

AUTOMATIC SCHEDULER:
A NEW APPROACH FOR
SCHEDULING
RIDES

by

AMEY SUDHIR BHALERAO

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2012

Copyright © by Amey Sudhir Bhalerao 2012

All Rights Reserved

ACKNOWLEDGEMENTS

I would like to thank Dr. Hao Che for allowing me to be a part of his excellent research group. I am also thankful to him for showing the trust and confidence in me and for supporting me during the entire process. His valuable suggestions helped me improve my problem solving skills and also helped me evolve as a research candidate. I am thankful to Dr. Mohan Kumar for his courses that helped me understand the way in which an algorithm needs to be designed and how a computer programming problem needs to be solved. I am also thankful to Dr. Chris Ding because his course Data Modeling and Design Techniques helped me in designing solutions to my problems. I appreciate that Dr. Mohan Kumar and Dr. Chris Ding took out their valuable time for being a part of my thesis panel.

Lastly I would like to thank my father Sudhir Bhalerao, my mother Smita Bhalerao, my sister Pallavi Bhalerao and my friends Vishal Subramani, Shekhar Menon and Samaya Madhavan for their unconditional support and encouragement to pursue my dreams and for providing me the highest level of confidence.

April 12, 2012

ABSTRACT

AUTOMATIC SCHEDULER
A NEW APPROACH FOR
SCHEDULING
RIDES

Amey Sudhir Bhalerao, M.S.

The University of Texas at Arlington, 2012

Supervising Professor: Hao Che

Scheduling plays an important role in a person's life. Even computers use the concept of scheduling when it comes to processing jobs. Scheduling is important in cases normally when there is large data to process and manual scheduling is not an easy option to go for. A new concept of developing a scheduler has been discussed in this thesis. The algorithm was designed keeping in mind the Para Transit agencies that operate on federal funds to provide cheap transport service to the clients.

The algorithm presented here is developed with the intention to reduce the delay faced by the client when making use of cheap transit service and at the same time the algorithm is designed to make sure that it will service the maximum number of clients with fewer requirements of buses.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
LIST OF ILLUSTRATIONS.....	vii
LIST OF TABLES	viii
Chapter	Page
1. INTRODUCTION.....	1
1.1 Background and Motivation.....	1
1.1.1 What is Para Transit Agency and What is it Not?	1
1.2 Automatic Scheduler	2
2. RELATED WORK.....	5
3. AUTO SCHEDULER	9
3.1 Design Approach.....	9
3.1.1 Approach.....	9
3.1.1.1 Practical Working of Auto Scheduler	12
3.2 The Algorithm	31
3.3 Experimental Setup and Analysis	32
4. CONCLUSION	35
5. FUTURE WORK.....	36
REFERENCES.....	37
BIOGRAPHICAL INFORMATION	39

LIST OF ILLUSTRATIONS

Figure	Page
3.1 Map Showing all the Sources and their Corresponding Destinations.....	13
3.2 Figure Representing Source 1 and Destination 1	13
3.3 Figure Representing Source 2 and Source 3 and Destination 2 and Destination 3	14
3.4 Figure Representing Source 4 and Source 5 and Destination 4 and Destination 5	14
3.5 Figure Representing Source 6 and Destination 6	14
3.6 Figure Representing Source 7 and Destination 7	14
3.7 Figure Representing Source 8 and Destination 8	14
3.8 Figure Representing Source 9 and Destination 9	15
3.9 Figure Representing Source 10 and Destination 10	14
3.10 Map Showing the Route Taken by Bus 1.....	26
3.11 Map Showing the Route Taken by Bus 2.....	27
3.12 Map Showing the Route Taken by Bus 3.....	28
3.13 Map Showing the Route Taken by Bus 4.....	29
3.14 Map Showing the Route Taken by Bus 5.....	30
3.15 Graph Showing Average Waiting Time	33
3.16 Graph Showing Average Number of Passengers Serviced.....	33
3.17 Graph Showing Number of Requests v/s Number of Buses.....	34
3.18 Graph Showing Comparison of Existing and Proposed Approach	34

LIST OF TABLES

Table	Page
3.1 A Table Displaying Client Pickup Times	15
3.2 A Table Representing Distance Between s1 and All Other Locations	16
3.3 A Table Representing Distance Between s2 and All Other Locations	17
3.4 A Table Representing Distance Between s3 and All Other Locations	18
3.5 A Table Representing Distance Between s4 and All Other Locations	19
3.6 A Table Representing Distance Between s5 and All Other Locations	20
3.13 A Table Representing Distance Between s6 and All Other Locations	21

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Scheduling is an important feature, be it job scheduling or scheduling an event or scheduling transport service for clients. The presented thesis work focuses on the development of an automatic scheduler that can be used for scheduling transport services to the clients. The automatic scheduler presented here is developed keeping in mind Para - Transit agencies.

1.1.1 What is Para Transit Agency and what it is not?

The Para - Transit agencies provide transportation service to passengers who are unable to use traditional public transit. Their duties include coordinating transportation providers (i.e. bus fleets, volunteers etc.) reporting to transit funders and managing transportation requests for their passenger clients. Many of these agencies act as brokers or call centers, taking passenger requests and working to fill the clients' needs by assigning the request to the appropriate provider. Several of the agencies manage programs that are funded with federal dollars or insurance and they must track performance indicators and file reports. The service provided by para transit agencies is different from the service provided by a metro. Metros have fixed route and fixed schedule. Para transit service does not have fixed route and schedule. The buses delivered by agencies are sent to serve the clients at the requested location at the requested time. There can be multiple pickups and drop offs for a given bus and each pickup has a requested pickup time associated with it. This makes the route and pickup time dynamic for service provided by para transit agencies. The service provided by para transit agencies should not be confused or compared with taxi service [16]. The services provided by such agencies are normally community services [16]. The para transit service resembles taxi service when it comes to picking up passengers. Similar to taxi service, para transit service also picks up a client from the requested location. The difference between the two services is that taxi is a

dedicated service where as para transit is a shared service. In taxi, once a passenger is picked up, the taxi need not go to other places to pick up some other passengers, however the para transit agencies need to consider pickups of other clients as well. Hence the service provided by para transit agencies is bound by pickup time and the number of clients being picked up and at the same time measures need to be taken to reduce the delay faced by the clients. The vehicle used by para transit agencies is normally 10 to 15 seater.

The current systems used by Para – Transit agencies lay more focus on company's interest giving lesser importance to clients. For instance, if a client 'A' wants to schedule a ride at 10:00 am, then the client must be ready to board any bus that reaches his place in between 09:45 to 10:15 am [16][14]. The other case might be that the bus might be visiting the neighborhood of client 'A' at 09:30 am, then in this case the client 'A' will be requested to board the bus by 09:30 am as it is not feasible to send a vehicle across the city to the clients place [14][16]. There are different scheduling approaches followed depending upon what the business is actually interested in. Unlike typical vehicle scheduling problem transit cannot have the property of visiting nodes exactly once. In some cases when source 1 is picked, source 2 might be closer to source 1 than destination 1. In that case the bus has to pickup source 1, source 2 and then come back and drop source 1 at destination 1 if destination 1 lies near source 1 but farther than source 2. Transit scheduling does not have criteria of leaving and returning to the same depot after serving the clients. Need was felt to develop a scheduler that will bridge the gap between agencies interest and customer satisfaction. The automatic scheduler presented here gives a way to schedule maximum number of clients in the same transport vehicle with minimum delay thereby taking care of company's interest as well as customer's interest.

1.2 Automatic Scheduler

Scheduling rides is the most important work for any para - transit agency. So when such an agency is looking for para – transit software, automatic scheduler is the module that can provide this feature and make scheduling easier for transit managers. More efficient the auto scheduler more profits can an organization make. Normally all the para – transit agencies

provide services at a very low cost. Sometimes there are people who voluntarily work for para – transit agencies. The clients of such agencies don't expect service of the quality of a taxi or metro, but they do expect a reliable service, where in an agency is in a position to pickup or drop a client at the time requested by the client [14]. However the research shows that this expectation of client is not fulfilled in many cases. The clients normally have to face delay. Delay is calculated by multiplying the trip time by 1.5 to 3. So if it takes 10 minutes to reach from point 'A' to point 'B', then the current systems normally assume that it is a better deal even if they drop a client in 15 to 30 minutes. Thus auto scheduler presented here is designed in such a way that it can meet the expectations of the clients and at the same time it is able to keep the agencies running by increasing their revenue.

On a high level, the auto scheduler takes input from the database. This input typically includes:

- Trip Id: Each client is assigned a unique trip id.
- Pickup Address: The address is geo – coded, meaning that latitude and longitude values are given to the auto scheduler.
- Destination Address: Similar to pickup address, destination address is also geo – coded.
- Pickup Time: For scheduling a ride, the transit manager needs the pickup time of the client.

For the purpose of scheduling, some approaches make use of map service providing software. Some of the well known map services providing software are Bing Maps, Google Maps, Yahoo Maps, Policy Map, MapQuest, Map Source Navigator, Delorme Street USA etc. Of these Bing Maps and Google Maps are more popular. The auto scheduler described in the following section is customer centric. It tries to strike a balance between the company's revenue expectations and customer's service expectations. To achieve this concept of dividing the time into one hour block is used. This has two advantages, firstly it has the client entries sorted according to time, so the clients who are up in the sorted list will be considered first and at the

same time the clients who are at the end of the list also do not lose importance. Secondly, the amount of data required to process is very small since the auto scheduler has to process the entries present in one hour window. Normally the agencies get 40 -150 requests daily. So even if an average of 100 clients is considered, the number of entries required to process is just around 8. In many of the current approaches, a database of all the trips is maintained. Client's source and destination coordinates are passed on to a system that calculates the distance between the source and the destination and at the same time it also calculates the time required for this journey. These details are stored in the database.

CHAPTER 2

RELATED WORK

Scheduling is an important aspect in many industries. The role of scheduling is very crucial in any kind of industry. This section deals with the areas that require scheduling. Concept of scheduling is used in the field of computer science to schedule job for processing [8]. Scheduling resources is very important in this process. Scheduling is used in transport systems for ensuring delivery of goods [7][9]. Scheduling is an important activity in public transport system including different transport media like bus, metro, taxi, cruise etc. [1][10] where this concept is used for scheduling rides for passengers. Scheduling has its application in entertainment industry. Here its application is to schedule movie tickets or to scheduling dates for celebrities. In corporate environment, scheduling is used for scheduling meetings, scheduling events, scheduling conference halls for events etc [6]. Food and beverage industry are require scheduling system to deliver food and beverages to different parts of country, state and city or sometimes even to different parts of the world [7][9]. Scheduling algorithms find place in educational institutes where they are used for course scheduling, time table scheduling, classroom scheduling etc [6]. The construction industry also requires scheduling to ensure delivery of construction tools and equipments, scheduling manpower (resources) etc. Airline industry uses scheduling to schedule flights for passengers. Sports industry uses scheduling approach to schedule matches considering availability of team and players and for scheduling venue for matches. Scheduling finds its application even in television industry where it is used for scheduling sitcoms, movies and sports – deals with event scheduling. In addition to these domains, scheduling has a crucial role in military operations, where it is used for scheduling troops to different locations, scheduling delivery of arms and armaments. Government agencies make use of scheduling for planning and development plans or for scheduling polls.

