ROUTING AND PLANNING FOR THE LAST MILE MOBILITY SYSTEM

NGUYEN VIET ANH

NATIONAL UNIVERSITY OF SINGAPORE

2012

ROUTING AND PLANNING FOR THE LAST MILE MOBILITY SYSTEM

NGUYEN VIET ANH

(B.Eng (Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF INDUSTRIAL & SYSTEMS ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

DECLARATION

I hereby declare that the thesis is my original work
and it has been written by me in its entirety.
I have duly acknowledged all the sources of information
which have been used in the thesis.
This thesis has also not been submitted for any degree in any
university previously.

Nguyen Viet Anh

5 July 2012

Acknowledgements

I am indebted to many people for making the time working on my M.Eng thesis an unforgettable experience.

I would like to express my deepest gratitude to Associate Professor Ng Kien Ming for all the guidance that you have offered me throughout the course of the project.

I would like to thank Assistant Professor Teo Kwong Meng for all the time and effort that you have spent on me and for giving me your valuable advice especially on the gist of research methodology.

I am also grateful to Professor Amedeo Odoni of the SMART for the original idea, as well as for the continual support during the course of this thesis.

I am also privileged to get to know Professor Melvyn Sim. Exploring new ideas with you was the highlight of my time in NUS. Thank you so much for your guidance both in research and in the daily life.

Many thanks to Dr Kim Sujin for giving me advice when I am lost. I am also grateful to the administrative staff from ISE, particularly Lai Chun, Victor and Weiting.

My friends in NUS are wonderful: Long, a Hieu, Markus, Dayu, Nugroho, Ashwani, Thanh Mai, Muchen, Yanhua, Ly Vu, Chun Kai and many others who have become my friends during the past two years. I really appreciate the time spent with the French DDP, as well as the DMP friends, especially Hu Xiang, Kok Cheong, Junwei, Lester, Amanda, Julius, Chloe, Timothy, you have taught me that there are more to life than just theoretical problems.

Finally, I would like to thank my family, especially my parents, for being so supportive and understanding during the entire year when I was working on this project. Thank you, Bong, for your beautiful smiles. Thank you, Thi Vu, for your encouragement without which I surely could not finish this thesis.

May we all be happy!

Summary

In this thesis, we introduce several routing and planning algorithms for a small Mobility on Demand system. This system aims to provide "to-door" service which connects customers between transportation hub (MRT, bus terminal...) and their desired destinations (workplace, home ...). Initially, we consider a single period model of the problem. We first consider the case where there are only Last Mile customers who want to go from transportation hub to their destinations. The problem is modeled as a special instance of the Vehicle Routing Problem with Time Windows, and a tabu search algorithm is proposed. We then extend our algorithm to take into account the First Mile customers who want to go from their current place (workplace, home ...) to the transportation hub. This extension also brings along a rule to schedule the vehicles: the vehicle might stop and wait for the customers under certain conditions. We complement the single period problem by studying the heterogeneous fleet problem with a heterogeneous tabu search and pre-, post-processing procedure. Next, we study the multi-period problem which is more relevant to the real life implementation of the problem. In this setup, the single period algorithm is used in each period to find the best routing for that particular period's demand. For this problem, we relax the schedule for the vehicle and use the heterogeneous fleet tabu search algorithm. We demonstrate the capability of our algorithm by using a real life demand taken from the Singapore public transport data. Finally, we consider the last mile problem under uncertain travelling time. We propose the lateness index, which evaluates the possibility of serving the customers on time. We show that the lateness index solution is a promising approach to solve the problem with uncertain travelling time.

Contents

List of Figures

1 Introduction

1.1 Motivation

Mobility on Demand (MOD) becomes increasingly important in the current social and economic context. Although the transportation network is not scalable, there has been a boom in the number of private vehicles during the last decade, which in turn results in traffic jam, noise, stress and health problem. The current public transport system, including buses and underground, is inefficient and cannot deal with the problem of aging societies where old or disabled people need a to-door service. Governments and enterprises are also facing problems regarding how to reduce road footprint and carbon footprint. Thus, a convenient, reliable and profitable MOD system is the future of urban transportation. Nevertheless, in order to implement such a system successfully in the practical context, it is essential to have a good operation planning system which routes and schedules the fleet in a reasonable manner.

In this project, we study a routing and scheduling algorithm for a MOD system which connects passengers from big transportation hubs (MRT stations, railway stations) to their desired final destinations. A simplistic geographical layout of the MOD system is shown in Figure 1.1. The idea of such system is inspired by the famous Last Mile Problem (LMP) in which a commuter's hardest and most time consuming part in his whole trajectory is actually the last mile portion.

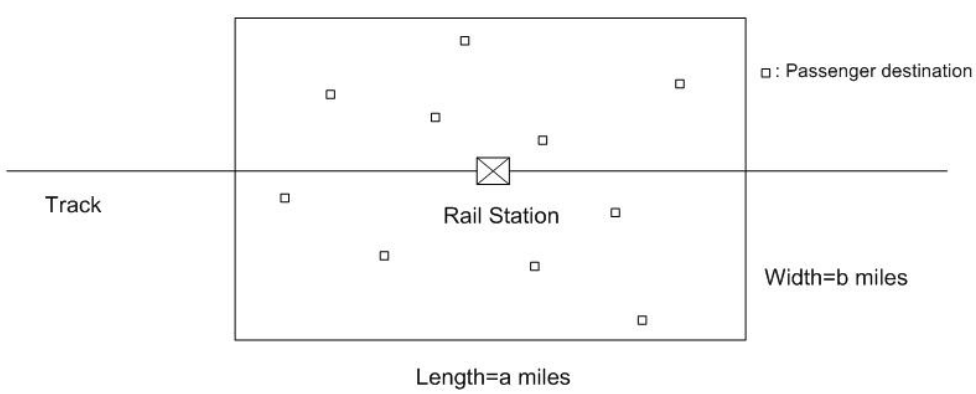

Figure 1.1: Geographical layout of the Last Mile Problem

1.1.1 Real life scenario

Our system consits of one central server and a fleet of vehicles (shuttle buses). A passenger will send an SMS to our server, stating his desired pickup point, destination and their time windows. After receiving the request, our server will assign the passenger to a suitable shuttle bus, and reply to the passenger the relevant information about the service. A good IT platform is crucial for the implementation of this transport on demand system. This platform should be able to:

- Receive and reply passengers' demands via SMS.
- Schedule the operation of each vehicle up to passengers' demands.
- Route the vehicle with the best possible route.

Among the three attributes above, the receiving and replying SMS involves the information technology (IT) system, while the scheduling and routing are operations management problems. For this reason, this thesis concentrates only on the scheduling and routing capabilities of the system.

1.1.2 Challenges

The LMP is a real world problem, so there are major challenges we need to overcome so as to achieve a pragmatic, efficient and implementable solution:

- The computational time must be low. In the real world, for a reasonable dispatch of the vehicles to serve passengers' demand throughout the day, the computational time can only be around a few minutes. Furthermore, it must provide good solutions even if the calculation time is restricted.
- The system must withstand high load due to passengers' demands during rush hours.

- The system should provide a good balance between the computational time of the algorithm and the quality of the solution.
- The system should be flexible, so that new features such as real time traffic data can be easily incorporated in the calculation.

1.2 Literature review

The problem, as described in the introduction, is a generalization of the vehicle routing problem with time windows (VRPTW), the multiple vehicle routing with pickup and delivery with time windows (VRPPDTW) and the Dial-A-Ride problem (DARP).

1.2.1 Studies on VRPTW

The VRPTW, which can be modeled as a multi-commodity network flow problem, attracts many exact algorithms and heuristics. Desrochers *et al.* (1992) proposed a column generation algorithm which managed to solve some 100 customer instances. Chen and Xu (2006) also proposed a dynamic column generation, and claimed that the algorithm outperforms other insertion heuristics. He reported a CPU computational time between 17 seconds and 5422 seconds. Nevertheless, since the solution space grows exponentially, it is unlikely that these exact algorithms will work for bigger instances.

In term of heuristics algorithm, Solomon (1987) and Potvin and Rousseau (1993) are fundamental work on the heuristics construction of VRPTW. Some other important heuristics include Chiang and Russell (1996) (simulated annealing), Russell (1995) (reactive tabu search). Lau *et al.* (2003) proposed a tabu search algorithm for the m-VRPTW where there is a limited number of vehicles.

1.2.2 Studies on VRPPDTW

There have been several proposed algorithms for the dynamic pickup and delivery problem with time windows. Nanry and Barnes (2000) proposed a reactive tabu search algorithm with soft constraints on the time windows and the vehicle capacity. They considered three move neighborhoods including single insertion, pair swapping between routes and within route insertion. The algorithm has been tested with their own instances involving 25, 50 and 100 customers. Lau and Liang (2001) used a similar set of neighborhood, and they also generated test cases from the Solomon (1987) test cases.

Ropke and Pisinger (2006) has proposed an adaptive large neighborhood search algorithm with several removal and insertion heuristics. He considered Shaw removal, random and worst removal heuristics. Two insertion heuristics are greedy insertion or regret insertion. The selection of heuristics is randomly done by considering a probability distribution whose weights are adjusted dynamically during the search. The algorithm has been tested with the test cases from Li and Lim (2001) which have from 50 to 500 requests. The author asserted that the algorithm outperformed previously proposed algorithms.

Cordeau *et al.* (2007) provided a survey on VRPPD with static or dynamic setup, as well as with time windows options and different assumptions on the size and capacity of the fleet.

1.2.3 Studies on DARP

A DARP is more passenger oriented which characterizes in a tighter time windows, stricter capacity constraint and a new customer ride time constraint.

For the dynamic DARP, Teodorovic and Radivojevic (2000) proposed a scheduling algorithm using fuzzy logic and fuzzy arithmetic. Their motivation comes from the fact that users (passengers, drivers etc.) have a fuzzy notion of time. They constructed a list of rules in order to perform insert or removal heuristics. The authors have generated their own test instances and the algorithm managed to solve an instance of 900 requests.

Mitrovic-Minic and Laporte (2004) studied the DARP under different scheduling strategies: drive-first, wait-first or dynamic waiting strategies. Berbeglia *et al.* (2010b) has recently proposed a hybrid tabu search and constraint programming routine to solve the dynamic DARP. The authors also considered four scheduling policies: basic, lazy, eager and hybrid scheduling.

A thorough survey on the DARP can be found in Cordeau and Laporte (2007) or Berbeglia *et al.* (2010a).

1.2.4 Studies on the VRP with heterogeneous fleet

An important variant of the VRP is the VRP with heterogeneous fleet, where the vehicles may have different capacity, different fixed or variable cost. When the number of vehicles is unlimited, we have the fleet size and mixed vehicle routing problem (FSMVRP), and when the number of vehicles is limited, we have the heterogeneous fixed fleet vehicle routing problem (HFFVRP). In this thesis, we are interested in the case where the fleet size is limited, thus the literature review considers only the HFFVRP instance. A comprehensive classification of this variant can be found in Paraskevopoulos *et al.* (2008).

The first method for the HFFVRP is proposed by Taillard (1996). At first, he used adaptive memory procedure to generate a large number of possible routes, and then column generation techniques were used to choose the best route among all the routes generated. Tarantilis *et al.* (2004) introduced a metaheuristics with 3 types of moves: 2-opt, 1-1 exchange and 1-0 exchange. His algorithm belongs to the stochastic search method, where in each iteration, the type of moves and the customers to be moved are randomly chosen using threshold accepting based methods. Li *et al.* (2007) proposed a record-to-record travel algorithm for the HFFVRP, with 4 types of move: 2-opt, or-opt, one point and two points. Brandão (2011) introduced a tabu search algorithm which use the GENI and US moves proposed by Gendreau *et al.* (1992), and a large neighborhood search in order to escape the local optimal solution.

1.2.5 Studies on VRP with uncertain travelling time

Uncertainty can come from different sources in the VRP: stochastic customers, stochastic demand or stochastic travelling time. A detailed review on the stochastic VRP can be found in Gendreau *et al.* (1996). In this thesis, we are interested in the VRP under uncertain travelling time.

Using stochastic programming and branch-and-cut, Laporte *et al.* (1992) proposed three different formulations for the VRP with stochastic travelling time and service time. Another approach using branch-and-cut with Monte Carlo simulation was developed by Kenyon and Morton (2003).

The stochastic VRP with soft time windows has been introduced more recently. Different heuristics algorithms were proposed instead of exact methods in order to solve mid to large scale instances. Among these algorithms, the most notable are genetic algorithm used by Ando and Taniguchi (2006) and tabu search used by Li *et al.* (2010).

1.3 Necessary attributes of a heuristic algorithm

Heuristic algorithms have been used widely for optimization problems where an exhaustive search is inefficient or impractical. However, when a heuristic algorithm is proposed in the literature, it is often the case that the algorithm is not reported objectively and evaluated scientifically. For this reason, it is usually difficult to assess the performance of the algorithm, and to compare one heuristic algorithm to the other. In this part, we survey different criteria to assess whether a heuristic algorithm makes a substantial contribution or not.

1.3.1 Speed

Speed is the main reason why researchers resort to heuristic algorithms instead of exact solution methods. However, according to Silberholz and Golden (2010), it is difficult to fairly compare the speed between algorithms due to the difference in computer hardware, programming languages, compilers and testing environments (multiple run, or run on distributed system). Apparently, the best way to compare two algorithms is to have the source codes, compile them on the same computer using the same compiler, and run them on the same computer. Nevertheless, this method is not always applicable due to several reasons. Firstly, two codes may use different programming languages. Secondly, the author may not want to divulge the code to the public. If the author can publish a Windows executable file of the algorithm, it will make the comparison easier and more reliable.

1.3.2 Accuracy

One important attribute of heuristic algorithm is that it should give satisfactory solutions. Above all, the solutions have to be feasible, that is the solutions have to satisfy all constraints of the problem. Another measure of the accuracy is the gap between the heuristic solution and the exact optimal solution or a good bound. However, for hard combinatorial problem, good bounds or exact optimal solution is not always ready; thus, most of the comparisons have been made with the best result found so far. Yet, Barr *et al.* (1995) stated that comparing solution quality between two algorithms is also a troublesome task. Very often, the author reports only the best solution found by tuning their parameters, or by running with different starting points. For this reason, it is desirable that the computational result is obtained in only one run, and the number of parameters is small.

1.3.3 Robustness

Robustness is the characteristic that an algorithm performs well on a large set of instances. According to Cordeau *et al.* (2002), users prefer an algorithm which gives a reasonably good result to all instance to another algorithm which performs extremely well on certain instances but poorly on the others. The deterioration in the solution quality is partly because of the probabilistic nature in the parameter or in the searching routine. Authors usually report the best result found by their algorithm, which gives a false idea on how well the algorithm really performs. To this extent, the algorithm has to be tested thoroughly using a large test set. Furthermore, if the quality of the solution does not differ too much, a deterministic algorithm should be preferred to a probabilistic algorithm.

1.3.4 Stability

Under real life scenarios, there are situations where the problem is over-constrained. For example, in vehicle routing, we may have limited number of vehicles, and these vehicles may not be able to serve all the customers. To handle this issue, Lau *et al.* (2003) proposed that under over-constrainedness, when the number of vehicles is reduced, the average number of customers served by each vehicle should be monotonically increasing.

1.3.5 Flexibility

As heuristic algorithm is used to solve real life problems, flexibility is a critical factor. Braysy and Gendreau (2005) suggested that a good heuristic algorithm should be able to handle changes in the objective function as well as in the constraints. Although, modifications to the algorithm are sometimes trivial, it is less evident how the performance of the algorithm will be affected by these changes. To show the flexibility of the algorithm, it is recommended that the algorithm should be tested with several variants of the problem, along with a clarification of the changes made.

1.3.6 Simplicity

The reason why a heuristic algorithm is not widely used in real life is that the algorithm is too complicated, or too hard to implement. An algorithm should be simple enough to be understood, with the exemplary case is the Clarke and Wright algorithm proposed by Clarke and Wright (1964). Another measure of the simplicity is the number of parameters involved. In fact, Silberholz and Golden (2010) indicated that the space of possible parameter combinations increases exponentially along with the number of parameters in the algorithm, and this makes tuning for a good set of parameters a tedious task. Furthermore, it is inevitable that there is a certain correlation among parameters, which makes understanding and analyzing the algorithm more difficult.

1.3.7 Reproducibility

Reproducibility is indeed another criterion for a good algorithm. To achieve this point, the algorithm should be well documented so that a reader can successfully construct a similar algorithm from the report. To this extent, Barr *et al.* (1995) suggested that the source code, the executable files and the solution to the test cases should be made publicly available. Furthermore, the source code should also be well documented and straightforward to be compiled.

Not all of the algorithms found in the literature survey are implemented and reported following the above criteria: most of them concentrate on reporting the accuracy and speed of the algorithms. By taking these criteria into account in both the development as well as the testing phase of the algorithm, we will demonstrate that our algorithm satisfies the desired criteria.

1.4 Contributions of the thesis

This thesis makes both practical and scientific contributions. Firstly, this thesis introduces a new variant of vehicle routing problems: the Last Mile Problem. The LMP is inspired by a real life problem encountered by transportation planners in big cities, and the solution to a LMP is a promising approach to future urban mobility. The importance of the LMP is further emphasized by its positive impact on societal, economic and environmental issues for future cities.

Secondly, using operations research techniques, this thesis demonstrates that the Last Mile mobility system can be reliable and profitable. In fact, from the practitioners' perspective, this thesis proposes a decision support system which assists the service provider to make both strategic and operational decisions. This system is tested using real life data taken from the public transportation data of Singapore. This system is essential in encouraging the urban transportation planners, enterprises and relevant parties to provide the Last Mile service to the passengers.

Finally, this thesis makes scientific contributions to the vehicle routing problems. This thesis proposes a tabu search heuristics for the LMP. This tabu search heuristics can handle various constraints including the heterogeneous fleet and different scheduling rules for passengers. In addition, this thesis also uses the tabu search heuristics to solve the LMP with uncertain travelling time by using a new index to assess the uncertainty called the lateness index. The main advantage of this lateness index approach is that it works even when the distributions of the travelling time are unknown or contain mixed distributions.

1.5 Structure of the thesis

- **Chapter 2: Algorithm for the single period problem** We start by describing our formulation for the basic LMP. The problem, which is initially modeled as a VRPPDTW, is then simplified as a special instance of the VRPTW with the time windows at the depot. After introducing an MIP formulation for the basic LMP, we propose a tabu search heuristics which is capable of solving the large instance of the problem. We then modify the algorithm to take into account the First Mile Problem (FMP) customers. Different scheduling rules are discussed, along with the parallel implementation of the algorithm which succeeds in reducing the computational time. We complete the single period problem by solving the heterogeneous fleet problem with a heterogeneous fleet tabu search routine and the pre-, post-processing procedure.

- **Chapter 3: Algorithm for the multi-period problem** In this chapter, we adapt the single period algorithm to solve the multi-period problem. First, in the multi-period setup, we can relax the scheduling rule without worsening any service quality. Secondly, we will use the heterogeneous fleet algorithm developed in the previous chapter to allow flexible fleet compositions for the service provider. Using rolling horizon, we solve a real life problem where real service demand is taken from the Singapore public transport database.

- **Chapter 4: Algorithm for the LMP under uncertain travelling time** In this chapter, we consider the Last Mile Problem with uncertain travelling time, where the travelling time between two nodes becomes a random variable. After characterizing the travelling time, we introduce the lateness index, a criteria to evaluate the quality of solution subject to meeting the customers' time windows. The tabu search heuristics is modified with the index, which is finally benchmarked with the static approach using mean travelling time and the 90th-percentile approach using the 90th-percentile travelling time.

- **Chapter 5: Conclusions** This chapter presents concluding remarks, and suggests future direction for research.

2 Algorithm for the single period problem

In this chapter, we first formulate the basic LMP as a VRPPDTW, and with a critical observation, we simplify the problem to a special instance of VRPTW. We then introduce a mixed integer programming model of the problem. A tabu search heuristics algorithm is also proposed to solve this basic formulation of the Last Mile problem. We continue by integrating the FMP customers in the algorithms along with different scheduling rules to meet the FMP customers' pickup time windows. The introduction of the scheduling makes the computation more extensive, so we propose to use parallel computing which exploit the modern computer structure to reduce the computational time. Finally, we will consider the heterogeneous fleet problem, where vehicles can have different capacities, fixed costs and variables costs. By modifying our tabu search routine, we propose a heterogeneous fleet tabu search routine as well as a pre-, post-processing process to solve the heterogeneous fleet problem.

2.1 The Last Mile Problem

2.1.1 Problem formulation

Customer definition

Each customer is characterized by his Pickup and Delivery location (p and d) and his time windows for pickup and delivery. Each time windows consists of two values: Ready Time and Due Time with the relationship: Ready Time $<$ Due Time. The customer is available for pickup or delivery within the interval from Ready Time to Due Time.

A customer is satisfied if: 1) The pickup is done before the delivery. 2) The actual pickup and delivery time must be in the time windows. 3) There is no capacity violation in the pickup, delivery as well as in the nodes in between.

System specification and simplification

Our system works under the following rules:

1. For every request i, the actual pickup time (APT) has to be between the Pickup Ready Time (PRT) and Pickup Due Time (PDT): $PRT_i \leq APT_i \leq PDT_i$.
2. We assume that the customer can be delivered early, the delivery ready time is hence relaxed.
3. For every request i, the actual delivery time (ADT) has to be before the Delivery Due Time (DDT): $ADT_i \leq DDT_i$.
4. The fleet is homogenous: all vehicles have the same capacity.
5. The vehicle returns to the depot after serving the last customer.
6. The vehicle departs right after all customers are onboard. This means the departure time of the vehicle is the maximum of the pickup ready time of its customers.

The general problem as described can only be modeled as a VRPPDTW, however, in a LMP, all the customers have the pickup locations positioned at the transportation hub, and hence the customers' pickup locations can be omitted. This is a critical observation since it helps simplifying the Last Mile Problem to a vehicle routing problem. Each customer is now characterized only by the more important delivery location, and without loss of generality, the fleet's depot can be set to coincide with the transportation hub and the customers' pickup locations. The LMP is thus considered as a special instance of the VRPTW with time windows in both the depot and the delivery nodes.

Mixed Integer Linear Programming Model

The LMP can be solved as a mixed integer programming problem. A detailed discussion on how to construct the mathematical problem for the general LMP as a

VRPPDTW is given in the appendix. Here, we present the MIP model for the basic LMP after simplification.

Let n be the total number of customers. Let D denote the delivery nodes with customer i is represented by a delivery node $i \in D$. Each node i has a time window $[l_i, u_i]$ and a demand $q_i < 0$ since it is always the delivery node. The pickup time windows for customer i is $[L_i, U_i]$. Let $N = D \cup \{0, 2n+1\}$ where $\{0, 2n+1\}$ denote the starting and the ending depot of the vehicles. The service time at node i is s_i, and the travel time between node i and node j is t_{ij}. Let V be the set of the available vehicles, every vehicle $v \in V$ has a finite capacity Q^v and is available during a period $[l_v, u_v]$.

x_{ij}^v is the decision variable, it equals 1 if the vehicle v travels from node i to node j, and 0 otherwise. S_i^v denotes the time the vehicle v reaches node i. y_i is the binary variable, $y_i = 0$ if the customer i is served, $y_i = 1$ otherwise. Q_i^v denote the current number of customers in vehicle v after the vehicle v visits node i. S_i^v denotes the time the vehicle v reaches node i.

The mathematical model is:

$$\text{minimize } \alpha \sum_{i \in D} y_i + \beta \sum_{v \in V} \sum_{i,j \in N} t_{ij} x_{ij}^v$$

subject to:

$$\sum_{v \in v} \sum_{j \in N} x_{ij}^v + y_i = 1 \qquad \forall i \in D \tag{2.1}$$

$$\sum_{j \in N} x_{0j}^v = 1 \qquad \forall v \in V \tag{2.2}$$

$$\sum_{j \in N} x_{j(2n+1)}^v = 1 \qquad \forall v \in V \tag{2.3}$$

$$\sum_{j \in N} x_{ji}^v - \sum_{j \in N} x_{ij}^v = 0 \qquad \forall i \in P, \forall v \in V \tag{2.4}$$

$$x_{ij}^v (S_i^v + s_i + t_{ij}) \leq S_j^v \qquad \forall i, j \in N, i, j \text{ are assigned to } v \tag{2.5}$$

$$l_v \leq S_0^v \leq u_v \qquad \forall v \in V \tag{2.6}$$

$$l_v \leq S_{2n+1}^v \leq u_v \qquad \forall v \in V \tag{2.7}$$

$$L_i \leq S_0^v \leq U_i \qquad \forall v \in V, i \text{ is assigned to } v \tag{2.8}$$

$$l_i \leq S_i^v \leq u_i \qquad \forall i \in D, i \text{ is assigned to } v \tag{2.9}$$

$$0 \leq Q_i^v \leq Q^v \qquad \forall i \in D, i \text{ is assigned to } v \tag{2.10}$$

$$Q_0^v = -\sum_i q_i \qquad \forall v \in V; i \text{ is assigned to } v \tag{2.11}$$

$$Q_j^v = (Q_i^v + q_j) x_{ij}^v \qquad \forall v \in V; i, j \text{ are assigned to } v \tag{2.12}$$

$$x_{ij}^v \in \{0, 1\} \qquad \forall i \in N, \forall j \in N, \forall v \in V \tag{2.13}$$

$$y_i \in \{0, 1\} \qquad \forall i \in D \tag{2.14}$$

The objective function is to minimize the weighted sum with parameters α and β of the number of unserved customers and the total distance travelled respectively. Constraint (2.1) ensures that the customer is either accepted or rejected. Constraint (2.2) and (2.3) ensure that the route for each vehicle starts and ends at the depot. Constraint (2.4) and (2.5) ensure the continuity of the route. Constraint (2.6) and

(2.7) ensure the vehicle is active within its own time windows. Constraint (2.8) and (2.9) ensures that the pickup and delivery is done within the time windows. Constraint (2.10) ensures the capacity is valid for each vehicle. Constraint (2.11) and (2.12) ensure the capacity continuity of the route.

