mins)		Aman Jaya	Menglembu	Station 18	Kampung Sengat	Taman Jubilee	Medan Gopeng	Kampung Bercham	Ulu Kinta	Taman Silibin
D e s t i n a t i o n	Aman Jaya	0.00	0.37	0.32	0.85	-0.15	0.25	1.73	1.54	0.60
	Menglembu	1.53	0.00	2.51	1.17	0.07	0.65	0.90	0.95	0.62
	Station 18	0.90	1.63	0.00	1.62	0.03	0.76	0.87	0.95	-0.02
	Kampung Sengat	1.04	0.34	1.21	0.00	-0.03	2.15	1.22	1.30	0.06
	Taman Jubilee	2.09	1.02	1.08	2.08	0.00	1.86	1.98	1.87	1.01
	Medan Gopeng	1.43	0.60	1.19	3.72	0.45	0.00	2.09	2.00	0.37
	Kampung Bercham	2.51	0.20	0.55	1.14	0.09	0.78	0.00	3.71	0.07
	Ulu Kinta	1.89	0.18	0.48	1.08	-0.12	0.69	3.00	0.00	0.01
	Taman Silibin	2.85	1.35	0.83	1.13	0.29	0.81	1.69	1.03	0.00
	Ipoh KTM	2.10	1.87	1.38	1.75	0.90	1.62	2.08	1.66	1.54

**Table 6 : Profit earned over time**

The values above were computed with the excel using the above parameters.

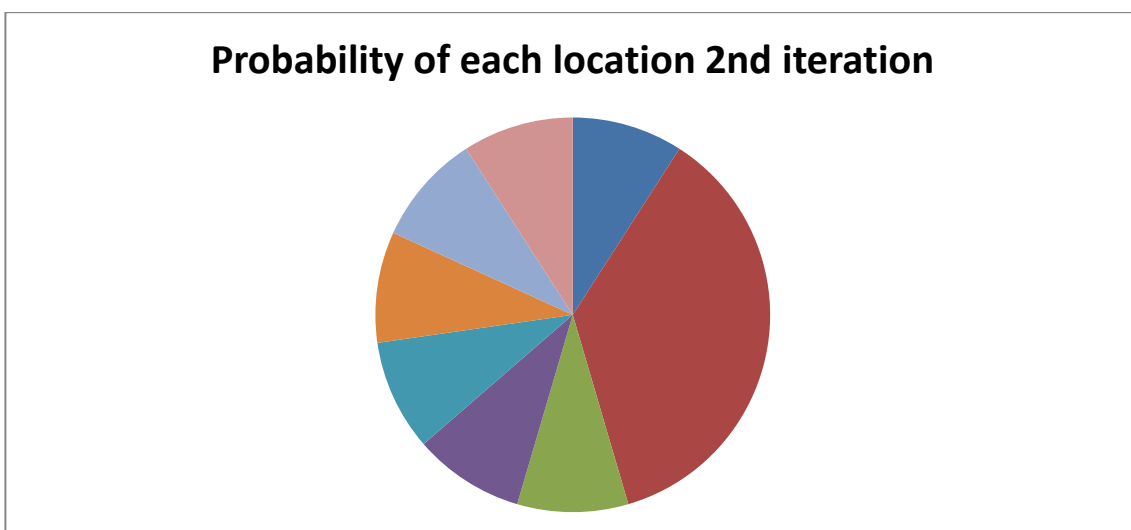
### How the ant colony algorithm function in selecting a destination

First of all we must understand at the source location, there are 8 other location to travel to not inclusive of the final destination. Initially all these 8 different locations have equal chance to be selected as no ants have explored any of these location yet.