This shows that scheduling is used in many domains to achieve different goals. So it means that the role of scheduling is different in every domain and that in turn means that the way scheduling is carried out in different domains is different as well. The way scheduling can be used to schedule events or to schedule classrooms, cannot be used for scheduling public transport system. The main reason for this is the parameters required for scheduling in each domain are different. The role of scheduling is important but it also varies depending on the intention for which it is used. For example, in classroom scheduling, class timings cannot be adjusted frequently, whereas in transportation industry, delivery of goods is adjusted many times due to various factors. So this shows that although scheduling is used in many domains, the way it is carried out is different.

The following section describes the algorithms that are used for transit scheduling. The approach discussed here deals with taxi scheduling [1]. This approach takes into consideration five predictors that are used to split the trip: Location zones, hourly window, day of week window, hourly day of week window and peak period window. There are two types under this, static zoning and dynamic zoning. In static zoning, the whole city is divided into squares of equal size. For each zone size, the historical trips need to be partitioned. They are partitioned according to each predictor. The average trip time is thus found out and stored in the database. Average trip durations and fares for each partition are stored. Based on this a trip prediction table is built. Whenever the trip duration and fare of an incoming request is to be calculated, the prediction table is looked up. So when a new request comes in, it is sent to prediction table for processing. The whole table consisting of many records is queried to find out the exact trip that considers all the five predictors as well. If an exact match is found, then a result is returned, else an unsuccessful prediction is returned. In the dynamic approach minimum zone size is found out to give a specified number of historical trips. The approach involves finding k nearest neighbors. This yields k similar trips from the historical dataset. The dynamic approach involves building distance metric between any two trips. Distance is calculated using GPS coordinates. In contrast to the static zoning it does not require explicit zones. It only requires k similar trips.

The performance of static zoning is more efficient than dynamic zoning as individual trip detail is not stored in the memory. However in this case a historical dataset set containing millions of records is searched to find out the exact match. Lot of processing needs to be done. In static zoning, smaller zones enable better prediction but they tend to a lot of database partitions. On the other hand bigger zones tend to generate less database partitions but they comparatively yield less accurate predictions. The system lacks error filters. Real world data needs to be cleaned before use. Proper partition of data is important.

In [5] for scheduling rides GPS has been used. The system is based on mobile phones with GPS functionality. The passenger who wants to make use of this service, requests for taxi using software installed on the phone. The drivers are also equipped with 3G/GPRS enabled phones and they can update their status. The request sent by the passenger is sent to the central server. The server has the location of all the passengers that are requesting the service and it also has the location of all the taxis along with their status, viz. vacant or occupied. The server distributes the information of the passenger and the taxi and helps in providing decision support for both, the passenger and the taxi driver. For the purpose of scheduling, it is assumed that drivers compete with each other to serve the passengers. In this model, it is assumed that when the location of passengers is given, the taxi driver will choose the passenger and the route to be taken independently. So matching of taxi and passengers is done such that none of them is able to find a better match than the existing one. In this algorithm all the elements in the payoff matrix of $\{E_{ij}\}$ into decreasing order. Then each item is considered from first to last. For each element E_{ij} where 'i' denotes taxi and 'j' denotes a passenger, it is checked whether a taxi or a passenger has been allocated or not. If neither of taxi or passenger is allocated, then this pair of 'ij' is created. So it mean taxi 'i' is paired with passenger 'j'. However in this scheduling approach, decision support is only provided for taxi drivers. It is not mandatory for the taxi driver to service the assigned passenger or even in the case if the assigned passenger is serviced, then it is not mandatory for the taxi to follow the assigned route. The scheduling approach

discussed here assumes that live traffic conditions or individual experience about taking the routes or favoritism for a specific area may influence scheduling.

The scheduling algorithm that is used for para transit services has been discussed here. This algorithm is a perfect match for comparing the proposed work since the existing approach and the proposed approach both aim at scheduling rides for para transit agencies. In this, the auto scheduler needs input information that typically consists of the client's trip id, his pickup and destination co-ordinates, his departure time, flexibility about time and travel time. Once this input is received, the scheduler passes the coordinates of all the clients to third party map service provider software [4]. Once this software receives the inputs it will return the distance between pickup and destination addresses along with the journey time. The system divides the whole city into polygons. Once the scheduler is able to find out the addresses of the clients lying in the same polygon, it tries to extract the time at which they wish to depart from their respective location. If the auto scheduler finds out that there are clients leaving the same polygon at the same time and departing towards the same polygon, it immediately schedules them on the same bus. If it does not find a match between the clients, it assigns a different bus for the given number of clients. There are a few shortcomings with this approach. If the clients lie in the same polygon and if their departure times are same (+/- 10 minutes) then they are scheduled for the same bus. So this case shows that a customer's trip has been delayed by ten minutes. If the departure times are not within a range of 10 minutes, this algorithm will assign a separate bus for all the clients lying in that region. Even if it assigns a bus for a client, the client will get delayed. Third, if a client 'A' lies just outside polygon1, but his departure time and destination polygon is same as that of say clients 'B', 'C' and 'D'. Then in this case auto scheduler will schedule 'A' on another bus since 'A' lies in a different polygon. So even if it was possible to schedule 'A' with 'B', 'C' and 'D', a separate bus was scheduled for 'A'.

CHAPTER 3
AUTO SCHEDULER
3.1 Design Approach

In this thesis, an auto scheduler has been developed for para transit agencies to provide an output that will achieve minimum delay and at the same time will try to use minimum number of buses to schedule clients. So a system has been developed to strike a balance between the company's revenue expectations and client's expectations. The approach requires the input data like trip id, pickup and destination address (geo coded values), pickup time and board time. The previous approaches on some occasions required user consent for taking multiple rides to reach the destination. So it also required input from the user asking his willingness to take multiple rides. The approach discussed here eliminates this inconvenience caused to the customer while on transit. So this is one of areas where the new approach has an edge over the current approach.

3.1.1 Approach

The proposed approach has a concept called as time window. Time is divided into 12 blocks with each block constituting a window of one hour [1]. Time window has been used to divide the clients and sort them according to their pickup times. So every window contains clients whose pickup lies in the given one hour block. So the scheduler just needs to schedule rides for the clients in one time window. The auto scheduler first sorts the entries according to time. So there will be in all 12 windows as the agencies normally operate for 12 hours a day. Then the auto scheduler tries to schedule rides for the people starting with the first time window of the day. Normally these agencies operate on fixed times, say starting from 7 in the morning to 7 o'clock in the evening. So initially the auto scheduler will get the data of all the clients lying in the time window 7:00 am to 8:00 am. At this point the scheduler is just interested in getting the clients lying in a given time frame. Once it gets these entries, it will take the first entry from

the given time window and it will try to find out the nearest neighbor. Nearest neighbor is a neighbor who has the minimum distance from point 'A', if point 'A' is considered to be the point of reference. A map service is used to find out the distance between the pickup and destination coordinates. All these distances are stored in a distance matrix. First entry of the day is considered as pivot element and all the distances between the first entry and the incoming entries are calculated. At the same time track is kept to check whether the time exceeds sixty minutes. If time exceeds sixty minutes, then the last entered source is removed. If time does not exceed sixty minutes, then the same bus is scheduled for this entry. This process repeats for all the entries in a given time window.

The explanation and dry running of the algorithm is presented here. For the entire explanation $s_1, s_2, s_3 \dots s_n$ will denote source/pickup address of a client whereas $d_1, d_2, d_3 \dots d_n$ will denote the destination address of the corresponding client. $s_1 \text{---} 4 \rightarrow s_2$ indicates that the distance between s_1 and s_2 is 4 miles. Assume there are 5 clients in the time window 7:00 am to 8:00 am. s_1 over here denotes the first client in the time window 7:00 – 8:00 am. s_2 denotes the second client in the time window 7:00 – 8:00 am and so on. So in other words s_1, s_2, s_3 are the notations that we use to identify the clients who appear in a time window in this sequence because of their pickup times. These are sorted entries. The main concept over here is finding the nearest neighbor in the given time frame. So for that whenever a new source comes in, its distance will be compared with the pivot source. The pivot source is nothing but the first entry in the time window. To explain the working of auto scheduler for three entries let's assume our sources are s_1, s_2 and s_3 . Let d_1, d_2 and d_3 be their corresponding destinations. The distance matrix already has the distance between a source and its corresponding destination. So auto scheduler will get the distance between s_1 and d_1, s_2 and d_2, s_3 and d_3 and s_n and d_n . Initially the auto scheduler decides to pick up s_1 and drop it at d_1 , since s_1 is the only client being considered currently. Now when s_2 comes in, the distance between s_1 and s_2 is extracted and compared with the distance between s_1 and d_2 . If $s_1s_2 < s_1d_2$ (i.e. if the distance between s_1 and s_2 is smaller than s_1 and d_2), then s_2 is placed before d_1 . So the order of pickup and drop