The model, unfortunately, is not linear because of the bilinear constraint (2.12). We might linearize the nonlinear constraints (see, for example, Chang (2000)), and then advanced techniques in solving large scale optimization problem (column generation, Benders decomposition) may be utilized to get an exact solution. However, since linearization creates a plethora of intermediate variables, and since the original model has already been exponentially hard, exact solution algorithm might not work well in this case. Furthermore, the running time constraint is also very critical. For these reasons, the traditional approach is not really applicable for this project.

2.1.2 Tabu search algorithm

Tabu search is a metaheuristic which uses local search with a wise memory management in order to avoid visiting the same solutions. More information about tabu search can be found in Glover and Laguna (1997). For the representation of the solution, a vehicle's route starts with node 0 which is the depot, follows by nodes representing the customers served by the vehicle, and ends with node 0 since the vehicle returns to the depot. In each iteration of the tabu search algorithm, we use neighborhood moves to explore the neighbors of the current solution in order to find a better solution. It is important to note that the tabu search can only find a local optimal to the problem.

Holding list

The holding list contains the list of requests which are not served by the current solution. The idea of the holding list was first proposed by Lau *et al.* (2003). At initialization, all customers are put in the holding list. A tabu search routine will then be called in order to insert the customers into the routes. A feasible solution exists when the routine manages to drive all customers out of the holding list. At intermediate steps, the routine might as well take out the customer from the route and insert it to the hold list. This action helps enlarge the neighborhood of the solution, and thus increase the quality of the solution.

Neighborhood moves

The single insertion from holding list (IH) move attempts to insert one request into the existing route. Each request in the holding list will be sequentially chosen; the algorithm then tries to insert the delivery node into any possible position in the route (remind that the pickup nodes coincide with the depot). Similarly, a single removal to the holding list (RH) move attempts to remove a request from the route and put it into the holding list. A switch with holding list (SH) move tries to switch one customer from one of the routes with one customer from the holding list.

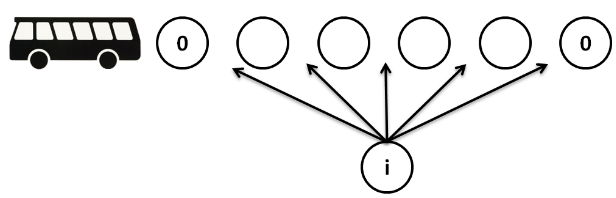

Figure 2.1: Example of possible insertion move

A transfer (T) move transfers request from one route to another route. A switch (S) move attempts to switch customer from one route with another customer in another route. An exchange (E) move will exchange subsequent customers between two routes. A flip (F) move will switch the order of two adjacent nodes in a route, which is meant to eliminate zigzag cross in the trajectory of the vehicle.

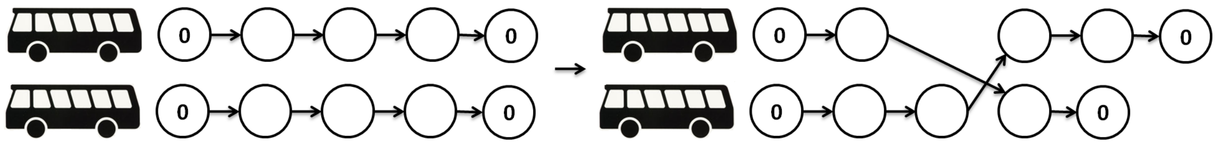

Figure 2.2: Example of an exchange move

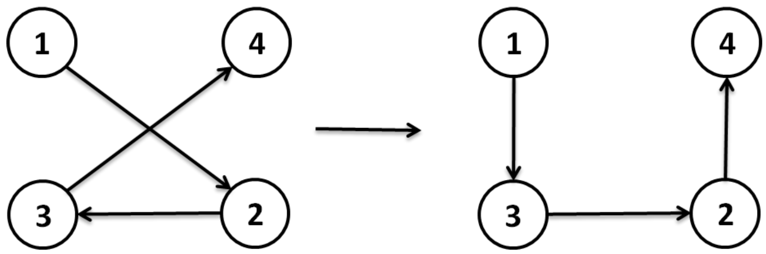

Figure 2.3: Example of a flip move

The hierachical cost

Normally, in a VRP, we use an objective function which is the weighted sum of individual criteria (eg. total distance, number of customers served etc.) to assess the quality of the solution. However, it is often challenging to determine a reasonable weight for each of the criteria. In order to handle this issue, a hierarchical cost is used to assess the quality of the solutions. The evaluating function will have the priorities given below:

1. Maximize the satisfied customers
2. Minimize the number of vehicles used
3. Minimize the distance travelled by the fleet

With this hierarchical cost, between two solutions, the solution which serves more customers is always better regardless of the number of vehicles used. With the same number of satisfied customers, a solution which uses less number of vehicles is always better regardless of the distance travelled. This hierarchical cost is supported by the fact that the satisfaction of the customers is the most important purpose of the Last

Mile mobility system. Furthermore, minimizing the number of vehicles used is advantageous if the algorithm is executed in real life environments since it will save the initial capital investments in the system.

Optimization routine

The optimization routine for the LMP is implemented as in Lau *et al.* (2003): let $numVeh$ be the current number of vehicles used, N be the maximum number of vehicles available, $CountLimit$ be the maximum number of iterations without improvement. The Tabu search routine is described in Algorithm 1 with the standard tabu search TS being described in the Appendix.

Algorithm 1 Tabu search routine

```
while holding list is empty or numVeh ≤ m do
  count = 0
  while count ≤ CountLimit do
    perform TS
    if better solution found then
      count = 0
    else
      count = count + 1
    end if
  end while
  numVeh = min(numVeh + 1, N)
end while
```

2.1.3 Experimental results

In this section, all tests are carried out on a desktop with Core 2 Duo E6750 @2.66GHz, 4.00GB RAM running on Windows Vista SP2 32-bit.

Performance on VRP test cases

There is no VRPTW, VRPPDTW or DARP test case which is applicable to our basic LMP algorithm. Instead, we use the test cases for the Vehicle Routing Problem

Table 2.1: Result with Christofides test cases

	Best	Our result
	Distance	Distance
vrpnc1	524.61	530.86
vrpnc2	835.26	833.4
vrpnc3	826.14	871.43
vrpnc4	1028.42	1039.22
vrpnc5	1291.29	1302.07
vrpnc11	1042.11	1118.05
vrpnc12	819.56	878.76

Table 2.2: Result with Taillard test cases

	Best	Our result
	Distance	Distance
tai75a	1618.36	1507.01
tai75b	1344.62	1311.29
tai75c	1291.01	1234.3
tai75d	1365.42	1259.57
tai100a	2041.34	1936.31
tai100b	1940.61	1887.81
tai100c	1406.22	1717.7
tai100d	1581.25	1565.76
tai150a	3055.23	2769.96
tai150b	2656.47	2720.14
tai150c	2341.84	2339.82
tai150d	2645.39	2500.02

(VRP). A very large time windows is added to each customer so that our basic LMP will solve the VRP test cases. We use Christofides *et al.* (1979) and Taillard (1993) test cases and the best solution reported is taken from Diaz (2012). The detailed results for the two sets of test cases are given in Table 2.1 and Table 2.2. The results show that our heuristics manages to get very good results on these test cases. This demonstrates the power of our algorithm, and more specifically our neighborhood definition as well as our tabu search routine.

LMP test cases

Although the Solomon test cases have become the standard test for VRPTW algorithms, they are not applicable to our algorithm since they do not take into account the pickup time windows at the depot. For this reason, we generate 50 test cases for the basic LMP algorithm. The test cases are generated with the following parameters:

- Each test case has 100 customers, each has demand of 1.
- Fleet consists of 30 vehicles of capacity 9.

- 70% of customers are ready for pickup at time 0, 30% are ready for pickup at time 100.
- The time windows' width for each customer (from ready time to due time) is 150.
- Delivery ready time is relaxed, it equals the pickup ready time.
- Delivery due time equals the pickup due time plus the distance between the pickup location (the depot) and the delivery location.

LMP versus mVRPTW

It is previously shown that the basic LMP is a special instance of the VRPTW with additional constraints on the time windows at the depot. Thus, we would like to compare the performance of our basic LMP with the m-VRPTW algorithm proposed by Lau *et al.* (2003). We use the test cases generated for the basic LMP, and relax the pickup time windows: every customer is now available at from time 0. The result of this test indicates the complexity introduced by incorporating the pickup time windows. The detail results are given in the appendix.

The frequency of the computational time for the two algorithms on the LMP test cases with relaxed time windows is shown in Figure 2.4. Adding the time windows at the depot creates a significant increase in the computational time of the basic LMP with respect to the mVRPTW algorithm. However, it should be noted that the case where all the passengers have the same pickup ready time corresponds to the worst case of the basic LMP algorithm. Furthermore, from the detailed results from the Appendix D.1 show that the objective values achieved by our LMP problem are very close to the values of the mVRPTW.

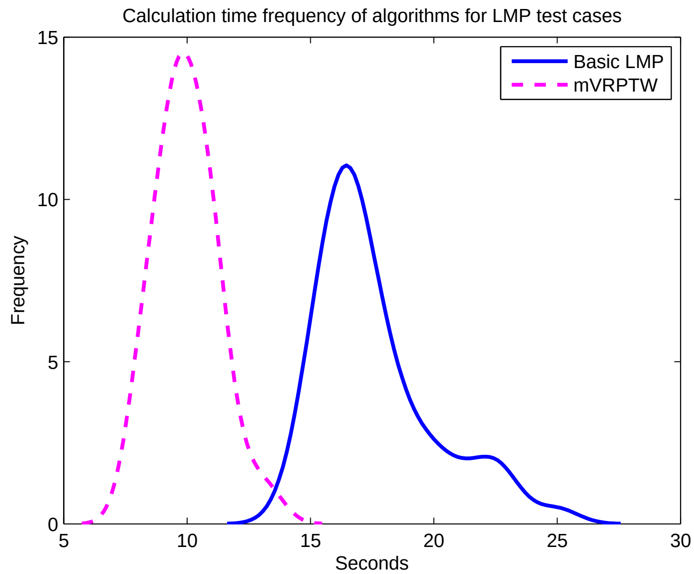

Figure 2.4: Frequency of computational time for relaxed LMP test cases

2.2 The Last and First Mile Problem

In this section, we are interested in solving a complementary problem to the LMP: the First Mile Problem (FMP). In the FMP, FMP customers request travel from their pickup location and they would like to go to the transportation hub. This problem arises when people want to travel from home to the MRT station in the morning, or from the workplace to the MRT station in the afternoon. By serving the FMP customers, the algorithm becomes complete and robust. Furthermore, since both LMP and FMP customers can be served at the same time, the daily operation cost (in term of distance travelled, number of vehicles used) is reduced, which in turn increases the profit to the service provider and reduce the cost to customers. This makes the system more attractive to both users and enterprises. For this reason, the extension of the algorithm is of great importance.

The FMP customers will have the same time windows structure as the LMP customers: the pickup has to be made between the Pickup Ready Time and the Pickup Due Time. The delivery ready time for FMP customers is again relaxed: the vehicle has to deliver customers before the Delivery Due Time, but it can deliver them earlier than the Delivery Ready Time. Similarly, the LMP customers are characterized by their pickup locations since their delivery locations are the depot, and hence can be omitted. For this reason, the integration of the FMP into the existing system does not destroy the special VRPTW structure.

2.2.1 Scheduling of the vehicles

Although the combination of the FMP does not pose any problem to the structure of the algorithm, it brings a big issue to the schedule of the routes. In the case of the basic LMP, we have assumed that the vehicle will not wait in its route. However, when we have FMP customers, some situations might arise:

- The vehicle may arrive at the FMP customer earlier than the customer's pickup ready time.
- If the vehicle serves only the FMP customers, there is difficulty in determining a reasonable departure time for the vehicle.

To address this issue, we propose the following scheduling principles for the problem: The vehicle should not wait when there is a passenger onboard. This scheduling principle is very practical since it guarantees the satisfaction of customers on the vehicle. With this scheduling, a vehicle might wait either at the depot, or at the first LMP customer in route assuming that it is not carrying any other customer.

In the following example, we assume that the vehicle serves 3 LMP customers and 1 FMP customers. In the first situation as depicted in Figure 2.5, the vehicle arrives at FMP1 early, and there is still LMP3 in the vehicle, so it is not allowed to wait. In this situation, we can schedule the vehicle to leave the depot later, so that it will arrive at FMP1 on time. In the second situation illustrated in Figure 2.6, the vehicle arrives at FMP1 early, but there is no customer onboard, so the vehicle can simply wait until FMP1 shows up. In the situation where all customers are LMP customers, the departure time of the vehicle is calculated in similar manner as in the first situation mentioned above.

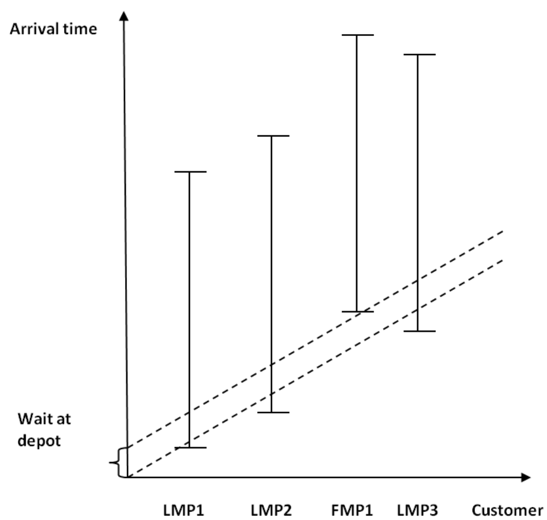

Figure 2.5: Wait at depot scenario

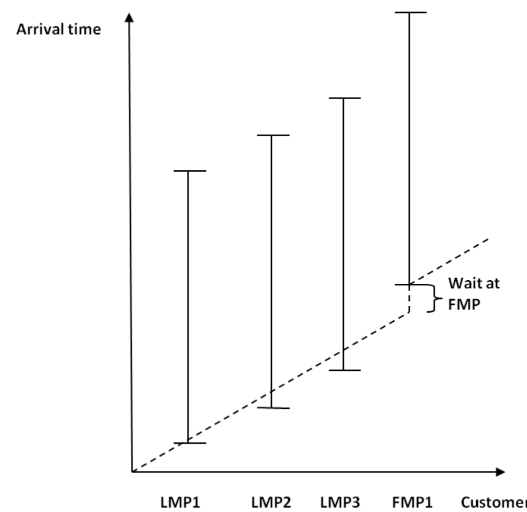

Figure 2.6: Wait at FMP scenario

Furthermore, with the FMP customers in route, it is important to check the time the vehicle returns to the depot. We need to assure that this return time is before the delivery due time of all FMP customers served by the vehicle.

2.2.2 Parallel version of the algorithm

Parallel computing is one of modern techniques in High Performance Computing which helps reduce the runtime needed to solve a problem, or increase the size of the problem that can be solved. Parallel computing is accomplished by dividing a complex and large task into smaller and easier to solve subproblems, these subproblems will be solved concurrently by using either many computers (grid computing), or a processor with multiple cores (MPI, OpenMP), or the graphic cards (GPGPU). As it has been indicated before, the computational time of our program has to be very low in order to be implemented dynamically. For this reason, we need to implement the algorithm in parallel to reduce the runtime of the system. Furthermore, the nature of our tabu search routine, which involves many evaluations of the neighborhood per iteration, is also a good case to apply parallelization.

More information and the technical terms of parallel computing can be found in Kumar *et al.* (1994).

Parallelization techniques

There are different strategies to parallelize an heuristic algorithm, the two most popular are: 1) at the same time, run several independent searches, and at the end, compare and take the best solutions; 2) run only one search routine, but in each iteration, divide the neighborhood into smaller neighborhoods, and evaluate these neighborhoods concurrently to find the best neighborhood solution. Clearly to see, the first strategy is meant to increase the quality of the solution, while the latter reduces the computational time. In this project, since the runtime constraint is important, the second approach is utilized.

In this project, parallel computing is implemented on one computer with multi-cores. As our tabu search standard procedure (as describe in the Appendix) consists of many for-loops, it is advisable to divide these for-loops so that they will be evaluated in parallel in different cores. The solution from each core will then be compared to choose the best neighborhood move available.

In parallel computing, the foremost issue is to eliminate the data dependency and the race condition. In brief, these situations arise when a variable is read/written by different cores, which results in the inconsistency in the value of the variable and the errors in the final solution. This issue is tackled by:

- Separating local and global variables: Each evaluation value is stored in local variable, and only when all the neighborhoods are evaluated, the local variables are then compared and the global variables are assigned accordingly.

- Eliminating the 'implicit' pointers which can be changed by other cores.

Scheduling is also an important issue of parallel computing. The schedule determines when and how the work is divided among cores. Typical scheduling strategies are:

- Static: the work is assigned with a developer defined parameter.
- Guided: the work is assigned with a decreasing amount of work among cores.
- Runtime: the work is assigned according to user input at runtime.
- Dynamic: the work is assigned dynamically to cores which are free.

For our tabu search procedure, the amount of work among iterations is highly volatile due to changes in the number of customers in each vehicle, as well as changes in the number of vehicles used. For this reason, it is challenging to determine the optimal chunk size for the whole program; the Static schedule is hence impractical. Similarly, it is hard to require the end user to specify a correct chunk size at runtime, thus the Runtime schedule will not be considered. Furthermore, we observe that even the amount of work in one iteration is not symmetrical: a vehicle with more customers requires more work than a vehicle with less customers, so the Guided schedule will not work efficiently. The Dynamic schedule stands out to be the most appropriate schedule to use for parallelization since it works well when there is high variability in the amount of work to assign.

In addition, the option "nowait" is also used. With this option, each core, after finishing its given work, directly receives new work to evaluate without waiting for other cores.

Parallelization implementation

The parallelization is implemented in each tabu search procedure, with the pseudocode described in Algorithm 2.

The parallel environment is created in line 2, and all variables declared afterwards are local (private) variables. In line 4, the 'for' loops are instructed to be run in parallel with nowait option and dynamic schedule. In line 6, a critical point is created to avoid any data dependency or race condition while comparing and writting the variables.

Algorithm 2 Parallelized tabu search routine

1: Declare global variables
2: Start the parallel computing environment (function call: #pragma omp parallel)
3: Declare local (private) variables
4: Set schedule to dynamic (function call: #pragma omp for nowait schedule(dynamic))
5: Evaluate the neighborhood
6: Synchronization (function call: #pragma omp critical)
7: Compare and choose the best move

2.2.3 Experimental result

Tests are carried out on a desktop with Core 2 Duo E6750 @2.66GHz, 4.00GB RAM running on Windows Vista SP2 32-bit. We concentrate on finding the computational time frequency the test cases, rather than finding the computational time average.

Results with LMP test cases

We test the basic LMP test cases on all three algorithms: the basic LMP, the LMP and FMP with waiting time, and the LMP and FMP with waiting time under OpenMP. The purpose behind this test is to see the increase in the complexity when we take into account the FMP customers, and also to see the performance improvements by using parallel computing. Since the basic LMP algorithm is the most suitable to handle these test cases, we predict that the basic LMP algorithm will dominate the two others. The detail result on each test case can be found in the appendix.

The computational time of three algorithms on the LMP test cases is shown in Figure 2.7. As being predicted, the basic LMP has the best performance. There is a big gap in the computational time between the basic LMP and the LMP+FMP with waiting time, which suggests that the scheduling has caused a significant increase in the complexity of the algorithm. The parallel version manages to reduce the computational time by 1.8 times of the original, which reduces the computational time to less than 30 seconds in all test cases.

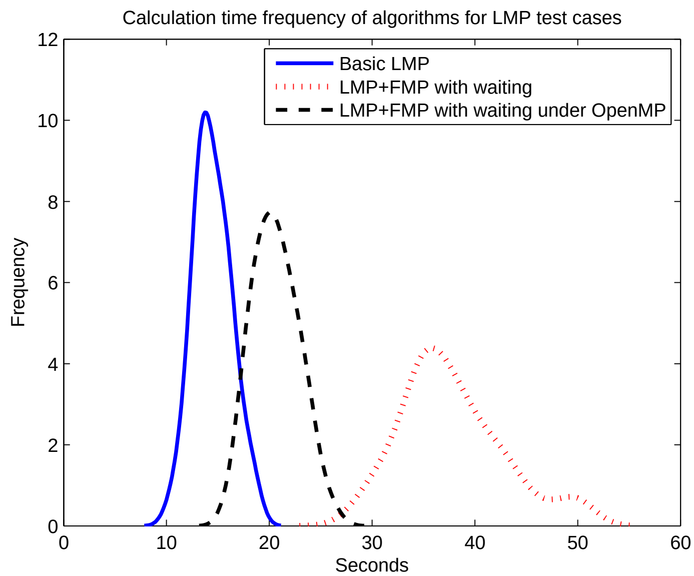

Figure 2.7: Frequency of computational time for LMP test cases

New test cases for LMP+FMP

We generate 50 test cases for the LMP + FMP algorithm:

- Each test case has 80 LMP customers and 20 FMP customers.
- Every customer has a demand of 1. The FMP customer is represented by demand of "-1", while LMP is represented as "1".
- Time windows and fleet is determined in the same way as for the LMP test cases.

Results with LMP+FMP test cases

Since the basic LMP does not handle FMP customers, and the LMP+FMP with waiting time runs slower than the parallel version, in this section, we test only the LMP + FMP with waiting time under OpenMP. The purpose of this test is to evaluate the real-life performance of the algorithm, thus we use the extension test cases which have both LMP and FMP customers.

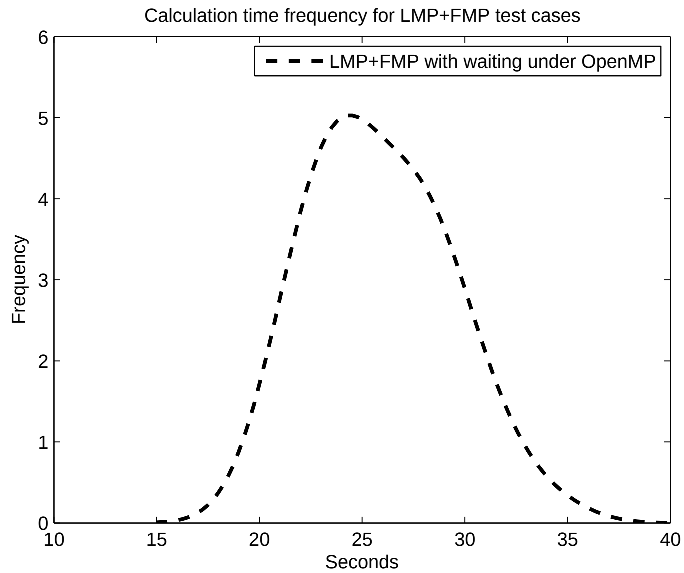

Figure 2.8: Frequency of computational time for LMP+FMP test cases

Figure 2.8 depicts the computational time of the algorithm on LMP+FMP test cases. The real life performance with the LMP+FMP test cases gives an average time of 28 seconds, and the maximum computational time is 40 seconds. This performance is regarded as satisfactory for real life implementation, however, there will be certain difficulties if this algorithm is used to solve larger instances, or if we would like to reduce the available computational time.

In conclusion, it is arguable that the scheduling of waiting for customers imposes a certain burden to the tabu search heuristics in terms of computational time. In real life implementation, it is highly recommended that the service provider simplify the waiting times. It can be achieved by considering the dynamic vehicle routing problem which will be introduced in Chapter 3. For the rest of this chapter, we will study the problem without scheduling of the vehicles.