**Figure 11 : Pie chart or roulette system representing probability of all choices on 1st iteration**

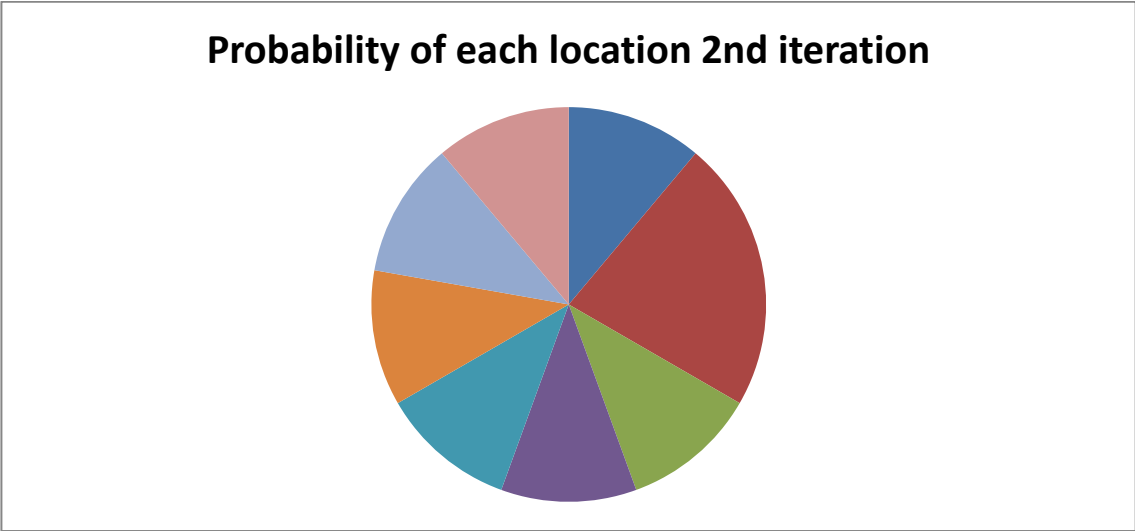
We assume that four ants are sent out to randomly choose any of these locations and of course there would be four observation found upon the return of the ants. There is a possibility that two or three or all the ants come back with the same solution but it is highly unlikely. They would come back with different observation and depending on our criteria, pheromone would be added to the best solution for that particular iteration. Take for example, the highest profit found for that iteration was the red section, as compared to the other random 3 section which were explored, for the next iteration the red section would grow in size and all the other section that were not chosen would reduce in size, thus causing the red section to be more likely to be chose in the next iteration.



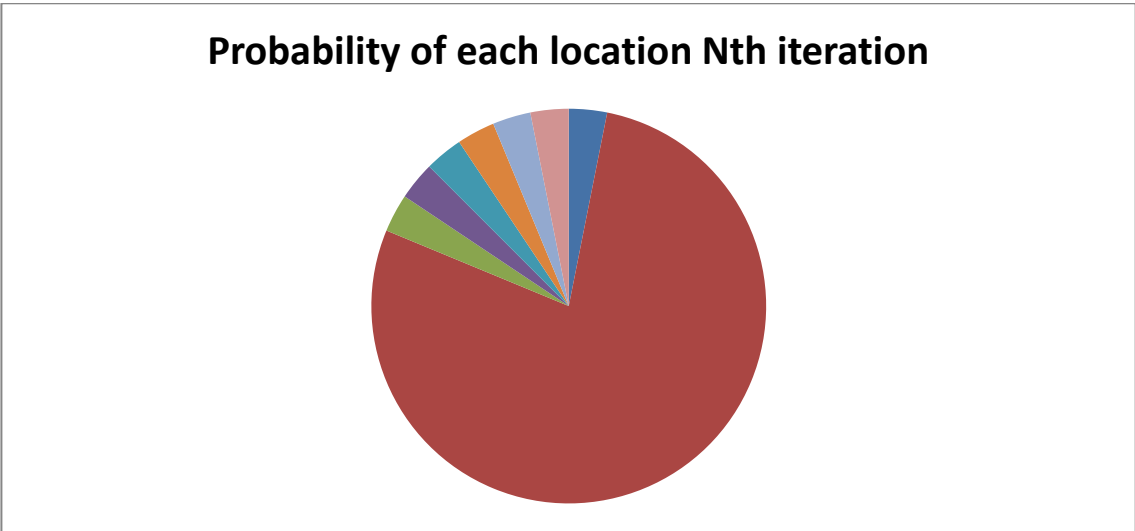
**Figure 12 : 2nd iteration**

This continue to occur with probability of other section increasing or the red section increasing further while the other section reduces even more. The decay rate also affects how much the best solution section increases or how much the other non-selected choices decrease.

Example of the second iteration with a lower decay rate.



