UNIVERSITAS INDONESIA

Aplikasi *Ant Colony System* untuk Menyelesaikan Masalah Penjadwalan Rute Dinamis Kendaraan dengan Unsur *Fuzzy* pada Waktu Pelayanan

TESIS

Nama: Budhi Sholeh Wibowo
NPM: 0906578535

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK, JAWA BARAT
JANUARI 2012

UNIVERSITAS INDONESIA

Aplikasi *Ant Colony System* untuk Menyelesaikan Masalah Penjadwalan Rute Dinamis Kendaraan dengan Unsur *Fuzzy* pada Waktu Pelayanan

TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Teknik

Nama: Budhi Sholeh Wibowo
NPM: 0906578535

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK, JAWA BARAT
JANUARI 2012

HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar

Nama	: Budhi Sholeh Wibowo
NPM	: 0906578535
Tanda Tangan	:
Tanggal	: 20 Januari 2012

HALAMAN PENGESAHAN

esis ini diajukan oleh

ama	: Budhi Sholeh Wibowo
PM	: 0906578535
ogram Studi	:Teknik Industri
dul tesis	: Aplikasi *Ant Colony System* untuk Menyelesaikan Masalah Penjadwalan Rute Dinamis Kendaraan dengan Unsur Fuzzy pada Waktu Pelayanan

elah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian rsyaratan yang diperlukan untuk memperoleh gelar Magister Teknik pada Program Studi eknik Industri, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

mbimbing	: Prof. Dr. Ir. Teuku Yuri M. Zagloel, M.Eng.Sc.	( )
mbimbing	: Ir. Fauzia Dianawati, M.Si	( )
nguji	: Farizal, Ph.D	( )
nguji	: Ir. Amar Rahman, MEIM	( )
nguji	: Ir. Erlinda Muslim, MEE	( )

itetapkan di	: Depok, Jawa Barat
nggal	: 20 Januari 2011

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

ebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

ama	: Budhi Sholeh Wibowo
PM	: 0906578535
epartemen	: Teknik Industri
akultas	: Teknik
enis karya	: Tesis

emi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas ndonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas arya ilmiah saya yang berjudul : Aplikasi Ant Colony System untuk Menyelesaikan Masalah Penjadwalan Rute Dinamis Kendaraan dengan Unsur Fuzzy pada Waktu elayanan.

engan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, engalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, an memublikasikan tugas akhir saya **untuk akses terbatas pada lingkungan internal ampus** dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai emilik Hak Cipta. Adapun untuk **akses secara umum/ eksternal kampus dapat ilakukan setelah 5 tahun** semenjak karya imiah ini disetujui sebagai syarat memperoleh elar magister teknik.

emikian pernyataan ini saya buat dengan sebenarnya.

Depok, 20 Januari 2012
Yang menyatakan,

Budhi Sholeh Wibowo

Abstrak

Kemajuan teknologi informasi telah membuat orang mampu untuk memproses data secara *real-time* dan juga mengurangi ketidakpastian permintaan dalam manajemen logistik. Meski demikian, untuk bidang jasa, durasi waktu pelayanan (*service time*) seringkali masih tidak dapat diidentifikasi secara pasti. Studi ini mengajukan aplikasi *ant colony system* untuk menyelesaikan masalah penjadwalan rute dinamis kendaraan (*dynamic vehicle routing problem*) dengan unsur ketidakpastian pada waktu pelayanan. Studi ini mengajukan kasus yang lebih realistik dengan mempertimbangkan jumlah kendaraan yang terbatas. Pada model, teori *fuzzy* dan pengukuran kredibilitas digunakan untuk menghadapi unsur ketidakpastian. Sebuah metode heuristik konstruktif bernama *clustered-insertion method* diperkenalkan untuk meningkatkan kualitas solusi yang dihasilkan. Algoritma yang diajukan diuji dengan lima kasus yang memiliki tingkat kedinamikan yang berbeda. Hasil perhitungan menunjukkan bahwa fuzzy-ACS adalah sebuah metode yang efektif untuk menyelesaikan masalah ini.

Keywords: *ant colony optimization*, *dynamic vehicle routing*, *fuzzy theory, limited vehicle number, uncertain service time.*

Abstract

Recent advance in information technology has allowed people to do real-time processing and reduced demand uncertainty in logistics management. However, in case of service field, the duration of service time still often cannot be identified in certain. This study proposes an application of ant colony system (ACS) to solve dynamic vehicle routing problem with uncertainty in service time. The attempt is made to present a more realistic problem by considering a limited number of vehicles. In the model, fuzzy theory and credibility measurement are used to deal with the uncertainty. An improved constructive heuristic called clustered-insertion method is also introduced to improve the solution quality. The proposed algorithm was tested for five instances with different degrees of dynamism. The computational results show that fuzzy-ACS is an effective method to deal with the problem.

Keywords: ant colony optimization, dynamic vehicle routing, fuzzy theory, limited vehicle number, uncertain service time.

Kata Pengantar

Alhamdulillahirabbil'alamin. Puji syukur kepada Allah, Tuhan Yang Maha Esa, karena atas berkat rahmat-Nya penulis dapat menyelesaikan karya ilmiah ini. Penulisan tesis ini dilakukan dalam rangka memenuhi syarat untuk memperoleh gelar *dual degree*; Magister Teknik (MT) dari Departemen Teknik Industri, Universitas Indonesia dan *Master of Business Administration* (MBA) dari Departemen Manajemen Industri, National Taiwan University of Science and Technology. Penulis menyadari tesis ini tidak dapat selesai tanpa bantuan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Prof. Ren-Jieh Kuo selaku pembimbing tesis yang telah memberikan kesabaran, kepercayaan, dan arahan kepada penulis selama melanjutkan studi di National Taiwan University of Science and Technology (NTUST).
2. Prof. Dr. Ir. Teuku Yuri M. Zagloel selaku ketua Departemen Teknik Industri, Universitas Indonesia (UI) yang telah memberikan kesempatan kepada penulis untuk menempuh program dual degree di UI dan NTUST.
3. Seluruh jajaran dosen dan staf di departemen Teknik Industri, UI dan NTUST
4. Orang tua dan keluarga yang telah memberikan doa dan dukungan pernuh kepada penulis selama menempuh studi S2.
5. Seluruh rekan angkatan dan sahabat atas segala bantuan dan dukungan.

Akhir kata, penulis berharap Allah berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu pengetahuan.

Depok, 20 Januari 2012

Budhi Sholeh Wibowo

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

Vehicle Routing Problem (VRP) is a well-known classic NP-hard problem and one of the most challenging combinatorial optimization tasks introduced by Dantzig and Wright in 1954 (Dantzig et al., 1954). In VRP, a fleet of vehicles with limited capacity have to be routed in order to serve a set of geographically dispersed customers at minimum cost. VRP plays important role in logistics, especially in the design and management of distribution system, since it has high effect on efficiency in resource management, service level, and client satisfaction (Brito et al., 2009).

The interest in VRP is motivated by its practical application and complexity. During five decades since it first introduced, many applications and variants have been considered. Each of them usually has its own objective and problem constraints. Some examples of VRP variants are VRP with time window (Solomon, 1987), VRP with backhaul (Toth and Vigo, 1999) and VRP with pick-up and delivery (Nanry and Barnes, 2000).

Nevertheless, most of the previous studies of VRP were modeled in static and deterministic case where all the information was known *a priori*. In real world applications, many scheduling problems are actually dynamic and changing in nature. New orders may appear over time and must be incorporated into an evolving schedule (Kilby et al., 1998). Therefore, the dispatcher is faced with a dynamic decision making for continuously scheduling the vehicle route based on the latest information.

In last decade, an increasing interest had been paid attention to dynamic vehicle routing problem (DVRP). The recent development in DVRP is motivated by the current advances in communication and information technologies which allow real-time processing of customer orders and vehicles control (Fleischmann et al., 2004). On the other hand, current customers increasingly expect quicker and more flexible fulfillment of their transportation request. Practical applications of the DVRP can be found in taxi-cab services, express mail delivery, emergency services and also repairman services (Larsen, 2001; Gendrau et al., 2006).

In such dynamic environment, an advanced mobile communication system between dispatcher and driver is certainly required. Information technology such as Global Positioning System (GPS) and Geographical Information System (GIS) play essential roles for data acquisition purpose. By utilizing those technologies, the dispatcher knows the position of the vehicle and customer at any given point in time. Afterward, the dispatcher can tell the driver which customer to be served next based on the latest information. Information technology can also help reduce information uncertainty in the problem. The dispatcher can obtain information regarding customer location and demand through an online communication. Therefore he/she will have certain information while planning the route schedule. The DVRP system architecture is illustrated in Figure 1.1.

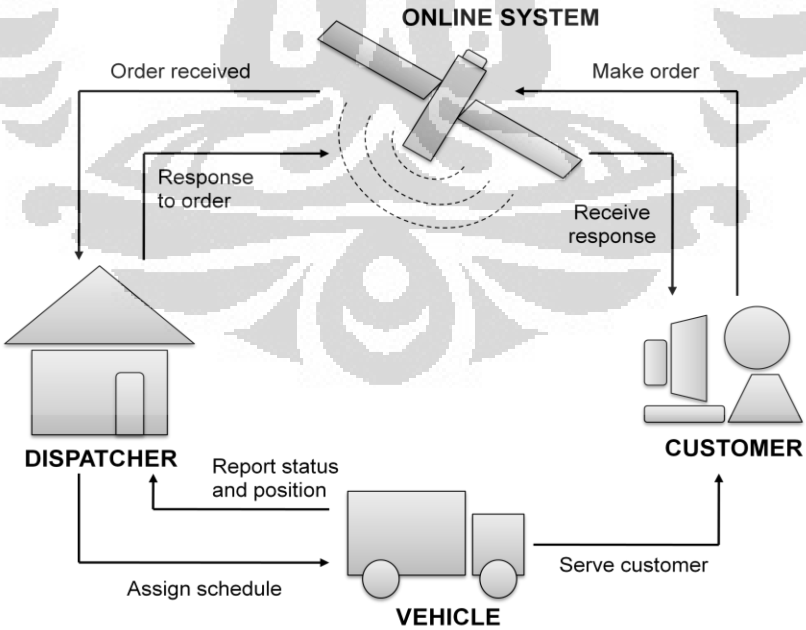

Figure 1.1 DVRP system architecture.