2.3 The Last and First Mile Problem with heterogeneous fleet

Another important variant of the vehicle routing problem is the heterogeneous fleet VRP. Since the heterogeneous fleet is necessary for real life implementation, in this section, we extend our algorithm to handle the heterogeneous fleet. More specifically, we consider two types of heterogeneity:

1. Different capacity
2. Different cost (both fixed cost and variable cost, where fixed cost is charged any time we use the vehicle, while the variable cost is charged with the distance travelled)

Two methods are used in order to handle the heterogeneous fleet: the pre- and post-processing, and the modifications in the tabu search routine

2.3.1 The heterogeneous tabu search procedure

The tabu search procedure described in the last section does not account for the different cost of the vehicles. For the heterogeneous tabu search procedure, for every move, when the penalty of the move is calculated, it is necessary to find the best vehicle to serve the route.

The algorithm to find the best vehicle if the move involves only one route is described in Algorithm 3, and the algorithm to find the best pair of vehicles if the move involves two routes is described in Algorithm(4).

Nevertheless, the practice of finding the lowest cost vehicles is computationally intensive. In the algorithm, we will run the standard tabu search first to construct a reasonably good initial solution, and then we use the heterogeneous tabu search to further improve the solution. The detail algorithm for the heterogeneous fleet is described in the Appendix C.

Algorithm 3 Heterogeneous fleet penalty for moves involving one route

Calculate the number of customers served by route A and the distance of route A
Let Υ be the set of available vehicle, let υ_A be the vehicle currently used to serve route A
for Every vehicle in $\{\Upsilon, \upsilon_A\}$ **do**
    Calculate the fixed cost and the variable cost of using this vehicle to serve route A
    if This vehicle has the lowest total cost **then**
        Choose this vehicle to serve route A
    end if
end for

Algorithm 4 Heterogeneous fleet penalty for moves involving two routes

Calculate the number of customers served by route A, B and the distance of route A, B
Let Υ be the set of available vehicle, let υ_A, υ_B be the vehicle currently used to serve route A,B
for Every pair of vehicles in $\{\Upsilon, \upsilon_A, \upsilon_B\}$ **do**
    Calculate the fixed cost and the variable cost of using this pair of vehicles to serve route A and B
    if This pair of vehicles has the lowest total cost **then**
        Choose this pair of vehicles to serve route A and B
    end if
end for

2.3.2 The preprocessing and postprocessing procedures

The use of pre- and post-processing is a simple idea to take into account the different capacity and cost for each vehicle. For the heterogeneous fleet, the order of the vehicles which are sequentially used in the standard tabu search procedure is critical. It is then beneficial to have a good order of vehicles to be use, and we may consider two ways of sorting the vehicles for the preprocessing:

1. Sort the vehicles in increasing fixed cost: for this rule, the vehicles with lower fixed cost will be used first. By using this rule, the manager expects to use low fixed cost vehicles, which results in a lower cost planning. A more complicated rule, for example, using weighted sum of the fixed cost and variable cost, can also be used.

2. Sort the vehicles in decreasing capacity: for this rule, the vehicles with higher capacity will be used first. By using high capacity vehicles first, the manager expects to use less number of vehicles, which results in more compact planning. This sorting is important in real life implementation where the service provider has only a fixed number of vehicles in the fleet.

Furthermore, we after running the tabu search heuristics, we can run a simple post-processing which might detect improvement over the routing plan as in Algorithm 5. Our complete algorithm can be summarized in the flow chart shown in Figure 2.9.

Algorithm 5 Post processing procedure

Sort all the route in increasing number of customers served
for Every route in the ordered list **do**
    Let Υ be the set of all unused vehicles
    Let υ be the vehicle currently used to serve this route
    Find a vehicle in the set $\{\Upsilon, \upsilon\}$ which has the lowest cost
    Use the found vehicle to serve this route
end for

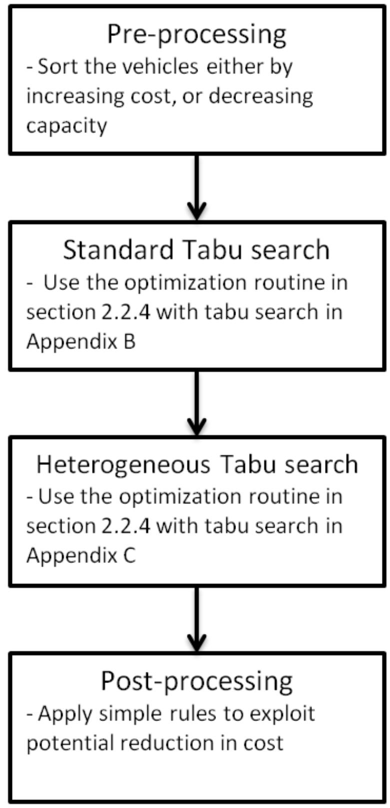

Figure 2.9: The optimization procedure for the heterogeneous fleet Last Mile Problem

2.3.3 Experimental results

We benchmark our heterogeneous fleet algorithm using two data sets which are taken from Taillard (1996). Both test sets contain 8 problems numbered from 13 to 20 with customers' position and demand taken from Christofides and Eilon (1969). The number of customers in each test case is from 50 to 100.

1. The first test set (VFM): the vehicles have different fixed cost and variable cost.

2. The second test set (VFMHE): the vehicles have different variable cost. The fixed cost is 0.

We report the total cost and the computational time reported by each paper, as well as the total cost and computational time of our algorithm running on a desktop computer with Core 2 Duo 2.67Ghz, 4GB RAM. For comparison purposes where

Table 2.3: Test result for VFM test cases

Pr	Size	Taillard (1996)		Gendreau et al. (1999)		Renaud et al. (2002)		Choi & Tcha (2007)		LMP	
		Total cost	Time(s)	Total cost	Time(s)	Total cost	Time(s)	Total cost	Time(s)	Total cost	Time(s)
13	50	2413.78	470	2408.41	724	2406.43	50	2406.36	8	2467.67	20.5
14	50	9119.03	570	9119.03	1033	9122.01	160	9119.03	36	9145.02	20.5
15	50	2586.37	334	2586.37	901	2618.03	45	2586.37	8	2640	16.15
16	50	2741.5	349	2741.5	815	2761.96	28	2720.43	7	2780.72	24.63
17	75	1747.24	2072	1749.5	1022	1757.21	652	1744.83	160	1807.33	42.84
18	75	2373.63	2744	2381.43	691	2413.39	1037	2371.49	46	2432.5	96.1
19	100	8661.81	12528	8675.16	1687	8687.31	1110	8664.29	890	8703.64	97.2
20	100	4047.55	2117	4086.76	1421	4094.54	307	4039.49	161	4175.5	85
Avg.		4211.36	2648	4218.52	1037	4232.61	423.6	4206.54	164	4269.0475	50.365

the total cost is important, we use the pre-processing rule of sorting the vehicles in increasing fixed cost.

For the VFM test cases, we compare our algorithms with those proposed by Taillard (1996), Gendreau *et al.* (1999), Renaud and Boctor (2002) and Choi and Tcha (2007). The result is reported in Table 2.3. The average deviation from the best solution found in the literature is only 1.5%, this can be due to low deviation in test case 14 and test case 19 where the total cost is very high which in turn results in deviations of only 0.2% and 0.4%. For the last three test cases, the computational time is around 90 seconds, but it is still reasonable when we compare with other algorithms, especially for the test case 19.

For the VFMHE test cases, we compare our algorithm with four algorithms solving the heterogeneous fixed fleet vehicle routing problem: Taillard (1996), Tarantilis *et al.* (2004), Li *et al.* (2007) and Brandão (2011). The results are reported in Table 2.4. We can see that our algorithm gives very fast computational time with competitive total cost: the computational time is kept at below 1 minute while the deviation of our cost from the best solution found is on average 2%. Our algorithm only has some problems with test case 19 where the deviation from the best solution found is 5%. Our algorithm has very good computational time with all 4 test cases of 50 customers.

Table 2.4: Test result for VFMHE test cases

Pr	Size	Taillard (1996)		Tarantilis et al.(2004)		Li et al. (2007)		Brandao (2011)		LMP	
		Total cost	Time(s)	Total cost	Time(s)	Total cost	Time(s)	Total cost		Total cost	Time(s)
13	50	1518.05	473	1519.96	843	1517.84	358	1517.84	56	1541.05	8.88
14	50	615.64	575	611.39	387	607.53	141	607.53	55	609.856	8.19
15	50	1016.86	335	1015.29	368	1015.29	166	1015.29	59	1030.965	10
16	50	1154.05	350	1145.52	341	1144.94	188	1144.94	94	1155.4	7.15
17	75	1071.79	2245	1071.01	363	1061.96	216	1061.96	206	1070.942	26.3
18	75	1870.16	2876	1846.35	971	1823.58	366	1831.36	198	1846.053	45.01
19	100	1117.51	5833	1123.83	428	1120.34	404	1120.34	243	1189.73	58.02
20	100	1559.77	3402	1556.35	1156	1534.17	447	1534.17	302	1591.648	59.16
Avg.		1240.48	2011	1236.21	607	1228.21	285	1229.18	151	1254.4555	28.67125

2.4 Conclusions

In this chapter, we propose various algorithms for the last mile mobility system. The algorithm is based on a tabu search heuristics for limited number of vehicles proposed in Lau *et al.* (2003). Our first algorithm serves only the last mile customers, with customers wanting to travel from the transportation hub to their desired destinations. We extend the algorithm to take into account the first mile customers. Next, we consider a schedule rule for the vehicle to meet the first mile customers' pickup time windows. The algorithms are then parallelized using OpenMP. Our algorithms are applicable because of their practical utilities and low computational times.

We also study the problem with heterogeneous fleet where vehicles can have different capacity, fixed cost and variable cost. This extension is very important when we consider the real life last mile mobility system. Along with the pre-, post-processing, we modify our existing tabu search routine into a heterogeneous fleet tabu search algorithm. Experimental results show that our algorithm performs well with existing test cases for the heterogeneous fleet with very good computational time.

3 The multi-period vehicle routing problem

In this chapter, we consider a real life implementation of the last mile mobility system. In practice, the last mile mobility system is a multi-period vehicle routing problem where customers' requests are revealed along the day and the decisions of assigning the customers to the vehicles as well as routing the vehicles have to be made according to the requests received so far. The planning horizon is divided into periods, and the vehicles will start to leave the depot at the beginning of each period. The vehicle, after serving all the customers assigned to it, returns to the depot and is available for the next period. The tabu search heuristics developed in the previous chapter can be utilized for the multi-period setup: at the beginning of each period, we use one of the algorithms to solve for the unserved requests revealed up to that period to get the assignment and the routing plan for the period, and we can continue the same practice for each period in the planning horizon. Nevertheless, major improvements are necessary to make the algorithms more robust: the schedule can be relaxed without deteriorating the service quality; furthermore, the algorithms need to be modified to make use of the heterogeneous fleet.

This chapter is motivated by the service provider's needs. Every company who wants to implement the last mile mobility system will have to face the profitability questions about the systems. Profitability can be ensured by good decisions made at the strategic and the operational level. In the strategic level, the service provider will have to decide how many vehicles they need to buy or rent, what is the capacity of each vehicle. In the operational level, the service provider needs to guarantee an efficient daily business, including good routing plans. For these reasons, the service provider requires a good decision support to assist them in making these hard decisions, and the aim of this chapter is to propose such decision support system.

The contributions of this chapter are, then, two-fold. First, we construct a system which, given a heterogeneous fleet and a multi-period demand, generates the assignment and routing plan for each period. In addition, the system is also a fleet sizing tool to be employed by the service provider: given the demand, the service provider may try different combinations of the fleet and determine which fleet composition is the most efficient and profitable to invest in. Using data from real life public transport system in Singapore, we demonstrate the capability of the system in assisting the service provider for a last mile mobility system based on Clementi area.

3.1 Relaxation of the schedule

3.1.1 Relaxation of the waiting time

In the last chapter, due to the time windows of the first mile customers, we propose a scheduling rule for the vehicle: it can wait if there is no passenger onboard. This scheduling rule is practical for a static problem: schedule the fleet for daily or weekly operations. However, in the multi-period vehicle routing setup, we are going to solve one problem instance for each period, so instead of waiting for the passenger, the vehicle should just go back to the depot as early as possible so that the vehicle is available for serving the next period. Furthermore, when we decompose the planning horizon into periods, the length of each period can be small enough so that we can assume that all first mile customers are ready to be served in this period. This, in turn, means the first mile customers should be at their pickup location at the beginning of the period. Further real life negotiations between customers and the service providers on the time windows at the point of making the requests also facilitate this practice.

3.1.2 Relaxation of the time windows at the depot

We relax further the time windows at the depot of both types of customers; more specifically, we relax the pickup time windows of the LMP customers, along with the delivery time windows of the FMP customers. Intuitively, it is equivalent to keeping the time windows only for the more important nodes of each type: delivery nodes for

LMP and pickup nodes for FMP. In real life, it is reasonable to consider that the LMP customers should reach their desired final destination on time, and the FMP customers are picked up within their available time windows. Furthermore, we observe that the LMP customers normally arrive in batch, so we can assume that they will have the same pickup time windows, and by carefully solve the new problem at a reasonable time frame, we can easily account for the pickup time windows of the LMP customers.

Thus, for the multi-period problem, we consider a LMP+FMP routing algorithm without waiting, time windows relaxed: if the vehicle arrives at the first mile customer early, it will not wait for that customer; furthermore, the time windows at the depot are relaxed.

3.2 Use of the heterogeneous fleet

For the multi-period problem, one of the most challenging problems is how to handle the fluctuation in demand at different time of the day: the demand is low during early morning or late evening, but we may experience very high demand during rush hours. Indeed, we may use vehicles of small capacity to serve the early morning or late evening customers, or to serve a small group of customers which is geographically close; while big capacity vehicles becomes profitable during rush hours when there are a large number of customers travelling together. The service provider may consider employing external, part-time fleet, for example, by using vacant school bus, or taxis, in order to meet the high peak demand. These ideas necessitate the use of a heterogeneous fleet.

3.3 Experimental result

3.3.1 Description of the real life data

We acquire a real data set from the Land Transport Authority of Singapore and the Singapore-MIT Alliance for Research and Technology. The data keep track of the trips of passengers each day from the ez-link card for both MRT and bus system. From this data set, we can easily recognize the travel pattern of the public transport in Singapore.

For the scope of this thesis, we study an example of a Last Mile Mobility System which can be implemented in the Clementi area with the transportation hub is the Clementi MRT station / Bus terminal. The services being considered include two main types 4 feeder bus services which run around the Clementi area: bus 96, 282, 284, and 285. The routes for these buses are described in the Figure 3.1. These four buses cover a substantial area in the Clementi region, including the National University of Singapore, several industrial zones as well as residences. We consider in this part both last mile passengers and first mile passengers in order to reflect truely the real life problem.

The demand frequency of the last and first mile passengers over a period of one week can be plotted as in Figure 3.2 and in Figure 3.3. From these figures, we can see that there is high demand during the morning and afternoon rush hours for week days. Furthermore, the peak for last mile passengers is higher in the morning, and the peak for first mile passengers is higher in the afternoon.

We concentrate, then, on the morning demand for day 2, since day 2 has the highest peaks for both last mile and first mile customers. We extract the data for each 5 minute time slot from 6.45am to 10am. The detail about the demand can be found in Table 3.1 and Figure 3.4.

We use the rolling horizon method to solve this multi-period problem: in each period, we solve the problem with the demand extracted from the real life data, under the assumption that all the passengers of each period have to be served (or equivalently, we cannot delay the passenger to the time slot after). For this problem of 40 periods, the rolling horizon method indicates that we solve 40 deterministic problems, one problem for each period. We use the word deterministic to differentiate the method

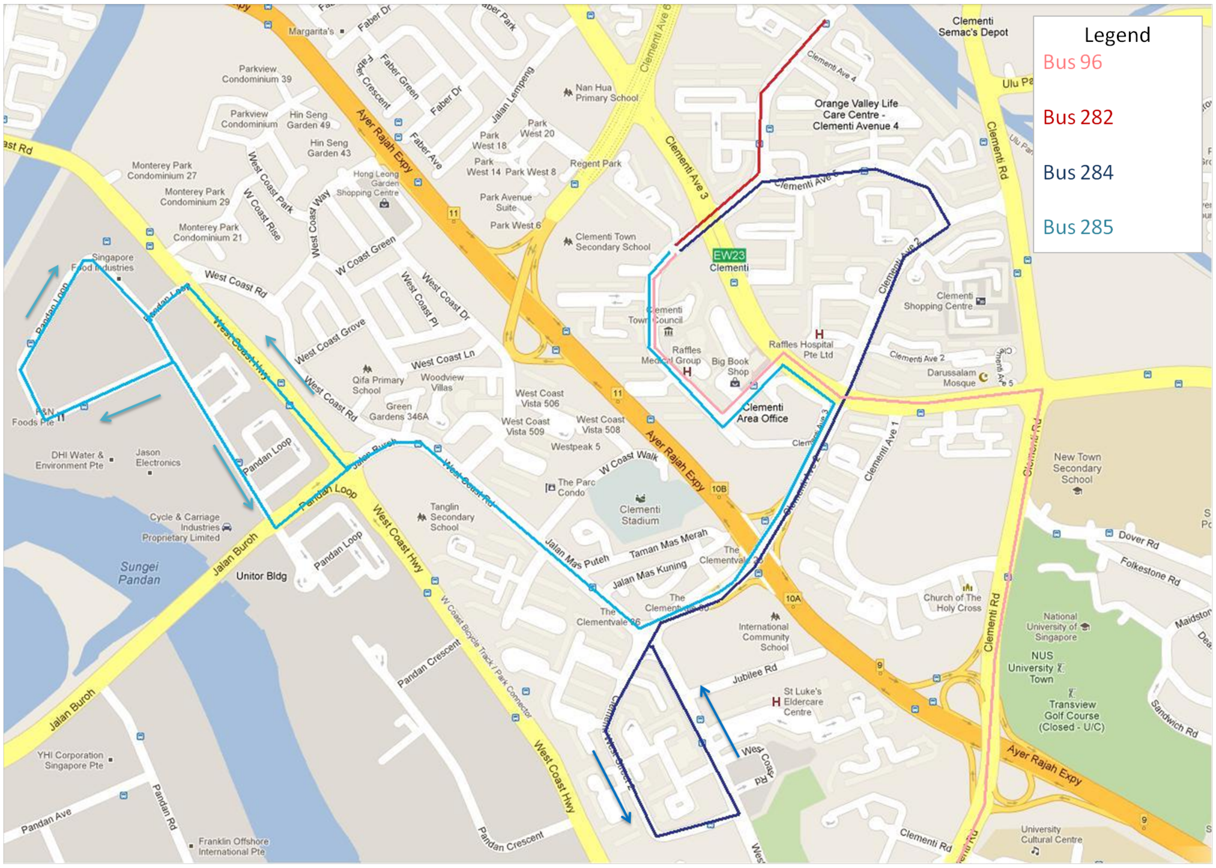

Figure 3.1: Map of feeder bus services in Clementi (Map taken from Google Maps)

from other dynamic/probabilistic methods where the demand arrives and is taken into account dynamically.

Furthermore, by using the real life data, we can extract the average speed of the vehicles and make the problem even more realistic. The average speed used in this part is 13.32km/h. We also require that each vehicle, after coming back to the depot, needs to rest for 5 minutes before it is available for another trip. This 5 minute buffer has several real life implications: it can represent the working condition constraint where the drivers need to rest after a certain duration of work, and it can also be used to offset possible adverse conditions such as traffic jams, bad weather which may affect the time the vehicles return to the depot.

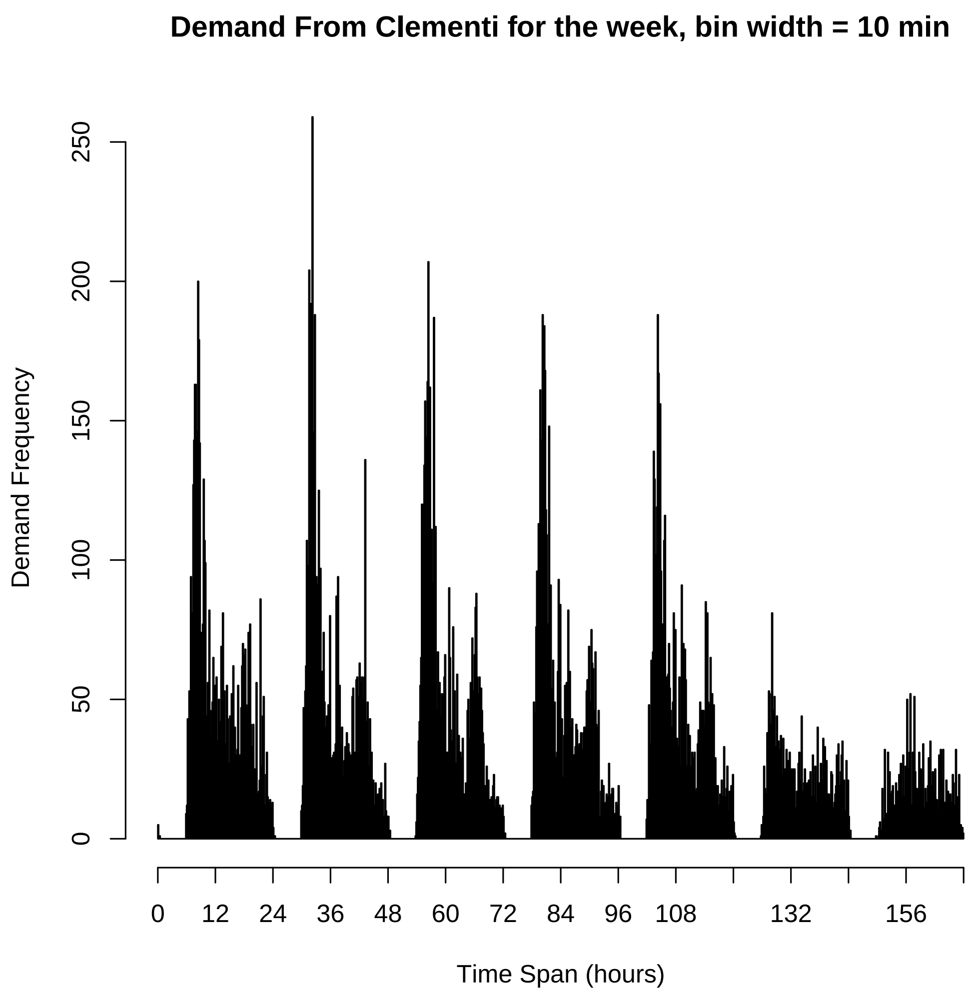

Figure 3.2: Demand frequency of last mile passengers from Clementi Bus interchange during one week

3.3.2 Test using internal fleet

In this section, we consider the situation where the service provider uses only their own fleet: the service provider buys or signs a full time contract to use the fleet for the last mile mobility system. In order to reduce the maintenance cost, it is suggested that the service provider use a homogeneous fleet. Here, we assume that the service provider has three choices on the capacity of the vehicles: 10, 20 or 30. The number of vehicles required for each fleet is depicted in Table 3.2, and the details about the

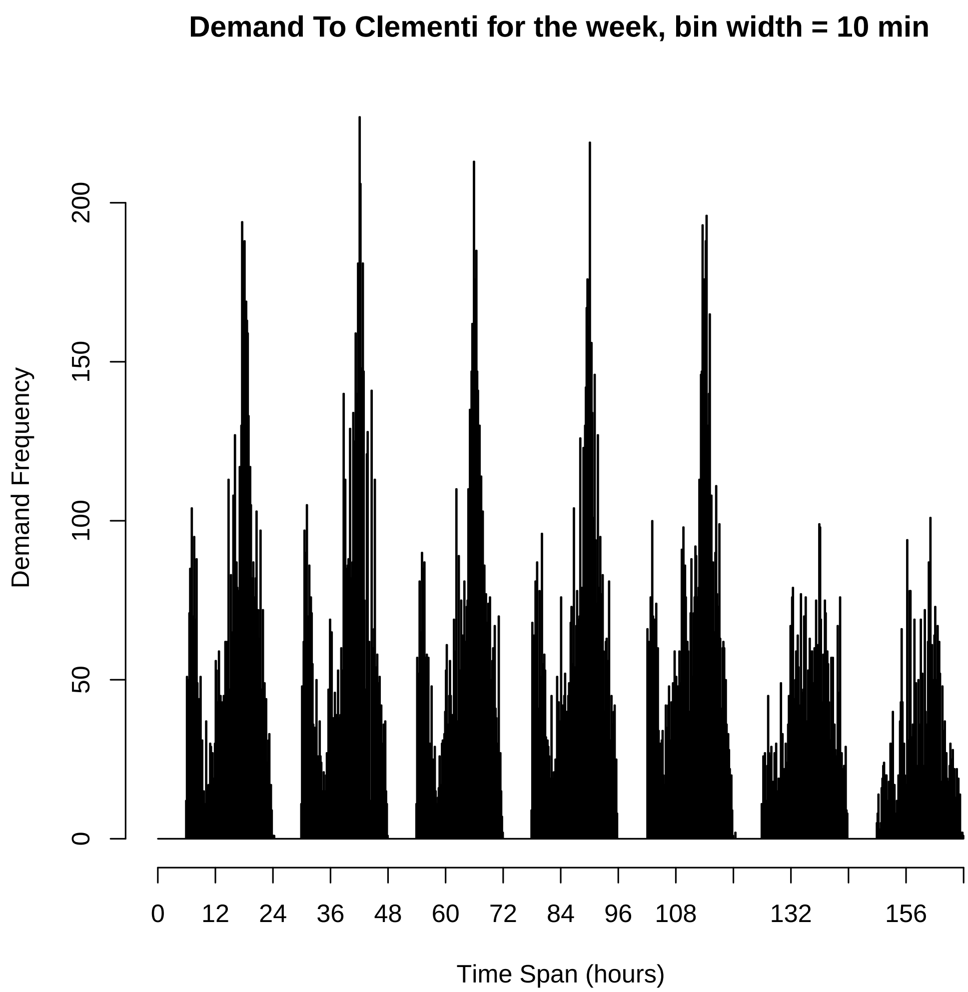

Figure 3.3: Demand frequency of first mile passengers to Clementi Bus interchange during one week

routing for each period are shown in the Table 3.3 and Table 3.4.