**Figure 13 : 2nd iteration with much lower decay rate**



**Figure 14 : Nth iteration for best possible solution**

From the pie chart above after N number of iteration we can safely assume that the red section is the best solution and that the location represented by the red section would provide the most profitability in terms of a balanced ratio between fare rate and demand of each location.

To begin realizing this solution we must first determine the initial source location and the final destination of the feeder bus system. Based on the table , aman jaya feeder station is located the furthest from the Ipoh KTM station thus making it a suitable choice to be chosen as the source location. Now that we fix the source location and the final destination which is the Ipoh KTM station, we must consider how many other choices of route is possible to be taken from aman jaya feeder station to the Ipoh KTM station. I tried testing for the number of possible route combination given the source and destination is fixed and I found an astounding 40320 total combinations. This translate into meaning that the pie chart show above would be split in 40320 different section. Initially, I tried coming up with the matlab coding for the solution above but soon I realise that the number of arrays and variables that I would need to define would be immeasurable. Hence, I decided to come up with the simplest demonstration of the ant colony algorithm for solving this problem. I would only consider 2 feeder station inclusive of the source and the final destination. The other reason why I do so I because of the total time taken to reach the KTM station from the source to the final destination would amount to nearly an hour and a half should I consider all the station in a singular route.

### Brute Force Method

Aman Jaya - A
Menglembu - B
Station 18 - C
Kampung Sengat - D
Taman Jubilee - E
Medan Gopeng - F
Kampung Bercham - G
Ulu Kinta - H
Taman Silibin - I
Ipoh KTM - J

**Table 7 : Feeder station indicator**

Route	Profit per mins
ABJ	1.53 + 1.87 = 3.4
ACJ	0.90 + 1.38 = 2.28
ADJ	1.04 + 1.75 = 2.79
AEJ	2.09 + 0.90 = 2.99
AFJ	1.43 + 1.62 = 3.05
<b>AGJ</b>	<b>2.51 + 2.08 = 4.59</b>
AHJ	1.89 + 1.66 = 0.23
AIJ	2.85 + 1.54 = 4.39

**Table 8 : The solution of two stations using brute force method**

```

z =
    4.5900

AGJ is the most optimal route
fx >>

```

**Figure 15 : Solution obtained from matlab**

To realise a much more complete, solution for all the possible route to take from the KTM station in Ipoh throughout all the nine different distinct location and back to the station again. I tried to come out with the coding for it by using a much simpler objective function as shown below:

$$Profit = (BF \times DM) - (FP \times DT)$$

Bus fee (ticket price per person) - BF is set to RM 3

Demand (number of passenger) - DM is set to 5 per station

Fuel price (burned per km) - FP is found to be RM 1 per km

Average travelling distance between feeder station - DT (refer to Table 4

Bus capacity (number of seats) - C is 45 in total

However, after trial and error, I am only able to come up with the first iteration for this solution implementation and not able to implement this pheromone update mechanism due to my limited programming abilities. The partial implementation can be found in the appendix



## **CHAPTER 4**

### **CONCLUSION**

In conclusion, this project will benefit the society as it will aim to improve public transportation in Ipoh as a start by introducing a software to optimise the bus feeder system services. If the feedback is positive, it can also be used to enhance the services of other modes of public transportation in different areas all over Malaysia.

The project is feasible to be used as a final year project as it is quite challenging. The time allocated to complete the project should be sufficient with the guidance of the project supervisor and the final year project committee. Quite a little bit of self study and effort would be require to have a successful completion of this project.