However, in case of on-site service (e.g. maintenance operation); it is often difficult to obtain a certain information regarding service time even through an online communication. Because even the customers itself often did not know how much time needed to serve their needs. Thus, the information regarding the service time will remain uncertain in the route planning. The dispatcher then should make estimation to handle the uncertain variable since the full information of it only can be known after the vehicle has finished the on-site operation.

There are numerous researches which proposed methods for handling uncertain variables in VRP. One of the ways is by considering them as random variables (Dror et al., 1989; Teodorovic and Pavkovic, 1992). This approach often referred as stochastic VRP. The basic input to solve such a problem is a probability density function which represents the uncertain variable distribution. Thus, in this kind of method, one needs to record numerous data of the specified variables in order to analyze and verify the distribution.

In case of dynamic problem, it is often difficult to know the distribution of the specified variables since new customers may arise over the time. Therefore, the dispatcher should make an approximation based on best available information to estimate the values of the uncertain variables from the new customer. Since approximation process often involves subjectivity and vagueness, one can apply fuzzy theory to deal with it. Fuzzy theory has been effectively implemented for dealing with uncertainty in VRP by Teodorovic and Pavkovic (1996), Zheng and Liu (2006), and Erbao and Mingyong (2009).

In 1992, Dorigo introduced a novel optimization method called Ant Colony Optimization (ACO) (Dorigo, 1992). ACO is a metaheuristic method which mimics the foraging behavior of real ant to solve optimization problem. ACO had been used widely to solve many difficult optimization problems with satisfactory result e.g. traveling salesman problem (Dorigo and Gambardella, 1996), vehicle routing problem (Rizolli et al., 2004), quadratic assignment problem (Gambardella et al., 1999), and job-scheduling (Colorni et al., 1994). Because of its flexibility, ACO is easily combined with other method such as fuzzy theory. Such realization can be seen in Kuo et al. (2003) and Teodorovic and Lucic (2007). They integrated ACO with fuzzy set theory to solve VRP with uncertain demand.

Since ACO is a class of heuristic methods, its computational result cannot be guaranteed to be optimal. However, they still produce high quality solution in very reasonable time. Thus, in many complex practical instances which require fast computation like real-time processing, heuristic methods is often more being preferred than exact method. In recent years, several heuristic methods have been proposed for DVRP, e.g. tabu search (Gendrau et al., 1998), ant colony systems (Montemanni et al., 2005), and genetic algorithm (Pankratz, 2005; Hanshar and Ombuki-Berman, 2007).

In previous study of DVRP, most of the researchers assumed that there are unlimited vehicles available to serve the entire requests in the day. In fact, this assumption does not always apply in general. In most real cases, the dispatcher only has a limited number of vehicles to be dispatched (Lau et al., 2003). Thus, he/she can only rely on the available capacity to serve all the customer requests. When there are a lot of requests coming in a day, the dispatcher should have filter mechanism to accept or reject the incoming request in order to optimize the available resources based on the latest information.

Thus, this study concerns about VRP in service field where the dispatcher is faced with dynamic requests and uncertainty in service time. The effort is made to present a more realistic problem by considering limited number of vehicles. Application of fuzzy theory and Ant Colony System (Fuzzy-ACS) is proposed to solve the developed problem. Even though, the hybrid of fuzzy theory and Ant Colony Optimization has been considered in previous researches, but its implementation in dynamic environment with limited resources has not been examined well. This reason has become the motivation to develop an application of fuzzy-ACS to solve DVRP with limited vehicles and uncertainty in service time.

1.2 Research Objectives

The objectives of this research are as follows:

(1) Develop VRP variant in service sector which has dynamic request, service time uncertainty, and limited number of vehicles.
(2) Develop fuzzy approximation method which is able to deal with limited resources and uncertainty in the problem.
(3) Design a hybrid implementation of ant colony system and fuzzy approximation to solve the specified problem with high quality solution.

1.3 Scope and Assumptions

The scope of the research is limited to the development of the hybrid intelligent algorithm of fuzzy sets theory and Ant Colony System for dynamic VRP with uncertainty in service time. In this problem, the assumptions are as follows:

(1) Customer location and travel time between customers can be known in certain (deterministic).
(2) The customers do not have specific time window. They would like to wait for the service until the end of the working time.

1.4 Research framework

The research is performed in a series of steps. The first step is to present the research background, objectives, and scope of the problem. The next step is to review and analyze the existing literatures related to the research e.g. DVRP, ant colony optimization, and fuzzy sets theory. Third step deals with model formulation of the problem and algorithm. The next step is to analyze the performance of the designed algorithm through computational experiment. The objective is to examine the behavior of the model in different environment. Finally, the result of the research is summarized and potential further research is discussed.

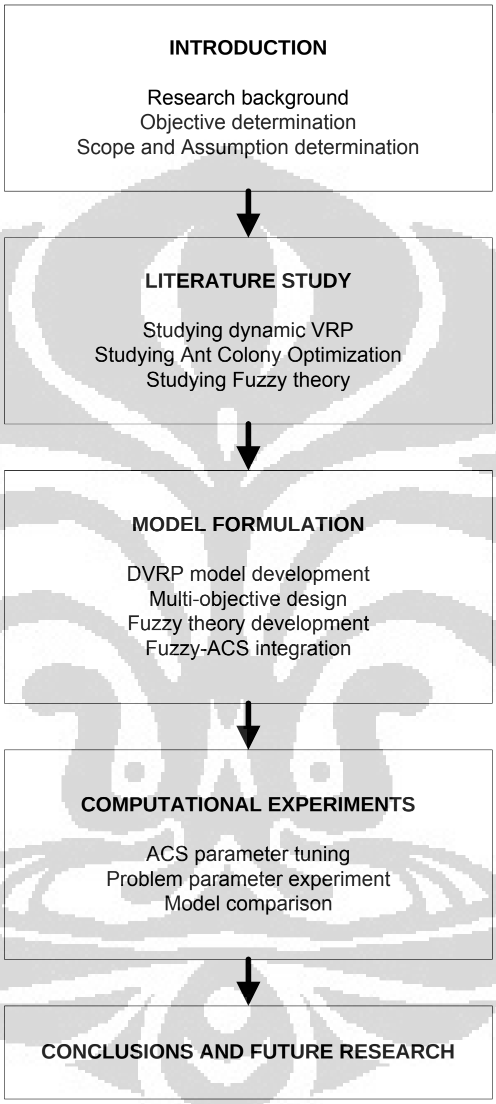

Figure 1.2 Research flowchart.

Chapter 2
Literature Review

2.1 Vehicle Routing Problem

Vehicle routing problem (VRP) plays a central role in logistics management. It consists of designing an optimal routes used by a fleet of vehicles stationed at a depot to serve a set of customers with known demands (Toth and Vigo, 2002). The work of Dantzig and Ramser (1959) is widely considered as the first scientific treatment of vehicle routing. It dealt with the routing of a fleet of gasoline delivery trucks between a bulk terminal and a large number of service stations supplied from the terminal.

VRP can be defined as directed graph $G = (V, A)$ where $V = \{v_0, v_1, ..., v_l\}$ is a vertex set, and $A = \{(v_i, v_j): i = j]\}$ is an arc set. Vertex v_0 denotes a depot at which m identical vehicles are based, and the remaining vertices of V represent cities. The value of m is either fixed at some constant, or bounded above by I_n. With every arc (v_i, v_j) is associated a non-negative distance c_{ij} (Gendrau et al., 1994). The objective of VRP is to set least cost routes of vehicles in such a way that:

a) every route starts and ends at the depot;
b) every customer is visited exactly once by exactly one vehicle;
c) the total demand of any vehicle route may not exceed the vehicle capacity;
d) the total length of any route (travel plus service times) may not exceed a preset bound;

The mathematical model of capacitated VRP is stated as follows (Toth and Vigo, 2002):

Minimize:

$$\min \sum_{i\in V} \sum_{j\in V} c_{ij} x_{ij} \qquad [2.1]$$

Subject to:

$$\sum_{h<i} x_{hi} + \sum_{j>i} x_{ij} = 2 \quad \forall i \in V \setminus \{0\} \qquad [2.2]$$

$$\sum_{j \in V \setminus \{0\}} x_{oj} = 2K \qquad [2.3]$$

$$\sum_{i \in S} \sum_{\substack{h<i \\ h \notin S}} x_{hi} + \sum_{i \in S} \sum_{\substack{j>i \\ j \notin S}} x_{ij} \geq 2r(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \theta \qquad [2.4]$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in V \setminus \{0\}, i < j \qquad [2.5]$$

$$x_{0j} \in \{0,1,2\} \quad \forall j \in V \setminus \{0\} \qquad [2.6]$$

where c_{ij} is the travel cost incurred on customer *i* to customer *j*, *K* is the number of vehicles, *C* denotes the loading capacity of vehicle, and d_i represents the demand at customer *i*. *S* is the customer set and *r(S)* denotes the minimum number of vehicles needed to serve set *S*.