We can see that the fleet of capacity 20 and 30 outperforms the fleet of capacity 10 on both the number of vehicles required, as well as on the routing for each period. By increasing the size of the vehicles from 10 to 20, we manage to reduce more than 30 vehicles; however, further increasing the size from 20 to 30 only reduces 10 vehicles. The fleet of 30 gives better or equal distance travelled each period to the fleet of 20. From these results, we recommend that the service provider choose the fleet of capacity 20 or 30, and disregard the choice of the fleet of capacity 10.

Table 3.1: Demand for each 5 minute time slot of day 2

Period	Time	LMP	FMP		Period	Time	LMP	FMP		Period	Time	LMP	FMP
1	6:45	53	32		14	7:50	102	17		27	8:55	48	5
2	6:50	44	58		15	7:55	91	53		28	9:0	36	8
3	6:55	18	43		16	8:0	66	29		29	9:5	60	7
4	7:0	45	47		17	8:5	55	27		30	9:10	9	11
5	7:5	60	22		18	8:10	125	12		31	9:15	82	15
6	7:10	52	22		19	8:15	133	24		32	9:20	25	6
7	7:15	43	9		20	8:20	72	12		33	9:25	26	2
8	7:20	27	67		21	8:25	75	0		34	9:30	64	8
9	7:25	52	18		22	8:30	39	30		35	9:35	61	29
10	7:30	113	12		23	8:35	100	5		36	9:40	72	19
11	7:35	88	11		24	8:40	71	16		37	9:45	17	7
12	7:40	57	21		25	8:45	117	8		38	9:50	21	9
13	7:45	75	55		26	8:50	40	45		39	9:55	76	15
										40	10:0	43	6

Table 3.2: Number of vehicles required for each capacity

	Capacity 10	Capacity 20	Capacity 30
Number of Vehicles Required	85	51	41

Table 3.3: Test result for internal fleet, period 1-20

	Capacity 10		Capacity 20		Capacity 30	
Period	Vehicles Used	Distance	Vehicles Used	Distance	Vehicles used	Distance
1	7	44.5	4	31.65	3	21.9
2	7	43.65	4	28.1	3	22.2
3	6	28.45	4	22.7	3	20.39
4	6	43.1	4	29.2	4	23.65
5	8	53.85	5	31.45	4	28.1
6	6	43.3	5	34.9	3	20.5
7	6	38.4	4	26.4	3	18.7
8	7	38.7	4	23.9	3	16.2
9	6	41.35	4	25.3	3	22.4
10	13	85.8	7	50.6	6	37.2
11	9	63.6	5	34.79	5	34.79
12	7	44	4	27.1	4	27.1
13	11	67.09	6	40.75	4	26.45
14	11	75	7	47.5	5	33.9
15	11	70.95	6	43.8	5	34.79
16	7	51.2	5	35.79	4	28.2
17	9	56.7	6	34.5	4	30.4
18	13	92.5	7	49.4	5	35
19	15	102.65	8	58.75	6	45
20	9	60.6	6	40.5	5	33.9

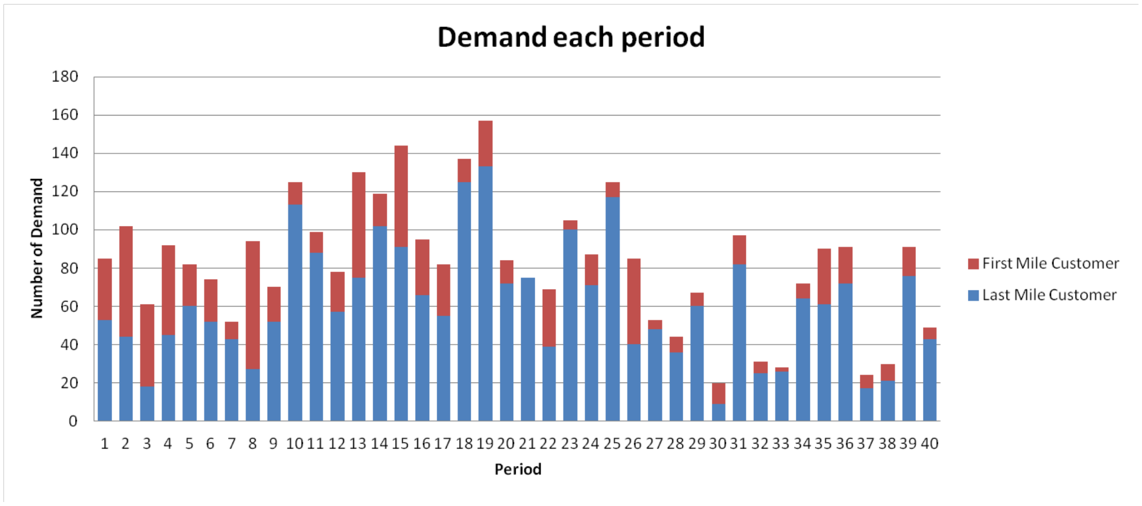

Figure 3.4: Number of Customers over time

3.3.3 Test using internal and external fleet

In this section, we consider the case where the service provider has the possibility to sign a contract to use the external fleet. The external fleet may include school bus, private bus, or taxis. The external will relieve the initial investment for the service provider; however, the fixed cost and variable cost for the external fleet are higher than the internal fleet.

Due to the results in the last part, we assume that the service provider will choose to buy the internal fleet of capacity 20. The external fleet consists of vehicles of capacity 20 and 10 as well as of taxis of capacity 4. Table 3.5 presents two fleet compositions which are used in the test. The fixed cost is charged whenever the vehicle is used, while the variable cost is charged per km distance travelled by the vehicle. The external vehicles with capacity 4 are taxis.

The results with two types of fleet composition are shown in Table 3.6 and Table 3.7. We can see that the fleet composition 2, which uses five more external vehicles of capacity 10, does not need to employ any taxi for the period routing. Due to this fact, the fleet composition 2 manages to perform better than the fleet composition 1 in both the fixed and variable costs.

Table 3.4: Test result for internal fleet, period 21-40

	Capacity 10		Capacity 20		Capacity 30	
Period	Vehicles Used	Distance	Vehicles Used	Distance	Vehicles used	Distance
21	9	59.65	5	36.2	4	29.5
22	7	45.85	5	27.1	4	27.1
23	11	78.4	6	43.9	4	29.5
24	9	63.3	6	41.2	5	34.5
25	13	93.3	8	55.9	6	41.5
26	8	46.1	5	25.25	3	21.8
27	6	41.5	4	27.1	3	20.39
28	5	32.79	4	26.1	3	19.39
29	8	49.35	5	29.25	4	22.55
30	2	13.1	2	13	2	13
31	10	65.59	6	38.79	5	32.1
32	4	27.8	3	21.1	2	14.4
33	4	25.2	3	18.5	2	11.8
34	7	47.9	4	27.8	3	21.1
35	10	62.2	6	38.04	4	27.8
36	9	57.05	5	33.4	4	26.7
37	2	14.4	2	14.4	2	14.4
38	4	26.2	3	19.5	3	19.5
39	10	61.95	5	33.4	4	26.7
40	5	34.5	3	21.1	3	21.1

Table 3.5: External Fleet Composition

	Fleet 1				Fleet 2			
	Capacity	No of Vehicles	Fixed cost	Variable cost	Capacity	No of vehicles	Fixed cost	Variable cost
Internal	20	30	100	10.0	20	30	100	10.0
External	20	5	200	15.0	20	10	200	15.0
External	10	10	175	12.5	10	10	175	12.5
External	4	19	250	25.0	4	0	250	25.0

Table 3.6: Test result for internal and external fleet, period 1-20

	Fleet 1		Fleet 2	
Period	Fixed cost	Variable cost	Fixed cost	Variable cost
1	400	316.5	400	316.5
2	400	281	400	281
3	400	227	400	227
4	400	292	400	292
5	500	314.5	500	314.5
6	500	349	500	349
7	400	264	400	264
8	675	324.8	675	324.8
9	475	275.3	475	264.1
10	700	506	700	506
11	575	362.9	575	362.9
12	400	271	400	271
13	775	450.8	775	450.5
14	1300	685.9	1150	603
15	700	471	700	435.7
16	1250	678.7	850	466.5
17	2700	1493.5	875	468
18	700	494	700	494
19	1125	710.6	1075	711.2
20	1800	1063	1000	455.9

Table 3.7: Test result for internal and external fleet, period 21-40

	Fleet 1		Fleet 2	
Period	Fixed cost	Variable cost	Fixed cost	Variable cost
21	1650	1114	575	381.2
22	650	372.5	850	380.2
23	675	458.2	600	439
24	950	510.7	1150	582
25	2450	1606.5	1250	694
26	400	295	500	252.4
27	400	271	400	271
28	400	261	400	261
29	500	292.5	500	292.5
30	200	130	200	153
31	600	387.9	600	387.9
32	300	211	300	211
33	300	185	300	185
34	400	278	400	278
35	750	406.1	750	406.1
36	500	334	500	334
37	200	144	200	144
38	300	195	300	195
39	500	334	500	334
40	300	211	300	211

The output of the system is a detailed planning for the morning rush hour which is decomposed into 40 periods of 5 minutes. Such a planning is given in the Figure 3.5. For each vehicle, the red horizontal stripe indicates that the vehicle is used for planning during these periods.

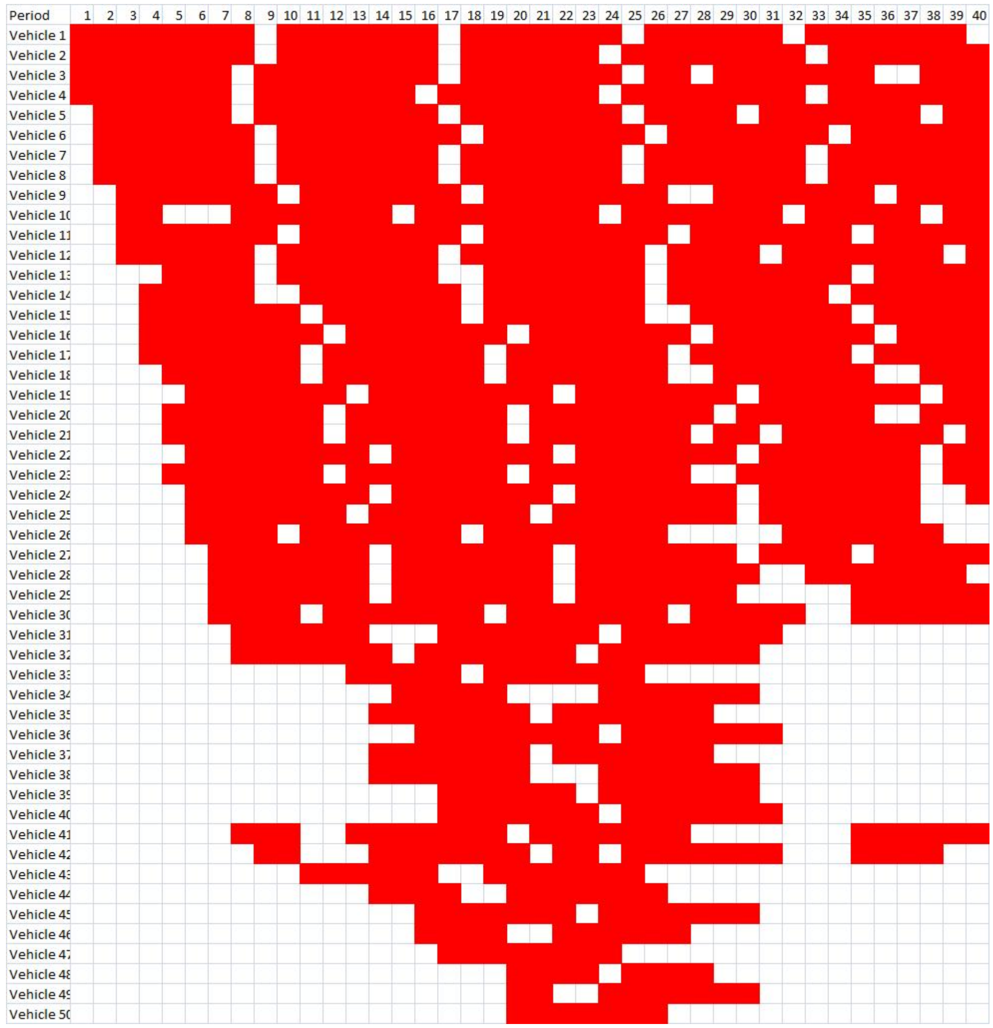

Figure 3.5: Morning rush hour planning for Fleet 2

3.4 Conclusions

In this chapter, we propose the multi-period algorithm for the last mile mobility system. We employ the heterogeneous fleet algorithm developed in the last chapter to solve the multi horizon using rolling horizon method. In the multi-period settings, we can relax the scheduling rule without deteriorating the quality of the service. The goal of this multi-period routing algorithm is two-fold: the service provider can use the algorithm for their daily operations, or they can use it for planning the fleet size, capacity and external resources. Our algorithm is tested using real life public transport data taken from the Land Transport Authority of Singapore. The algorithm is tested under two scenarios: an internal fleet and an internal-external fleet. The results suggest that the service provider should invest in an internal fleet of capacity 20 or 30 instead of 10. Furthermore, we also suggest the service provider to sign an external fleet contract in order to reduce the initial capital investments. By carefully configuring the fleet compositions, the service provider will find an efficient fleet size for the system. A good fleet composition and a reasonable daily routing will better ensure the profitability and the viability of the last mile mobility system, which is the critical factor for the service provider to decide whether to enter the market.

4 The Last Mile Problem with uncertain travelling time

The stochastic VRP has been an active field of research during the last few years. The problem is highly important as the travelling time is always subject to uncertainty due to possible congestion on the route. In the Last Mile mobility system, the problem is even more critical since the system is implemented in high density region (city center, etc.) which may experience tremendous fluctuations in the travelling time. Furthermore, in the multi-period setup, uncertain travelling time may cause the vehicle to return late to the depot, which in turns results in further negative changes in the multi-period planning. In this chapter, we consider a satisficing approach to the stochastic VRP. After characterizing the travelling time and the lateness index, we demonstrate changes to the tabu search, and then test the satisficing approach against the static mean travelling time approach.

4.1 Problem formulation

In this chapter, we consider a last mile mobility system where there are only last mile customers. When the travelling time is certain, we have imposed a strict time windows in order to satisfy a certain level of service quality to the customers. In the case of uncertain travelling time, we assume that the time windows are soft, and there is a penalty if the time windows are violated. The late time windows become a time target to achieve: we want the travelling time to meet the target as high as possible. In this chapter, we use the satisficing measure approach introduced by Brown and Sim (2009).

4.1.1 Modeling travelling time

Our model of uncertainty is defined by a state space Ω and a σ-algebra $\mathcal{F}$ of events in Ω. We model the travelling time between node i and node j as a random variable, more specifically, as an affine function of independently distributed factors $\tilde{z}_1, \ldots, \tilde{z}_K$, i.e

$$\tilde{t}_{ij} = t_{ij}^0 + \sum_{k=1}^{K} t_{ij}^k \tilde{z}_k$$

We define the set $\mathcal{T}$ of all attainable travelling time as:

$$\mathcal{T} = \{\tilde{t} \,|\, \exists (t^0, \boldsymbol{t}) \in \mathbb{R}^{K+1} : \tilde{t}(\omega) = t^0 + \sum_{k=1}^{K} t^k \tilde{z}^k(\omega)\}$$

The definition of $\tilde{z}_k$ is general, that is, the probability distribution of $\tilde{z}_k$ belongs to some family of distributions $\mathbb{F}_k$. $\mathbb{F}_k$ can be well defined distributions (normal distribution, gamma distribution), or it can take some ambiguous distributions. For an example of ambiguous distributions, we can assume that $\mathbb{F}_k$ contains all possible distributions with bounded support $\tilde{z}_k \in [\underline{z_k}, \overline{z_k}]$ and with mean support $[\underline{\mu}_k, \overline{\mu}_k] \in [\underline{z}^k, \overline{z}^k]$.

$$\mathbb{F}_k = \left\{ \mathbb{P}_k \middle| \mathbb{P}_k \left(\tilde{z}^k \in [\underline{z}^k, \overline{z}^k] \right) = 1, \mathbb{E}_{\mathbb{P}_k} \left(\tilde{z}^k \right) \in [\underline{\mu}_k, \overline{\mu}_k] \right\} \tag{4.1}$$

4.1.2 Lateness index

Definition 1 *Given a time target, $\tau \in \mathbb{R}$, the lateness index (LI), $\rho_\tau : \mathcal{T} \to [0, +\infty)$ is defined by*

$$\rho_\tau(\tilde{t}) = \sup\{a > 0 : C_a(\tilde{t}) \le \tau\}$$

where the function $C_a(\tilde{t}) : \mathcal{T} \to \mathbb{R}$ is defined by

$$C_a(\tilde{t}) = \sup_{\mathbb{P} \in \mathbb{F}} \left(\frac{1}{a} \log \mathbb{E}_{\mathbb{P}}[\exp(a\tilde{t})] \right) = \frac{1}{a} \log \sup_{\mathbb{P} \in \mathbb{F}} \left(\mathbb{E}_{\mathbb{P}}[\exp(a\tilde{t})] \right)$$

It is important to note that the lateness index is similar to the entropic satisficing measure proposed by Brown and Sim Brown and Sim (2009). Furthermore, we assume that

$$\tilde{t} = t^0 + \sum_{k=1}^{K} t^k \tilde{z}^k$$

Since each $\tilde{z}_1, \ldots, \tilde{z}^K$ are independent, $C_a(\tilde{t})$ can be simplified as:

$$C_a(\tilde{t}) = C_a(t^0 + \sum_{k=1}^{K} t^k \tilde{z}^k) = t^0 + \frac{1}{a}\sum_{k=1}^{K} \log \sup_{\mathbb{P}_k \in \mathbb{F}_k} \left(\mathbb{E}_{\mathbb{P}_k}[\exp(at^k \tilde{z}^k)]\right) = t^0 + \sum_{k=1}^{K} C_a(t^k \tilde{z}^k)$$

Lemma 1 *$C_a(\tilde{t})$ is a non-decreasing function of $a > 0$.*

Proof: For $a_1 > a_2 > 0$,

$$\begin{aligned}
C_{a_1}(\tilde{t}) = \quad & \sup_{\mathbb{P} \in \mathbb{F}} \frac{1}{a_1} \log \mathbb{E}_{\mathbb{P}}[\exp(a_1 \tilde{t}))] \\
= \quad & \sup_{\mathbb{P} \in \mathbb{F}} \frac{1}{a_1} \log \mathbb{E}_{\mathbb{P}} \left[(\exp(a_2 \tilde{t}))^{\frac{a_1}{a_2}} \right] \\
\geq \quad & \sup_{\mathbb{P} \in \mathbb{F}} \frac{1}{a_1} \log \left(\mathbb{E}_{\mathbb{P}} \left[\exp(a_2 \tilde{t}) \right] \right)^{\frac{a_1}{a_2}} \\
= \quad & \sup_{\mathbb{P} \in \mathbb{F}} \frac{1}{a_2} \log \mathbb{E}_{\mathbb{P}}[\exp(a_2 \tilde{t}))] \\
= \quad & C_{a_2}(\tilde{t})
\end{aligned}$$

where the inequality comes from the Jensen's inequality. ■

Next, we characterize the function $C_a(t^k z^k)$ for different family of distributions $\mathbb{F}_k$:

Theorem 2 • *If $\mathbb{F}_k$ contains a single normal distribution with mean μ_k and standard deviation σ_k, then*

$$C_a(t^k \tilde{z}^k) = t^k \mu_k + \frac{1}{2}(t^k)^2 \sigma_k^2 a$$

- *If $\mathbb{F}_k$ contains a single uniform distribution in $[\underline{z}^k, \overline{z}^k]$, then*

$$C_a(t^k \tilde{z}^k) = \frac{1}{a} \log[\frac{\exp(at^k \overline{z}^k) - \exp(at^k \underline{z}^k)}{at^k(\overline{z}^k - \underline{z}^k)}]$$

- *If $\mathbb{F}_k$ contains a single gamma distribution with shape α and scale θ, and $t^k > 0$, then*

$$C_a(t^k \tilde{z}^k) = -\frac{\alpha t^k}{a} \log(1 - a\theta)$$

- *If $\mathbb{F}_k$ contains all possible distribution with bounded support and bounded mean support as in (4.1), then*

$$C_a(t^k \tilde{z}^k) = \begin{cases} \frac{1}{a} \log \frac{(\overline{z}^k - \underline{\mu}^k)\exp(at^k \underline{z}^k) + (\underline{\mu}^k - \underline{z}^k)\exp(at^k \overline{z}^k)}{\overline{z}^k - \underline{z}^k} & \text{when } t^k < 0 \\ \frac{1}{a} \log \frac{(\overline{z}^k - \overline{\mu}^k)\exp(at^k \underline{z}^k) + (\overline{\mu}^k - \underline{z}^k)\exp(at^k \overline{z}^k)}{\overline{z}^k - \underline{z}^k} & \text{when } t^k \geq 0 \end{cases}$$

Proof: The first three equations are derived from the moment generating functions of the normal and gamma distribution. For the third equation, let $\lambda = at^k$, we first determine the expressions for $\sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\exp(\lambda \tilde{z}^k) \right)$. First we formulate it as a convex optimization problem:

$$\begin{aligned} \max_{\mathbb{P}} \quad & \mathbb{E}_{\mathbb{P}}\left(\exp(\lambda\tilde{z}^k)\right) \\ \text{s.t.} \quad & \mathbb{E}_{\mathbb{P}}(1) = 1 \\ & \mathbb{E}_{\mathbb{P}}(\tilde{z}^k) \leq \overline{\mu}^k \\ & \mathbb{E}_{\mathbb{P}}(\tilde{z}^k) \geq \underline{\mu}^k \\ & \mathbb{P}\left(\{z^k \in [\underline{z}^k, \overline{z}^k]\}\right) = 1 \end{aligned}$$

By weak duality, we have:

$$\begin{aligned} \max_{\mathbb{P}\in\mathbb{F}} \mathbb{E}_{\mathbb{P}}\left(\exp(\lambda\tilde{z}^k)\right) \leq \min \quad & y_0 + \overline{\mu}^k y_1 - \underline{\mu}^k y_2 \\ \text{s.t.} \quad & y_0 + z^k y_1 - z^k y_2 \geq \exp(\lambda z^k) \quad \forall z^k \in [\underline{z}^k, \overline{z}^k] \\ & y_1, y_2 \geq 0 \end{aligned} \tag{4.2}$$

Since $\exp(\lambda z^k) + (y_2 - y_1)z^k$ is a convex function in z^k, we note that

$$\begin{aligned} y_0 \geq & \max_{z^k \in [\underline{z}^k, \overline{z}^k]} \left(\exp(\lambda z^k) + (y_2 - y_1)z^k\right) \\ = & \max\left\{\exp(\lambda\underline{z}^k) + (y_2 - y_1)\underline{z}^k, \exp(\lambda\overline{z}^k) + (y_2 - y_1)\overline{z}^k\right\} \end{aligned} \tag{4.3}$$