## REFERENCES

- [1] W. Pepping, "The Shortest Route : Developing a Bicycle Route Planner for the Netherlands," April 2009.
- [2] D. Srinivasan, L. Wei Lup, X. German, E. Taylor, S.H. Ong, S.Chandima Nanayakkara, "Genetic Algorithm base Route Planner for Large Urban Street Networks," in *Congress on Evolutionary Computation*, 2007.
- [3] V. Ganapathy, L. Ooi Chong, S. Chin Yun, "Improved Genetic Algorithm based Optimum Path Planning for Mobile Robot," in *Control, Automation, Robotics and Vision*, Singapore, 2010.
- [4] C. Sanderson, C. Hocaoglu, "Planning Multiple Paths with Evolutionary Speciation," in *Evolutionary Computation.*, 2002, pp. 169-191.
- [5] B. Alfarra, M. Alhanjouri, "Ant Colony versus Genetic Algorithm based on Travelling Salesman Problem," June 1995.
- [6] J. Hanumanthappa, D.H Manjaiah, K. Thippeswamy, "A Study on Contrast and Comparison between Bellman - Ford Algorithm and Djikstra's Algorithm".
- [7] J. Singh, Shivani, "Route Planning in Vanet By Comparitive Study of Algorithms," *International Journal of Advance Research in Computer Science and Software Engineering*, vol. 3, no. 7, July 2013.
- [8] P. Valsalal, S. Hemalatha, "Identification of Optimal Path in Power System Network using Bellman Ford Algorithm," August 2012.
- [9] C. Ting Chen, J. Yang Li, R. Chang Chen, "A Genetic Algorithm for Planning Travel Route with Mimimum Transportation Carbon Footprint," in *Information and Business Intelligence.*, 2011, pp. 57-63.

[10] H. Kanoh, "Dynamic Route Planning for Car Navigation Systems using Virus Genetic Algorithms," *International Journal of Knowledge-based and Intelligent Engineering Systems*, pp. 65-78, 2007.

[11] <https://maps.google.com.my/>.

[12] <http://en.wikipedia.org>.

## APPENDIX

```
N=4; %Number of ants
p=8; %Permissible discrete values
r=[0:1/(p-1):1]; %Defining the initial probability of the roulette system
n=1
i=1:n %Iteration
```

```
%Range representation of the probability division of the roulette system
```

```
x11=[0 1/p];
x12=[1/p 2/p];
x13=[2/p 3/p];
x14=[3/p 4/p];
x15=[4/p 5/p];
x16=[5/p 6/p];
x17=[6/p 7/p];
x18=[7/p 1];
```

```
%Iteration 1
```

```
%Random generating four random values in the range of 0 to 1
```

```
n = 1:N
r0 = rand(N,1)
r1 = r0(1,1)
r2 = r0(2,1)
r3 = r0(3,1)
r4 = r0(4,1)
```

```
rall=[r1 r2 r3 r4]
```

```
%Defining the value of x to be used as the input for y for each individual range
```

```
if r1>=0 && r1<=1/p
x1=3.4;
elseif r1>1/p && r1<=2/p
x1=2.28;
elseif r1>2/p && r1<=3/p
x1=2.79;
elseif r1>3/p && r1<=4/p
x1=2.99;
elseif r1>4/p && r1<=5/p
x1=3.05;
elseif r1>5/p && r1<=6/p
x1=4.59;
elseif r1>6/p && r1<=7/p
x1=0.23;
elseif r1>7/p && r1<=1
x1=4.39;
```

```
end
```

```
if r2>=0 && r2<=1/p
x2=3.4;
elseif r2>1/p && r2<=2/p
x2=2.28;
elseif r2>2/p && r2<=3/p
x2=2.79;
elseif r2>3/p && r2<=4/p
x2=2.99;
```

```

elseif r2>4/p && r2<=5/p
x2=3.05;
elseif r2>5/p && r2<=6/p
x2=4.59;
elseif r2>6/p && r2<=7/p
x2=0.23;
elseif r2>7/p && r2<=1
x2=4.39;

end

if r3>=0 && r3<=1/p
x3=3.4;
elseif r3>1/p && r3<=2/p
x3=2.28;
elseif r3>2/p && r3<=3/p
x3=2.79;
elseif r3>3/p && r3<=4/p
x3=2.99;
elseif r3>4/p && r3<=5/p
x3=3.05;
elseif r3>5/p && r3<=6/p
x3=4.59;
elseif r3>6/p && r3<=7/p
x3=0.23;
elseif r3>7/p && r3<=1
x3=4.39;

end

if r4>=0 && r4<=1/p
x4=3.4;
elseif r4>1/p && r4<=2/p
x4=2.28;
elseif r4>2/p && r4<=3/p
x4=2.79;
elseif r4>3/p && r4<=4/p
x4=2.99;
elseif r4>4/p && r4<=5/p
x4=3.05;
elseif r4>5/p && r4<=6/p
x4=4.59;
elseif r4>6/p && r4<=7/p
x4=0.23;
elseif r4>7/p && r4<=1
x4=4.39;

end

yall=[x1 x2 x3 x4]

for i = 1
c=2
d=(c*max(yall))/min(yall)
e=1+d
f=(p-1)*(1*0.5)+e
g=(1*0.5)/f
h=e/f
end