Equation [2.1] determines the objective function of the problem. Equation [2.2] and [2.3] impose that exactly two edges are incident into each vertex associated with a customer and that 2*K* edges are incident into the depot vertex, respectively. Equation [2.4] represents the capacity-cut constraint of the problem which imposes both the connectivity of the solution and the vehicle capacity requirements by forcing that a sufficient number of edges enter each subset of vertices.

VRP may have additional constraints that will lead to different variants. Those variants are basically constructed by modification in one or more of VRP's elements. There are four elements which construct the model variants: the road network, the vehicles, the customers, and the uncertainties in the model (Rizolli et al., 2007). These elements can be set in different ways. For example, people may consider asymmetric road network, a heterogeneous vehicles, time windows and different type of customer request (pick-up or delivery). Besides, some uncertainties also can be considered into the model i.e. uncertainty in the demand, and travel times. Some examples of VRP variants are VRP with time windows (Solomon, 1987), VRP with backhaul (Toth and Vigo, 1999), VRP with pick-up and delivery (Nanry and Barnes, 2000), and stochastic VRP (Dror et al., 1989).

2.2 Dynamic Vehicle Routing Problem

Dynamic Vehicle Routing Problem (DVRP) is one of VRP variants which consider a dynamic decision process. It is inspired by the dynamic environment of real distribution system where the orders arrive randomly in time and the dispatching of vehicle is a continuous process of forming tours and collecting demands (Bertsimas and Ryzin, 1990). The development of DVRP is highly motivated by recent advances in communication and information technologies which allow people to do real time processing of customer orders and vehicle dispatching (Gendrau et al., 1999; Fleischmann et al., 2004). Some practical applications of DVRP can be found in taxi-cab services, express mail delivery, emergency services and also repairman services.

Psaraftis (1988) and Hanshar et al. (2003) give clear differences between static VRP and dynamic VRP. According to these researchers, static VRP is a class of problem where all the routing information is known in advance before the optimization process begun. Hence, no new information relevant to routing is obtained during the optimization. On the other hand, in the DVRP some information may exist to the planner before the optimization begins and some others information may revealed over time during the optimization process. Powel et al. (1995) classified a problem as dynamic if one or more of its parameters are a function of time and if the model is solved repeatedly as new information is received.

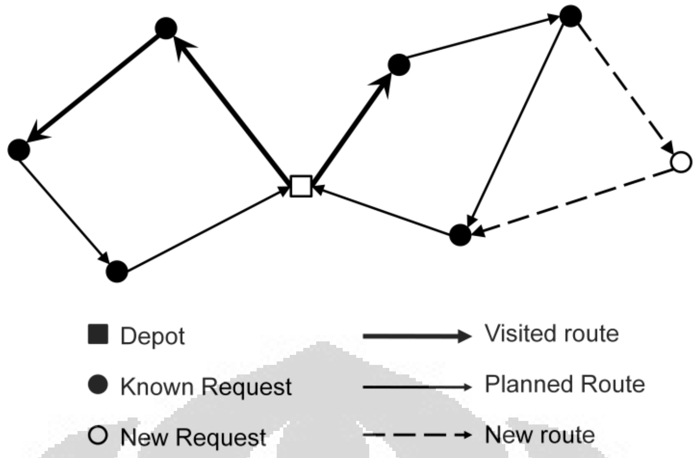

Figure 2.1 Graphical Representation of DVRP.

As a result, in DVRP, the dispatcher should continuously plan a route for vehicle to serve both static and dynamic requests. Ideally, new customers should be inserted without changing the order of planned route. However, in practice the insertion of new customer will usually be a much more complicated task (Larsen et al., 2002). This is because some constraints in the scheduling process, like the capacity and position of vehicles and customer time windows.

Due to the immediate request over time, the dispatcher is faced with a dynamic decision process. The decision may result in committing some orders while rejecting the others at a time. The rejected customers are usually followed by an offer to serve the customer in the following day or to be served by other companies (Meissel, 2011).

The objectives of DVRP can be different from one to another. It depends on the interest and problem type. A decision maker may prefer in maximizing his benefits i.e. the ratio of the demand served per time while others may prefer in service quality i.e. minimizing the total waiting time of customers (Bertsimas and Ryzin, 1990; Larsenet al., 2002; Meissel, 2011).

During the last two decades there are a growing number of papers addressing the dynamic version of VRP. Psaraftis (1988) elaborated a survey on DVRP which give a basic characteristic to the problem. Bertsimas and Ryzin (1990) introduced several policies to tackle the dynamic traveling repairman problem (DTRP). Kilby et al. (1998) proposed a way to split dynamic problem into several static sub-problems. Gendrau et al. (1999) studied about dynamic courier mail service problem with soft time windows. Larsen et al. (2001) investigated various problem of DVRP with different level of dynamism. Montemanni et al. (2005), Hanshar and Ombuki-Berman (2007), and Garrido and Riff (2010) introduced several metaheuristic applications to obtain fast and good quality solutions to specific problems.

2.2.1 Technical requirement

Larsen (2001) pointed out some required technologies when dealing with real-life applications of dynamic VRP. They are:

a) mobile communication system;
b) Global Positioning System (GPS);
c) Geographical Information System (GIS).

In a dynamic environment, a mobile communication system between dispatch center and driver is certainly required. It will help the dispatcher communicate with the driver about the updated schedule. Information technology such as Global Positioning System (GPS) and Geographical Information System (GIS) will also play an essential role for data acquisition purpose. GPS will help the dispatcher maintain information about vehicle status and position while GIS will provide the position of the customers and the path to reach them. However, the utilization of GPS and GIS may prove to be infeasible due to the operational cost of this method. Alternatively, the driver could send a message to update about his current status and position to the dispatch center each time he finishes the service for a customer (Larsen, 2001).

2.2.2 Degree of dynamism

The main sources of dynamism in the vehicle routing are the online arrival of customer during the operation. Dynamism will give more complexity to the problem. Lund et al. (1996) introduced a ratio to measure the dynamism of the problem called *degree of dynamism* (*dod*). This ratio measures the number of dynamic requests relative to the total number of request.

$$dod = \frac{\text{number of dynamic request}}{\text{Total number of request}} \qquad [2.7]$$

According to the formula, the problem is more dynamic if the above proportion is much closer to 1. If *dod* = 0, then the problem is static and if *dod* = 1, the problem is fully dynamic. This measure does not take into account the arrival times of dynamic request.

Larsen et al. (2002) proposed a new method to measure the dynamism called *effective degree of dynamism (edod)* which consider the disclosure time of request. Let us consider a problem where the planning horizon starts at time 0 and ends at time *T*. The advance requests are received before the beginning of the planning horizon or at time 0 at the latest. The time the *i*'*th* immediate request received is denoted t_i, where $0 < t_i \leq T$ .The number of immediate requests received during the planning horizon is denoted n_{imm} and the number of advance requests is denoted n_{adv}. The total number of requests, n_{tot} is therefore $n_{adv}+n_{imm}$. We now define the following measure as the *effective degree of dynamism*, (*dod*):

$$edod = \frac{\sum_{i=1}^{N_{imm}} \left(\frac{t_i}{T} \right)}{N_{total}} \qquad [2.8]$$

Equation [2.8] represents the average of how late the immediate requests are received compared to the latest possible time these requests could be received. If *edod* = 0, then the problem is in a pure dynamic system and if *edod* = 1, the problem is in a pure static system.

2.3 DVRP Applications

DVRP has a lot of variants. Most of them are motivated by the real world application where route construction and new information are processed during the operation day. Pillac (2011) divide the DVRP variant into three categories: transport of good, service and transport of person.

2.3.1 Transport of Good

Transport of good is the most well known application in VRP especially in logistics. In this category a fleet of vehicles is routed to pick or deliver some goods from customers into specified locations. Each vehicle has a specific capacity. They cannot carry goods more than their capacity.

The dynamic application of this category can be found in courier mail service (Gendrau et al., 1999) which offers to pick-up mail and/or packages at one location and deliver the goods safely at another location within a certain time limit. Montemanni et al. (2003) also gave a simulation of realistic DVRP case in Switzerland where the vehicle should pick some packages in widely scattered locations and bring it back to the depot.

2.3.2 Services

Common applications in dynamic routing of service vehicle can be found mainly in maintenance operations. In this category, vehicle routes will fulfill the service requests made by customers without any capacity constraint. This problem is often referred as Dynamic Traveling Repairmen Problem (DTRP) which has been studied intensively by Bertsimas and Ryzin (1991) and Larsen et al. (2002).

In DTRP, a repairman should create a route to serve a set of service request which arrive dynamically over time. Every demand i requires an amount of service times with specific duration, s_i. The system time, T_i, of demand i is defined as an elapsed time between the arrival of the demand i and the time the repairman completes the service of the demand.

The waiting time, W_i, of demand i is defined as an elapsed time from the demand arrive until the service start.

$$T_i = W_i + s_i \qquad [2.9]$$

In case of service, the customer's waiting time is often more important than the travel cost. Hence, in DTRP the objective function is to design a route which minimizes the average system time $\bar{\bar{T}}$.

$$\bar{T} = \frac{\sum_{i=1}^{n} T_i}{n} \qquad [2.10]$$

The other application of this category also can be found in emergency service like police, fireman and ambulance. In those situations, the strategy is to assign the best vehicle (for instance the nearest) to the new request. Therefore the location analysis for deciding where to locate the vehicles and crews become a main issue for the dispatching problem (Larsen, 2001).