Thus,

$$\begin{aligned} & \max_{\mathbb{P}\in\mathbb{F}} \mathbb{E}_{\mathbb{P}}\left(\exp(\lambda\tilde{z}^k)\right) \\ \leq & \min_{y_1, y_2 \geq 0} \max \left\{ \begin{array}{l} \exp(\lambda\underline{z}^k) + (\overline{\mu}^k - \underline{z}^k)y_1 + (\underline{z}^k - \underline{\mu}^k)y_2, \\ \exp(\lambda\overline{z}^k) + (\overline{\mu}^k - \overline{z}^k)y_1 + (\overline{z}^k - \underline{\mu}^k)y_2 \end{array} \right\} \end{aligned} \tag{4.4}$$

The optimal value of y_1 and y_2 should equate the two terms in (4.4), so

$$y_1 - y_2 = \frac{\exp(\lambda\overline{z}^k) - \exp(\lambda\underline{z}^k)}{\overline{z}^k - \underline{z}^k}$$

When $\lambda < 0$, substitute y_2 in terms of y_1 in (4.4) we have:

$$
\begin{aligned}
&\max_{\mathbb{P}\in\mathbb{F}} \mathbb{E}_{\mathbb{P}}\left(\exp(\lambda\tilde{z}^k)\right)\\
\leq &\min_{y_1\geq 0}\left\{\exp(\lambda\underline{z}^k) + (\overline{\mu}^k - \underline{z}^k)y_1 + (\underline{z}^k - \underline{\mu}^k)\left(y_1 - \frac{\exp(\lambda\overline{z}^k) - \exp(\lambda\underline{z}^k)}{\overline{z}^k - \underline{z}^k}\right)\right\}\\
= &\min_{y_1\geq 0}\left\{\exp(\lambda\underline{z}^k) + (\overline{\mu}^k - \underline{\mu}^k)y_1 + (\underline{z}^k - \underline{\mu}^k)\left(-\frac{\exp(\lambda\overline{z}^k) - \exp(\lambda\underline{z}^k)}{\overline{z}^k - \underline{z}^k}\right)\right\}\\
= &\frac{(\overline{z}^k - \underline{\mu}^k)\exp(\lambda\underline{z}^k) + (\underline{\mu}^k - \underline{z}^k)\exp(\lambda\overline{z}^k)}{\overline{z}^k - \underline{z}^k}
\end{aligned}
$$

When $\lambda \geq 0$, substitute y_1 in terms of y_2 in (4.4) we have:

$$
\begin{aligned}
&\max_{\mathbb{P}\in\mathbb{F}} \mathbb{E}_{\mathbb{P}}\left(\exp(\lambda\tilde{z}^k)\right)\\
\leq &\min_{y_1\geq 0}\left\{\exp(\lambda\underline{z}^k) + (\overline{\mu}^k - \underline{z}^k)\left(y_2 + \frac{\exp(\lambda\overline{z}^k) - \exp(\lambda\underline{z}^k)}{\overline{z}^k - \underline{z}^k}\right) + (\underline{z}^k - \underline{\mu}^k)y_2\right\}\\
= &\min_{y_1\geq 0}\left\{\exp(\lambda\underline{z}^k) + (\overline{\mu}^k - \underline{\mu}^k)y_2 + (\overline{\mu}^k - \underline{z}^k)\left(\frac{\exp(\lambda\overline{z}^k) - \exp(\lambda\underline{z}^k)}{\overline{z}^k - \underline{z}^k}\right)\right\}\\
= &\frac{(\overline{z}^k - \overline{\mu}^k)\exp(\lambda\underline{z}^k) + (\overline{\mu}^k - \underline{z}^k)\exp(\lambda\overline{z}^k)}{\overline{z}^k - \underline{z}^k}
\end{aligned}
$$

The optimal distribution can be achieved under a two point distribution:

$$
\begin{cases}
\mathbb{P}(\tilde{z}^k = \underline{z}^k) = \frac{\overline{z}^k - \overline{\mu}^k}{\overline{z}^k - \underline{z}^k}, \mathbb{P}(\tilde{z}^k = \overline{z}^k) = \frac{\overline{\mu}^k - \underline{z}^k}{\overline{z}^k - \underline{z}^k}, \text{ when } \lambda \geq 0\\
\mathbb{P}(\tilde{z}^k = \underline{z}^k) = \frac{\overline{z}^k - \underline{\mu}^k}{\overline{z}^k - \underline{z}^k}, \mathbb{P}(\tilde{z}^k = \overline{z}^k) = \frac{\underline{\mu}^k - \underline{z}^k}{\overline{z}^k - \underline{z}^k}, \text{ when } \lambda < 0
\end{cases}
$$

This completes the proof. ■

Definition 2 *For a routing plan of N customers, for each customers i, $i = 1, \ldots, N$, there is a time target τ_i, the overall lateness index of the routing plan is defined as:*

$$\varrho = \sum_{i=1}^{N} \rho_{\tau_i}(\tilde{t}_i)$$

where $\tilde{t}_i$ denotes the travelling time to customer i.

The travelling time $\tilde{t}_i$ to customer i is difficult to defined if we use linear mixed integer formulation. Nevertheless, when we use heuristics algorithm, $\tilde{t}_i$ can be easily computed based on the routing plan, in fact, it equals to the sum of the travelling time of all previous arcs connecting the depot to the customers. For example, consider the following plan, where one of the routes has the vehicle travelling from the depot (node 0) to customer 1, customer 2, customer 3 and back to the depot.

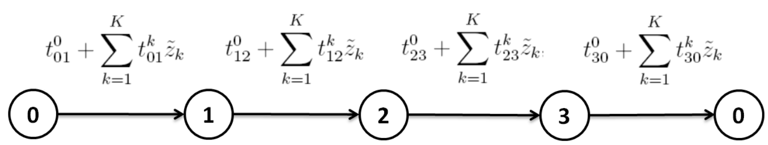

Figure 4.1: Example of a vehicle route serving customer 1, 2 and 3

$$\tilde{t}_{01} = t^0_{01} + \sum_{k=1}^{K} t^k_{01}\tilde{z}_k$$

$$\tilde{t}_{12} = t^0_{12} + \sum_{k=1}^{K} t^k_{12}\tilde{z}_k$$

$$\tilde{t}_{23} = t^0_{23} + \sum_{k=1}^{K} t^k_{23}\tilde{z}_k$$

Thus, the travelling times to the three customers are:

$$\begin{aligned}
\tilde{t}_1 =& \quad t^0_{01} + \sum_{k=1}^{K} t^k_{01}\tilde{z}_k \\
\tilde{t}_2 =& \quad t^0_{01} + t^0_{12} + \sum_{k=1}^{K} (t^k_{01} + t^k_{12})\tilde{z}_k \\
\tilde{t}_3 =& \quad t^0_{01} + t^0_{12} + t^0_{23} + \sum_{k=1}^{K} (t^k_{01} + t^k_{12} + t^k_{23})\tilde{z}_k
\end{aligned}$$

According to Lemma (1), when we know $\tilde{t}_i$, we can bisect on a to find $\rho_{\tau_i}(\tilde{t}_i)$, and thus, ϱ can also be computed. The bisection algorithm to find the lateness index for Customer i with target τ_i and travelling time $\tilde{t}_i$ is described in Algorithm 6, where M is a very large number.

Algorithm 6 Bisection algorithm to find lateness index

$upper = M, lower = 0$
while $upper - lower > \epsilon$ **do**
  $a = (upper + lower)/2$
  if $C_a(\tilde{t}_i) \leq \tau_i$ **then**
    $lower = a$
  else
    $upper = a$
  end if
end while
return $(upper + lower)/2$

4.2 The tabu search heuristics

The lateness index introduced in the last section can be easily implemented within the tabu search heuristics introduced in the first chapter. The lateness index becomes one of the costs in the hierarchical cost, and the order will be:

1. Maximize the satisfied customers
2. Minimize the number of vehicles used
3. Maximize the overall lateness index
4. Minimize the distance travelled by the fleet

4.3 Experimental result

Since there is no testing standard for the vehicle routing problem under uncertain or stochastic travelling time, for the experiments, we will use randomly generated test cases which consist of only last mile customers. Without loss of generality, we assume that the service time is known with certainty. For any two customer i and j whose mutual distance is d_{ij}, we model the travelling time from i to j, $\tilde{t}_{ij}$, as an independent random variable with normal and gamma distribution.

1. Normal distribution: $\tilde{t}_{ij}$ follows a normal distribution with

$$\mathbb{E}(\tilde{t}_{ij}) = d_{ij}, \sigma(\tilde{t}_{ij}) = \sqrt{d_{ij}}$$

2. Gamma distribution: $\tilde{t}_{ij}$ follows a gamma distribution of shape α_{ij} and scale θ_{ij} with

$$\alpha_{ij} = d_{ij}, \theta_{ij} = \theta = 1$$

   This gamma distribution will have mean d_{ij} and standard deviation $\sqrt{d_{ij}}$, which is of the same magnitude as the above normal distribution.

It is important to note that, in the case of normal distribution, the arrival time to each customer follows a normal distribution as the sum of normal random variables is again a normal random variable. Furthermore, in the case of gamma distribution, the arrival time to each customer follows a gamma distribution as the sum of gamma random variables with the same scale parameter is again a gamma random variable. As all the travelling time is model with $\theta = 1$, the arrival time to each customer follows a gamma distribution with scale θ. Even though the normal distribution is more popular and easier to understand, gamma distribution is a better way of modeling stochastic travelling time for two main reasons: first, gamma distribution is nonnegative, and second, practitioners may want to model the arrival process of vehicles at each road junctions as a Poisson process, which results to the fact the travelling time between two junctions is a gamma distribution.

To benchmark the solution of the lateness index algorithm, we compare the solution of the lateness index algorithm with the solutions of two algorithms originating from the deterministic basic last mile algorithms introduced in chapter 2:

1. Deterministic Mean algorithm: where we use the mean value d_{ij} for the distance between customer i and j.

Table 4.1: Distance results by each algorithm

	Normal distribution			Gamma distribution		
Test case	Mean	90th-percentile	Lateness index	Mean	90th-percentile	Lateness index
101	1397.87	1489.78	1646.02	1397.87	1482.9	1538.99
102	1407.48	1515.78	1647.66	1407.48	1554.84	1626.25
103	1427.87	1478.37	1583.87	1427.87	1537.49	1656.96

2. Deterministic 90th-percentile algorithm: where we use, with the abuse of notation:

$$\overline{d}_{ij} = \inf\{d \mid \quad \mathbb{P}_{ij}(\tilde{t}_{ij} < d) \geq 0.90\}$$

In fact, in the mean algorithm, we use only the information about the mean of $\tilde{t}_{ij}$ and disregard any other information about the distribution of $\tilde{t}_{ij}$. On the other hand, the 90th-percentile algorithm is equivalent to the chance constrained formulation of the problem: for any feasible solution of the 90th-percentile algorithm, each customer will be served with the probability of being in time at least 90%.

We use three basic last mile test cases to test the three algorithms. For each test case, every algorithm uses 12 vehicles to carry all the customers. The distance travelled is reported in the Table 4.1. The mean algorithm gives the lowest distance travelled while the lateness index algorithm gives the highest distance travelled for all three test cases. We can see that in order to satisfy the chance constraints, the 90th-percentile results in a higher cost of the distance travelled than the mean algorithm.

Next, we assess the quality of the solution subject to the stochastic travelling time. Since the probability that the time windows are violated can be very close to zero, we compare the natural logarithm of the probability of violating the time windows for each customer of each test case. We plot the empirical cumulative distribution instead of reporting statistical values to have a full perspective on the quality of the solution. Furthermore, we use first order stochastic dominance to assess whether the one solution is better than another: if the empirical cumulative distribution plot of solution A is above that of solution B, we say that A is (first-order) stochastically dominant to B or equivalently, solution A is better than solution B.

For both normal and gamma distribution, the lateness index solution has better quality than the mean solution for all test cases. However, neither the lateness index solution nor the 90th-percentile solution is better than the other. In fact, the 90th-percentile solution ensures that all the violation probability of each customer is less than 10%, however, for lateness index and mean solution, there is no upper bound for the violation probability. Despite the fact that the lateness index solution may have high violation probability customers, it is worthy to note that the number of these customers is always less than 10.

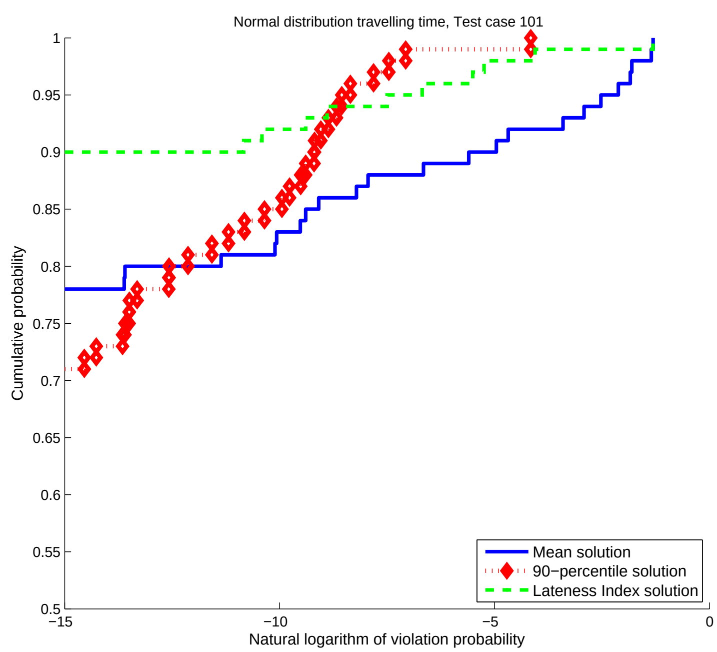

Figure 4.2: Empirical cumulative distribution of the natural logarithm of violation probability for each customer with normal distribution, test case 101

In spite the fact that the lateness index algorithm cannot perform better than the 90th-percentile algorithm, the lateness index is still a promising approach. The strength of the lateness index algorithm lies in its analytical foundation, which make the lateness index a practical algorithm for real life implementation, especially in the following cases:

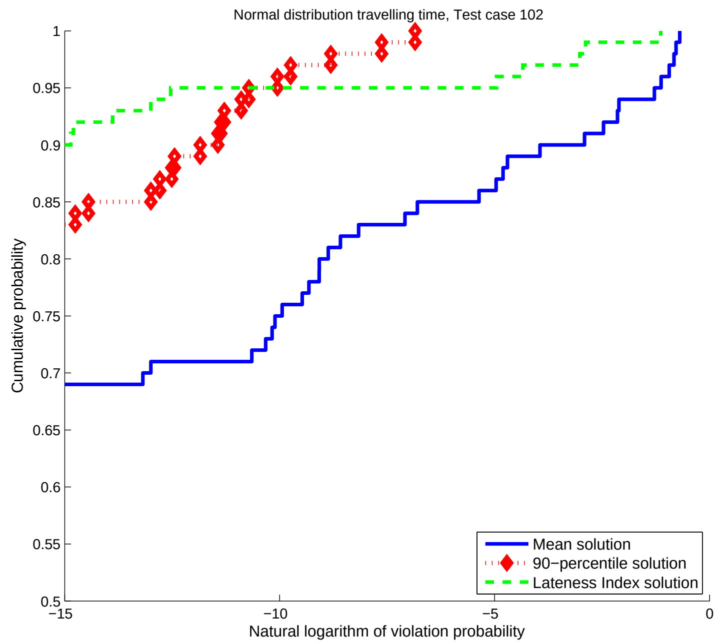

Figure 4.3: Empirical cumulative distribution of the natural logarithm of violation probability for each customer with normal distribution, test case 102

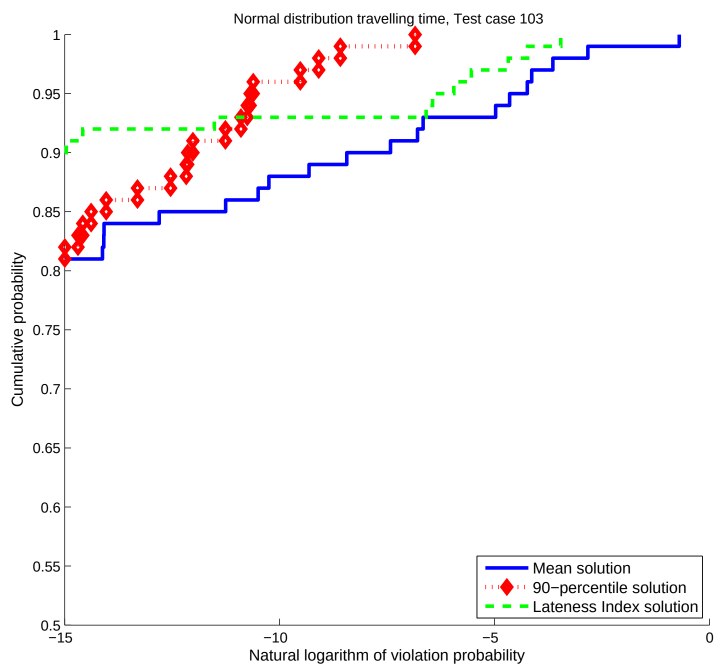

Figure 4.4: Empirical cumulative distribution of the natural logarithm of violation probability for each customer with normal distribution, test case 103

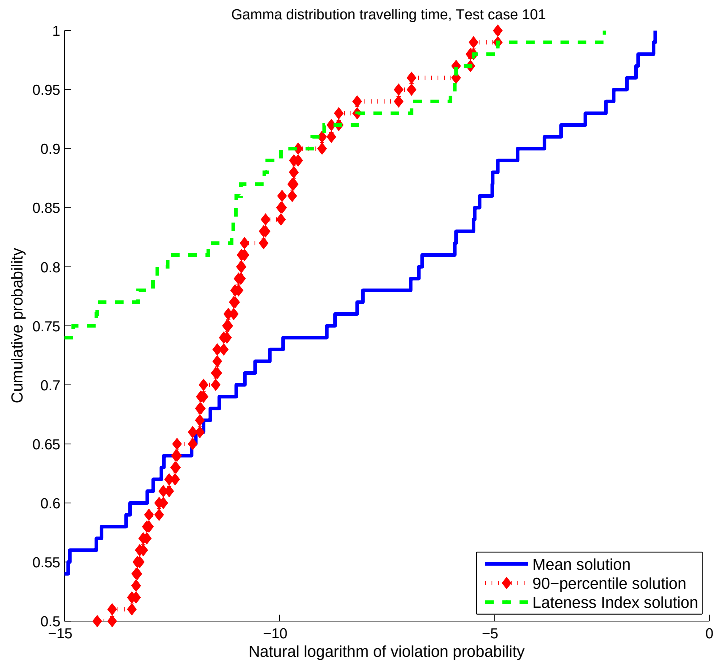

Figure 4.5: Empirical cumulative distribution of the natural logarithm of violation probability for each customer with gamma distribution, test case 101

- Each travelling time $\tilde{t}_{ij}$ follows a different type of distribution. In this case, it is very hard to think of a naive algorithm which can work well in all cases due to the complexity of the sum of probability distributions. However, if we can factor each $\tilde{t}_{ij}$ into an affine combination of primitive random variable as in (4.1.1), we can use the lateness index algorithm without any problem.

- Each travelling time $\tilde{t}_{ij}$ follows an unknown distribution. In this case, we can use the lateness index algorithm along with Theorem 2.

Furthermore, in the chance constrained formulation, it is tricky to define a good value for the percentile. A too high number may make the problem become infeasible since there is no possible route which can satisfy the passenger's time windows to that probability level. On the other hand, the lateness index does not require the users to input the probability parameter, which makes the lateness index algorithm a more applicable algorithm when we need to deal with the complex real life scenarios.

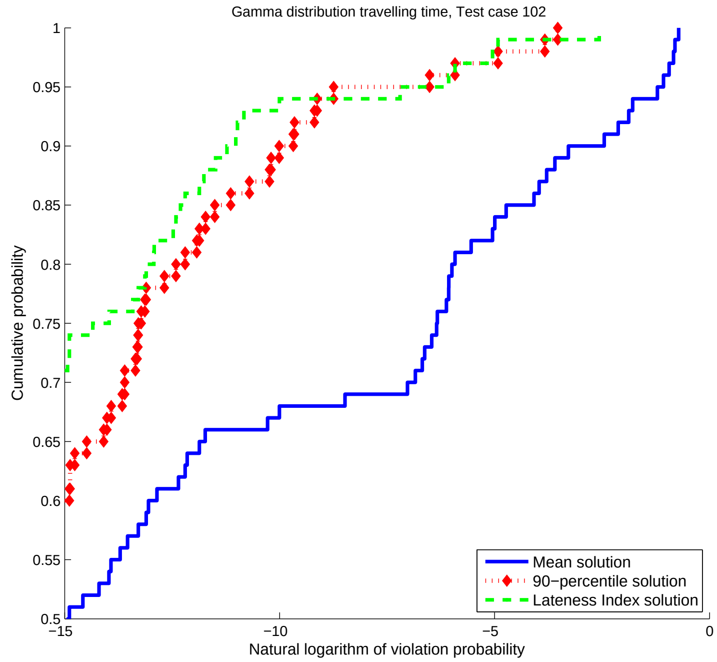

Figure 4.6: Empirical cumulative distribution of the natural logarithm of violation probability for each customer with gamma distribution, test case 102

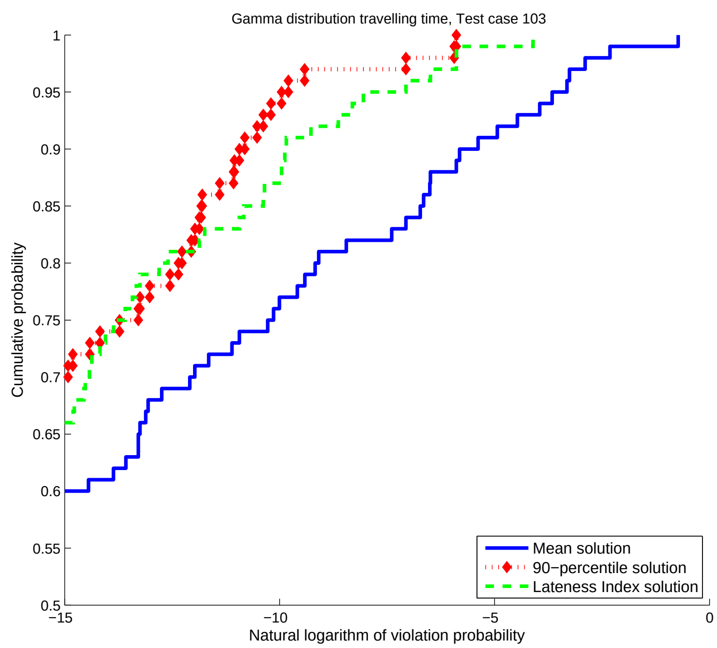

Figure 4.7: Empirical cumulative distribution of the natural logarithm of violation probability for each customer with normal distribution, test case 103

4.4 Conclusions

In this chapter, we have studied the last mile problem under uncertain travelling time. We propose, based on the satisficing measure, the lateness index with the delivery time of the passenger becomes the target. We give the analytical solution of the lateness index under several important distributions such as the normal distribution and the gamma distribution. We also consider the situation where the travelling time can admit an ambiguous distribution. The lateness index can be easily integrated in the tabu search routine as one of the hierarchical cost. We test the lateness index algorithm with Mean and 90th-percentile algorithm where the travelling time follows a normal and gamma distribution. The experimental result demonstrates that the lateness index is very promising in solving the problem under uncertain travelling time. The lateness index is also practical for real life implementation, where there are more complications over the travelling time.

5 Conclusions

In this thesis, we introduce a new instance of the vehicle routing problem: the last mile problem. Numerous contributions are made to the routing and planning of the last mile system. First, we consider the problem for a single period setup. Using an existing limited vehicle tabu search algorithm for the vehicle routing problem with time windows, we propose a new algorithm for the basic last mile problem. We then extend the algorithm so as to handle the first mile customers as well. The algorithm is also implemented using parallel computing under the OpenMP framework. We complete the single period problem by studying a heterogeneous fleet algorithm. The heterogeneous fleet is handled by using a heterogeneous tabu search routine and a preprocessing, postprocessing procedure.

In addition, we also study the multi-period problem, which is more relevant to the real life implementation of the last mile mobility system. Besides relaxing the scheduling for the vehicles, we use the heterogeneous fleet algorithm with a rolling horizon policy to solve the multi-period problem. Using real life public transport data in Singapore, we demonstrate the usefulness of our algorithm in assisting the service provider in making both strategic and operational level decisions. In strategic level, the service provider needs to determine a good fleet composition to run the service, while in the operational level, the service provider has to give reasonable routing plan for daily business. Good strategic and operational decisions ensure the profitability of the last mile mobility system, which is of critical factor to involve companies in providing the service. Our study also suggests that the service provider should seek flexibility in the system by involving external fleet such as school buses or taxi in order to handle fluctuating demand and reduce initial cost.