```

Iteration needs to be implemented here for pheromone update which seems to pose some problem

```
z = max(yall)

if z == 3.4
disp('AbJ is the most optimal route')
elseif z == 2.28
disp('ACJ is the most optimal route')
elseif z == 2.79
disp('ADJ is the most optimal route')
elseif z == 2.99
disp('AEJ is the most optimal route')
elseif z == 3.05
disp('AFJ is the most optimal route')
elseif z == 4.59
disp('AGJ is the most optimal route')
elseif z == 0.23
disp('AHJ is the most optimal route')
elseif z == 4.39
disp('AIJ is the most optimal route')
end
```

Coding for the wholesome implementation of four ants randomly choosing random combinations of the possible route from the Ipoh KTM station throughout all nine stations and back to the Ipoh KTM station at the end

```
Ant1a = randperm(9,1);
Ant1b = [1 2 3 4 5 6 7 8 9];
Ant1b (:,Ant1a) = [];
ROUTE1 = Ant1b (randperm(numel(Ant1b))));
```

```
Ant2a = randperm(9,1);
Ant2b = [1 2 3 4 5 6 7 8 9];
Ant2b (:,Ant2a) = [];
ROUTE2 = Ant2b (randperm(numel(Ant2b))));
```

```
Ant3a = randperm(9,1);
Ant3b = [1 2 3 4 5 6 7 8 9];
Ant3b (:,Ant3a) = [];
ROUTE3 = Ant3b (randperm(numel(Ant3b))));
```

```
Ant4a = randperm(9,1);
Ant4b = [1 2 3 4 5 6 7 8 9];
Ant4b (:,Ant4a) = [];
ROUTE4 = Ant4b (randperm(numel(Ant4b))));
```

```
a = [10 5.7 11.6 10.3 1.7 5.9 9.2 11.4 3.8];
z = Ant1a;
y = Ant2a;
x = Ant3a;
v = Ant4a;
a1 = a(Ant1a);
a2 = a(Ant2a);
a3 = a(Ant3a);
a4 = a(Ant4a);
```

```
if z == 1;y == 1;x == 1;v == 1;
b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
```

```

z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 2;y == 2;x == 2;v == 2;
b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 3;y == 3;x == 3;v == 3;
b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 4;y == 4;x == 4;v == 4;
b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];
z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 5;y == 5;x == 5;v == 5;
b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 6;y == 6;x == 6;v == 6;
b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 7;y == 7;x == 7;v == 7;
b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];
z1 = ROUTE1(1);

```

```

y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 8;y == 8;x == 8;v == 8;
b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
elseif z == 9;y == 9;x == 9;v == 9;
b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
z1 = ROUTE1(1);
y1 = ROUTE2(1);
x1 = ROUTE3(1);
v1 = ROUTE4(1);
b1 = b(z1);
b2 = b(y1);
b3 = b(x1);
b4 = b(v1);
end

if z1 == 1;y1 == 1;x1 == 1;v1 == 1;
b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 2;y1 == 2;x1 == 2;v1 == 2;
b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 3;y1 == 3;x1 == 3;v1 == 3;
b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 4;y1 == 4;x1 == 4;v1 == 4;
b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];

```



```

z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 5;y1 == 5;x1 == 5;v1 == 5;
b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 6;y1 == 6;x1 == 6;v1 == 6;
b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 7;y1 == 7;x1 == 7;v1 == 7;
b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 8;y1 == 8;x1 == 8;v1 == 8;
b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
elseif z1 == 9;y1 == 9;x1 == 9;v1 == 9;
b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
z2 = ROUTE1(2);
y2 = ROUTE2(2);
x2 = ROUTE3(2);
v2 = ROUTE4(2);
c1 = b(z2);
c2 = b(y2);
c3 = b(x2);
c4 = b(v2);
end

if z2 == 1;y2 == 1;x2 == 1;v2 == 1;