2.3.3 Transport of Person

Taxi cab and dial-a-ride service are the most common applications in this category. For the taxi cab service, the dynamic request is very high. The taxi should pick customers in specified location and deliver them to a certain place. This kind of service is also provided by the dial-a-ride service, especially to serve the elderly and handicapped people. The dynamic version of dial a ride problem had been studied by Psaraftis (1983) where each of customers has a specific time window and the objective is to minimize the time needed to serve all customers.

2.4 DVRP Solution Methods

VRP can be solved by exact methods or heuristic approaches. The use of exact methods can obtain the optimal solutions. However, since VRP is a NP hard problem, the computational effort required to solve this problem increases exponentially with the problem size. Therefore, nowadays heuristics approaches are mostly preferred for tackling a real-life instance. It is used to find good solution, but not necessarily guaranteed optimal using a reasonable amount of computational time. In this section, we will focus on some reviews about DVRP solution methods.

2.4.1 Exact solution methods

The first known application to the optimization of dynamic routing was the work of Psaraftis (1983). He developed a dynamic programming approach for dial-a-ride problem (DARP). Bertsimas and Ryzin (1990) used queuing model to test several policies on dynamic traveling repairman problem (DTRP). The recent exact solution method was proposed by Yang et al. (2004). They used mixed integer programming approach combined with simple local rule to solve real-time multi-vehicle truckload pickup and delivery.

Pillac et al. (2011) gave valuable comment to exact solution method. According to the researchers, in such a dynamic environment, critical information is revealed over time, meaning that a complete instance is only known in the end of horizon time. As a consequence, an optimal solution can only be found a-posteriori and exact method only provide an optimal solution for the current state, lacking of any guarantee that the solution will be optimal once new data become available.

2.4.2 Heuristics methods

Because the optimal solution only can be found a-posteriori, most of real life problems rely on heuristics approaches to give a good solution in reasonable time. The basic strategy in heuristics methods is to decompose the dynamic problem into a series of static problems

(Bianchi, 2000). By doing so, the optimization process is expected to run unhindered for a given time and results in a good and feasible solution.

Kilby et al. (1998) introduced the concept of *time-step* and *cutoff time* in dynamic problem. *Time-step* is a way to decompose a dynamic problem into a sequence of static problems. It divides the time horizon into several periods. In each period, a static problem is created and solved respectively. In other side, *cutoff time* is a way to change the level of dynamism in the problem. It is expressed as a fraction of the working day. Any request with an arrival time before the cutoff times is treated as if they arrived yesterday. These concepts had been followed by Montemanni et al. (2005), Hanshar and Ombuki-Berman (2007), and Garrido and Riff (2010). By adopting those concepts, many constructive and improvement heuristics methods that works on static VRP will also work in DVRP.

Here we list some heuristics methods which have been applied to the DVRP:

a) *Insert and Improve Algorithm*

Kilbyet al. (1998) used an *insert and improve* algorithm which combine cheapest insertion method with 2-opt, 3-opt, and Or-opt. The cheapest insertion method is a cost-based function heuristics which able to construct an initial solution in VRP. It expands a current routeR_c by sequentially insert an unrouted customer k within an edge a-b, that minimize the total cost of route.

$$\arg\min_k \left\{ C_r * (a, k*, b) = c_{ak} + c_{bk} - c_{ab} \right\} \quad [2.11]$$

The researchers tested their algorithm to some new benchmark data sets which were created by modifying the standard VRP data sets of Christofides et al. (1979), Taillard (1994) and Fisher et al. (1996). It was done by adding three new data types to the basic VRP data sets i.e. available time of customers, duration of each visit, and the working day periods. Unfortunately, the researchers did not show the optimization results of the data sets. They only showed the impact of varying the degree of dynamism and the length of commit horizon to the total travel cost.

b) *Tabu Search*

Gendrau et al. (1999) applied tabu search heuristics to solve DVRP with time windows. They modified the tabu search so it can adapt to the dynamic case. The strategy is to re-optimize the route every time a new customer revealed. As a consequence, they allowed the diversion of vehicle from its current destination to adapt the new planned route. However, the researchers noted that this strategy did not apply in general. It must be carefully addressed in what case the diversion should be allowed.

c) *Ant Colony System*

Montemanni et al. (2005) developed an ant colony system (ACS) heuristics with pheromone trace to transfer characteristics of good solution into the next time step. The researcher also introduced the concept of *event manager* in dynamic problem. Event manager is an interface between the architecture and the external world. This module helps the dispatcher to handle dynamic orders from customers and transform it into a sequence of static problem. The researchers used Kilby data sets to test their algorithm and compare the result to *Greedy Randomized Adaptive Search Procedure* (GRASP) (Resende and Ribeiro, 2003). The computational results showed that the ACS heuristics is able to give a good solution and outperform some of the GRASP result.

d) *Genetic Algorithm*

Hanshar and Ombuki-Berman (2007) used genetic algorithm (GA) to solve DVRP. They created a new chromosome representation to deal with the dynamicity. It consisted of two types of nodes: positive node representing a single customer (who has yet to be assigned to a vehicle) and a negative node representing a group of clustered customers that have been already committed to a given vehicle. In the experiment, they compared their algorithm to the Ant Colony System (Montemanni et al., 2003). Numerical result showed that GA outperformed almost all of the ACS result in Kilby data sets.

e) *Hyperheuristics*

Garrido and Riff (2010) implemented evolutionary hyper-heuristics method to DVRP. Hyper-heuristics is a high level heuristics which manages a set of low level heuristics in a common framework. The researchers set genetic algorithm as the high level heuristics. Each gene in the chromosome represented a set of constructive and improvement heuristics for a specified number of customers. They selected four heuristics for the initial solution, i.e. saving method, sequential insertion method, cluster-first route-second, and dynamic insertion method. For the improvement heuristics, they selected five different methods, namely 3-opt, Or-opt, string cross, string relocate and string exchange. The result show that evolutionary hyper-heuristics was able to compete with the previous algorithm such as GRASP, ACS, Tabu Search, and GA. Some of the results even gave new best known solutions to the benchmark problems.

2.5 Ant Colony Optimization

Ant Colony Optimization (ACO) is a metaheuristic method which mimics the foraging behavior of real ant for solving computational problem through graph. ACO was first initially introduced by Dorigo (1992) and has been successfully applied to solve many of optimization problems, e.g. traveling salesman problem (Dorigo et al., 1996) quadratic assignment (Gambardella et al., 1999), and job-shop scheduling (Colorniet al., 1994).

In ACO, an artificial ant constructs a solution by visiting a series of nodes on a graph. They select the next node according to two parameters: trails and attractiveness. The attractiveness η_{ij} of a move from node i to j is computed according to a heuristic which expresses the a priori desirability of the move. In a shortest path problem, the attractiveness can be expressed as the inverse of the distance. The trail level τ_{ij} of a move depends on the pheromone level, and it represents a dynamic indication *a posteriori* of its goodness. In other words, the trail will show the promising direction to explore. When the constructive procedure has finished, the pheromone information is updated according to the following equation:

$$\tau_{ij} = (1-\rho).\tau_{ij} + \Delta\tau_{ij} \qquad [2.12]$$

whereρ denotes the evaporation rate of the posterior pheromone and $\Delta\tau_{ij}$ denotes the amount of pheromone deposited to the edge *i-j*. The evaporation rate ρ will avoid the ant to be trapped into local optima due to the strength of the pheromone level in the trail.

Based on those two parameters, an ant selects the next node to be visited by a probabilistic random proportional rule. It is formulated by:

$$p_{ij} = \begin{cases} \dfrac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{h\epsilon\Omega}\left[\tau_{ih}\right]^{\alpha}\left[\eta_{ih}\right]^{\beta}}, & \text{if } j\epsilon\Omega \\ 0, & \textit{otherwise} \end{cases} \qquad [2.13]$$

where p_{ij} is the probability of moving to *j*f rom*i*, and is the set of nodes which are feasible to be visited from *i*. The parameters α and β are the weights of trails and visibility which determine the relative importance of the parameters. Ants construct their solutions in parallel. At the end of each constructive phase (iteration) the entire set of computed solutions is used to update the pheromone trail by following equations:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad [2.14]$$

Gambarella and Dorigo (1997) proposed a modified version of ACO called Ant Colony System (ACS) to enhance the optimization process. They introduce the concept of *global update*, *local update*, and *pseudo-random rule*.

In those concepts, pheromone update is performed much more frequently. Every time an ant finished constructing a solution, local update is performed to exploit the solution candidate. At the end of iteration, only the best solution is used to update the global pheromone and transferred into the next iteration. Let denotes global pheromone matrix as

ξ_{ij} and Ω^* is the best solution found in the colony. Then the global pheromone update can be formulated as follow:

$$\varsigma_{i^*j^*} = (1-\rho).\varsigma_{i^*j^*} + \Delta\varsigma_{i^*j^*} \quad i^* \text{ and } j^* \in \Omega^* \qquad [2.15]$$

Initialize pheromone values;

While (*stopping criterion is not met*)

  For *each ant k*

    *S = {*1, 2, 3,*…n};*

    While (*S≠ ∅*)

      Choose next node j based on equation [2.16];

      S = S\{j};

      Update local pheromone based on equation [2.11];

      i = j;

    End While

  End For

  Calculate Fitness Function;

  Return the best solution in population Ω^*;

  Update global pheromone based on equation [2.15];

End While

Return the best solution;

Figure 2.2 Ant Colony System pseudo-code.