Finally, we introduce a tabu search heuristics for the last mile problem under uncertain travelling time. After characterizing the travelling time as an affine function of random variables, we use the lateness index to evaluate the possibility that a solution meet the customers' time windows. The lateness index can be incorporated as one of the cost in the tabu search heuristics. The experimental results show that the lateness index method gives better protection under uncertainty than the mean method. The lateness index approach is also practical for real life implementation since it can solve instances where the travelling time distribution is ambiguous, or where each travelling time follows a different type of distribution.

Although the research has reached its aims, there are some unavoidable limitations. Firstly, the fleet composition for the multi period last mile problem is currently implemented as service provider's inputs. Ideally, an optimization routine, possibly by searching algorithms, can be implemented to suggest the best fleet compositions to the service provider. Secondly, the current implementation of the satisficing measure approach cannot deal with the early time windows of the customers. Finally, because of the time limit, the research cannot test and compare different satisficing measures. A thorough comparison of the solutions under other measures such as the Conditional Value-at-Risk based, or the Bernstein based satisficing measures may uncover subtle criteria to choose the most appropriate measure for real life implementation.

5.1 Areas for future studies

5.1.1 The multi-period problem with uncertain travelling time

A natural extension to this thesis is to use the lateness index algorithm developed in chapter 4 for the multi-period problem in chapter 3. This would constitute an ideal decision support platform for the service provider. The integration is straightforward, however, due to many complicating factors such as the multi-period settings, heterogeneous fleet and uncertain travelling time, analyzing the solution requires a comprehensive framework which is out of the scope of this thesis.

5.1.2 Dynamic last mile problem

The multi-period problem is currently solved by dividing the demand into separate periods using the rolling horizon policy. A better solution might involve dynamic vehicle routing techniques where the demand is revealed stochastically: given the service region, we assume that the passenger will show up and demand a service at a random position following a certain distribution. The routing and scheduling of the vehicles at a period will have to take into account the stochastic demand in the future so as to better utilize the vehicles.

5.1.3 VRP with uncertain travelling time under stricter time windows

The last mile problem under uncertainty considers the time windows to be soft, furthermore, the time windows contain only the late delivery time. A more general case will impose a time windows with early time: the delivery has to be made after a certain time. This case arises more frequently under the logistics - supply chain setup, and it corresponds to a more general case of vehicle routing problem with time windows. The extension of the lateness index to this more general case is an interesting problem for further studies.

References

Ando, N. and Taniguchi, E. (2006). Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, **6**, 293–311.

Barr, R., Golden, B., Kelly, J., Resende, M. and Stewart, W. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, **1**, 9–32.

Berbeglia, G., Cordeau, J.F. and Laporte, G. (2010a). Dynamic pickup and delivery problems. *European Journal of Operational Research*, **202**, 8–15.

Berbeglia, G., Cordeau, J.F. and Laporte, G. (2010b). A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing*.

Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers and Operations Research*, **38**, 140–151.

Braysy, O. and Gendreau, M. (2005). Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, **39**, 119–139.

Brown, D.B. and Sim, M. (2009). Satisficing measures for analysis of risky positions. *Management Science*, **55**, 71–84.

Chang, C.T. (2000). An efficient linearization approach for mixed-integer problems. *European Journal of Operational Research*, **123**, 652 – 659.

Chen, Z.L. and Xu, H. (2006). Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, **40**, 74–88.

Chiang, W.C. and Russell, R.A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, **63**, 3–27.

Choi, E. and Tcha, D.W. (2007). A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, **34**, 2080–2095.

Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle dispatching problem. *Operational Research Quarterly*, **20**, 309–318.

Christofides, N., Mingozzi, A., Toth, P. and Sandi, C. (1979). *Combinatorial optimization*. John Wiley, Chichester.

Clarke, G. and Wright, J.W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**, 568–581.

Cordeau, J.F. and Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, **153**, 29–46.

Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y. and Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 512–522.

Cordeau, J.F., Laporte, G., Potvin, J.Y. and Savelsbergh, M.W. (2007). *Handbook in Operations Research and Management Science*, vol. 14, chap. Transportation on Demand. Elsevier.

Desrochers, M., Desrosiers, J. and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, **40**, 342–354.

Diaz, B.D. (2012). The VRP website, available at http://neo.lcc.uma.es/radi-aeb/webvrp/.

Gendreau, M., Hertz, A. and Laporte, G. (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, **40**, 1086–1094.

Gendreau, M., Laporte, G. and Seguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, **88**, 3–12.

Gendreau, M., Laporte, G., Musaraganyi, C. and Taillard, E.D. (1999). A tabu search heuristic for the heterogenous fleet vehicle routing problem. *Computers and Operations Research*, **26**, 1153–1173.

Glover, F. and Laguna, M. (1997). *Tabu search*. Kluwer.

Kenyon, A.S. and Morton, D.P. (2003). Stochastic vehicle routing with random travel times. *Transportation Science*, **37**, 69–82.

Kumar, V., Grama, A., Gupta, A. and Karpis, G. (1994). *Introduction to Parallel Computing: Design and Analysis of Parallel Algorithms*.

Laporte, G., Louveaux, F. and Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science*, **26**, 161–170.

Lau, H.C. and Liang, Z. (2001). Pickup and delivery with time windows: Algorithms and test case generation. *International Journal on Artificial Intelligence tools*, **1**, 455–472.

Lau, H.C., Sim, M. and Teo, K.M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operations Research*, **148**, 559–569.

Li, F., Golden, B. and Wasil, E. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, **34**, 2734–2742.

Li, H. and Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows. In *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '01, 160–, IEEE Computer Society, Washington, DC, USA.

Li, X., Tian, P. and Leung, S.C. (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, **125**, 137 – 145.

Mitrovic-Minic, S. and Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, **38**, 635–655.

Nanry, W.P. and Barnes, W.J. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, **34**, 107–121.

Paraskevopoulos, D.C., Repoussis, P.P., Tarantilis, C.D., Ioannou, G. and Prastacos, G.P. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, **14**, 425–455.

Potvin, J.Y. and Rousseau, J.M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, **66**, 331340.

Renaud, J. and Boctor, F.F. (2002). A sweep-based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, **140**, 618–628.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, **40**, 455–472.

Russell, R.A. (1995). Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science*, **29**, 156166.

Silberholz, J. and Golden, B. (2010). Comparison of metaheuristics. In M. Gendreau, J.Y. Potvin, F.S. Hillier and C.C. Price, eds., *Handbook of Metaheuristics*, vol. 146

of *International Series in Operations Research and Management Science*, 625–640, Springer US.

Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, **35**, 254–265.

Taillard, E. (1993). Parallel iterative search methods for vehicle routing problems. *Networks 23*, 661–673.

Taillard, E. (1996). A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO Operations Research*, **33**, 1–14.

Tarantilis, C.D., Kiranoudis, C.T. and Vassiliadis, V.S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, **152**, 148–158.

Teodorovic, D. and Radivojevic, G. (2000). A fuzzy logic approach to dynamic dial-a-ride problem. *Fuzzy Sets and Systems*, **116**, 23–33.

A Mathematical formulation for VRPPDTW

Let n be the total number of customers. Let P and D denote the pickup and delivery nodes: the i-th customer is represented by a pickup node $i \in P$ and a delivery node $(i+n) \in D$. Each node i has a time window $[l_i, u_i]$ and a demand q_i, with $q_i > 0$ for $i \in P$, and $q_i = -q_{i+n}$. Let $N = P \cup D \cup \{0, 2n+1\}$ where $\{0, 2n+1\}$ denote the starting and the ending depot of the vehicles. The service time at node i is s_i, and the travel time between node i and node j is t_{ij}.

Let v be the set of the available vehicles, every vehicle $k \in K$ has a finite capacity Q^v and is available during a period $[l_v, u_v]$. Q_i^v denote the current number of customers in vehicle v after the vehicle v visits node i.

x_{ij}^v is the decision variable, it equals 1 if the vehicle v travels from node i to node j, and 0 otherwise. y_i is the binary variable, $y_i = 0$ if the customer i is served, $y_i = 1$ otherwise. S_i^v denotes the time the vehicle v reaches node i.

Several costs can be computed:

The travel time is: $\sum_{v \in V} \sum_{i \in N} \sum_{j \in N} t_{ij} x_{ij}^v$.

If node i is assigned to vehicle v, the time windows violation for each node i, whose time windows is $[l_i, u_i]$, is computed as: $C_i^{TW} = \max\{0, l_i - S_i^v, S_i^v - u_i\}$

If node i is assigned to vehicle v, the capacity violation at i is computed as: $C_i^{Cap} = \max\{0, Q_i^v - Q^v\}$

The total time windows and capacity violation for the solution will be: $C^{TW} = \sum_{i \in N} C_i^{TW}$, $C^{Cap} = \sum_{i \in N} C_i^{Cap}$

The mathematical model is:

$$\text{minimize} \sum_{i \in P} y_i$$

subject to:

$$\sum_{v \in V} \sum_{j \in N} x_{ij}^v + y_i = 1 \qquad \forall i \in P \qquad (A.1)$$

$$\sum_{j \in N} x_{ij}^v - \sum_{j \in N} x_{(i+n)j}^v = 0 \qquad \forall i \in P, \forall v \in V \qquad (A.2)$$

$$\sum_{j \in N} x_{0j}^v = 1 \qquad \forall v \in V \qquad (A.3)$$

$$\sum_{j \in N} x_{j(2n+1)}^v = 1 \qquad \forall v \in V \qquad (A.4)$$

$$\sum_{j \in N} x_{ji}^v - \sum_{j \in N} x_{ij}^v = 0 \qquad \forall i \in P, \forall v \in V \qquad (A.5)$$

$$x_{ij}^v (S_i^v + s_i + t_{ij}) \leq S_j^v \qquad \forall i, j \in N, i, j \text{ are assigned to } v \qquad (A.6)$$

$$l_v \leq S_0^v \leq u_v \qquad \forall v \in V \qquad (A.7)$$

$$l_v \leq S_{2n+1}^v \leq u_v \qquad \forall v \in V \qquad (A.8)$$

$$l_i \leq S_i^v \leq u_i \qquad \forall i \in P \cup D, i \text{ is assigned to } v \qquad (A.9)$$

$$S_i^v + t_{i(i+n)} \leq S_{(i+n)}^v \qquad \forall i \in P, i \text{ is assigned to } v \qquad (A.10)$$

$$0 \leq Q_i^v \leq Q^v \qquad \forall i \in P \cup D, \text{i is assigned to v} \qquad (A.11)$$

$$Q_j^v = (Q_i^v + q_j) x_{ij}^v \qquad \forall v \in V; i, j \text{ are assigned to } v \qquad (A.12)$$

$$x_{ij}^v \in \{0, 1\} \qquad \forall i \in N, \forall j \in N, \forall v \in V \qquad (A.13)$$

$$y_i \in \{0, 1\} \qquad \forall i \in P \qquad (A.14)$$

The objective function is to minimize the number of customers which are not served. Constraint (A.1) ensures that the customer is either accepted or rejected. Constraint (A.2) ensures that the pickup and delivery is served by the same vehicle. Constraint (A.3) and (A.4) ensure that the route for each vehicle starts and ends at the depot. Constraint (A.5) and (A.6) ensure the continuity of the route. Constraint

(A.7) and (A.8) ensure the vehicle is active within its own time windows. Constraint (A.9) ensures that the pickup and delivery is done in the time windows. Constraint (A.10) ensures that the pickup node is visited before the delivery node. Constraint (A.11) ensures the capacity is valid for each vehicle. Constraint (A.12) ensures the capacity continuity of the route.

B Standard Tabu search procedure

```
for every route A do
   for every route B do
      for every customer in A do
         for every customer in B do
            for every of the 4 moves: T, S, E do
               Check the feasibility of the move
               if feasible then
                  Find the penalty
                  if this move is the best up to now then
                     Choose this move as the next move
                  end if
               end if
            end for
         end for
      end for
   end for
   for every customer in A do
      Check the flip feasibility
      if feasible then
         Find penalty
         if this move is the best up to now then
            Choose the flip move as the next move
         end if
      end if
      for every customer in the holding list do
         for every of the 3 moves: IH, RH, SH do
            Check the move feasibility
            if feasible then
               Find penalty
               if this move is the best up to now then
                  Choose this move as the next move
               end if
            end if
         end for
      end for
   end for
end for
```

C Tabu search procedure for heterogeneous fleet

```
for every route A do
  for every route B do
    for every customer in A do
      for every customer in B do
        for every of the 4 moves: T, S, E do
          Check the feasibility of the move
          if feasible then
            Find the two best vehicles to serve two routes A and B
            Find the penalty with respect to the two best vehicles found.
            if this move is the best up to now then
              Choose this move as the next move
            end if
          end if
        end for
      end for
    end for
  end for
  for every customer in A do
    Check the flip feasibility
    if feasible then
      Find the best vehicle to serve the route
      Find penalty with respect to the best vehicle found
      if this move is the best up to now then
        Choose the flip move as the next move
      end if
    end if
    for every customer in the holding list do
      for every of the 3 moves: IH, RH, SH do
        Check the move feasibility
        if feasible then
          Find the best vehicle to serve the route
          Find penalty with respect to the best vehicle found
          if this move is the best up to now then
            Choose this move as the next move
          end if
        end if
      end for
    end for
  end for
end for
```

D Test result

D.1 Test results of basic LMP and m-VRPTW with relaxed LMP test cases

	mVRPTW			LMP		
No.	Vehicle	Distance	Computation Time	Vehicle	Distance	Computation Time
1	12	1708.94	10.76	12	1694.36	17.72
2	12	1680.72	13.46	12	1698.63	17.03
3	12	1637.31	9.84	12	1613.16	16.08
4	12	1664.17	9.64	12	1694.29	16.91
5	12	1709.08	8.72	12	1679.09	16.62
6	12	1638.69	9.53	12	1646.12	22.45
7	12	1691.07	11.04	12	1700.78	17.18
8	12	1655.78	10.83	12	1700.28	15.96
9	12	1666.56	9.38	12	1679.44	18.56
10	12	1585.36	10.02	12	1621.09	16.55
11	12	1598.83	11.12	12	1637.42	16.61
12	12	1644.58	9.45	12	1650.37	19.56
13	12	1609.53	11.08	12	1578.59	19.78
14	12	1677.06	10.81	12	1698.83	18.36
15	12	1711.53	8.42	12	1726.19	16.58
16	12	1698.77	12.9	12	1758.41	15.16
17	12	1618.91	10.53	12	1630.41	19.58
18	12	1718.36	9.35	12	1716.48	15.16
19	12	1662.09	10.11	12	1624.97	24.99
20	12	1697.01	8.05	12	1672.02	21.12
21	12	1684.62	9.8	12	1704.18	16.08
22	12	1692.45	9.77	12	1711.05	16.57
23	12	1775.46	8.19	12	1761.88	16.32
24	12	1643.95	11.37	12	1654.79	20.16
25	12	1647.9	10.3	12	1671.56	15.66

	mVRPTW			LMP		
No.	Vehicle	Distance	Computation Time	Vehicle	Distance	Computation Time
26	12	1701.06	10.47	12	1668.63	18.19
27	12	1691.1	8.89	12	1699	16.13
28	12	1733.5	9.88	12	1746.72	15.65
29	12	1630.86	8.8	12	1655.2	18.38
30	12	1673.73	9.63	12	1667.49	16.83
31	12	1702.72	8.78	12	1735.23	14.21
32	12	1624.65	8.02	12	1620.02	15.12
33	12	1655.65	12	12	1629.05	22.1
34	12	1668.58	11.03	12	1680.67	23.03
35	12	1652.12	10.39	12	1667.37	17.28
36	12	1739.04	9.86	12	1753.32	15.24
37	12	1699.02	9.02	12	1701.99	15.79
38	12	1641.37	8.77	12	1634.18	22.65
39	12	1645.76	8.97	12	1636.56	15.68
40	12	1653.93	9.55	12	1656.22	17.3
41	12	1623.43	10.83	12	1618.5	21.12
42	12	1703.75	10.3	12	1721.11	14.91
43	12	1691.64	9.38	12	1684.55	18.38
44	12	1631.91	12.03	12	1641.55	16.38
45	12	1589.78	11	12	1632.74	17.36
46	12	1640.48	10.09	12	1649.37	16.65
47	12	1775.1	8.25	12	1764.95	17.19
48	12	1690.75	7.75	12	1666.59	16.4
49	12	1729.32	11.28	12	1720.11	15.13
50	12	1701.92	9.61	12	1703.32	17.96
Average	12	1672.198	9.981	12	1677.577	17.6362

D.2 Test results of basic LMP, LMP+FMP with waiting, and LMP+FMP with waiting under OpenMP with LMP test cases

	LMP			LMP+FMP			OMP dynamic		
No.	Vehicle	Distance	Time	Vehicle	Distance	Time	Vehicle	Distance	Time
1	12	1789.45	18.01	12	1789.45	49.99	12	1791.56	17.68
2	12	1797.26	18.23	12	1797.26	49.56	12	1794.26	21.64
3	12	1724.32	12.98	12	1724.32	34.81	12	1677.23	20.15
4	12	1750.46	11.74	12	1750.46	30.83	12	1741.77	23.75
5	12	1813.34	17.31	12	1813.34	46.76	12	1810.25	21.86
6	12	1732.33	12.76	12	1732.33	32.83	12	1732.33	17.76
7	12	1808.42	15.91	12	1808.42	42.41	12	1808.42	22.58
8	12	1787.84	13.42	12	1787.84	36.59	12	1787.84	22.45
9	12	1778.28	14.79	12	1778.28	38.25	12	1794.89	17.48
10	12	1677.71	13.41	12	1677.71	35.43	12	1677.71	18.97
11	12	1685.08	16.35	12	1685.08	43.08	12	1685.08	22.71
12	12	1786.04	11.31	12	1786.04	30.03	12	1786.04	16.73
13	12	1717.98	15.99	12	1717.98	42.75	12	1729.02	19.94
14	12	1791.97	16.77	12	1791.97	44.57	12	1756.39	23.51
15	12	1830.89	10.61	12	1830.89	28.55	12	1830.89	18.29
16	12	1800.14	16.59	12	1800.14	44.43	12	1833.24	24.02
17	12	1742.33	12.77	12	1742.33	33.21	12	1742.33	20.4
18	12	1832.99	14.63	12	1832.99	39.09	12	1818.65	23.27
19	12	1734.03	13.54	12	1734.03	34.39	12	1734.03	18.37
20	12	1748.41	13.75	12	1748.41	36.54	12	1749.09	20.2
21	12	1810.01	13.38	12	1810.01	35.59	12	1849.95	21.37
22	12	1795.05	12.77	12	1795.05	34.16	12	1795.05	18.12
23	12	1913.03	13.9	12	1913.03	35.24	12	1913.03	18.74
24	12	1727.57	13.83	12	1727.57	36.71	12	1730.12	23.31
25	12	1748.11	15.53	12	1748.11	40.81	12	1748.11	21.67

	LMP			LMP+FMP			OMP dynamic		
No.	Vehicle	Distance	Time	Vehicle	Distance	Time	Vehicle	Distance	Time
26	12	1804.52	14.19	12	1804.52	37.6	12	1804.52	19.99
27	12	1824.98	12.84	12	1824.98	32.89	12	1824.98	17.82
28	12	1828.41	18.36	12	1828.41	49.8	12	1828.41	26
29	12	1709.99	13.95	12	1709.99	36.87	12	1717.71	21.16
30	12	1777.63	15.06	12	1777.63	38.81	12	1795.56	20.61
31	12	1769.94	15	12	1769.94	37.61	12	1769.94	20.12
32	12	1718.11	13.25	12	1718.11	33.66	12	1701.14	23.9
33	12	1667.84	16.19	12	1667.84	41.07	12	1685.67	20.05
34	12	1783.2	16.06	12	1783.2	41.76	12	1774.61	22.73
35	12	1779.47	13.64	12	1779.47	35.5	12	1779.47	21.45
36	12	1795.68	15.46	12	1795.68	39.37	12	1789.91	19.36
37	12	1734.47	13.65	12	1734.47	34.99	12	1734.47	21.65
38	12	1725.86	15.05	12	1725.86	37.66	12	1725.86	20.18
39	12	1769.71	14.08	12	1769.71	35.12	12	1769.71	18.94
40	12	1699.1	12.11	12	1699.1	30.54	12	1689.49	18.24
41	12	1720.71	13.9	12	1720.71	34.96	12	1720.71	18.67
42	12	1800.87	12.79	12	1800.87	33.67	12	1800.87	19.77
43	12	1812.12	13.47	12	1812.12	35.25	12	1812.12	21.24
44	12	1704.66	15.38	12	1704.66	40.27	12	1704.3	21.99
45	12	1736.33	14.16	12	1736.33	35.54	12	1736.33	18.87
46	12	1743.86	16.1	12	1743.86	42.22	12	1740.93	24.86
47	12	1826.01	14.67	12	1826.01	38.28	12	1801.7	16.4
48	12	1750.23	15.71	12	1750.23	39.34	12	1750.23	20.84
49	12	1801.44	12.45	12	1801.44	31.07	12	1801.44	19.25
50	12	1727.83	15.41	12	1727.83	38.53	12	1727.83	20.17
Avg	12	1766.72	14.46	12	1766.72	37.78	12	1766.104	20.58

D.3 Test results of LMP+FMP with waiting under OpenMP with LMP+FMP test cases

	LMP Only			FMP Only			Combined LMP+FMP		
No.	Vehicle	Distance	Time	Vehicle	Distance	Time	Vehicle	Distance	Time
1	9	1511.95	9.93	4	719.81	0.36	10	1832.48	27.82
2	9	1696.89	10.07	4	619.11	0.33	10	1696.66	30.19
3	9	1648.41	6.95	4	731.07	0.34	9	1699.44	27.7
4	9	1517.67	8.05	4	637.6	0.32	10	1662.16	23.11
5	9	1517.85	9.49	4	650.85	0.25	11	1819.28	27.05
6	9	1532.93	8.54	4	712.99	0.39	10	1746.08	21.11
7	9	1690.11	8.93	4	671.47	0.29	11	1810.17	34.3
8	9	1512.33	13.67	4	628	0.47	10	1666.48	31.42
9	9	1505.91	9.09	4	712.36	0.35	11	1744.16	24.59
10	9	1515.36	8.45	4	840.27	0.51	10	1701.83	20.57
11	9	1499.79	8.75	4	608.21	0.41	10	1598.66	21.95
12	9	1571.59	7.09	4	653.74	0.43	10	1701.83	20.55
13	9	1551.46	12.84	4	687.55	0.29	10	1791.89	22.28
14	9	1408.9	9.48	4	724.26	0.31	10	1618.75	30
15	9	1631.72	9.47	4	628.53	0.33	10	1727.7	22.64
16	9	1529.71	7.84	4	609.62	0.35	10	1719.06	28.87
17	9	1460.96	10.11	5	837.51	0.32	10	1708.16	25.57
18	9	1493.15	8.94	4	736.12	0.52	10	1670.41	28.08
19	9	1556.98	8.73	4	587.46	0.48	10	1740.79	29.03
20	9	1488.64	10.66	4	609.19	0.32	11	1783.61	23.98
21	9	1549.71	10.66	4	673.31	0.34	10	1688.2	27.62
22	9	1429.41	9.13	4	742.41	0.48	10	1696.15	25.11
23	9	1475.5	7.7	4	571.42	0.39	11	1746.28	28.6
24	9	1577.33	10.8	4	686.31	0.32	9	1780.79	21.65
25	9	1539.88	8.09	4	712.6	0.47	10	1723.95	22.7

	LMP Only			FMP Only			Combined LMP+FMP		
No.	Vehicle	Distance	Time	Vehicle	Distance	Time	Vehicle	Distance	Time
26	9	1418.29	9.32	4	549.02	0.41	9	1612.28	23.7
27	9	1553.02	9.88	4	609.68	0.4	10	1660.67	27.75
28	9	1534.98	9.78	4	590.41	0.43	10	1707.2	25.61
29	9	1538.36	10.83	4	684.45	0.45	10	1641.09	28.3
30	9	1523.41	9.18	4	627.81	0.36	9	1853.25	23.95
31	9	1563.06	7.93	4	730.84	0.39	10	1736.55	24.93
32	9	1480.74	8.46	4	705.05	0.4	10	1746.44	23.75
33	9	1612.05	7.44	4	713.01	0.3	11	1754.42	22.5
34	9	1528.11	7.76	4	661.76	0.5	10	1664.33	24.9
35	9	1518.8	7.74	4	647.7	0.4	10	1680.04	24.94
36	9	1527.51	11.03	4	647.16	0.3	10	1705.59	30.99
37	9	1544.96	9.63	4	698.76	0.79	11	1795.33	26.45
38	9	1536.97	8.87	4	634.17	0.32	10	1708.17	23.38
39	9	1468.64	10.43	4	683.31	0.35	10	1687.93	24.73
40	9	1386.58	10.42	4	637.33	0.29	10	1633.28	28
41	9	1539.18	9.54	4	535.72	0.5	11	1679.58	32.09
42	9	1573.97	11	3	516.06	0.74	10	1679.98	22.26
43	9	1548.94	8.26	4	645.58	0.29	10	1677.26	24.19
44	9	1466.61	8.57	4	614.71	0.42	10	1643.47	30.05
45	9	1430.34	9.97	4	604.94	0.35	10	1669.37	26.03
46	9	1649.86	8.15	4	707.62	0.57	10	1711.71	28.34
47	9	1607.27	9.18	4	716.89	0.46	10	1813.86	22.11
48	9	1594.46	10.44	4	587.61	0.3	10	1754.09	23.72
49	9	1425.53	9.01	5	797.71	0.35	9	1657.93	25.86
50	9	1477.65	8.47	4	730.81	0.29	10	1680.75	27.88
Avg	9	1529.27	9.30	4.02	665.40	0.39	10.06	1712.59	25.86