```

```

b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 2;y2 == 2;x2 == 2;v2 == 2;
b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 3;y2 == 3;x2 == 3;v2 == 3;
b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 4;y2 == 4;x2 == 4;v2 == 4;
b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 5;y2 == 5;x2 == 5;v2 == 5;
b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 6;y2 == 6;x2 == 6;v2 == 6;
b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 7;y2 == 7;x2 == 7;v2 == 7;
b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];

```

```

z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 8;y2 == 8;x2 == 8;v2 == 8;
b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
elseif z2 == 9;y2 == 9;x2 == 9;v2 == 9;
b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
z3 = ROUTE1(3);
y3 = ROUTE2(3);
x3 = ROUTE3(3);
v3 = ROUTE4(3);
d1 = b(z3);
d2 = b(y3);
d3 = b(x3);
d4 = b(v3);
end

if z3 == 1;y3 == 1;x3 == 1;v3 == 1;
b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 2;y3 == 2;x3 == 2;v3 == 2;
b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 3;y3 == 3;x3 == 3;v3 == 3;
b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 4;y3 == 4;x3 == 4;v3 == 4;

```

```

b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 5;y3 == 5;x3 == 5;v3 == 5;
b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 6;y3 == 6;x3 == 6;v3 == 6;
b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 7;y3 == 7;x3 == 7;v3 == 7;
b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 8;y3 == 8;x3 == 8;v3 == 8;
b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
elseif z3 == 9;y3 == 9;x3 == 9;v3 == 9;
b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
z4 = ROUTE1(4);
y4 = ROUTE2(4);
x4 = ROUTE3(4);
v4 = ROUTE4(4);
e1 = b(z4);
e2 = b(y4);
e3 = b(x4);
e4 = b(v4);
end

```

```

if z4 == 1;y4 == 1;x4 == 1;v4 == 1;
    b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
    z5 = ROUTE1(5);
    y5 = ROUTE2(5);
    x5 = ROUTE3(5);
    v5 = ROUTE4(5);
    f1 = b(z5);
    f2 = b(y5);
    f3 = b(x5);
    f4 = b(v5);
elseif z4 == 2;y4 == 2;x4 == 2;v4 == 2;
    b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
    z5 = ROUTE1(5);
    y5 = ROUTE2(5);
    x5 = ROUTE3(5);
    v5 = ROUTE4(5);
    f1 = b(z5);
    f2 = b(y5);
    f3 = b(x5);
    f4 = b(v5);
elseif z4 == 3;y4 == 3;x4 == 3;v4 == 3;
    b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
    z5 = ROUTE1(5);
    y5 = ROUTE2(5);
    x5 = ROUTE3(5);
    v5 = ROUTE4(5);
    f1 = b(z5);
    f2 = b(y5);
    f3 = b(x5);
    f4 = b(v5);
elseif z4 == 4;y4 == 4;x4 == 4;v4 == 4;
    b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];
    z5 = ROUTE1(5);
    y5 = ROUTE2(5);
    x5 = ROUTE3(5);
    v5 = ROUTE4(5);
    f1 = b(z5);
    f2 = b(y5);
    f3 = b(x5);
    f4 = b(v5);
elseif z4 == 5;y4 == 5;x4 == 5;v4 == 5;
    b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
    z5 = ROUTE1(5);
    y5 = ROUTE2(5);
    x5 = ROUTE3(5);
    v5 = ROUTE4(5);
    f1 = b(z5);
    f2 = b(y5);
    f3 = b(x5);
    f4 = b(v5);
elseif z4 == 6;y4 == 6;x4 == 6;v4 == 6;
    b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
    z5 = ROUTE1(5);
    y5 = ROUTE2(5);
    x5 = ROUTE3(5);
    v5 = ROUTE4(5);
    f1 = b(z5);
    f2 = b(y5);
    f3 = b(x5);
    f4 = b(v5);
elseif z4 == 7;y4 == 7;x4 == 7;v4 == 7;