In ACS, an ant used a pseudo-random rule instead of random-proportional rule for selecting the next node. In this rule, an ant in node *i* select the best state as the next node *j* with probability q_0, and select with a proportional rule with probability 1-q_0. It is formulated as follow:

$$j = \begin{cases} \arg\max\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}, & \text{if } q < q_0 \\ \text{eq.[2.14]}, & \textit{otherwise} \end{cases} \qquad [2.16]$$

In each of the iteration, the algorithm will keep the best solution in the colony. If the new best solution found, the previous best solution will be replaced. When the terminating condition is reached, the algorithm will return to the best solution as the result of the optimization.

2.6 Fuzzy Sets Theory

Fuzzy set is a set without a crisp or clearly defined boundary. Fuzzy set was first introduced by Zadeh (1965) to deal with a class of object that does not have precisely defined criteria of membership. In optimization problem, fuzzy sets theory help to deal with a class of problem where the constraints or the objective function cannot be valued in precise way (Brito et al., 2009). Therefore, besides the stochastic approach, fuzzy logic often becomes a preferred method to handle uncertainty in a problem.

A fuzzy set admits the possibility of partial membership. Each elements of fuzzy set have a membership degree from 0 to 1 which indicates the certainty that the element belongs to a set. If an element has membership degree equal to 0 then it means that the element is definitely not a member of the set. If the membership degree is greater than 0 and less than 1, then it falls on fuzzy boundary of the set.

2.6.1 Membership Function

To define a membership degree in a fuzzy set, a membership function is often used. Membership function (MF) is a curve that defines how each point in the input space is mapped to a membership degree between 0 and 1.

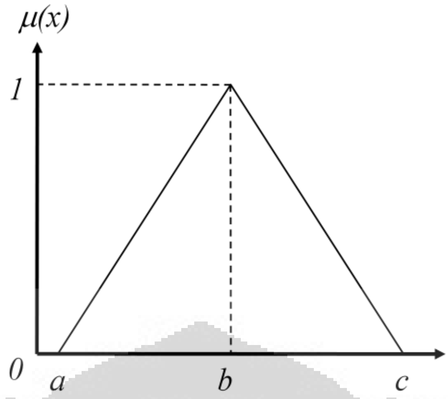

Figure 2.3 Triangular membership function.

There is several ways to state a membership function for a fuzzy set. The simplest way is by using a triangular membership function. Triangular MF is specified by 3 parameters {*a*, *b*, *c*} where *a* denotes the lower bound, *b* denotes the most likely to happen, and *c* denote the upper bound. The membership degree, $\mu(x)$, in triangular MF is defined by the following equation:

$$\mu(x;a,b,c)=\begin{cases}0, & \text{if } x<a \\ \dfrac{x-a}{b-a}, & \text{if } a\le x<b \\ \dfrac{c-x}{x-b}, & \text{if } b\le x<c \\ 1, & \text{if } c<x\end{cases} \qquad [2.17]$$

2.6.2 Logical Operations

Zadeh (1965) defined the logical operations of fuzzy sets as follows:

a) Two fuzzy sets *A* and *B* is equal if and only if $f_A(x) = f_B(x)$ for all a*x* in *X*.
b) The complement of fuzzy set *A* is denoted by *A'* and is defined by:

$$f_A{}'(x)=1-f_A(x) \qquad [2.18]$$

c) The union of two fuzzy sets *A* and *B* is fuzzy set *C*, written as $C = A \cup B$ whose membership function is related to those of A and B by:

$$f_C(x) = \max\left[f_A(x), f_B(x)\right] \qquad x \in X \qquad [2.19]$$

d) The intersection of two fuzzy sets *A* and *B* is fuzzy set *C*, written as $C = A \cap B$ whose membership function is related to those of A and B by:

$$f_C(x) = \min\left[f_A(x), f_B(x)\right] \qquad x \in X \qquad [2.20]$$

2.6.3 If-Then Rules

One of the benefits of fuzzy theory is its approximate reasoning capability using a simple linguistic statement of *if-then rules*. These rules are used to formulate the conditional statements which contain a fuzzy logic. A single fuzzy *if-then rule* assumes the form:

if *x* is *A* then y is *B*

where *A* and *B* are linguistic values defined by the fuzzy sets. The if-part of the rule "*x* is *A*" is called the *premise*, while the then-part of the rule "*y* is *B*" is called the *conclusion*. In order to apply a fuzzy logic, one should transform the premise statement into a membership degree between 0 and 1. Therefore the degree can be used to shape the output or the conclusion of the fuzzy.

2.6.4 Credibility Measure Theory

Credibility measure theory was first proposed by Liu (2004) which used to measure a fuzzy event. The credibility of a fuzzy event is defined as the average of its possibility and necessity. In the fuzzy theory, it is obvious that a fuzzy event may fail even though its possibility achieves 1, and hold even though its necessity is 0. However, the fuzzy event must hold if its credibility is 1, and fail if its credibility is 0. Thus, the credibility measure plays a role like a probability measure.

For a triangular fuzzy set, $A = \{r_1, r_2, r_3\}$, the credibility Cr is defined as follow:

$$Cr\{A \geq r\} = \begin{cases} 1, & r \leq r_1 \\ \dfrac{2r_2 - r_1 - r}{2(r_2 - r_1)}, & r_1 < r \leq r_2 \\ \dfrac{r_3 - r}{2(r_3 - r_2)}, & r_2 < r \leq r_3 \\ 0, & r \geq r_3 \end{cases} \qquad [2.21]$$

2.7 Fuzzy Optimization in VRP

In a real world VRP, the decision making process is often faced with a high degree of uncertainty. The required information is not always available at the beginning of the problem. Therefore, many researchers treated the uncertain variable as a random variable (Teodorovic and Lucic, 2007). These problems are known in the literature as stochastic VRP. The basic input data to solve such a problem are the probability density functions of the random variables.

Teodorovic and Pavkovic (1996) stated two main weaknesses in the stochastic approach. First, the need of huge amount of data to verified the probability density functions as the input data. Second, stochastic approach cannot deal with a new variable which does not have a historical data. In other words, the information is often not precise enough. In this case, one should use approximates value based on the best available information could be use. The approximate value can be expressed in fuzzy numbers which able to deal with imprecise data. Therefore, a fuzzy approach is also an appropriate way to handle the uncertainty. There are a growing number of researches which had been applied fuzzy sets theory in VRP. Most of them fall into these categories: (1) fuzzy demands, and (2) fuzzy times.

2.7.1 Fuzzy Demands

In this case, the dispatcher is faced with imprecise information or uncertainty regarding the amount of demand at some nodes. The idea is to treat the uncertain demand as a fuzzy numbers (Teodorovic and Pavkovic, 1996; Kuo et al. (2004); Teodorovic and Lucic, 2007)

Let us denote the vehicle capacity by C and the fuzzy numbers representing demand at *i*-th node by D. After serving *k* nodes, the available capacity A_k equal:

$$A_k = C - \sum_{i=1}^{k} D_i \qquad [2.22]$$

If the demand D_i is represented by a triangular fuzzy numbers (d_{1i}, d_{2i}, d_{3i}) then the available capacity, A_k is also a triangular fuzzy numbers.

$$A_k = \left(C - \sum_{i=1}^{k} d_{3i}, C - \sum_{i=1}^{k} d_{2i}, C - \sum_{i=1}^{k} d_{1i} \right) \qquad [2.23]$$

The strength of preference, p_k for the vehicle to serve the next node after it has served *k* nodes depends on available capacity A_k. Hence, the approximate reasoning algorithm could be stated as follows:

Rule 1:
IF the *available capacity* is *small*
THEN the *preference strength* is *low*.

Rule 2:
IF the *available capacity* is *medium*
THEN the *preference strength* is *medium*.

Rule 3:
IF the *available capacity* is *large*
THEN the *preference strength* is *high*.

Zheng and Liu (2006) and Erbao and Mingyong (2009) develop a fuzzy chance constraint model with a credibility measure to solve the fuzzy VRP problem. They used subjective parameter *Cr** which indicate the behavior of decision maker toward the risk. Lower value of parameter *Cr** indicates the dispatcher desire to use vehicle remaining capacity as best as it can while a higher value indicates the risk aversion behavior of the dispatcher.

2.7.2 Fuzzy Travel Times

In many practical problems, a travel time between two locations in routing problems is often imprecise in advance because of the road conditions or traffic congestion. Therefore it also can be considered as fuzzy numbers. This kind of problem had been studied by Hong and Xu (2008) and Brito et al. (2010).

Unlike the fuzzy demands which only deal with a fuzzy constraint problem, the fuzzy times is often also be faced with a fuzzy objective function (Brito et al., 2009). In that case, obviously, the objective value is also become a fuzzy number. Brito et al. (2009) proposed to follow Harrera and Vardegay (1995) method to use Third Yager's index to solve the problem. The Third Yager's index is a linear ranking function that, applied to a triangular number $\tilde{t} = Tr(t^1, t^2, t^3)$ is given by:

$$g(\tilde{t}) = (t^1 + 2t^2 + t^3) \quad [2.24]$$

Then, if each fuzzy travel time$\tilde{t}_{ij}^k$ is a triangular fuzzy number $Tr(t_{ij}^{1k}, t_{ij}^{2k}, t_{ij}^{3k})$and u_i^k is a crisp number equivalent to a triangular fuzzy number$Tr(u_i^k, u_i^k, u_i^k)$, the objective function can be replaced by the following crisp function:

$$\min\left(\sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n}\left(t_{ij}^{1k}x_{ij}^k + 2t_{ij}^{2k}x_{ij}^k + t_{ij}^{3k}x_{ij}^k\right) + \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n}4u_i^k x_{ij}^k\right) \quad [2.25]$$

Chapter 3
Model Formulation

3.1 Problem Description

Consider a dynamic VRP where a specified number of vehicles are dispatched from a depot to serve static and dynamic requests from customers. The dispatcher should manage a “tentative route schedule” which incorporates all requests currently known. When the new request arrived, then the schedule should be adjusted according to the new information. The dispatcher may also improve the schedule as long as it does not interfere with decisions that have been already committed to. The vehicles are operated within a given working time period, originating and terminating at the depot such that:

a) Each vehicle service one route;
b) Each customer is visited exactly once
c) The start time of each vehicle route is greater than or equal to 0;
d) The end time of each vehicle route is less than or equal to working time period.