D.4 Target, Mean and Standard deviation values of each customer for test case 101 with normal distribution

	Mean solution			90th percentile solution			Lateness Index solution		
Customer	Target	Mean	Std	Target	Mean	Std	Target	Mean	Std
1	117.000	46.660	6.557	117.000	50.830	6.557	117.000	50.360	6.000
2	75.000	29.180	5.000	75.000	29.180	5.000	75.000	29.180	4.690
3	125.000	59.310	7.000	125.000	78.290	8.062	125.000	63.490	6.856
4	55.000	21.930	4.000	55.000	21.930	4.000	55.000	21.930	4.000
5	125.000	60.020	6.856	125.000	49.770	6.708	125.000	50.540	6.164
6	42.000	16.470	3.742	42.000	16.120	4.000	42.000	16.420	3.162
7	95.000	54.190	5.916	95.000	46.900	6.403	95.000	38.380	4.796
8	125.000	79.880	7.746	125.000	58.940	7.071	125.000	50.420	5.657
9	145.000	74.430	7.810	145.000	103.860	8.775	145.000	65.860	6.928
10	55.000	21.210	4.472	55.000	21.190	4.000	55.000	21.500	4.243
11	135.000	88.720	8.602	135.000	67.500	6.633	135.000	63.220	6.403
12	140.000	76.990	7.416	140.000	79.100	7.937	140.000	76.250	7.000
13	105.000	44.140	6.000	105.000	70.310	7.348	105.000	68.380	6.928
14	155.000	71.800	7.810	155.000	76.040	6.928	155.000	83.640	7.810
15	130.000	61.180	6.856	130.000	98.980	8.660	130.000	62.600	6.557
16	110.000	45.070	6.000	110.000	49.960	6.164	110.000	86.580	7.746
17	120.000	88.010	8.426	120.000	90.280	8.185	120.000	52.280	6.164
18	150.000	64.350	7.416	150.000	69.240	7.550	150.000	67.050	6.245
19	110.000	101.300	8.888	110.000	48.660	5.745	110.000	47.280	5.916
20	112.000	99.500	8.888	112.000	56.720	6.245	112.000	52.440	6.000
21	20.000	7.070	2.000	20.000	13.540	3.000	20.000	7.070	2.000
22	102.000	41.310	6.000	102.000	75.740	7.071	102.000	59.580	6.245
23	122.000	60.880	6.325	122.000	85.190	8.185	122.000	59.500	7.071
24	87.000	77.740	7.874	87.000	34.790	5.000	87.000	35.100	5.196
25	100.000	80.820	7.810	100.000	73.370	7.616	100.000	40.990	5.099
26	67.000	30.350	4.583	67.000	26.990	4.899	67.000	26.840	4.472
27	47.000	25.000	4.000	47.000	18.380	4.000	47.000	20.280	3.606
28	97.000	47.240	5.477	97.000	69.120	7.348	97.000	38.960	5.099
29	157.000	84.350	8.426	157.000	103.080	8.775	157.000	87.050	7.416
30	127.000	58.430	7.000	127.000	62.600	7.000	127.000	62.130	6.481
31	122.000	58.640	6.245	122.000	87.430	8.246	122.000	57.260	7.000
32	80.000	40.240	5.099	80.000	31.960	4.690	80.000	31.960	4.690
33	32.000	12.730	2.828	32.000	19.200	3.606	32.000	12.080	3.000
34	132.000	54.290	6.708	132.000	59.180	6.856	132.000	77.360	7.141
35	137.000	63.820	7.280	137.000	77.490	7.810	137.000	57.990	5.831
36	87.000	64.320	6.557	87.000	38.040	5.745	87.000	34.890	5.000
37	47.000	18.360	4.000	47.000	18.360	4.000	47.000	18.360	3.606
38	92.000	68.300	7.000	92.000	36.410	5.000	92.000	39.730	5.385
39	122.000	62.710	7.280	122.000	60.060	6.481	122.000	75.260	6.856
40	35.000	21.700	3.606	35.000	13.420	3.000	35.000	13.420	3.000
41	70.000	36.120	4.690	70.000	27.840	4.243	70.000	27.840	4.243
42	127.000	51.660	6.856	127.000	55.830	6.856	127.000	55.360	6.325
43	90.000	84.690	8.185	90.000	37.770	5.292	90.000	37.720	5.385
44	65.000	25.960	5.000	65.000	42.840	5.745	65.000	26.780	4.123
45	105.000	45.430	6.325	105.000	51.990	6.403	105.000	82.850	8.367
46	40.000	16.750	3.464	40.000	15.030	3.000	40.000	15.030	3.000
47	35.000	13.150	3.000	35.000	22.850	3.873	35.000	13.780	2.828
48	117.000	71.880	7.550	117.000	85.550	8.062	117.000	49.930	5.477
49	12.000	4.120	2.000	12.000	4.410	1.414	12.000	4.410	1.414
50	112.000	47.760	6.403	112.000	46.420	6.403	112.000	59.600	6.403

	Mean solution			90th percentile solution			Lateness Index solution		
Customer	Target	Mean	Std	Target	Mean	Std	Target	Mean	Std
51	120.000	68.390	7.141	120.000	91.770	8.426	120.000	55.390	6.245
52	7.000	2.000	1.000	7.000	2.000	1.000	7.000	2.000	1.000
53	110.000	94.090	8.660	110.000	84.200	7.937	110.000	54.490	6.245
54	87.000	36.470	5.477	87.000	46.580	6.083	87.000	47.270	6.325
55	132.000	54.820	6.928	132.000	58.990	6.928	132.000	58.520	6.403
56	30.000	11.750	2.828	30.000	11.750	2.828	30.000	11.700	3.000
57	130.000	83.700	7.681	130.000	72.390	7.681	130.000	63.870	6.403
58	102.000	40.920	5.831	102.000	70.270	7.416	102.000	82.860	7.483
59	95.000	77.850	7.280	95.000	61.840	7.071	95.000	46.240	5.477
60	45.000	17.200	4.000	45.000	18.460	3.464	45.000	18.410	3.606
61	12.000	4.470	2.000	12.000	4.470	2.000	12.000	4.470	2.000
62	87.000	34.960	5.831	87.000	51.840	6.481	87.000	38.660	5.196
63	120.000	88.800	7.937	120.000	86.380	8.185	120.000	68.970	6.708
64	87.000	38.050	5.657	87.000	55.800	6.708	87.000	38.050	5.657
65	90.000	43.150	6.000	90.000	50.700	6.403	90.000	43.150	6.000
66	77.000	30.180	5.099	77.000	30.180	5.099	77.000	30.180	4.796
67	120.000	96.340	8.832	120.000	59.880	6.325	120.000	55.600	6.083
68	127.000	55.780	6.557	127.000	64.600	7.280	127.000	54.780	6.481
69	70.000	27.660	5.000	70.000	34.100	5.000	70.000	35.180	5.099
70	142.000	67.230	7.141	142.000	56.980	7.000	142.000	107.490	8.602
71	17.000	6.470	2.236	17.000	6.470	2.236	17.000	6.320	2.000
72	120.000	54.650	7.000	120.000	68.120	6.782	120.000	67.200	6.557
73	147.000	81.650	8.367	147.000	85.890	7.550	147.000	140.430	10.630
74	110.000	85.330	8.185	110.000	75.070	7.937	110.000	92.830	8.124
75	97.000	39.040	5.657	97.000	65.210	7.071	97.000	73.480	7.211
76	115.000	46.310	6.164	115.000	64.880	7.141	115.000	77.470	7.211
77	150.000	63.350	7.348	150.000	68.240	7.483	150.000	66.050	6.164
78	77.000	31.180	5.196	77.000	31.180	5.196	77.000	31.180	4.899
79	122.000	86.600	8.367	122.000	91.690	8.246	122.000	53.690	6.245
80	115.000	56.490	6.403	115.000	65.890	7.348	115.000	51.540	6.083
81	90.000	81.860	8.124	90.000	40.600	5.385	90.000	39.220	5.568
82	92.000	43.350	5.477	92.000	36.060	6.000	92.000	41.250	5.477
83	57.000	25.880	4.123	57.000	22.520	4.472	57.000	22.820	3.742
84	10.000	3.000	1.000	10.000	3.000	1.000	10.000	3.000	1.000
85	117.000	67.950	6.633	117.000	78.120	7.937	117.000	66.570	7.348
86	125.000	62.470	7.071	125.000	81.450	8.124	125.000	61.590	7.000
87	120.000	67.570	7.348	120.000	86.550	8.367	120.000	75.040	7.550
88	97.000	59.850	6.245	97.000	38.590	6.000	97.000	39.360	5.385
89	102.000	64.300	6.708	102.000	73.700	7.616	102.000	43.730	5.745
90	122.000	50.680	6.245	122.000	69.700	7.550	122.000	59.880	6.782
91	92.000	45.590	5.568	92.000	38.300	6.083	92.000	39.010	5.385
92	35.000	13.890	3.000	35.000	17.750	3.317	35.000	13.890	3.000
93	60.000	23.920	4.243	60.000	33.620	4.899	60.000	24.450	4.243
94	112.000	76.110	7.616	112.000	65.850	7.348	112.000	83.580	7.810
95	122.000	54.080	6.708	122.000	52.740	6.708	122.000	65.920	6.708
96	117.000	56.180	6.708	117.000	54.770	6.403	117.000	84.110	8.000
97	85.000	73.300	7.280	85.000	34.380	5.657	85.000	34.730	5.000
98	107.000	101.500	8.944	107.000	54.720	6.164	107.000	50.440	5.916
99	70.000	34.900	5.000	70.000	39.100	5.385	70.000	30.180	4.690
100	112.000	55.190	6.708	112.000	74.170	7.810	112.000	66.980	7.280

D.5 Target, Mean and Standard deviation values of each customer for test case 102 with normal distribution

	Mean solution			90th percentile solution			Lateness Index solution		
Customer	Target	Mean	Std	Target	Mean	Std	Target	Mean	Std
1	150.000	70.860	7.483	150.000	84.810	8.185	150.000	74.870	7.280
2	117.000	67.860	7.000	117.000	68.370	7.211	117.000	53.570	5.745
3	120.000	52.360	5.916	120.000	84.760	8.544	120.000	115.770	9.110
4	40.000	15.030	3.000	40.000	15.030	3.000	40.000	15.030	3.000
5	102.000	83.570	7.681	102.000	59.660	6.856	102.000	48.880	5.477
6	80.000	31.910	5.000	80.000	34.180	4.899	80.000	33.740	4.690
7	102.000	79.330	7.416	102.000	53.780	6.481	102.000	53.120	5.831
8	70.000	47.910	6.083	70.000	31.780	5.292	70.000	31.780	5.292
9	80.000	36.020	4.899	80.000	44.390	5.916	80.000	35.210	5.099
10	92.000	41.130	5.831	92.000	36.070	6.000	92.000	44.270	5.916
11	102.000	49.690	6.000	102.000	71.170	7.550	102.000	88.160	8.367
12	60.000	23.020	4.000	60.000	33.400	5.385	60.000	23.020	4.000
13	72.000	49.070	5.745	72.000	29.870	4.472	72.000	31.090	4.690
14	102.000	59.600	6.164	102.000	53.610	6.633	102.000	43.130	5.099
15	145.000	100.130	8.888	145.000	103.630	9.110	145.000	71.220	6.481
16	72.000	55.470	7.000	72.000	37.370	5.657	72.000	28.760	4.243
17	67.000	27.890	4.243	67.000	45.910	5.831	67.000	27.890	4.243
18	77.000	51.600	6.245	77.000	44.820	5.831	77.000	34.440	4.583
19	72.000	30.680	5.000	72.000	28.280	5.000	72.000	38.940	5.831
20	92.000	37.680	5.477	92.000	56.060	6.325	92.000	37.400	4.796
21	87.000	47.130	5.916	87.000	44.780	5.745	87.000	42.170	5.099
22	42.000	16.120	4.000	42.000	16.120	4.000	42.000	16.120	4.000
23	132.000	64.400	6.633	132.000	72.720	8.000	132.000	55.290	6.245
24	75.000	29.180	5.385	75.000	29.300	5.196	75.000	45.510	5.657
25	127.000	71.890	6.928	127.000	65.900	7.348	127.000	60.390	5.916
26	57.000	23.980	3.873	57.000	23.980	3.873	57.000	23.540	3.606
27	40.000	15.270	3.162	40.000	15.270	3.162	40.000	15.000	3.000
28	105.000	77.920	7.348	105.000	55.190	6.557	105.000	54.530	5.916
29	145.000	124.430	9.950	145.000	79.000	8.124	145.000	94.540	8.888
30	77.000	32.890	4.690	77.000	50.910	6.164	77.000	32.890	4.690
31	87.000	37.360	5.099	87.000	55.380	6.481	87.000	37.360	5.099
32	57.000	24.250	4.243	57.000	24.250	4.243	57.000	24.250	4.243
33	85.000	83.780	7.810	85.000	36.420	5.000	85.000	35.980	4.796
34	125.000	59.160	6.856	125.000	73.110	7.616	125.000	53.180	5.568
35	95.000	38.910	5.000	95.000	71.450	7.141	95.000	83.650	7.141
36	30.000	11.660	3.000	30.000	11.660	3.000	30.000	11.660	3.000
37	127.000	80.630	7.681	127.000	61.790	7.071	127.000	88.270	8.124
38	65.000	64.270	7.000	65.000	25.340	5.000	65.000	25.340	5.000
39	55.000	38.080	5.385	55.000	21.950	4.472	55.000	21.950	4.472
40	10.000	3.610	1.000	10.000	3.610	1.000	10.000	3.610	1.000
41	87.000	83.590	7.483	87.000	38.720	5.385	87.000	52.340	6.083
42	37.000	14.000	3.000	37.000	17.320	3.162	37.000	14.000	3.000
43	125.000	110.410	9.110	125.000	59.300	6.325	125.000	59.300	6.325
44	75.000	30.180	5.477	75.000	30.300	5.292	75.000	46.510	5.745
45	62.000	54.900	6.083	62.000	24.040	4.000	62.000	24.040	4.000
46	75.000	29.180	5.385	75.000	29.300	5.196	75.000	45.510	5.657
47	117.000	117.010	9.165	117.000	84.630	7.681	117.000	57.230	5.916
48	77.000	55.210	6.325	77.000	41.210	5.745	77.000	30.830	4.472
49	7.000	6.610	1.414	7.000	2.000	1.000	7.000	2.000	1.000
50	137.000	68.750	7.348	137.000	100.960	8.307	137.000	113.590	9.539

	Mean solution			90th percentile solution			Lateness Index solution		
Customer	Target	Mean	Std	Target	Mean	Std	Target	Mean	Std
51	127.000	55.130	6.325	127.000	81.100	7.810	127.000	66.520	6.782
52	122.000	91.190	8.660	122.000	94.690	8.888	122.000	62.280	6.164
53	127.000	67.190	6.856	127.000	54.370	6.083	127.000	56.560	5.657
54	132.000	80.890	8.124	132.000	84.390	8.367	132.000	89.150	8.660
55	62.000	24.040	4.000	62.000	24.230	4.123	62.000	24.040	4.000
56	92.000	39.920	5.568	92.000	53.820	6.245	92.000	39.640	4.899
57	120.000	69.860	7.071	120.000	66.370	7.141	120.000	55.570	5.831
58	115.000	104.510	8.832	115.000	50.560	7.000	115.000	55.450	6.325
59	87.000	35.440	5.385	87.000	58.300	6.403	87.000	35.160	4.690
60	75.000	29.430	5.099	75.000	36.280	5.196	75.000	29.450	4.243
61	107.000	61.840	6.245	107.000	55.850	6.708	107.000	45.370	5.196
62	100.000	43.810	6.325	100.000	73.040	7.000	100.000	68.270	7.071
63	117.000	58.750	6.708	117.000	62.110	6.928	117.000	97.220	8.888
64	42.000	17.270	3.317	42.000	17.270	3.317	42.000	17.050	3.606
65	95.000	39.250	5.196	95.000	46.260	5.568	95.000	46.260	5.568
66	57.000	42.080	5.745	57.000	25.950	4.899	57.000	25.950	4.899
67	105.000	76.880	7.211	105.000	45.430	5.745	105.000	59.050	6.403
68	72.000	28.430	5.000	72.000	35.280	5.099	72.000	28.450	4.123
69	125.000	88.010	8.000	125.000	58.450	6.481	125.000	86.230	7.416
70	102.000	41.810	6.245	102.000	71.040	6.928	102.000	44.010	5.196
71	155.000	84.050	7.937	155.000	85.660	7.746	155.000	85.660	7.746
72	115.000	48.140	6.403	115.000	45.740	6.403	115.000	56.400	7.071
73	72.000	31.560	4.472	72.000	36.240	5.385	72.000	29.080	4.243
74	85.000	34.180	5.831	85.000	33.620	5.000	85.000	36.380	4.690
75	15.000	5.830	2.000	15.000	5.830	2.000	15.000	5.830	2.000
76	37.000	14.320	3.000	37.000	14.320	3.000	37.000	14.320	3.000
77	107.000	43.760	5.831	107.000	50.610	5.916	107.000	43.780	5.099
78	77.000	30.440	5.000	77.000	30.410	5.000	77.000	45.420	5.477
79	92.000	54.210	5.831	92.000	40.420	5.099	92.000	37.740	4.690
80	165.000	107.940	9.110	165.000	111.440	9.327	165.000	77.960	7.141
81	112.000	50.620	6.557	112.000	45.560	6.708	112.000	60.450	6.633
82	130.000	65.400	6.708	130.000	71.720	7.937	130.000	56.290	6.325
83	97.000	38.650	6.164	97.000	67.880	6.856	97.000	40.850	5.099
84	132.000	110.010	8.944	132.000	77.630	7.416	132.000	64.230	6.245
85	57.000	23.320	4.472	57.000	30.960	4.690	57.000	30.960	4.690
86	130.000	97.440	8.602	130.000	59.510	7.348	130.000	68.500	7.000
87	82.000	36.030	4.899	82.000	48.430	5.916	82.000	38.050	4.690
88	112.000	106.750	8.888	112.000	52.800	7.071	112.000	53.210	6.245
89	85.000	41.680	5.292	85.000	37.260	5.000	85.000	34.580	4.583
90	137.000	103.700	8.888	137.000	66.010	6.633	137.000	66.010	6.633
91	102.000	90.680	8.246	102.000	71.840	7.681	102.000	50.710	6.403
92	120.000	70.800	6.928	120.000	57.980	6.164	120.000	52.950	5.568
93	80.000	56.400	6.403	80.000	40.270	5.657	80.000	40.270	5.657
94	90.000	36.140	5.477	90.000	42.990	5.568	90.000	36.160	4.690
95	132.000	68.360	7.550	132.000	65.960	7.550	132.000	76.620	8.124
96	12.000	4.120	2.000	12.000	4.240	1.414	12.000	4.240	1.414
97	165.000	83.230	8.062	165.000	97.180	8.718	165.000	82.710	7.616
98	95.000	67.030	7.071	95.000	48.190	6.403	95.000	56.540	6.708
99	127.000	51.570	6.164	127.000	58.420	6.245	127.000	72.180	6.403
100	82.000	43.680	5.385	82.000	35.260	4.899	82.000	32.580	4.472

D.6 Target, Mean and Standard deviation values of each customer for test case 103 with normal distribution

	Mean solution			90th percentile solution			Lateness Index solution		
Customer	Target	Mean	Std	Target	Mean	Std	Target	Mean	Std
1	42.000	18.400	3.606	42.000	18.710	3.464	42.000	18.400	3.606
2	50.000	19.100	4.000	50.000	24.410	4.472	50.000	19.100	4.000
3	107.000	106.860	9.165	107.000	42.490	6.000	107.000	55.600	5.916
4	127.000	50.240	6.557	127.000	50.360	6.557	127.000	74.590	7.141
5	17.000	6.710	2.000	17.000	6.710	2.000	17.000	6.710	2.000
6	112.000	66.930	7.071	112.000	56.850	6.708	112.000	60.900	6.481
7	75.000	36.210	5.292	75.000	31.070	4.796	75.000	31.070	4.796
8	97.000	61.160	7.141	97.000	63.280	6.928	97.000	43.680	5.745
9	127.000	52.980	6.481	127.000	81.770	7.746	127.000	55.890	6.083
10	120.000	50.080	6.325	120.000	67.120	7.280	120.000	95.480	8.185
11	137.000	74.280	7.483	137.000	60.470	6.782	137.000	64.140	6.403
12	102.000	48.530	5.831	102.000	43.390	5.385	102.000	43.390	5.385
13	30.000	11.660	3.000	30.000	11.710	2.828	30.000	11.710	2.828
14	10.000	3.000	1.000	10.000	3.000	1.000	10.000	3.000	1.000
15	140.000	76.090	7.681	140.000	103.070	8.775	140.000	57.980	7.000
16	115.000	58.820	6.782	115.000	45.850	6.403	115.000	47.680	6.325
17	42.000	16.120	4.000	42.000	16.600	3.162	42.000	16.600	3.162
18	130.000	71.370	7.141	130.000	71.390	7.000	130.000	64.180	7.071
19	152.000	87.430	8.062	152.000	86.990	8.367	152.000	79.120	7.550
20	90.000	74.090	7.416	90.000	62.170	6.856	90.000	67.930	7.483
21	85.000	41.820	5.477	85.000	36.680	5.000	85.000	36.680	5.000
22	155.000	64.380	7.141	155.000	70.370	7.141	155.000	74.040	6.782
23	100.000	74.140	7.348	100.000	70.210	7.280	100.000	44.770	5.745
24	125.000	60.320	6.403	125.000	82.440	7.681	125.000	54.180	6.403
25	52.000	20.590	4.472	52.000	21.070	3.742	52.000	21.070	3.742
26	110.000	53.890	6.403	110.000	43.810	6.000	110.000	47.860	5.745
27	112.000	44.690	6.000	112.000	61.730	7.000	112.000	51.300	6.083
28	127.000	65.890	7.071	127.000	52.920	6.708	127.000	90.300	8.602
29	125.000	59.930	6.928	125.000	60.050	6.928	125.000	64.900	6.782
30	122.000	65.950	7.141	122.000	89.390	8.000	122.000	48.270	5.745
31	37.000	24.400	4.123	37.000	24.710	4.000	37.000	14.320	3.000
32	105.000	41.900	6.083	105.000	71.220	6.856	105.000	44.560	5.292
33	112.000	57.490	6.481	112.000	45.570	5.831	112.000	51.330	6.557
34	117.000	64.400	7.211	117.000	64.520	7.211	117.000	60.430	6.481
35	95.000	80.220	7.616	95.000	64.130	7.000	95.000	38.690	5.385
36	137.000	101.300	8.485	137.000	85.730	8.185	137.000	88.180	8.485
37	57.000	31.100	5.000	57.000	33.220	4.690	57.000	33.640	4.690
38	80.000	38.210	5.385	80.000	33.070	4.899	80.000	33.070	4.899
39	130.000	57.080	6.633	130.000	74.120	7.550	130.000	58.370	6.403
40	157.000	82.040	7.810	157.000	70.500	7.616	157.000	65.380	7.348
41	130.000	57.080	6.633	130.000	74.120	7.550	130.000	58.370	6.403
42	120.000	72.650	7.483	120.000	80.190	7.746	120.000	52.180	6.164
43	140.000	77.090	7.746	140.000	64.100	7.348	140.000	58.980	7.071
44	72.000	34.120	5.099	72.000	46.610	5.657	72.000	29.160	5.000
45	67.000	26.590	4.472	67.000	26.840	4.123	67.000	26.840	4.123
46	165.000	71.210	7.280	165.000	81.330	8.124	165.000	91.120	8.062
47	110.000	63.320	6.782	110.000	51.400	6.164	110.000	57.160	6.856
48	117.000	95.680	8.660	117.000	53.670	6.708	117.000	66.780	6.633
49	60.000	23.870	4.123	60.000	23.870	4.123	60.000	23.870	4.123
50	147.000	64.150	6.928	147.000	81.190	7.810	147.000	65.440	6.708