```

```

b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];
z5 = ROUTE1(5);
y5 = ROUTE2(5);
x5 = ROUTE3(5);
v5 = ROUTE4(5);
f1 = b(z5);
f2 = b(y5);
f3 = b(x5);
f4 = b(v5);
elseif z4 == 8;y4 == 8;x4 == 8;v4 == 8;
b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
z5 = ROUTE1(5);
y5 = ROUTE2(5);
x5 = ROUTE3(5);
v5 = ROUTE4(5);
f1 = b(z5);
f2 = b(y5);
f3 = b(x5);
f4 = b(v5);
elseif z4 == 9;y4 == 9;x4 == 9;v4 == 9;
b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
z5 = ROUTE1(5);
y5 = ROUTE2(5);
x5 = ROUTE3(5);
v5 = ROUTE4(5);
f1 = b(z5);
f2 = b(y5);
f3 = b(x5);
f4 = b(v5);
end

if z5 == 1;y5 == 1;x5 == 1;v5 == 1;
b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
z6 = ROUTE1(6);
y6 = ROUTE2(6);
x6 = ROUTE3(6);
v6 = ROUTE4(6);
g1 = b(z6);
g2 = b(y6);
g3 = b(x6);
g4 = b(v6);
elseif z5 == 2;y5 == 2;x5 == 2;v5 == 2;
b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
z6 = ROUTE1(6);
y6 = ROUTE2(6);
x6 = ROUTE3(6);
v6 = ROUTE4(6);
g1 = b(z6);
g2 = b(y6);
g3 = b(x6);
g4 = b(v6);
elseif z5 == 3;y5 == 3;x5 == 3;v5 == 3;
b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
z6 = ROUTE1(6);
y6 = ROUTE2(6);
x6 = ROUTE3(6);
v6 = ROUTE4(6);
g1 = b(z6);
g2 = b(y6);
g3 = b(x6);
g4 = b(v6);

```

```

elseif z5 == 4;y5 == 4;x5 == 4;v5 == 4;
    b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];
    z6 = ROUTE1(6);
    y6 = ROUTE2(6);
    x6 = ROUTE3(6);
    v6 = ROUTE4(6);
    g1 = b(z6);
    g2 = b(y6);
    g3 = b(x6);
    g4 = b(v6);
elseif z5 == 5;y5 == 5;x5 == 5;v5 == 5;
    b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
    z6 = ROUTE1(6);
    y6 = ROUTE2(6);
    x6 = ROUTE3(6);
    v6 = ROUTE4(6);
    g1 = b(z6);
    g2 = b(y6);
    g3 = b(x6);
    g4 = b(v6);
elseif z5 == 6;y5 == 6;x5 == 6;v5 == 6;
    b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
    z6 = ROUTE1(6);
    y6 = ROUTE2(6);
    x6 = ROUTE3(6);
    v6 = ROUTE4(6);
    g1 = b(z6);
    g2 = b(y6);
    g3 = b(x6);
    g4 = b(v6);
elseif z5 == 7;y5 == 7;x5 == 7;v5 == 7;
    b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];
    z6 = ROUTE1(6);
    y6 = ROUTE2(6);
    x6 = ROUTE3(6);
    v6 = ROUTE4(6);
    g1 = b(z6);
    g2 = b(y6);
    g3 = b(x6);
    g4 = b(v6);
elseif z5 == 8;y5 == 8;x5 == 8;v5 == 8;
    b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
    z6 = ROUTE1(6);
    y6 = ROUTE2(6);
    x6 = ROUTE3(6);
    v6 = ROUTE4(6);
    g1 = b(z6);
    g2 = b(y6);
    g3 = b(x6);
    g4 = b(v6);
elseif z5 == 9;y5 == 9;x5 == 9;v5 == 9;
    b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
    z6 = ROUTE1(6);
    y6 = ROUTE2(6);
    x6 = ROUTE3(6);
    v6 = ROUTE4(6);
    g1 = b(z6);
    g2 = b(y6);
    g3 = b(x6);
    g4 = b(v6);
end

```

```

if z6 == 1;y6 == 1;x6 == 1;v6 == 1;
    b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
elseif z6 == 2;y6 == 2;x6 == 2;v6 == 2;
    b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
elseif z6 == 3;y6 == 3;x6 == 3;v6 == 3;
    b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
elseif z6 == 4;y6 == 4;x6 == 4;v6 == 4;
    b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
elseif z6 == 5;y6 == 5;x6 == 5;v6 == 5;
    b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
elseif z6 == 6;y6 == 6;x6 == 6;v6 == 6;
    b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);