In this problem, we consider fleet of vehicle which provides on-site service to the customers. Each customer requires a unique service time whose duration cannot be known precisely until the server has finished the operation. For simplicity, we assume that the travel time between customers is deterministic and can be known in certain. We also assume that the customers do not have a specific time windows. After they sent a request, they will wait for the service until the end of the day.

Since the service time is uncertain, the dispatcher should make estimation about the total time needed to accomplish the planned schedule. The estimated total time is consisted of total travel time between customer and total estimated service time at the customer. It should not exceed the working time period.

In dynamic environment, service requests may appear randomly over time. Hence, the dispatcher is faced by decision, whether to accept those requests and serve them in the day or reject them due to capacity limit.

This model considers two objectives. The first objective is to maximize the number of customers served. The more customers served in one day, the more benefits are the company received. The second objective aims to minimize the average waiting time of the customers. The customer waiting time is defined as the elapsed time from the request arrives until the service start. The shorter the waiting time, the more satisfaction the customer has. Therefore, the dispatcher should plan a schedule which compromises both objectives.

3.2 Mathematical Model

According to the problem description, the mathematical model for the problem can be formulated as follow:

Definitions:

T = length of working period

V' = set of the known pending orders (non-visited customers)

n = number of the customers in V'

m = number of vehicles

e_{ij} = travel time from customer i to customer j.

$\tilde{u}_j$ = estimated service time at customer j

$\tilde{a}_j^k$ =estimated time of vehicle k arrived at customer j

r_j = request arrival time of customer j

$\tilde{w}_j$ = estimated waiting time of customer j

$$\tilde{w}_j = \tilde{a}_j^k - r_j \quad [3.1]$$

Decision Variables:

$$y_j^k = \begin{cases} 1 & \text{if customer } j \text{ is visited by vehicle } k \\ 0 & \textit{otherwise} \end{cases} \quad [3.2]$$

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ move from } i \text{ to } j \\ 0 & \textit{otherwise} \end{cases} \quad [3.3]$$

Objective Functions:

$$\max F_1 = \sum_{k=1}^{m} \sum_{j=0}^{n} y_j^k \quad [3.4]$$

$$\min \tilde{F}_2 = \frac{\sum_{i=1}^{n} \tilde{w}_i}{n} \quad [3.5]$$

Constraints:

$$\sum_{i=0}^{n} \sum_{j=0}^{n} \left(e_{ij} + \tilde{u}_j\right) x_{ij}^k \leq T, \quad \forall k = 1,\ldots,m; \quad [3.6]$$

$$\sum_{k=1}^{m} y_j^k = 1, \quad \forall j = 1,2,\ldots,n; \quad [3.7]$$

$$\sum_{i=0}^{n} x_{ij}^k - \sum_{j=0}^{n} x_{ji}^k = 0, \quad \forall j = 0,1,\ldots,n; \; k = 1,\ldots,m \quad [3.8]$$

$$\sum_{j=1}^{n} x_{j0}^k = m, \quad \forall k = 1,\ldots,m; \quad [3.9]$$

$$\sum_{j=0}^{n} x_{ij}^k = y_j^k, \quad \forall j = 0,1\ldots,n; \; k = 1,\ldots,m; \quad [3.10]$$

Equation [3.4] is the first objective function which seeks to maximize the number of customers served during the day. Equation [3.5] is the second objective function which intends to minimize the average of customer waiting time. Equation [3.6] ensures that the total time to finish the schedule does not exceed the working time period. Equation [3.7] and [3.8] guarantee that each customer is only visited once. Equation [3.9] makes sure that all of the vehicles is used during the tour. Equation [3.10] expresses the relationship between two decision variables.

3.3 System Architecture

Recent advance in information technology has motivated and allowed people to do real-time processing. By this technology, customers may order a service at any time during the working day. The dispatcher then should plan a schedule for the vehicles to serve the order from the customer. Due to the dynamic environment, the scheduling process should consider recent status and position of the vehicles before assigning them to the next schedule. This information can be acquired through mobile communication between dispatcher and the drivers.

Scheduling dynamic requests is a complex task. Thus, one needs to apply a strategy to handle the dynamicity of the problem. In this model, we incorporate an *event scheduler*. Event scheduler is a module which helps the dispatcher manage dynamic request from customer by transforming it into a sequence of static problems. This can be done by aggregating customer requests for a specified period and treats them as a static problem in the scheduling process.

One can specify the length of the period by dividing the working day T into a several *time slices* t_s. The length of time slice t_s is formulated by following equation:

$$t_s = \frac{T}{n_{ts}} \qquad [3.11]$$

Number of time slices n_{ts} is a parameter which is defined by the user. Montemanni et al. (2003) showed that large n_{ts} will make the system more responsive to the request but does not lead to satisfactory result because the optimization is restarted too often. On the other hand, small value of n_{ts} will make the system cannot take the advantage of new information. Therefore, the tuning parameter of n_{ts} should be carefully taken.

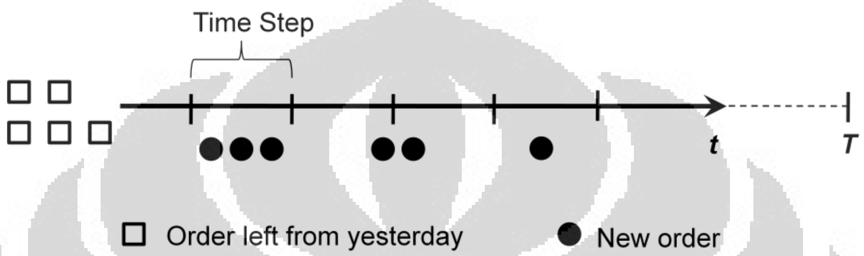

Figure 3.1 The concepts of time slice.

Figure 3.1 illustrates the concept of time slice. It divides the working day into several periods called time slices. In each of time slice, the problem will be similar to a static VRP but with different vehicle location and starting time. This static instance will be the basic input for the optimization process. The optimization will be run at the end of each time step for a limited computational time. Since the problem is highly constrained by the time, the system then should make estimation whether the remaining available time will be enough to serve all of the non-visited requests in the day. If it is, then the dispatcher will make a promise to the customer. Otherwise, the dispatcher will reject some of requests. In the real applications, the decision to reject customers is usually followed by an offer to serve them in the following day or to be served by other company.

This filter mechanism was not considered in the previous researches of DVRP, e.g. Montemanni et et al. (2003), Hanshar and Ombuki-Berman (2007), and Garrido and Riff (2010). It is because their researches were highly concerned to determine the best optimization method in DVRP instead of real-world application. This study proposes a model which combines the solution-driven algorithm by considering real-world application.

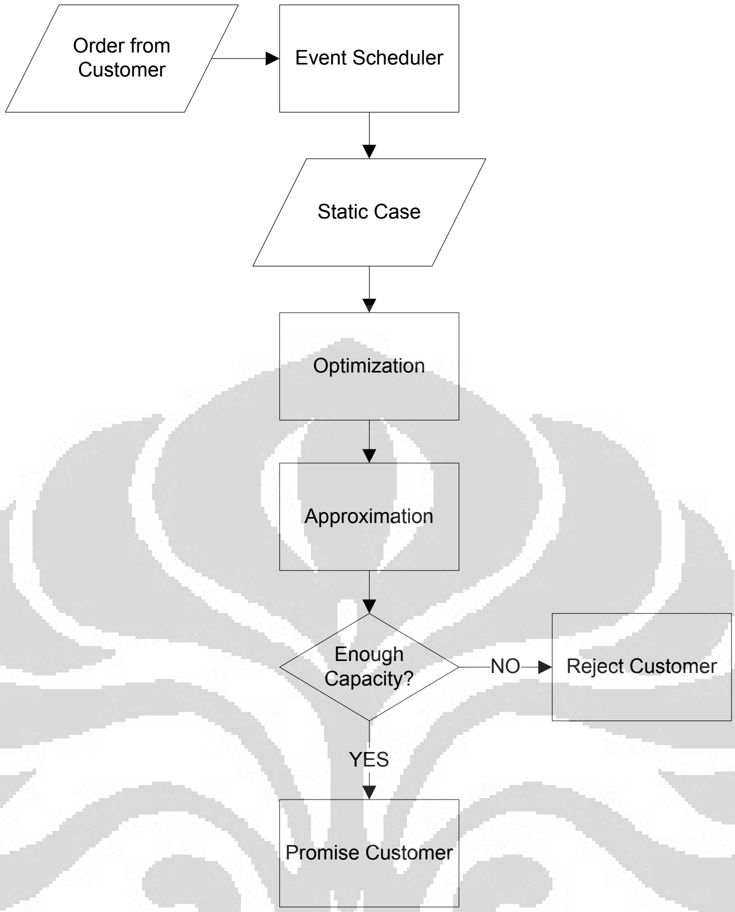

Figure 3.2 DVRP system flowchart

The dispatcher will assign the vehicle to serve the customer based on the planned schedule. It is assumed that once a vehicle is assigned to a dedicated customer, it cannot be diverted to another location. In every time slice, the drivers will report their status and condition to the dispatcher. This information will be used as consideration in planning the next schedule. Sometimes, the vehicle has already finished the entire schedule while there is still available time left before the working time end. In this case, we assume that the dispatcher will tell the driver to wait in the customer location until the next schedule exists or the working time has end. At the end of the working time, all of the vehicles should return to the depot. If there is any promised customer who has not been visited until the end of the working period, then it will be considered as a fail order. This failure indicates that the dispatcher has a bad estimation in the scheduling process. It is better to reject a customer than to promise them while in fact we cannot serve their requests.

```
Procedure Event Scheduler;
Set Time = 0;
Set all vehicles position at the depot;
PendingOrders = orders left from previous day;
While(Time ≤ T)
    StaticProblem = PendingOrders;
    Execute Fuzzy-ACS for StaticProblem;
    RejectedOrders = orders with estimated processing time >T;
    NewOrders = orders with arrival time between Time and (Time + TimeSlice);
    Time = Time + TimeSlice;
    CommitOrders = orders with actual processing time ≤Time;
    PendingOrders = PendingOrders \ CommitOrders + NewOrders;
    Update vehicle's starting position;
End While
Return all the vehicles to the depot;
```

Figure 3.3 Pseudo-code for event scheduler.