off is like, $s_1 \rightarrow s_2 \rightarrow d_1$. Now as client with source s_2 is on board, his drop off point is also considered. The destination of client 2 will be d_2 . When placing a destination, the auto scheduler will just compare its distance with all the entries after its corresponding source. So the way it works is, now the scheduler has the order like $s_1 \rightarrow s_2 \rightarrow d_1$. When the scheduler wants to place d_2 , it will find out the distance between s_2 and d_1 and compare it with s_2d_2 . If $s_2d_2 > s_2d_1$, then d_2 will be placed after d_1 . So now the auto scheduler has the order, $s_1 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2$. Now auto scheduler calculates the time it will require if it schedules s_1 and s_2 in the same bus. If this time is less than 60 minutes, it will go ahead and look for client s_3 . Now when s_3 is trying to get into the picture, auto scheduler will calculate the distance between s_1 and s_3 . It will compare this distance with s_1s_2 . If $s_1s_3 < s_1s_2$, the auto scheduler will place s_3 in between s_1 and s_2 and then calculate the distance between s_3 and s_2 . If this be the case, the temporary route will be $s_1 \rightarrow s_3 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2$. Now when d_3 is to be placed, scheduler will compare the distance s_3d_3 with s_3s_2 . If $s_3d_3 > s_3s_2$, then the scheduler will go ahead and find out the distance between s_2d_3 and compare it with s_2d_1 . If $s_2d_3 > s_2d_1$, then scheduler will go ahead and find out the distance between d_1d_3 and compare with d_1d_2 . If $d_1d_3 > d_1d_2$, then it will place d_3 after d_2 . So the temporary route will be $s_1 \rightarrow s_3 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2 \rightarrow d_3$. Now the auto scheduler calculates the time it will require if it schedules all these three clients on the same bus. If the total time is less than 60, it will go ahead and try to schedule s_4 . If the time exceeds 60 minutes, it will drop the most recent entry. So it will drop s_3 and now it will try to schedule s_4 on the same bus. This procedure repeats until all the entries in the window are considered. If at the end the auto scheduler realizes that none of the other entries in combination with s_1s_2 can reach their destination within 60 minutes, then it will just schedule s_1 and s_2 in the same bus. In the next iteration it will take s_3 as the pivot element and repeat the whole procedure to find out how many clients it can schedule in conjunction with s_3 . This process is repeated till all the entries in the given time window are serviced. To sum up, the approach processes the requests in a given one hour block. The entries in the bus are sorted according to their pickup times, so their priority is decided according to pickup time. So a client

who lies first in the time window will be scheduled first, but that does not mean he will also be dropped first. The nearest neighbor concept will decide the location of destination in the queue. so if the destination of client 'A' lies last in the given queue it means that his destination point was farthest and hence the entries lying in between the source of client 'A' and destination of 'A' must be serviced. So there will never be a case where in 'A' is kept on board even if his drop off point was the closest. The sixty minute limit helps to reduce the delay faced by the client as the entries are scheduled keeping in mind sixty minute limit.

3.1.1.1 Practical Working of Auto Scheduler

The following example takes into consideration ten clients who lie in the time window 7:00 am to 8:00 am. The example deals with clients residing in Dallas Fort Worth Metroplex. The naming conventions are as discussed above, viz. s1 will be the first client in the given time window and d1 will be its corresponding destination. Similarly s2 will be the second client in the given time window and d2 will be its destination and so on.

The following map shows the addresses of all the clients that are going to be considered in the explanation. All the maps have been taken from Google Maps.

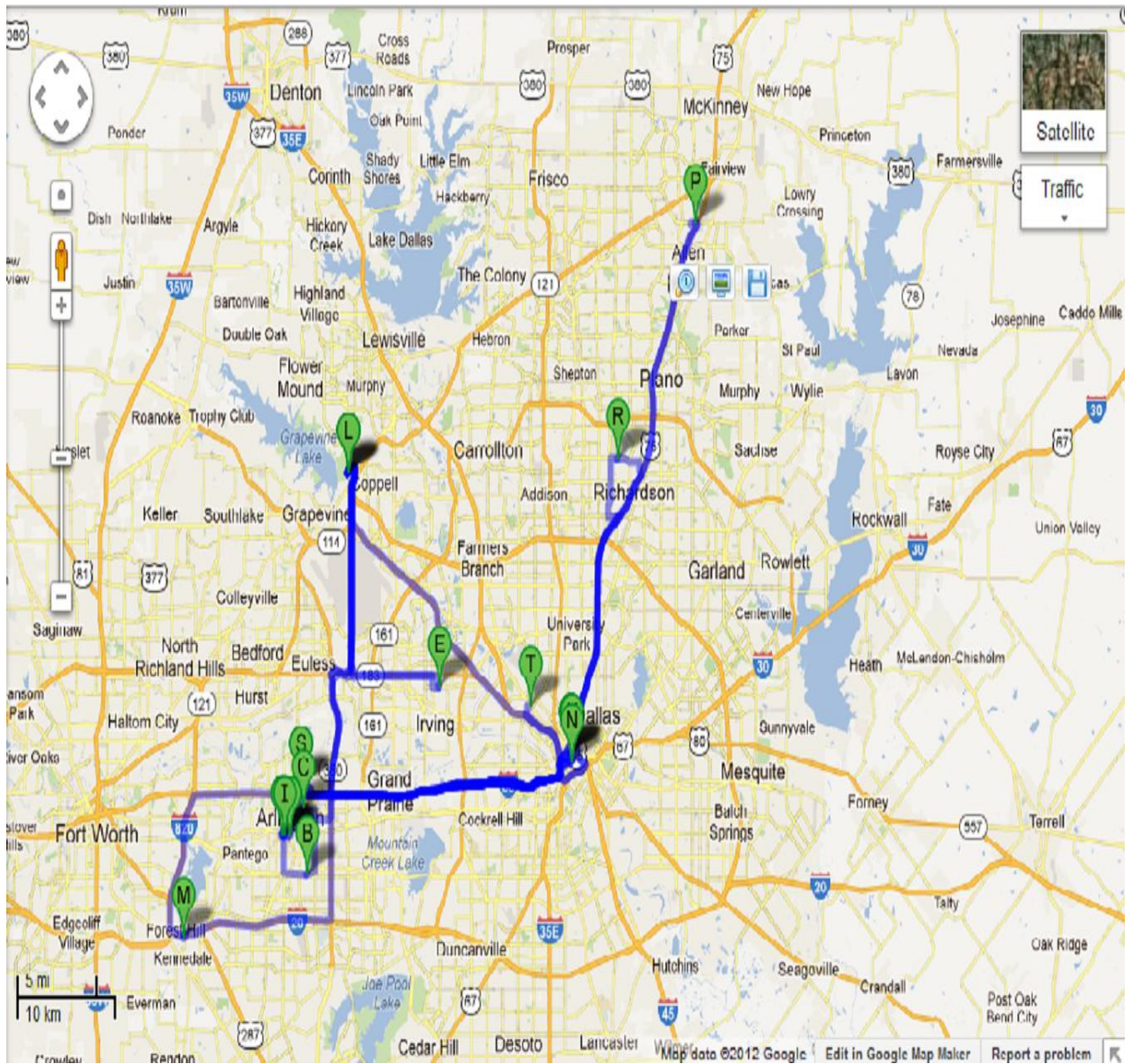


Figure 3.1 Map showing all the sources and their corresponding destinations.

Below are the figures of ten source/destination pairs:



University oaks

Fiesta Mart (Pioneer)

Figure 3.2 Figure Representing Source 1 and Destination 1

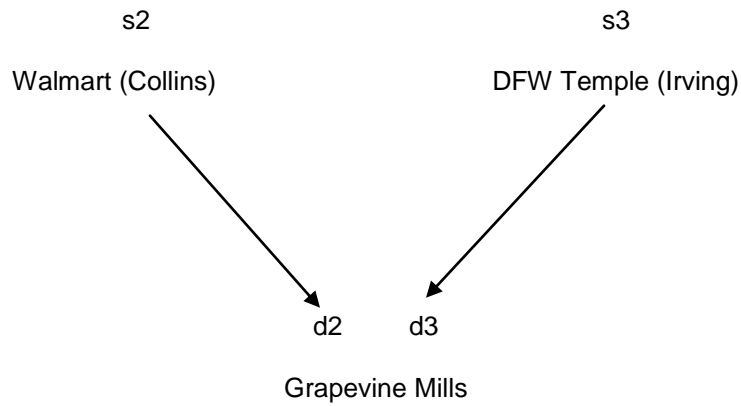


Figure 3.3 Figure Representing Source 2 and Source 3 and Destination 2 and Destination 3

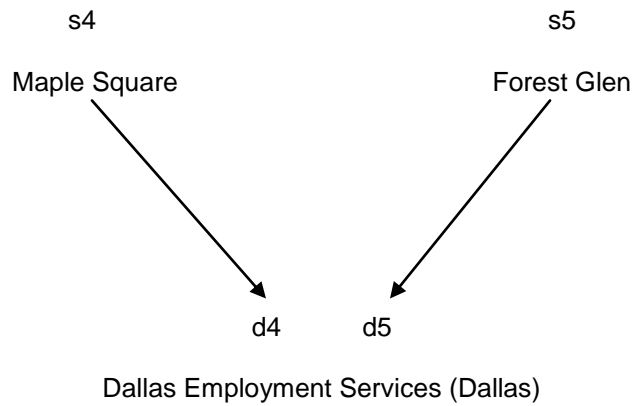


Figure 3.4 Figure Representing Source 4 and Source 5 and Destination 4 and Destination 5

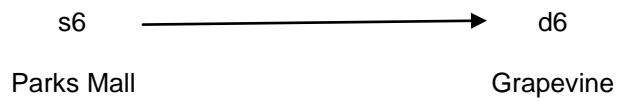


Figure 3.5 Figure Representing Source 6 and Destination 6

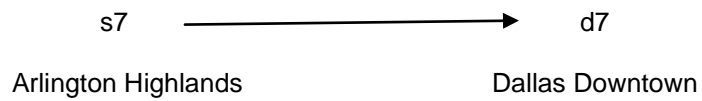


Figure 3.6 Figure Representing Source 7 and Destination 7

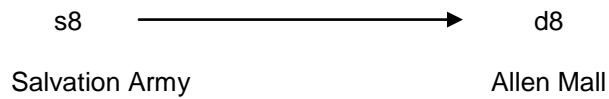


Figure 3.7 Figure Representing Source 8 and Destination 8

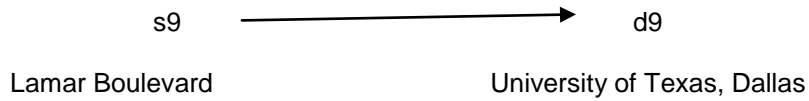


Figure 3.8 Figure Representing Source 9 and Destination 9

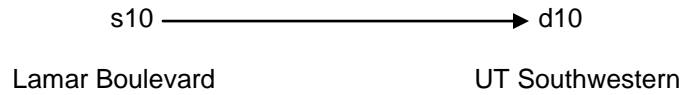


Figure 3.9 Figure Representing Source 10 and Destination 10

The table drawn below shows the clients sorted according to their pickup times.

Table 3.1 A Table Displaying Client Pickup Times

Client	Pickup Time
s1	8:00
s2	8:10
s3	8:12
s4	8:15
s5	8:18
s6	8:30
s7	8:30
s8	8:30
s9	8:42
s10	8:50

The way the proposed approach works is described below. At every stage the algorithm seeks to find the nearest neighbor and it also keeps on calculating the time to see if it's less than sixty minutes.