```



```

elseif z6 == 7;y6 == 7;x6 == 7;v6 == 7;
    b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
elseif z6 == 8;y6 == 8;x6 == 8;v6 == 8;
    b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
elseif z6 == 9;y6 == 9;x6 == 9;v6 == 9;
    b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
    z7 = ROUTE1(7);
    y7 = ROUTE2(7);
    x7 = ROUTE3(7);
    v7 = ROUTE4(7);
    h1 = b(z7);
    h2 = b(y7);
    h3 = b(x7);
    h4 = b(v7);
end

if z7 == 1;y7 == 1;x7 == 1;v7 == 1;
    b = [0 15.3 22.4 25.5 12.7 17.6 12.8 16.4 7.6];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);
elseif z7 == 2;y7 == 2;x7 == 2;v7 == 2;
    b = [15 0 6.1 17.3 7.1 9.9 16.8 16.8 7.6];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);
elseif z7 == 3;y7 == 3;x7 == 3;v7 == 3;
    b = [22 6.1 0 11.8 7.5 8.6 14.8 16.3 14.7];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);

```

```

    i4 = b(v8);
elseif z7 == 4;y7 == 4;x7 == 4;v7 == 4;
    b = [19 13.9 12.8 0 8.8 4.8 12.3 13.8 12.6];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);
elseif z7 == 5;y7 == 5;x7 == 5;v7 == 5;
    b = [10.7 7.4 7.7 8.8 0 4.8 7.2 10.8 5.5];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);
elseif z7 == 6;y7 == 6;x7 == 6;v7 == 6;
    b = [16.6 9.9 8.6 4.4 4.9 0 8.4 9.9 8.7];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);
elseif z7 == 7;y7 == 7;x7 == 7;v7 == 7;
    b = [10.4 14.6 14.7 12.3 7.9 8.2 0 5.6 14.5];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);
elseif z7 == 8;y7 == 8;x7 == 8;v7 == 8;
    b = [15.6 16.8 16.8 17.7 10.8 10.4 5.6 0 19.7];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);
elseif z7 == 9;y7 == 9;x7 == 9;v7 == 9;
    b = [7.2 7.6 14.7 12.8 4.3 8.5 12.9 14.1 0];
    z8 = ROUTE1(8);
    y8 = ROUTE2(8);
    x8 = ROUTE3(8);
    v8 = ROUTE4(8);
    i1 = b(z8);
    i2 = b(y8);
    i3 = b(x8);
    i4 = b(v8);

```

```

end

if z8 == 1;y8 == 1;x8 == 1;v8 == 1;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 2;y8 == 2;x8 == 2;v8 == 2;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 3;y8 == 3;x8 == 3;v8 == 3;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 4;y8 == 4;x8 == 4;v8 == 4;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 5;y8 == 5;x8 == 5;v8 == 5;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 6;y8 == 6;x8 == 6;v8 == 6;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 7;y8 == 7;x8 == 7;v8 == 7;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 8;y8 == 8;x8 == 8;v8 == 8;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
elseif z8 == 9;y8 == 9;x8 == 9;v8 == 9;
    c = [12.6 5.6 7.8 10.5 2.1 5.4 10.4 11.6 3.6];
    j1 = c(z8);
    j2 = c(y8);
    j3 = c(x8);
    j4 = c(v8);
end

ROUTE1 = [Ant1a ROUTE1];
ROUTE2 = [Ant2a ROUTE2];
ROUTE3 = [Ant3a ROUTE3];

```

ROUTE4 = [Ant4a ROUTE4];

Path1 = ROUTE1

Path2 = ROUTE2

Path3 = ROUTE3

Path4 = ROUTE4

DistanceTravelled1 = a1 + b1 + c1 + d1 + e1 + f1 + g1 + h1 + i1 + j1

DistanceTravelled2 = a2 + b2 + c2 + d2 + e2 + f2 + g2 + h2 + i2 + j2

DistanceTravelled3 = a3 + b3 + c3 + d3 + e3 + f3 + g3 + h3 + i3 + j3

DistanceTravelled4 = a4 + b4 + c4 + d4 + e4 + f4 + g4 + h4 + i4 + j4

Profit1 = 135 - DistanceTravelled1

Profit2 = 135 - DistanceTravelled2

Profit3 = 135 - DistanceTravelled3

Profit4 = 135 - DistanceTravelled4