3.4 Single Aggregate Objective Function

From the problem description, it is known that the dispatcher is faced with a multi-objective problem. The first objective F_1 is to maximize the number of customers served while the second objective F_2 is to minimize the average customer waiting time. This study treats this multi-objective function by aggregating the multiple objectives into a single objective function:

$$\max \tilde{F} = \frac{\left[F_1\right]^{\omega_1}}{\left[\tilde{F}_2\right]^{\omega_2}} \qquad [3.12]$$

where ω_1 and ω_2 denote the relative importance of two objectives, respectively. These parameters are subjective based on the preference of the decision maker. If the first objective is more important than the second objective, then one can set the ω_1 higher than ω_2 and vice versa. Because the weight values are subjective, this study assumes that the weights have equal values, $\omega_1 = \omega_2 = 1$.

3.5 Fuzzy Approximation

The actual service time is only known after the vehicle finished the service operation at a customer. Therefore, the dispatcher should make approximation for the planning purpose. The proposed method treats the service time as fuzzy number before the actual value is known in certain. It is assumed that each service time in customer i can be represented as triangular fuzzy numbers, $\widetilde{U}_i = \{u_{1i}, u_{2i}, u_{3i}\}$, where u_{1i} is the left boundary of service time and u_{3i} is the right boundary of service time. These boundaries will give subjective approximation range to the dispatcher such that the next service time will not be less than u_{1i} or greater than u_{3i}. The value of u_{2i} corresponds to a grade of membership 1, which also can be determined by subjective estimation.

3.5.1 Adaptive Fuzzy Sets

In the static VRP, the membership of the fuzzy set is fixed and used for all the approximation process (Zheng and Liu, 2006; Teodorovic and Lucic, 2007; Cao and Lai, 2010). However, in dynamic problem, we can receive feedback after particular realization. Therefore we may update the fuzzy set based on our new knowledge. This adaptive procedure will adjust the approximation range value of service time for the next customer.

Figure 3.4 illustrates the procedure for updating the fuzzy set. In this procedure, the triangular fuzzy set is designed based on the available data. Every time the new data regarding the actual service time is known, it will be recorded in database and used as the basic input for designing the fuzzy sets.

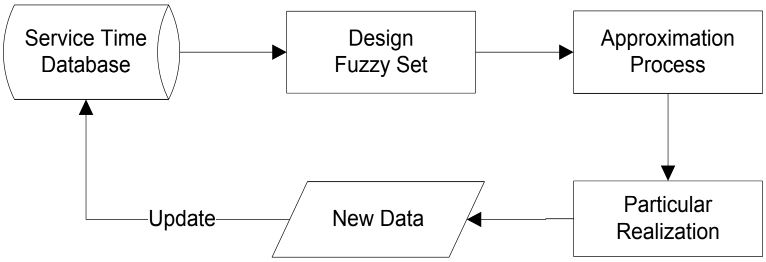

Figure 3.4 Procedure for updating fuzzy set.

Since the range of actual service time is uncertain, there is a possibility that it will go far away beyond the approximation range. This kind of extreme data will make the approximation range unstable. Therefore, in order to create a robust approximation range, we design the triangular fuzzy set based on *quartile* method. *Quartile* is a statistical method for summarizing data which give more robust measure in the presence of extreme value instead of *range* and *standard deviation*. The rules for designing the triangular fuzzy set are described as follows:

a) Left boundary u_{1i} is determined by the first quartile Q_1 of the database minus half of the inter-quartile range.

$$u_{1i} = Q_1 - 0.5(Q_3 - Q_1) \tag{3.13}$$

b) The value of u_{2i} is determined by the second quartile Q_2 (median) of the database.

$$u_{2i} = Q_2 \tag{3.14}$$

c) Right boundary u_{3i} is determined by the third quartile Q_3 plus half of the inter-quartile range.

$$u_{3i} = Q_3 + 0.5(Q_3 - Q_1) \tag{3.15}$$

3.5.2 Fuzzy Logic

In the approximation process, it is clear that increasing the number of customer served along the route will decrease the available remaining time. After serving n customer, the available remaining time, $\tilde{A}_n^k$, of vehicle k is defined as follows:

$$\tilde{A}_n^k = T - \sum_{i=0}^{n}\sum_{j=0}^{n}\left(e_{ij} + \tilde{u}_j\right) \qquad [3.16]$$

Since the service time is represented by triangular fuzzy numbers then the available remaining time is also a triangular fuzzy numbers $\tilde{A}_n = \{a_{1n}, a_{2n}, a_{3n}\}$ which are determined by:

$$\tilde{A}_n^k = \left(T - \sum_{i=0}^{n}\sum_{j=0}^{n}\left(e_{ij} + u_{3j}\right), T - \sum_{i=0}^{n}\sum_{j=0}^{n}\left(e_{ij} + u_{2j}\right), T - \sum_{i=0}^{n}\sum_{j=0}^{n}\left(e_{ij} + u_{1j}\right)\right) \qquad [3.17]$$

Therefore, we can state the fuzzy logic as follows:

IF the *remaining available time* is large and *service time* at the next node is short
THEN the *chance* to serve the next customer is big.

IF the *remaining available* time is small and *service time* at the next node is long
THEN the *chance* to serve the next customer is small.

From the statements above, we can conclude that the greater the difference between the available remaining time and the service time at the next customer, the greater is the chance to send the vehicle to serve the next customer.

3.5.3 Credibility Measure

The credibility theory developed by Liu (2004) is applied to measure the chance to serve the next customer. Denote the credibility of the fuzzy event by Cr, where $Cr \in [0, 1]$. When $Cr = 0$, the vehicle should terminate the tour and return to the depot. On the other hand, when $Cr = 1$, it can be completely sure that the vehicle is capable to serve the next customer within the working time period. The credibility that the travel time to next customer and its service time do not exceed the available remaining time is measured by following equations:

$$Cr = Cr\left\{e_{j,j+1} + \tilde{u}_{j+1} \le \tilde{A}_j\right\}$$

$$= Cr\left\{\left(\left[e_{j,j+1} + u_{1,j+1} - a_{3,j}\right],\ \left[e_{j,j+1} + u_{2,j+1} - a_{2,n}\right],\ \left[e_{j,j+1} + u_{3,j+1} - a_{1,j}\right]\right) \le 0\right\}$$

$$Cr = \begin{cases} 0 & \text{if } u_{1,n+1} \ge a_{3,n} \\ \dfrac{a_{3,n} - e_{j,j+1} + u_{1,n+1}}{2\left(a_{3,n} - u_{1,n+1} + u_{2,n+1} - a_{2,n}\right)} & \text{if } u_{1,n+1} \le a_{3,n}, u_{2,n+1} \ge a_{2,n} \\ \dfrac{u_{3,n+1} - e_{j,j+1} - a_{1,n} + 2\left(u_{2,n+1} - a_{2,n}\right)}{2\left(a_{2,n} - u_{2,n+1} + u_{3,n+1} - a_{1,n}\right)} & \text{if } u_{2,n+1} \le a_{2,n}, u_{3,n+1} \ge a_{1,n} \\ 1 & \text{if } u_{3,n+1} \le a_{1,n} \end{cases} \qquad [3.18]$$

where $e_{j,,j+1}$ represents the travel time between current location and the next location, $\tilde{u}_{j+1}$ denotes the triangular fuzzy service time in next location and $\tilde{A}_j$ represents the available remaining time at the current location.

3.5.4 Fuzzy Chance Constrained Programming

For the decision making purpose, we model the problem using fuzzy chance constrained programming. Let us denote the dispatcher preference index as Cr^* which indicates the decision maker behavior toward the risk. The decision maker will choose lower value of Cr^* when he/she is a risk taker. This indicates the preference to use the remaining time as much as possible, even though there is a chance where the vehicle fails to serve the promised customer. When the decision maker chooses the higher value of Cr^*, it indicates that he/she likes to play safe and prefer to avoid risk even there is a chance to utilize the available remaining time.