The auto scheduler starts with building a distance matrix for all the entries in the given time window.

Below is a distance matrix for the entries in the time window 7:00 am to 8:00 am. Also a map showing all the sources and destinations is shown.

Table 3.2 A Table Representing Distance Between s1 and All Other Locations

Source	Source/ Destination	Distance
S1	S1	0
	S2	3
	S3	18
	S4	0.6
	S5	0.4
	S6	21.7
	S7	10.2
	S8	0
	S9	3.4
	S10	3.4
	D1	3.5
	D2	21.7
	D3	21.7
	D4	21.7
	D5	21.7
	D6	21.7
	D7	21.2
	D8	47.5
	D9	36.8
	D10	24.6

Table 3.3 A Table Representing Distance Between s2 and All Other Locations

Source	Source/ Destination	Distance
S2	S2	0
	S3	14.9
	S4	2
	S5	2.4
	S6	19.4
	S7	16.5
	S8	2.4
	S9	1.3
	S10	3.5
	D1	3.5
	D2	20
	D3	20
	D4	19.4
	D5	19.4
	D6	20
	D7	18.8
	D8	45.2
	D9	34.4
	D10	22.3

Table 3.4 A Table Representing Distance Between s3 and All Other Locations

Source	Source/ Destination	Distance
S3	S3	0
	S4	17.2
	S5	17.6
	S6	10.9
	S7	29.7
	S8	17.6
	S9	14.6
	S10	14.6
	D1	18.7
	D2	15.8
	D3	15.8
	D4	10.9
	D5	10.9
	D6	15.8
	D7	11.2
	D8	31.3
	D9	19.9
	D10	7.2

Table 3.5 A Table Representing Distance Between s4 and All Other Locations

Source	Source/ Destination	Distance
S4	S4	0
	S5	0.3
	S6	21.6
	S7	10.1
	S8	0.4
	S9	3.3
	S10	3.3
	D1	3.1
	D2	21.1
	D3	21.1
	D4	21.6
	D5	21.6
	D6	21.1
	D7	21.1
	D8	47.4
	D9	36.7
	D10	24.5

Table 3.6 A Table Representing Distance Between s5 and All Other Locations

Source	Source/ Destination	Distance
S5	S5	0
	S6	21.4
	S7	9.8
	S8	0.2
	S9	3
	S10	3
	D1	3.1
	D2	21.4
	D3	21.4
	D4	21.4
	D5	21.4
	D6	21.4
	D7	20.8
	D8	47.2
	D9	36.4
	D10	24.3

Table 3.7 A Table Representing Distance Between s6 and All Other Locations

Source	Source/ Destination	Distance
S6	S6	0
	S7	32.3
	S8	22
	S9	18.9
	S10	18.9
	D1	22.5
	D2	22.3
	D3	22.3
	D4	0.4
	D5	0.4
	D6	22.3
	D7	0.6
	D8	26.5
	D9	15.8
	D10	4

Similarly tables are built for the remaining sources and destinations.

The algorithm will first extract the first entry of the time window, call it s1. As s1 is the first entry, scheduler will immediately schedule s1 on one bus. So s1d1 will be in the first bus. Now s2 is the next entry in the given time window. Algorithm will calculate the distance s1s2 and compare it with s1d1. If $s1d1 > s1s2$, then s2 will lie after s1, i.e. $s1 \rightarrow s2 \rightarrow d1$, else $s1 \rightarrow d1 \rightarrow s2$. (kindly refer table 2.1) In the given example, $s1 \rightarrow 3.5 \rightarrow d1$, $s1 \rightarrow 3 \rightarrow s2$, $s2 \rightarrow 3.5 \rightarrow d1$, $s1s2 < s1d1$, so we take the temporary route as: $s1 \rightarrow s2 \rightarrow d1$. The total time required for this is 20 minutes. As this is less than 60 minutes, scheduling s1 s2 on the same bus can be considered. Now consider the destination of client 2, i.e. d2. When trying to place the destination, the distance between the destination and the corresponding source is compared with all the entries that lie ahead of the corresponding source. So in the given example, when trying to place d2, the distance s2d2 will be compared with s2d1. Here $s2 \rightarrow 20 \rightarrow d2$, $s2 \rightarrow 3.5 \rightarrow d1$, so $s2d1 < s2d2$. So d2 will lie ahead of d1. (kindly refer table 2.1 and table 2.2) The total time required for this is 47 minutes, and the

temporary order will be $s_1 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2$. Since $47 \leq 60$, scheduler can go ahead and try to schedule s_3 on the same bus. When trying to place s_3 , the distance s_1s_3 will be calculated and compare with s_1s_2 . In the given condition, $s_1-3 \rightarrow s_2$, $s_1-18 \rightarrow s_3$, $s_2-3.5 \rightarrow d_1$, $s_2-14.9 \rightarrow s_3$, $s_3-18.7 \rightarrow d_1$, $d_1-21.5 \rightarrow d_2$ so $s_1s_2 < s_1s_3$, $s_2d_1 < s_2s_3$, but $s_3d_1 < d_1d_2$, therefore s_3 will be placed ahead of d_1 . So the order after s_3 has been finalized is $s_1 \rightarrow s_2 \rightarrow d_1 \rightarrow s_3 \rightarrow d_2$. The total time for this is 67 minutes. Since this is more than 60 minutes, s_3 is removed and s_4 is considered for scheduling. When placing s_4 , the distance s_1s_4 is calculated and compare with s_1s_2 . $s_1-0.4 \rightarrow s_4$, $s_1-3 \rightarrow s_2$, $s_1s_4 < s_1s_2$, so s_4 is placed in between s_1 and s_2 . So the order after this iteration is $s_1 \rightarrow s_4 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2$. The total time required is still less than 60 minutes. So d_4 will be considered. Compare s_4d_4 with s_4s_2 . $S_4-2.0 \rightarrow s_2$, $S_4-21.6 \rightarrow d_4$, $s_2-3.5 \rightarrow d_1$, $d_1-21.5 \rightarrow d_2$, $d_1-21.5 \rightarrow d_4$, $s_4s_2 < s_4d_4$, so now s_2d_4 is compared with s_2d_1 . Since $s_2d_1 < s_2d_4$, d_1d_4 is compared with d_1d_2 . Since d_1d_2 is equal to d_1d_4 , d_4 will be placed after d_2 . The total time required exceeds 60 minutes. So discard s_4 . Now place s_5 . The distance s_1s_5 is calculated and compared with s_1s_2 . $S_1-3 \rightarrow s_2$, $s_1-0.4 \rightarrow s_5$, $s_1s_5 < s_1s_2$, therefore s_5 will be placed in between s_1 and s_2 . Now the distance between s_5 and s_2 needs to be calculated. Rest all the distances will remain the same. $s_5-2.4 \rightarrow s_2$. So the order after s_5 has been placed will be $s_1 \rightarrow s_5 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2$. The total time required is 47. Since $47 \leq 60$, scheduler will try to place d_5 on the same bus. When d_5 is to be placed, the algorithm will get the distance for all the entries that lie ahead of s_5 . So it will initially compare s_5d_5 with s_5s_2 . $S_5-21.4 \rightarrow d_5$, $s_5-2.4 \rightarrow s_2$. Since $s_5s_2 < s_5d_5$, compare s_2d_1 with s_2d_5 . $S_2-3.5 \rightarrow d_1$, $s_2-19.4 \rightarrow d_5$. Since $s_2d_1 < s_2d_5$ d_1d_2 is compared with d_1d_5 . $D_1-21.5 \rightarrow d_2$, $d_1-21.5 \rightarrow d_5$. So d_5 will be placed after d_2 . So the order after placing d_5 will be $s_1 \rightarrow s_5 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2 \rightarrow d_5$. The total time for this journey is 77 minutes. Since it exceeds 60 minutes, s_5 will be discarded. Now s_6 needs to be placed. $S_1-21.7 \rightarrow s_6$, $s_1-3 \rightarrow s_2$, since $s_1s_2 < s_1s_6$, d_1d_2 needs to be compared with s_6d_1 . $D_1-21.5 \rightarrow d_2$, $s_6-22.5 \rightarrow d_1$. Since $d_1d_2 < s_6d_1$, s_6 will be placed ahead of d_2 . The temporary order will be $s_1 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2 \rightarrow s_6$. The time required for this journey is 74 minutes. Since it exceeds 60 minutes, no attempt to place d_6 will be made. S_6 will be ignored and s_7 will