	Mean solution			90th percentile solution			Lateness Index solution		
Customer	Target	Mean	Std	Target	Mean	Std	Target	Mean	Std
51	105.000	75.800	7.348	105.000	70.660	7.000	105.000	70.660	7.000
52	145.000	71.450	7.416	145.000	63.300	6.856	145.000	66.970	6.481
53	102.000	40.660	5.831	102.000	42.220	5.385	102.000	79.310	8.124
54	112.000	75.650	7.550	112.000	83.190	7.810	112.000	49.180	6.083
55	97.000	69.220	7.416	97.000	71.340	7.211	97.000	82.830	7.616
56	100.000	82.930	7.810	100.000	71.590	7.483	100.000	41.900	5.745
57	47.000	28.520	4.583	47.000	18.440	4.000	47.000	18.440	3.606
58	132.000	83.170	8.000	132.000	96.750	8.544	132.000	71.310	7.280
59	75.000	31.560	4.899	75.000	29.730	5.000	75.000	31.560	4.899
60	155.000	64.210	7.000	155.000	91.990	8.602	155.000	84.120	7.810
61	110.000	85.680	8.062	110.000	43.870	5.745	110.000	89.610	8.660
62	42.000	16.160	4.000	42.000	16.160	4.000	42.000	16.160	4.000
63	100.000	39.740	5.385	100.000	41.730	5.196	100.000	41.730	5.196
64	117.000	58.710	6.782	117.000	54.410	6.164	117.000	67.220	7.483
65	122.000	53.850	6.633	122.000	53.970	6.633	122.000	70.980	7.071
66	100.000	39.060	5.831	100.000	39.180	5.831	100.000	39.250	5.831
67	115.000	60.950	6.856	115.000	52.170	6.083	115.000	69.460	7.550
68	112.000	46.660	6.164	112.000	49.700	6.083	112.000	91.070	7.874
69	127.000	51.500	6.481	127.000	61.620	6.481	127.000	60.010	7.211
70	55.000	21.630	4.000	55.000	21.630	4.000	55.000	21.630	4.000
71	160.000	73.640	7.550	160.000	90.680	8.367	160.000	74.930	7.348
72	90.000	45.640	6.083	90.000	35.560	5.657	90.000	39.610	5.385
73	107.000	71.340	7.416	107.000	46.340	5.745	107.000	42.880	5.385
74	110.000	46.900	5.657	110.000	47.720	5.385	110.000	51.720	5.568
75	132.000	70.370	7.071	132.000	72.390	7.071	132.000	65.180	7.141
76	30.000	11.400	3.000	30.000	11.710	2.828	30.000	11.400	3.000
77	75.000	46.990	6.164	75.000	49.110	5.916	75.000	29.510	4.472
78	125.000	50.500	6.403	125.000	62.620	6.557	125.000	59.010	7.141
79	127.000	99.130	8.602	127.000	90.430	8.307	127.000	64.990	7.000
80	145.000	83.640	8.124	145.000	100.680	8.888	145.000	78.820	7.280
81	105.000	42.900	5.477	105.000	44.890	5.292	105.000	47.720	5.385
82	15.000	5.000	2.000	15.000	5.000	2.000	15.000	5.000	2.000
83	77.000	31.290	5.000	77.000	49.440	5.745	77.000	31.990	5.099
84	95.000	77.000	7.681	95.000	37.220	5.000	95.000	37.220	5.000
85	147.000	97.690	8.426	147.000	89.340	8.246	147.000	91.790	8.544
86	62.000	28.860	4.899	62.000	30.980	4.583	62.000	31.400	4.583
87	122.000	108.370	8.718	122.000	78.660	7.937	122.000	102.230	9.055
88	90.000	57.040	6.856	90.000	59.160	6.633	90.000	39.560	5.385
89	65.000	26.170	4.472	65.000	31.480	4.899	65.000	26.170	4.472
90	105.000	44.900	5.568	105.000	49.720	5.477	105.000	49.720	5.477
91	57.000	23.470	4.472	57.000	25.590	4.123	57.000	22.090	4.000
92	122.000	56.150	6.164	122.000	51.010	5.745	122.000	51.010	5.745
93	72.000	37.500	5.385	72.000	39.620	5.099	72.000	40.040	5.099
94	57.000	22.970	4.243	57.000	22.800	4.000	57.000	22.800	4.000
95	170.000	77.040	7.550	170.000	75.500	7.874	170.000	70.380	7.616
96	60.000	28.400	4.899	60.000	23.350	4.000	60.000	24.330	4.123
97	72.000	39.130	5.099	72.000	28.000	5.000	72.000	28.070	5.000
98	97.000	38.900	6.000	97.000	39.490	5.099	97.000	39.490	5.099
99	32.000	12.040	3.000	32.000	12.040	3.000	32.000	12.040	3.000
100	77.000	42.180	5.477	77.000	38.550	5.292	77.000	30.410	5.000

D.7 Target and Mean values of each customer for test case 101 with gamma distribution of scale parameter $\theta = 1$

	Mean solution		90th percentile solution		Lateness index solution	
Customer	Target	Mean	Target	Mean	Target	Mean
1	117.000	46.660	117.000	61.950	117.000	50.000
2	75.000	29.180	75.000	42.730	75.000	29.180
3	125.000	59.310	125.000	78.880	125.000	65.090
4	55.000	21.930	55.000	22.120	55.000	22.120
5	125.000	60.020	125.000	81.790	125.000	62.530
6	42.000	16.470	42.000	16.320	42.000	16.120
7	95.000	54.190	95.000	45.240	95.000	48.170
8	125.000	79.880	125.000	57.280	125.000	72.530
9	145.000	74.430	145.000	94.710	145.000	69.980
10	55.000	21.210	55.000	22.720	55.000	21.210
11	135.000	88.720	135.000	85.010	135.000	61.320
12	140.000	76.990	140.000	63.050	140.000	92.690
13	105.000	44.140	105.000	70.080	105.000	45.350
14	155.000	71.800	155.000	93.550	155.000	84.130
15	130.000	61.180	130.000	82.930	130.000	66.860
16	110.000	45.070	110.000	71.140	110.000	74.310
17	120.000	88.010	120.000	81.130	120.000	56.400
18	150.000	64.350	150.000	97.260	150.000	76.220
19	110.000	101.300	110.000	59.910	110.000	52.410
20	112.000	99.500	112.000	74.230	112.000	50.540
21	20.000	7.070	20.000	7.070	20.000	7.070
22	102.000	41.310	102.000	65.590	102.000	51.820
23	122.000	60.880	122.000	54.190	122.000	55.430
24	87.000	77.740	87.000	36.320	87.000	34.810
25	100.000	80.820	100.000	54.140	100.000	41.180
26	67.000	30.350	67.000	26.670	67.000	26.990
27	47.000	25.000	47.000	18.380	47.000	18.380
28	97.000	47.240	97.000	38.120	97.000	39.360
29	157.000	84.350	157.000	72.080	157.000	75.250
30	127.000	58.430	127.000	82.570	127.000	61.770
31	122.000	58.640	122.000	56.430	122.000	57.670
32	80.000	40.240	80.000	40.220	80.000	32.360
33	32.000	12.730	32.000	12.080	32.000	12.080
34	132.000	54.290	132.000	87.210	132.000	60.120
35	137.000	63.820	137.000	77.180	137.000	67.160
36	87.000	64.320	87.000	35.110	87.000	38.040
37	47.000	18.360	47.000	20.970	47.000	18.360
38	92.000	68.300	92.000	51.480	92.000	40.220
39	122.000	62.710	122.000	71.310	122.000	67.500
40	35.000	21.700	35.000	13.420	35.000	13.820
41	70.000	36.120	70.000	36.100	70.000	28.240
42	127.000	51.660	127.000	66.950	127.000	55.000
43	90.000	84.690	90.000	43.270	90.000	41.520
44	65.000	25.960	65.000	33.670	65.000	26.780
45	105.000	45.430	105.000	61.470	105.000	47.700
46	40.000	16.750	40.000	16.700	40.000	15.520
47	35.000	13.150	35.000	13.150	35.000	13.780
48	117.000	71.880	117.000	63.080	117.000	50.120
49	12.000	4.120	12.000	4.410	12.000	4.120
50	112.000	47.760	112.000	71.350	112.000	60.090

	Mean solution		90th percentile solution		Lateness index solution	
Customer	Target	Mean	Target	Mean	Target	Mean
51	120.000	68.390	120.000	75.720	120.000	74.070
52	7.000	2.000	7.000	2.000	7.000	2.000
53	110.000	94.090	110.000	52.700	110.000	59.620
54	87.000	36.470	87.000	34.930	87.000	39.510
55	132.000	54.820	132.000	70.110	132.000	58.160
56	30.000	11.750	30.000	11.700	30.000	11.700
57	130.000	83.700	130.000	56.340	130.000	85.980
58	102.000	40.920	102.000	45.560	102.000	46.600
59	95.000	77.850	95.000	51.620	95.000	37.850
60	45.000	17.200	45.000	25.760	45.000	18.410
61	12.000	4.470	12.000	4.470	12.000	4.470
62	87.000	34.960	87.000	34.740	87.000	35.780
63	120.000	88.800	120.000	70.330	120.000	79.460
64	87.000	38.050	87.000	44.150	87.000	38.240
65	90.000	43.150	90.000	39.050	90.000	35.390
66	77.000	30.180	77.000	41.730	77.000	30.180
67	120.000	96.340	120.000	77.390	120.000	53.700
68	127.000	55.780	127.000	77.550	127.000	73.150
69	70.000	27.660	70.000	30.080	70.000	33.280
70	142.000	67.230	142.000	69.930	142.000	80.770
71	17.000	6.470	17.000	6.320	17.000	6.320
72	120.000	54.650	120.000	79.370	120.000	59.440
73	147.000	81.650	147.000	103.400	147.000	68.390
74	110.000	85.330	110.000	44.210	110.000	96.410
75	97.000	39.040	97.000	47.600	97.000	40.250
76	115.000	46.310	115.000	50.950	115.000	51.990
77	150.000	63.350	150.000	98.260	150.000	75.220
78	77.000	31.180	77.000	44.140	77.000	31.180
79	122.000	86.600	122.000	82.540	122.000	57.810
80	115.000	56.490	115.000	63.290	115.000	52.030
81	90.000	81.860	90.000	40.440	90.000	44.350
82	92.000	43.350	92.000	41.080	92.000	59.010
83	57.000	25.880	57.000	22.200	57.000	22.520
84	10.000	3.000	10.000	3.000	10.000	3.000
85	117.000	67.950	117.000	47.120	117.000	48.360
86	125.000	62.470	125.000	82.040	125.000	61.930
87	120.000	67.570	120.000	62.740	120.000	56.830
88	97.000	59.850	97.000	39.580	97.000	42.510
89	102.000	64.300	102.000	55.480	102.000	44.220
90	122.000	50.680	122.000	75.270	122.000	54.280
91	92.000	45.590	92.000	38.840	92.000	56.770
92	35.000	13.890	35.000	16.500	35.000	13.890
93	60.000	23.920	60.000	24.450	60.000	24.450
94	112.000	76.110	112.000	54.200	112.000	48.290
95	122.000	54.080	122.000	77.670	122.000	66.410
96	117.000	56.180	117.000	58.040	117.000	88.230
97	85.000	73.300	85.000	46.480	85.000	35.220
98	107.000	101.500	107.000	72.230	107.000	48.540
99	70.000	34.900	70.000	35.080	70.000	28.280
100	112.000	55.190	112.000	70.800	112.000	49.810

D.8 Target and Mean values of each customer for test case 102 with gamma distribution of scale parameter $\theta = 1$

	Mean solution		90th percentile solution		Lateness index solution	
Customer	Target	Mean	Target	Mean	Target	Mean
1	150.000	70.860	150.000	103.230	150.000	78.700
2	117.000	67.860	117.000	65.870	117.000	58.990
3	120.000	52.360	120.000	51.400	120.000	63.380
4	40.000	15.030	40.000	15.030	40.000	15.030
5	102.000	83.570	102.000	44.440	102.000	50.280
6	80.000	31.910	80.000	33.910	80.000	39.620
7	102.000	79.330	102.000	48.680	102.000	44.400
8	70.000	47.910	70.000	34.190	70.000	31.780
9	80.000	36.020	80.000	32.760	80.000	39.910
10	92.000	41.130	92.000	46.920	92.000	48.970
11	102.000	49.690	102.000	56.700	102.000	62.470
12	60.000	23.020	60.000	25.500	60.000	23.020
13	72.000	49.070	72.000	28.640	72.000	35.790
14	102.000	59.600	102.000	46.970	102.000	43.790
15	145.000	100.130	145.000	104.220	145.000	96.540
16	72.000	55.470	72.000	28.640	72.000	28.640
17	67.000	27.890	67.000	26.930	67.000	27.890
18	77.000	51.600	77.000	30.530	77.000	30.640
19	72.000	30.680	72.000	30.680	72.000	38.140
20	92.000	37.680	92.000	43.470	92.000	46.860
21	87.000	47.130	87.000	55.220	87.000	35.110
22	42.000	16.120	42.000	16.120	42.000	16.120
23	132.000	64.400	132.000	63.440	132.000	75.420
24	75.000	29.180	75.000	36.710	75.000	29.070
25	127.000	71.890	127.000	64.230	127.000	56.080
26	57.000	23.980	57.000	23.710	57.000	23.710
27	40.000	15.270	40.000	15.000	40.000	15.000
28	105.000	77.920	105.000	50.090	105.000	45.810
29	145.000	124.430	145.000	79.920	145.000	81.670
30	77.000	32.890	77.000	31.930	77.000	43.910
31	87.000	37.360	87.000	36.400	87.000	48.380
32	57.000	24.250	57.000	24.250	57.000	24.250
33	85.000	83.780	85.000	36.150	85.000	37.380
34	125.000	59.160	125.000	57.020	125.000	67.000
35	95.000	38.910	95.000	50.630	95.000	81.780
36	30.000	11.660	30.000	15.630	30.000	11.660
37	127.000	80.630	127.000	69.190	127.000	73.810
38	65.000	64.270	65.000	33.560	65.000	25.340
39	55.000	38.080	55.000	24.360	55.000	21.950
40	10.000	3.610	10.000	5.000	10.000	3.610
41	87.000	83.590	87.000	46.810	87.000	35.900
42	37.000	14.000	37.000	14.000	37.000	14.000
43	125.000	110.410	125.000	66.170	125.000	51.970
44	75.000	30.180	75.000	35.710	75.000	30.070
45	62.000	54.900	62.000	28.670	62.000	24.700
46	75.000	29.180	75.000	36.710	75.000	29.070
47	117.000	117.010	117.000	78.440	117.000	64.060
48	77.000	55.210	77.000	34.140	77.000	34.400
49	7.000	6.610	7.000	2.000	7.000	2.000
50	137.000	68.750	137.000	75.760	137.000	81.530

	Mean solution		90th percentile solution		Lateness index solution	
Customer	Target	Mean	Target	Mean	Target	Mean
51	127.000	55.130	127.000	78.600	127.000	65.650
52	122.000	91.190	122.000	57.160	122.000	67.530
53	127.000	67.190	127.000	77.990	127.000	52.300
54	132.000	80.890	132.000	74.530	132.000	76.280
55	62.000	24.040	62.000	24.040	62.000	24.230
56	92.000	39.920	92.000	45.710	92.000	44.620
57	120.000	69.860	120.000	63.870	120.000	56.990
58	115.000	104.510	115.000	61.410	115.000	56.180
59	87.000	35.440	87.000	41.230	87.000	49.100
60	75.000	29.430	75.000	36.090	75.000	36.280
61	107.000	61.840	107.000	49.210	107.000	46.030
62	100.000	43.810	100.000	63.440	100.000	45.890
63	117.000	58.750	117.000	65.760	117.000	71.530
64	42.000	17.270	42.000	17.000	42.000	17.000
65	95.000	39.250	95.000	46.260	95.000	38.930
66	57.000	42.080	57.000	28.360	57.000	25.950
67	105.000	76.880	105.000	53.120	105.000	42.610
68	72.000	28.430	72.000	35.090	72.000	35.280
69	125.000	88.010	125.000	80.350	125.000	72.200
70	102.000	41.810	102.000	65.440	102.000	43.890
71	155.000	84.050	155.000	91.060	155.000	96.830
72	115.000	48.140	115.000	48.140	115.000	83.010
73	72.000	31.560	72.000	30.400	72.000	29.080
74	85.000	34.180	85.000	41.710	85.000	36.260
75	15.000	5.830	15.000	8.240	15.000	5.830
76	37.000	14.320	37.000	14.320	37.000	14.320
77	107.000	43.760	107.000	63.030	107.000	50.610
78	77.000	30.440	77.000	36.230	77.000	34.910
79	92.000	54.210	92.000	41.580	92.000	38.400
80	165.000	107.940	165.000	96.410	165.000	92.670
81	112.000	50.620	112.000	56.410	112.000	58.460
82	130.000	65.400	130.000	64.440	130.000	76.420
83	97.000	38.650	97.000	59.200	97.000	40.730
84	132.000	110.010	132.000	71.440	132.000	87.130
85	57.000	23.320	57.000	30.960	57.000	23.630
86	130.000	97.440	130.000	76.650	130.000	63.250
87	82.000	36.030	82.000	51.100	82.000	39.230
88	112.000	106.750	112.000	63.650	112.000	53.940
89	85.000	41.680	85.000	38.420	85.000	35.240
90	137.000	103.700	137.000	72.880	137.000	58.680
91	102.000	90.680	102.000	59.140	102.000	63.760
92	120.000	70.800	120.000	74.380	120.000	48.690
93	80.000	56.400	80.000	42.680	80.000	40.270
94	90.000	36.140	90.000	55.410	90.000	42.990
95	132.000	68.360	132.000	62.000	132.000	63.750
96	12.000	4.120	12.000	4.120	12.000	4.120
97	165.000	83.230	165.000	90.860	165.000	77.460
98	95.000	67.030	95.000	53.310	95.000	57.930
99	127.000	51.570	127.000	70.840	127.000	67.920
100	82.000	43.680	82.000	37.210	82.000	33.240

D.9 Target and Mean values of each customer for test case 103 with gamma distribution of scale parameter $\theta = 1$

	Mean solution		90th percentile solution		Lateness index solution	
Customer	Target	Mean	Target	Mean	Target	Mean
1	42.000	18.400	42.000	19.240	42.000	18.710
2	50.000	19.100	50.000	19.100	50.000	19.100
3	107.000	106.860	107.000	64.760	107.000	86.050
4	127.000	50.240	127.000	50.430	127.000	65.920
5	17.000	6.710	17.000	7.240	17.000	6.710
6	112.000	66.930	112.000	45.970	112.000	58.070
7	75.000	36.210	75.000	31.160	75.000	32.140
8	97.000	61.160	97.000	43.580	97.000	54.090
9	127.000	52.980	127.000	67.200	127.000	60.050
10	120.000	50.080	120.000	56.590	120.000	53.200
11	137.000	74.280	137.000	75.450	137.000	81.340
12	102.000	48.530	102.000	43.480	102.000	44.460
13	30.000	11.660	30.000	11.710	30.000	11.710
14	10.000	3.000	10.000	3.000	10.000	3.000
15	140.000	76.090	140.000	86.800	140.000	65.980
16	115.000	58.820	115.000	47.680	115.000	48.710
17	42.000	16.120	42.000	16.120	42.000	16.600
18	130.000	71.370	130.000	70.500	130.000	90.160
19	152.000	87.430	152.000	81.660	152.000	95.560
20	90.000	74.090	90.000	41.590	90.000	42.070
21	85.000	41.820	85.000	36.770	85.000	37.750
22	155.000	64.380	155.000	85.350	155.000	91.240
23	100.000	74.140	100.000	53.180	100.000	50.860
24	125.000	60.320	125.000	60.500	125.000	74.790
25	52.000	20.590	52.000	20.590	52.000	21.070
26	110.000	53.890	110.000	48.700	110.000	71.110
27	112.000	44.690	112.000	51.200	112.000	61.710
28	127.000	65.890	127.000	85.390	127.000	55.780
29	125.000	59.930	125.000	60.120	125.000	71.960
30	122.000	65.950	122.000	51.950	122.000	65.530
31	37.000	24.400	37.000	14.320	37.000	14.320
32	105.000	41.900	105.000	43.970	105.000	52.710
33	112.000	57.490	112.000	58.190	112.000	76.740
34	117.000	64.400	117.000	64.590	117.000	67.490
35	95.000	80.220	95.000	38.690	95.000	44.780
36	137.000	101.300	137.000	89.340	137.000	72.610
37	57.000	31.100	57.000	29.720	57.000	22.360
38	80.000	38.210	80.000	33.160	80.000	34.140
39	130.000	57.080	130.000	63.590	130.000	78.990
40	157.000	82.040	157.000	65.170	157.000	73.190
41	130.000	57.080	130.000	63.590	130.000	60.200
42	120.000	72.650	120.000	77.020	120.000	51.820
43	140.000	77.090	140.000	85.800	140.000	64.460
44	72.000	34.120	72.000	38.640	72.000	29.160
45	67.000	26.590	67.000	26.250	67.000	26.840
46	165.000	71.210	165.000	76.000	165.000	123.460
47	110.000	63.320	110.000	52.360	110.000	82.570
48	117.000	95.680	117.000	72.810	117.000	69.770
49	60.000	23.870	60.000	23.870	60.000	26.810
50	147.000	64.150	147.000	99.870	147.000	67.270

	Mean solution		90th percentile solution		Lateness index solution	
Customer	Target	Mean	Target	Mean	Target	Mean
51	105.000	75.800	105.000	50.440	105.000	71.730
52	145.000	71.450	145.000	78.280	145.000	84.170
53	102.000	40.660	102.000	42.440	102.000	46.710
54	112.000	75.650	112.000	74.020	112.000	48.820
55	97.000	69.220	97.000	59.330	97.000	40.550
56	100.000	82.930	100.000	41.900	100.000	60.420
57	47.000	28.520	47.000	18.440	47.000	18.440
58	132.000	83.170	132.000	79.720	132.000	58.380
59	75.000	31.560	75.000	31.560	75.000	29.920
60	155.000	64.210	155.000	86.660	155.000	90.560
61	110.000	85.680	110.000	55.050	110.000	72.200
62	42.000	16.160	42.000	16.160	42.000	16.160
63	100.000	39.740	100.000	41.140	100.000	55.540
64	117.000	58.710	117.000	59.190	117.000	58.290
65	122.000	53.850	122.000	54.040	122.000	62.310
66	100.000	39.060	100.000	39.250	100.000	53.250
67	115.000	60.950	115.000	56.950	115.000	60.530
68	112.000	46.660	112.000	60.880	112.000	66.370
69	127.000	51.500	127.000	53.190	127.000	51.080
70	55.000	21.630	55.000	21.630	55.000	24.570
71	160.000	73.640	160.000	90.380	160.000	76.760
72	90.000	45.640	90.000	40.450	90.000	35.560
73	107.000	71.340	107.000	46.560	107.000	50.830
74	110.000	46.900	110.000	68.190	110.000	61.370
75	132.000	70.370	132.000	71.500	132.000	89.160
76	30.000	11.400	30.000	12.240	30.000	11.710
77	75.000	46.990	75.000	29.410	75.000	39.920
78	125.000	50.500	125.000	54.190	125.000	50.080
79	127.000	99.130	127.000	73.400	127.000	52.060
80	145.000	83.640	145.000	80.380	145.000	86.760
81	105.000	42.900	105.000	64.190	105.000	49.550
82	15.000	5.000	15.000	5.000	15.000	5.000
83	77.000	31.290	77.000	41.470	77.000	31.990
84	95.000	77.000	95.000	37.440	95.000	41.710
85	147.000	97.690	147.000	85.730	147.000	69.000
86	62.000	28.860	62.000	27.480	62.000	24.600
87	122.000	108.370	122.000	75.290	122.000	58.560
88	90.000	57.040	90.000	39.460	90.000	49.970
89	65.000	26.170	65.000	26.170	65.000	25.000
90	105.000	44.900	105.000	66.190	105.000	47.550
91	57.000	23.470	57.000	22.090	57.000	22.090
92	122.000	56.150	122.000	51.100	122.000	52.080
93	72.000	37.500	72.000	36.120	72.000	30.430
94	57.000	22.970	57.000	23.020	57.000	22.800
95	170.000	77.040	170.000	70.170	170.000	78.190
96	60.000	28.400	60.000	23.350	60.000	24.330
97	72.000	39.130	72.000	28.070	72.000	42.070
98	97.000	38.900	97.000	38.900	97.000	38.900
99	32.000	12.040	32.000	12.040	32.000	12.040
100	77.000	42.180	77.000	30.580	77.000	32.070