be tested for scheduling. Compare s_1s_7 with s_1s_2 . $S_1-10.2 \rightarrow s_7$, $s_1-3 \rightarrow s_2$. $S_1s_2 < s_1s_7$. Now compare s_2d_1 with s_2s_7 . $S_2-16.5 \rightarrow s_7$, $s_2-3.5 \rightarrow d_1$. $s_2d_1 < s_2s_7$, so compare d_1d_2 with d_1s_7 . $D_1-10.7 \rightarrow s_7$, $d_1-21.5 \rightarrow d_2$. So s_7 will be placed in between d_1 and d_2 . The temporary order after placing s_7 will be $s_1 \rightarrow s_2 \rightarrow d_1 \rightarrow s_7 \rightarrow d_2$. The time required for this journey is 83 minutes. Since it is more than 60 minutes, we discard s_7 . To place s_8 , the distance s_1s_8 is calculated and compared with s_1s_2 . $S_1-0 \rightarrow s_8$. $S_1-3.5 \rightarrow s_2$. $s_1s_8 < s_1s_2$. So s_8 will be placed in between s_1 and s_2 . The temporary order after placing s_8 will be $s_1 \rightarrow s_8 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2$. The time required for this journey is 44 minutes. Since it is less than 60 minutes, attempts are made to place d_8 . To place d_8 , all the entries that lie ahead of s_8 are compared. Initially s_8d_8 will be compared with s_8s_2 . $S_8-47.2 \rightarrow d_8$, $s_8-2.4 \rightarrow s_2$. $s_8s_2 < s_8d_8$, so s_2d_1 will be compared with s_2d_8 . $S_2-3.5 \rightarrow d_1$, $s_2-45.2 \rightarrow d_8$. $s_2d_1 < s_2d_8$, so now d_1d_8 will be compared with d_1d_2 . $D_1-21.5 \rightarrow d_2$, $d_1-47.1 \rightarrow d_8$. Since $d_1d_2 < d_1d_8$, d_8 will be placed ahead of d_2 . The temporary order will be $s_1 \rightarrow s_8 \rightarrow s_2 \rightarrow d_1 \rightarrow d_2 \rightarrow d_8$. The time required for this journey is 78 minutes. Since this is greater than 60, we discard s_8 . Now place s_9 . For this s_1s_9 will be compared with s_1s_2 . $s_1-3.4 \rightarrow s_9$, $s_1-3 \rightarrow s_2$. s_9 will be placed in after s_2 . The order after placing s_9 will be $s_1 \rightarrow s_2 \rightarrow s_9 \rightarrow d_1 \rightarrow d_2$. The total time required for this is 57 minutes. Since it is less than 60, attempts are made to place d_9 . To place d_9 , all the distances and entries after s_9 are calculated and compared. Compare s_9d_9 with s_9d_1 . $S_9-34.1 \rightarrow d_9$, $s_9-4.6 \rightarrow d_1$. $s_9d_1 < s_9d_9$. So compare d_1d_9 with d_1d_2 . $D_1-38.8 \rightarrow d_9$, $d_1-21.5 \rightarrow d_2$. Since $d_1d_2 < d_1d_9$, d_9 will be placed after d_2 . The order after placing d_9 will be, $s_1 \rightarrow s_2 \rightarrow s_9 \rightarrow d_1 \rightarrow d_2 \rightarrow d_9$. The total time required for this journey is 89 minutes. Since this is greater than 60 minutes, s_9 is discarded. To place s_{10} , s_1s_{10} is compared with s_1s_2 . $S_1-3.4 \rightarrow s_{10}$, $s_1-3 \rightarrow s_2$. Now compare s_2d_1 with s_2s_{10} . $S_2-1.3 \rightarrow s_{10}$, $s_2-3.5 \rightarrow d_1$, so s_{10} will be placed in between s_2 and d_1 . The temporary order after placing s_{10} will be $s_1 \rightarrow s_2 \rightarrow s_{10} \rightarrow d_1 \rightarrow d_2$. The time required for this journey is 57 minutes. Since it is less than 60 minutes, attempts are made to place d_{10} . To place d_{10} , all the entries after s_{10} will be considered. Compare $s_{10}d_{10}$ with $s_{10}d_1$. $S_{10}-22 \rightarrow d_{10}$, $s_{10}-4.6 \rightarrow d_1$. $s_{10}d_1 < s_{10}d_{10}$, so compare d_1d_2 with d_1d_{10} . $D_1-24.1 \rightarrow d_{10}$, $d_1-21.5 \rightarrow d_2$. Since $d_1d_2 < d_1d_{10}$, d_{10} will be

placed ahead of d2. So the order will be $s1 \rightarrow s2 \rightarrow s10 \rightarrow d1 \rightarrow d2 \rightarrow d10$. The time required for this journey is 83 minutes. Since it is greater than 60, s10 is eliminated. At this point all the entries in the time window 7:00 am to 8:00 am are compared with the pivot element s1. The clients that can be scheduled safely on one bus are s1 and s2.

The output will be, Bus 1: $s1 \rightarrow s2 \rightarrow d1 \rightarrow d2$

After s1 and s2 have been scheduled for bus number 1, s3 becomes the pivot element. Now all the distances will be calculated with respect to s3. Initially since s3 is the only entry s3d3 will be scheduled for bus number 2. When s4 tries to get in, s3s4 will be compared with s3d3. $S3 \rightarrow 17.2 \rightarrow s4$, $s3 \rightarrow 15.8 \rightarrow d3$. Since $s3d3 < s3s4$, s4 will be placed after d3. So the order will be, $s3 \rightarrow d3 \rightarrow s4$. The time required for this journey is 53 minutes. Since it is less than 60 minutes, attempts will be made to place d4. In order to place d4, all the entries that lie ahead of s4 will be compared. Since s4 is the last entry, d4 will be placed after s4. So the order after putting d4 will be $s3 \rightarrow d3 \rightarrow s4 \rightarrow d4$. The total time required for this journey is 82 minutes. Since it is greater than 60, s4 is eliminated. To place s5, compare s3s5 with s3d3. $S3 \rightarrow 17.6 \rightarrow s5$, $s3 \rightarrow 15.8 \rightarrow d3$. Since $s3d3 < s3s5$, s5 will be placed after d3. So the order will look like $s3 \rightarrow d3 \rightarrow s5$. The time required for this is 52 minutes. D5 will be placed s5. The total time required for this will be 80 minutes. Since it is greater than 60, s5 will be discarded. Now to place s6, s3s6 will be compared with s3d3. $s3 \rightarrow 10.9 \rightarrow s6$, $s3 \rightarrow 15.8 \rightarrow d3$. Since $s3s6 < s3d3$, s6 will be placed in between s3 and d3. The time required for this is 43 minutes. So the order will be $s3 \rightarrow s6 \rightarrow d3$. Now to place d6, s6d6 will be compared with s6d3. $S6 \rightarrow 22.3 \rightarrow d6$, $s6 \rightarrow 22.3 \rightarrow d3$. So d6 will be placed ahead of d3. The time required for this journey is 43 minutes. So s3 and s6 can be safely scheduled on the same bus. The order after scheduling s3 and s6 will be $s3 \rightarrow s6 \rightarrow d3 \rightarrow d6$. Since the total time required for this journey is less than 60 minutes, attempts will be made to place s7. To place s7, s3s7 will be compared with s3s6. $S3 \rightarrow 29.7 \rightarrow s7$, $s3 \rightarrow 10.9 \rightarrow s6$. Since $s3s6 < s3s7$, s6s7 will be compared with s6d3. $S6 \rightarrow 32.3 \rightarrow s7$, $s6 \rightarrow 22.3 \rightarrow d3$. So now s7 will be placed ahead of d6. The order will be $s3 \rightarrow s6 \rightarrow d3 \rightarrow d6 \rightarrow s7$. The total time required for this is 79 minutes. So s7 will be discarded. On placing s8 the time exceeds 60 minutes. So s8 will be discarded. Similarly with

s9 and s10, the time goes beyond 60 minutes, so s9 and s10 are discarded. So the bus 2 will contain s3 and s6.

Output for Bus 2

s3->s6->d3->d6

Now s4 will act as the pivot. All the distances will be calculated with respect to s4. Following the steps described above, s4, s5 and s8 will be scheduled for bus 3.

Output for Bus 3

S4->s5->s8->d4->d5->d8

Similarly bus 4 will contain s7 and s10.

Output for Bus 4

s7->s10->d7->d10

On the same lines s9 will be placed on bus 5. Since s9 is the only client who does not gel well with other clients in the given time window, a separate bus will be scheduled for him.

Output for Bus 5

s9->d9

Thus for scheduling these ten clients five buses are required in the proposed approach. For the similar dataset seven buses were required in the previous approach. This shows that the number of buses required to service for same number of clients with the same source destination pairs is less in the proposed approach. The average delay in the proposed approach was found to be around 9 minutes, whereas in the previous approach it was around 11 minutes. The following figure shows the path taken by Bus1.

'A' denotes the source of client 1 i.e. s1, 'B' denotes the source of client 2, i.e. s2, 'C' denotes the destination of client 1 i.e. d1 and 'D' denotes the destination of client 2 i.e. d2.

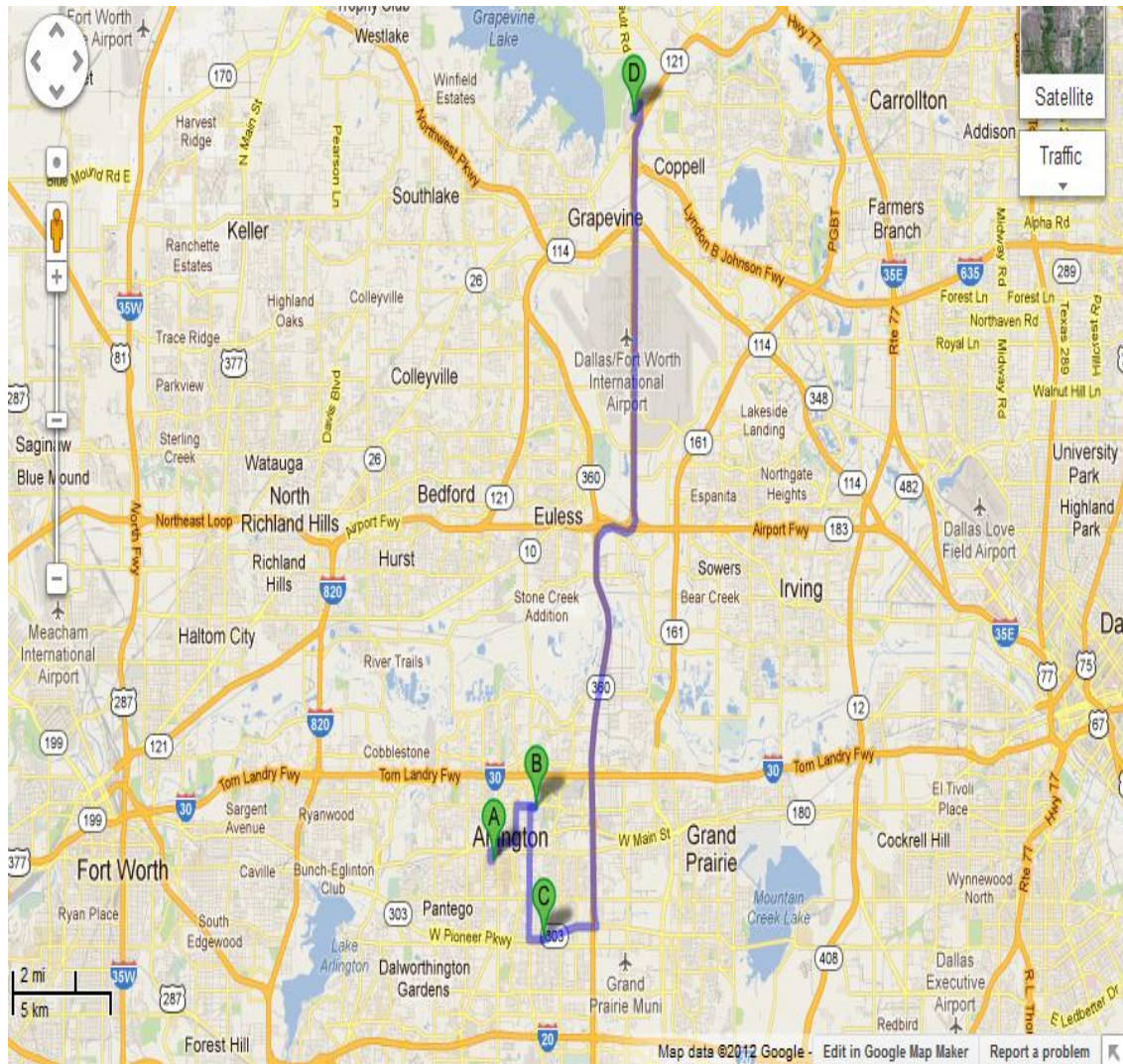


Figure 3.10 Map Showing the Route Taken by Bus 1

The following figure shows the path taken by bus 2.

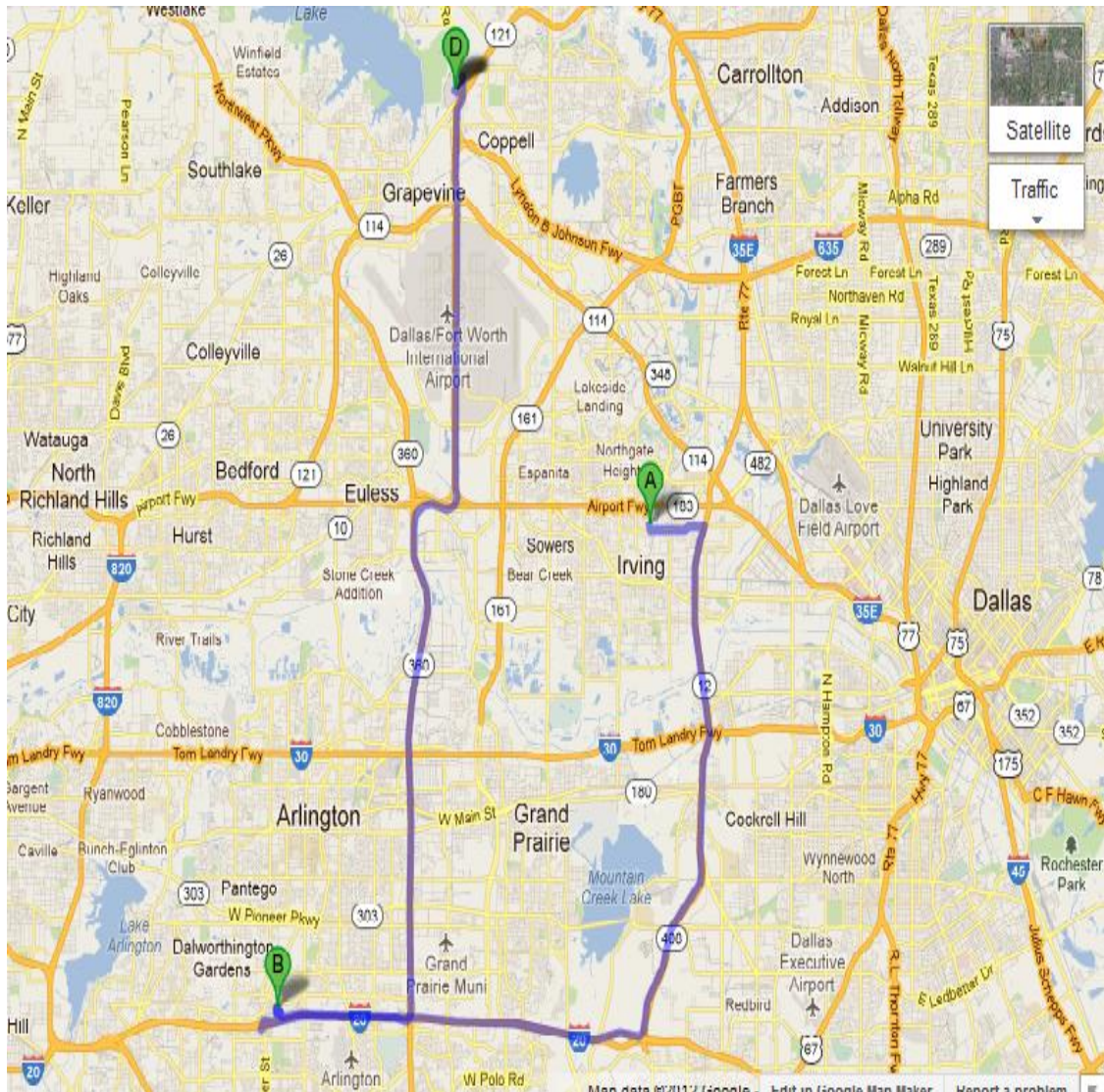


Figure 3.11 Map Showing the Route Taken by Bus 2

The following figure shows the path taken by bus 3.

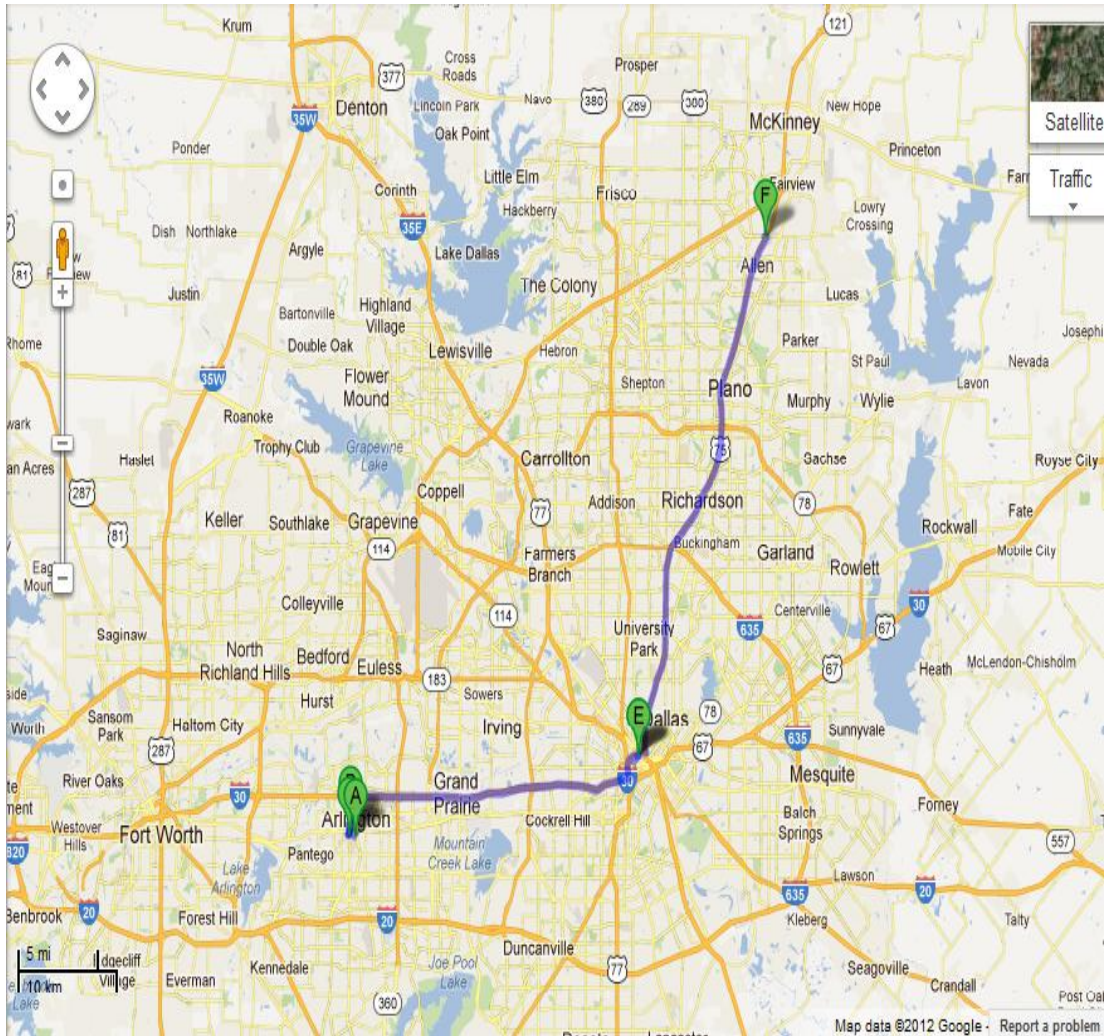


Figure 3.12 Map Showing the Route Taken by Bus 3

The following figure shows the path taken by bus 4.

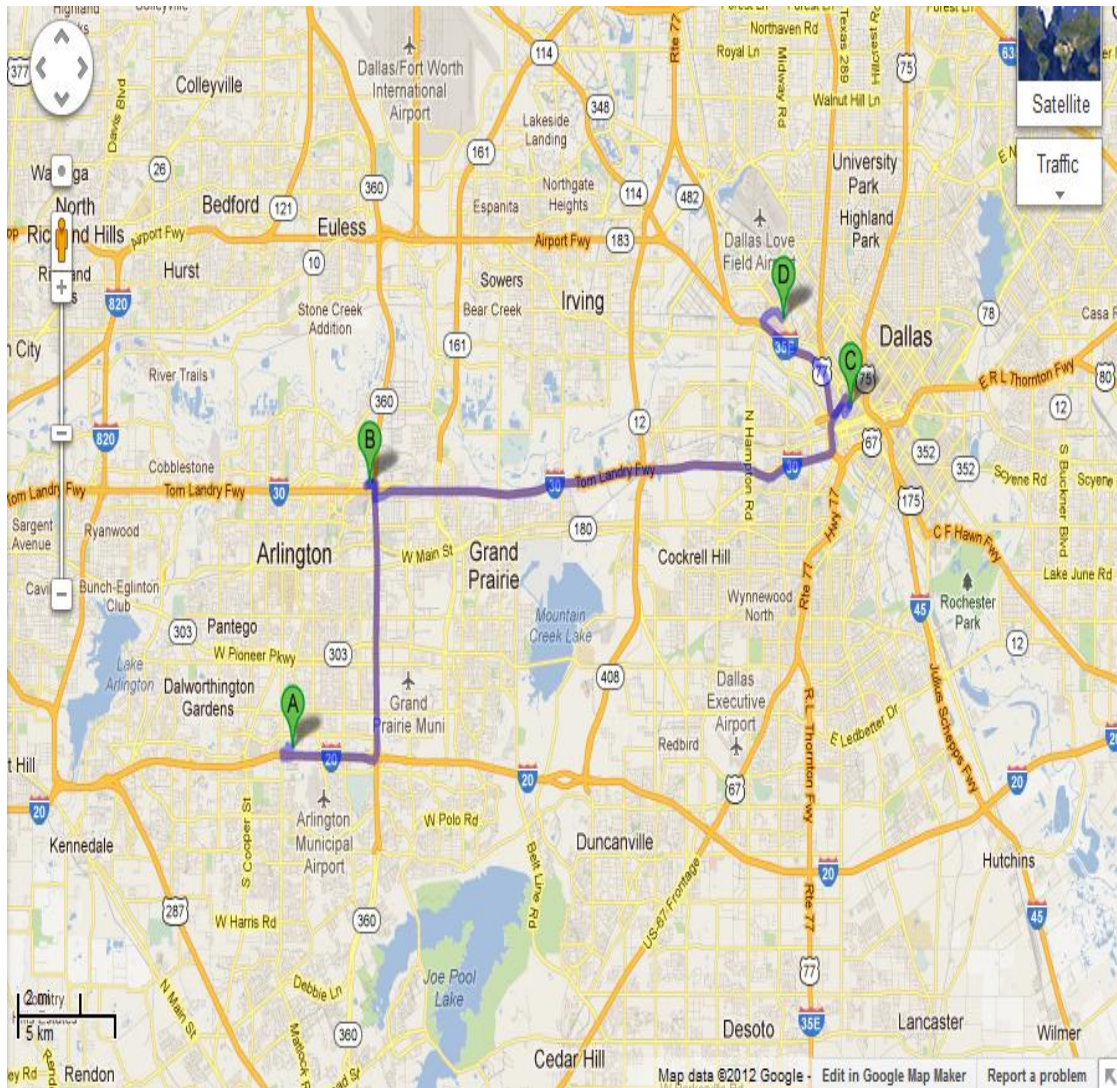


Figure 3.13 Map Showing the Route Taken by Bus 4

The following figure shows the path taken by bus 5.

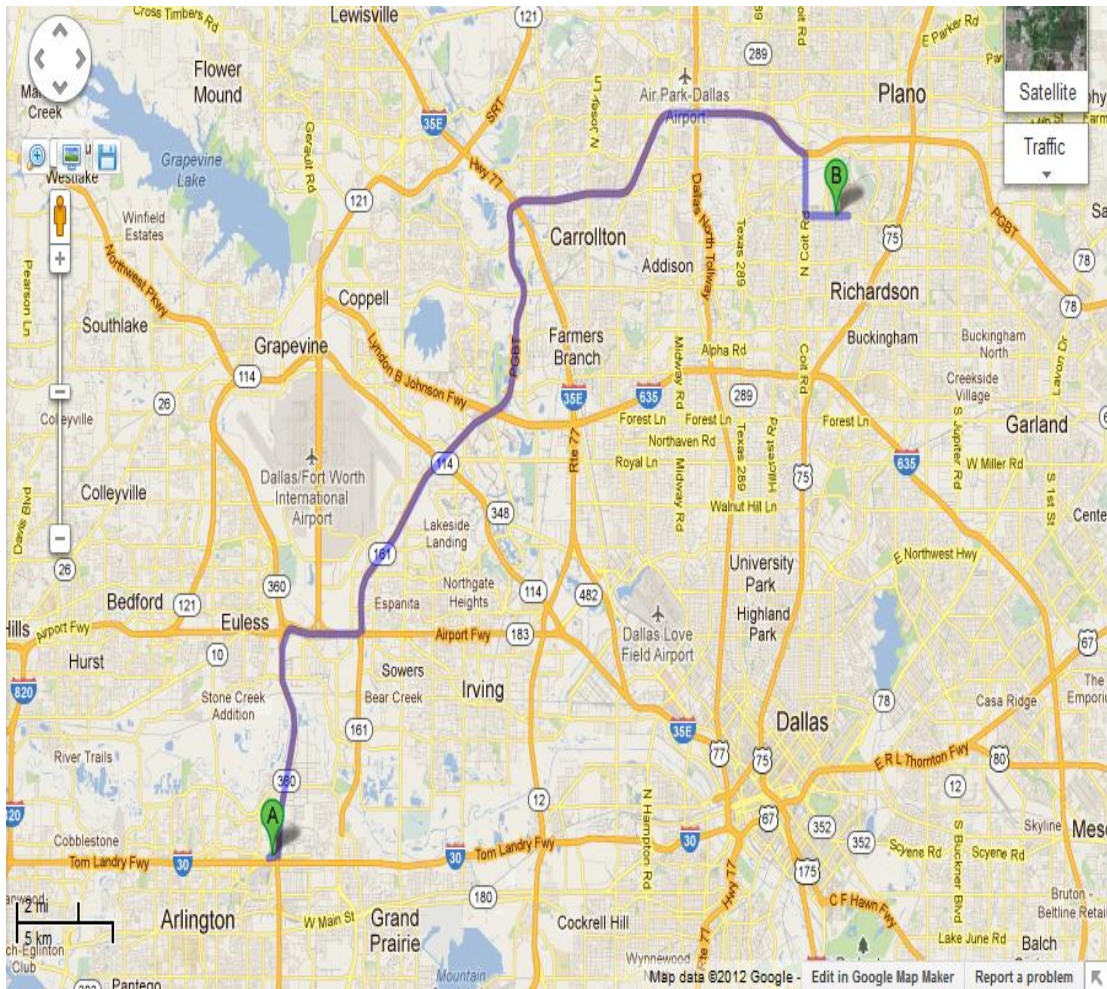


Figure 3.14 Map Showing the Route Taken by Bus 5

3.2 The Algorithm

1. Sort (entries)
2. $s \leftarrow$ Extract entries based on window size
3. for $i = 0$ to number of entries
 - a. count = 0;
 - b. for $j = 0$ to number of entries
 - b.i. Distance_Matrix[i][count++] = Calculate_Distance(s[0,i],s[0,j])
 - b.ii. Distance_Matrix[i][count++] = Calculate_Distance(s[0,i],s[1,j])
 - b.iii j++

```

        c. i++
4. time = 0;
5. for i = 0 to number of entries
    a. Queue Q = new Queue()
    b. time += Q.AddSource(s[0][i])
    c. if(time>60)
        c.i. Q.RemoveLastEnteredSource()
        c.ii. continue
    d. time += Q.AddDestination(s[1][i])
    e. if(time>60)
        e.i. Q.RemoveLastEnteredSource ()
        e.ii. Q.RemoveLastEnteredDestination()
        e.iii. continue
    f. i++
6. RemoveSandDFromEntries()
7. BusSchedule(Q)

```

The algorithm starts with sorting the requests for the given day with respect to pickup time. Then the algorithm extracts the entries for the given time frame. For the number of requests in the given time window, the algorithm calculates the distance required to go from every source to every source and destination. This information is stored in a distance table for easy retrieval. Once the distance table is built, the algorithm builds a queue to add the sources and destinations to schedule them for a ride. Addsource method correctly places the source in the queue and returns the time required for the total journey. If the time required is less than sixty minutes, then the algorithm tries to fit the destination of the corresponding source in the queue. If the time does not exceed sixty minutes, then the source and destination is placed in a queue, they are removed from the request pool and then the scheduler tries to schedule more requests on the same bus. If the time exceeds sixty minutes on adding a source, then the last

entered source is removed. If the tie does not exceed sixty minutes on addition of a source, but exceeds when a destination is added, then the last entered source and destination are removed from the queue.

3.3 Experimental Setup and Analysis

For the purpose of analysis a dataset with varied client information was used as input. The input included client's pickup address, destination address and pickup time. In the initial phases a database containing the trip distances and trip time was used [4]. It was used to obtain the results of the scheduler. The results obtained from the experiments were verified the by doing manual testing. After the manual testing passed, the algorithm was run on different data sets to find out the average number of clients scheduled on a bus and to determine the average waiting time a client has to face.

The system was tested for four hundred entries to find the average number of buses required to schedule the clients. Normally the transit agencies service 40 – 150 clients every day. Smaller agencies normally service not more than 40 clients while big agencies normally service not more than 150 clients. Research on trip time showed that average trip time of a customer is around 20 minutes [16]. This factor was considered while placing a sixty minute hold while scheduling buses. If only the clients with average trip time request for scheduling, one bus can service minimum two clients and maximum three clients in an hour. The number of clients serviced per hour was found to be 2.7. The seating capacity of para transit is generally between 10 and 15. For the purpose of calculating the average waiting time a dataset of 250 entries was taken. The standard deviation as depicted in graph [3.14] was observed to be 4.124. A one tailed graph showing 95% confidence interval for the average delay is shown below. The delay time in the existing system varies between 15 to 30 minutes. In the proposed system the average waiting time varies between 8.2 to 15 minutes.

The graph showing 95% confidence interval from which it can be inferred that the delay faced by the client cannot exceed 15 minutes.

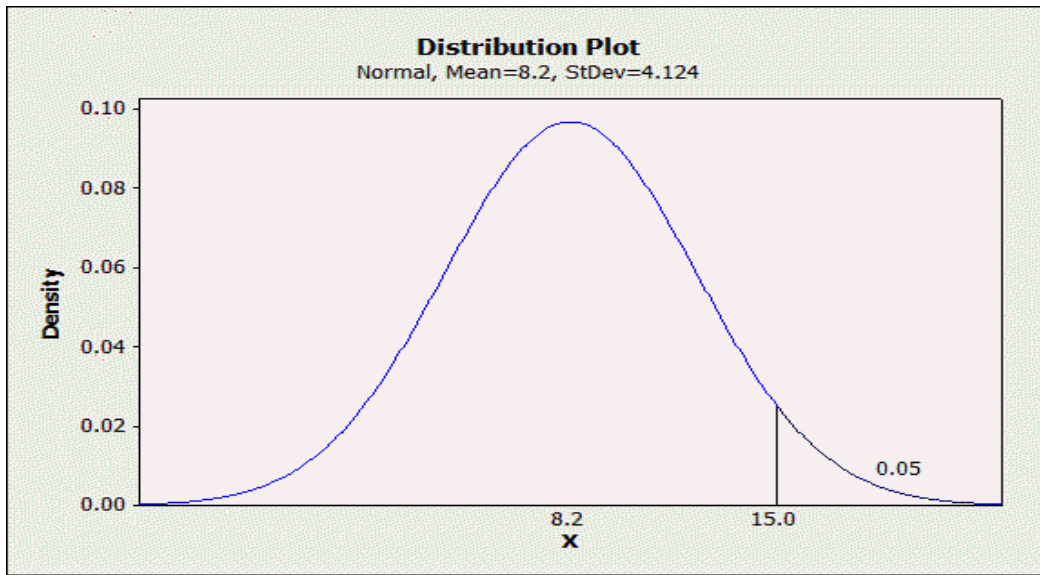


Figure 3.15 Graph Showing Average Waiting Time

The graph showing 95% confidence interval for the number of clients serviced by the bus is shown below.

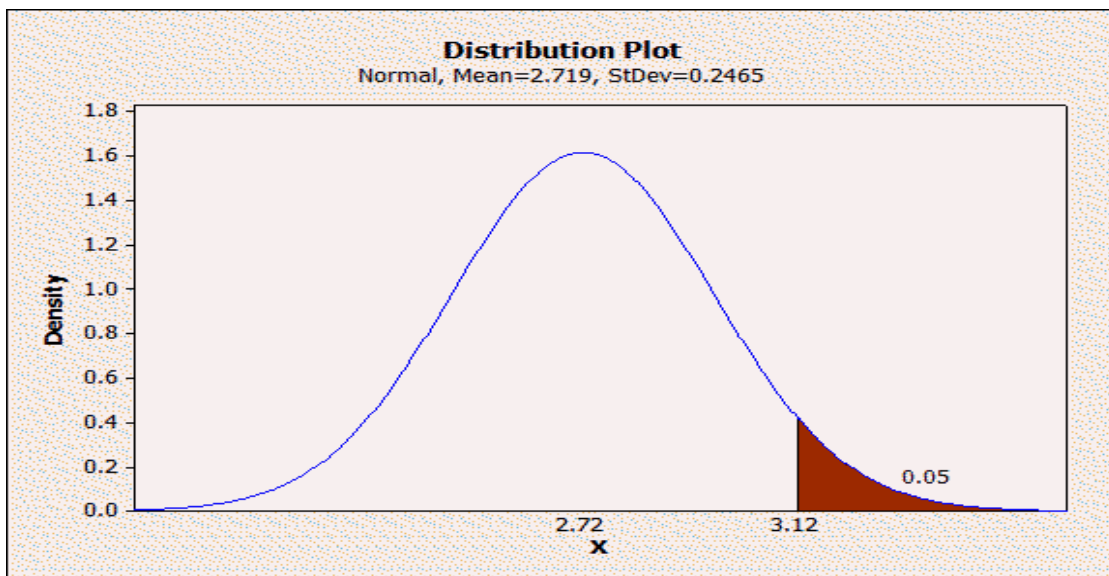


Figure 3.16 Graph Showing Average Number of Passengers Serviced

A graph displaying the number of requests v/s number of buses is shown below.

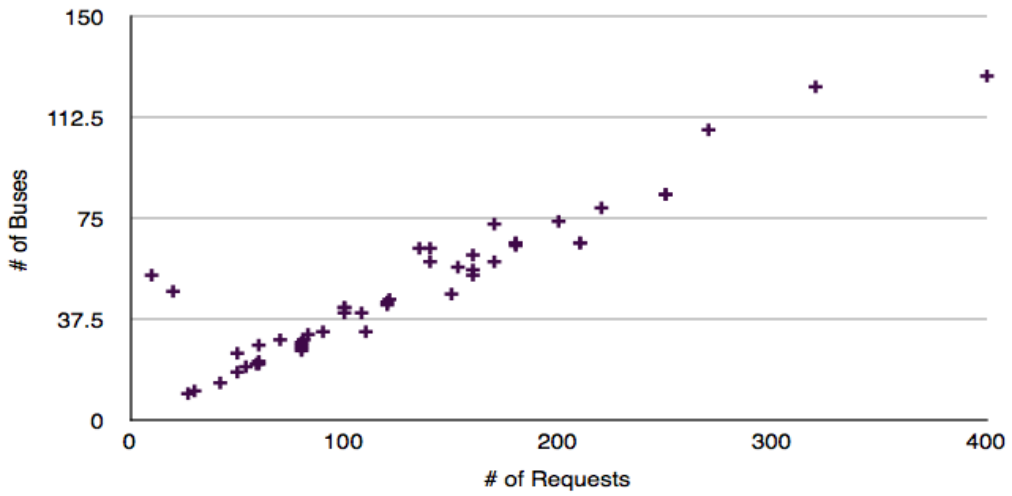


Figure 3.17 Graph Showing Number of Requests v/s Number of Buses

The following graph shows the time required for a client to reach his destination. Comparison has been made with the existing system and the proposed system.

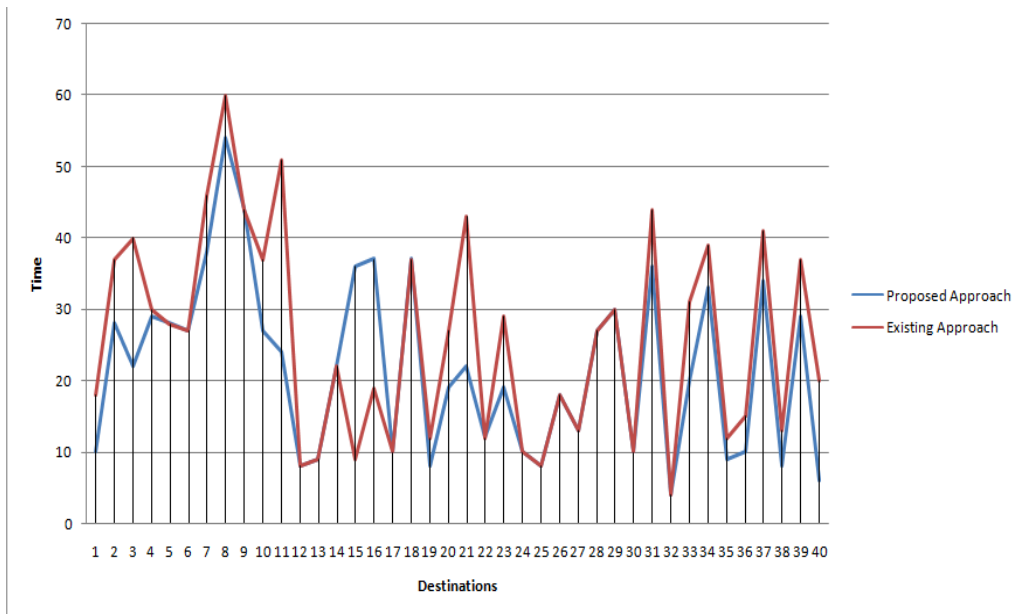


Figure 3.18 Graph Showing Comparison of Existing and Proposed Approach

CHAPTER 4

CONCLUSION

Para transit agencies schedule rides for their clients at an economical rate. The service provided by them is similar to taxi service however the main difference is that taxi is a dedicated service whereas para transit service is shared service. In the presented work, an auto scheduler has been developed for transit scheduling that can be used by para transit agencies. The researched work shows that out of all the domains where scheduling is used, the approach used for scheduling is different in different domains. So the work presented in this thesis comes under the category of public transport systems on a broad scale. The above algorithms used for scheduling show that scheduling approach for para transit agencies is different than the scheduling used in public transport systems. Many scheduling algorithms exist however they cannot be directly used for implementing scheduling for para transit services. In the presented work, algorithm has been developed that will minimize the wait period faced by the client and will require least number of buses for servicing. From the information provided by a para transit agency [16], the average trip time of a customer is 20 minutes. This information has been used while placing a sixty minute cap on the journey time. This approach has helped to achieve low wait time and efficient utilization of buses is also achieved. The efficiency of the algorithm has been backed up by suitable experiments and the results show that the average wait time of a client is 8.2 minutes with a standard deviation of 4.124 where as the average number of clients serviced by bus is 2.7 with a standard deviation of 0.246

CHAPTER 5

FUTURE WORK

The future work of the system should involve getting live traffic updates. Getting live traffic updates will not change the output given by the scheduler. The scheduler requires pickup time, pickup address and destination address while scheduling the rides. So traffic in a particular area, or a particular road under construction will not affect the output provided by the auto scheduler in most of the cases. The advantage of having live traffic updates will be that before a driver leaves for scheduling the clients; the driver can take what route to take to service these clients if there is some problem on the standard route. The auto scheduler presented here guides the agency in deciding the order in which the clients must be serviced and not the route that should be taken to service the clients. Addition of live traffic updates will increase the quality of service, but it will not affect the output of scheduling in most of the cases.

One interesting way in which this scheduler can be used is in the deployment of “DakNet”. DakNet is a project run in rural parts of India and Cambodia to provide low cost digital communication. For this buses equipped with internet facility are sent to villages. When the bus comes near the kiosk data gets exchanged. By making use of scheduler, buses can be scheduled to leave from Taluka place (town) to village such that the villagers keep on getting updated information over and over again and also they stay connected to internet.

REFERENCES

- [1] Rajesh Krishna Balan, Nguyen Xuan Khoa, and Lingxiao Jiang, 'Real-Time Trip Information Service for a Large Taxi Fleet'
- [2] Ziqi Liao, 'Taxi Dispatching Via Global Positioning System'
- [3] Dong YongFeng, Li NaNa, Dong ZhengChao, Gu Junhua, Li Weini, 'Research on model and its solving algorithm for transit scheduling problem'
- [4]<http://mobilitat.getmyip.com:8080/WebElements.ScheduleTestData.pretty>
- [5] Ruibin Bai, Jiawei Li, Jason A D Atkin, Graham Kendall, 'A Game Theoretic Approach for Taxi Scheduling with Street Hailing'
- [6] Solving the Course Scheduling Problem Using Simulate Annea in E. Ayca and T. Ayav
- [7] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. When is "nearest neighbor" meaningful? International Conference on Database Theory (ICDT), Jerusalem, Israel
- [8] Hong Jiang, Tianwei NI 'PB-FCFS--A Task Scheduling Algorithm Based on FCFS and Backfilling Strategy for Grid Computing'
- [9] Raidi, G.R, 'Solving the Railway Traveling Salesman Problem via Transformation into Classical Traveling Salesman Problem'
- [10] Morgan, M.J.W, 'Capacited Vehicle Routing: Perturbing the Landscape to Fool an Algorithm'
- [11] <http://www.itsmarta.com/bus-schedules.aspx>
- [12]<http://mycommute.itsmarta.com/hiwire>
- [13]<http://www.trailblazertransit.com/faq.cfm>
- [14]http://www.mjc2.com/Frameset_products_routing.htm
- [15]www.microsoft.com/maps/developers
- [16] <http://www.easylift.org/dial-a-ride/pdf-of-riders-guide/how-easy-lift-schedules-your-ride/>
- [17]<http://code.google.com/apis/maps/documentation/javascript/>

[18]<http://www.1stwebdesigner.com/wp-content/uploads/2010/06/distance-finder/source-distance-finder-google-maps.php>

BIOGRAPHICAL INFORMATION

Amey Sudhir Bhalerao received his Bachelors' degree in Computer Engineering from University of Pune in 2010. He is pursuing Masters' in Computer Science from University of Texas Arlington. His research interests include vehicle routing and schedule development.