Therefore the dispatcher can make the decision to serve the next customer based on the credibility and the preference index. If $Cr \ge Cr^*$ holds then the vehicle is sent to the next customer, otherwise the vehicle is returned to the depot due to the working time limit. By using this approach, we can substitute the problem constraint in equation [3.5] by the following equation:

$$Cr\left(\sum_{i=0}^{n}\sum_{j=0}^{n}\left(e_{ij}^{k} + \tilde{u}_{j}^{k} \right) x_{ij}^{k} \le T \right) \ge Cr^{*} \quad \forall k = 1, 2.., m \qquad [3.19]$$

3.6 Hybrid Fuzzy-Ant Colony System

According the problem constructed in the above subsection, this study proposes the hybrid of fuzzy set theory and Ant Colony System (ACS) algorithm to solve the problem. The characteristic of ACS which sequentially constructs a solution is naturally suitable for fuzzy logic implementation. In this ACS, a group of artificial ants will construct the solution by visiting node in the graph one by one. Every time an ant visits a node, it will compute the chance to serve the next node by using fuzzy approximation. The value of the chance will be used to make a decision whether to commit a request or even reject it to satisfy the problem constraints. The implementation of Fuzzy-ACS is described in details in the following subsection.

3.6.1 Initial Trail

Initial trail plays important role to enhance the performance of ACS. In this model, the initial trail is created heuristically by sequential insertion method. This study proposes modification to the cheapest insertion heuristics by combining it with clustering method. Consider *m* number of empty routes. The algorithm will construct the route by sequentially inserting non-routed customer to the closest route in random sequence. The closest route is determined by the Euclidian distance between non-routed customer and centroid of the routes. The centroid of routes, ctd_k, is defined as the average of all customer locations in the route *k*. It is formulated as follows:

$$ctd_k = \left\{ \frac{\sum_{i=1}^{n_k} x_i}{n_k}, \quad \frac{\sum_{i=1}^{n_k} y_i}{n_k} \right\} \qquad [3.20]$$

where x_i and y_i denote the location coordinates of customer i, and n_k represents number of customers in route k. Once the centroid is known, one can compute the Euclidian distance dt_{ik} between new customer i and the centroid in route k by using following equation:

$$dt_{ik} = \sqrt{\left(x_i - x_{ctd}^k\right)^2 + \left(y_i - y_{ctd}^k\right)^2} \qquad [3.21]$$

The customer is inserted to the closest route and positioned within an edge in the route which minimizes the total cost of the route as formulated in equation [2.11]. The insertion process is done by considering capacity constraint in each route. Thus, the customer will not be inserted to the route which has reached its capacity limit. Afterwards, the whole process is repeated until all the known customer is scheduled in the route.

This heuristic solution is used as the initial trail for guiding the ant at its first tour. It is done by depositing an amount of pheromone to every linked edge in the initial solution. Hence the ant will use it as consideration for constructing the next solution in the optimization process. Figure 3.6 illustrates the insertion process of new customer using clustered insertion method.

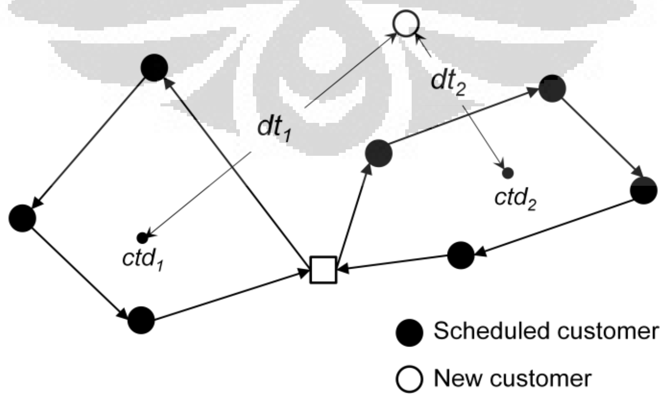

Figure 3.5 Inserting new customer to the closest route.

```
Procedure Clustered Insertion;
Set number of route = m;
Random shuffle PendingOrders;
For each PendingOrders i =1,2… n;
    STEP 1: Approximate route feasibility to receive new customer
        For each route k = 1, 2 … m
            Calculate Cr_k with Equation[3.17];
            If(Cr_k ≥ Cr*)
                route k = feasible;
            End
        End
        If (FeasibleRoute = ∅)
            Continue;
        End

    STEP 2: Calculate the distance between customer i and centroid of FeasibleRoute g
        For each FeasibleRoute g = 1, 2 … h
            Ctd_g = equation [3.20];
            dt_ig = equation [3.21];
        End

    STEP 3: Define route with the shortest distance to customer i
    STEP 4: Define the cheapest position in the route using equation [2.11];
    STEP 5: Insert the customer to the cheapest position;
End
```

Figure 3.6 Pseudo-codes for clustered-insertion method.

3.6.2 Solution Construction

In the basic ACS, one ant will create one solution and divide the tour by capacity constraint. In this problem, the solution should have a certain number of routes since we only operate a specified number of vehicles in the day (see equation [3.8]) Therefore, the basic approach in the ACS which terminates the route after it reaches the capacity limit will not suitable for this problem because the generated solutions may have different number of routes. Besides, in dynamic problem, each route contains specified information about vehicle last position and starting time. Therefore, we need to adjust the basic ACS so it can deal with the characteristic of the problem.

In the proposed approach, a solution will be constructed by a group of ants instead of one (see figure 3.7). Each ant in the group will represent a vehicle and sequentially construct the solution by selecting the next node in turn. If an ant has selects a node to be visited in its route, the other ant cannot choose it for its route. Hence an ant only can choose the unselected node to construct its route. This process is repeated until no available node left or the route reaches its capacity limit.

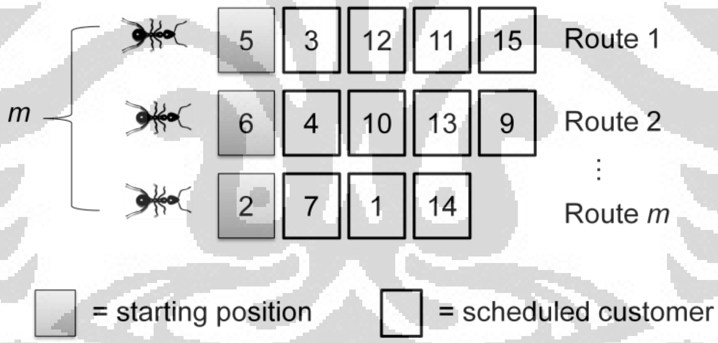

Figure 3.7 Solution representation.

By using this approach, we can assign the customer requests such that every vehicle has roughly equal work load for each of time slice. This consideration is important in DVRP since the problem is highly time constrained. Thus, unbalanced workload will make the capacity unutilized and create inefficiency in the schedule. Besides, this approach also allows us to keep the last information in the current step and pass it into the next step. Therefore, in each of time slice, the algorithm will optimize the schedule based on the passed information.

In order to simplify the algorithm, the pseudo random rule in equation [2.17] is modified as the following equation:

$$j = \begin{cases} \arg\max\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}, & \text{if } q < q_0 \\ \text{random } (j \in \Omega), & \textit{otherwise} \end{cases} \qquad [3.22]$$

where τ_{ij} denotes the pheromone level and η_{ij} denotes the attractiveness. This rule will choose the best considered node as the next customer with a probability q_0 and will select a node randomly with a probability $(1 - q_0)$. By a pure random selection, it is intended to give bigger chance for the ants to explore the space solution in the iteration process.

3.6.3 Pheromone Update

Pheromone update is one of the main features in ant colony optimization. By updating the pheromone, we can pass the information from the previous solution to the next. This procedure will intensify the search process in the potential solution.

This model follows *Ant System* concept (Gambrella and Dorigo, 1996) which incorporates global and local updates. Local update is executed every time an ant completes a solution, while global update is performed only after all ants complete the tour. In global update, only the best solution in the colony is used for the updating process. Therefore in the next iteration, the first group of ants will only use the information from the previous best solution.

The initial pheromone is set with a constant Q. After that, the pheromone matrix, τ_{ij}, is updated by using the following rule:

$$\tau_{ij} = (1 - \rho).\tau_{ij} + Q \qquad [3.23]$$

where ρ denotes the evaporation rate of the pheromone. The bigger the rate, the bigger chance is the ants to forget the previous trail. This rate will maintain the ants so they will not get trapped into local optima so fast.

3.6.4 Local Search

For speeding up the exploitation of search space, we apply *random swap* local search every time the ants complete a solution. This local search will randomly select two points in a route, i and j with $i \neq j$ and then swap the node's position. If the new position of nodes results in better solution, they will replace the old solution. This process is repeated until the local search cannot improve the solution for a number of iteration.

```
Procedure Random Swap;

Foreach route k = 1,2, .. m
    Set NonImprove = 0;
    While (NonImprove ≤ MaxNumber)
        Select two points i and j in the route randomly, where i ≠ j;
        Swap the position;
        Compute NewFitnessValue;
        If (NewFitnessValue < OldFitnessValue)
            Replace OldSolution with NewSolution;
        Else
            NonImprove = NonImprove + 1;
    End While
End For
```

Figure 3.8 Random swap pseudo-codes

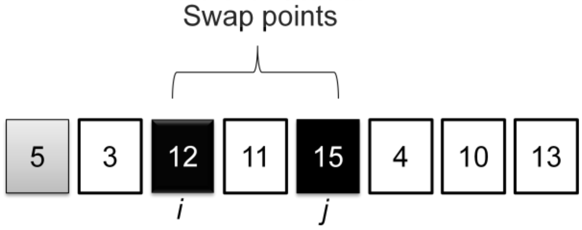

Figure 3.9 Random swap local search.

3.6.5 Fitness Value

In ACS, the best solution is found by comparing the fitness value of the solution in each of iteration. Because the fitness value is in fuzzy numbers, we need to transform it first into a crisp value so it can be used for comparison. This study follows Brito et al. (2009) to use the Third Yager's index to convert the fuzzy number into a crisp value (see equation [2.25]). If each fuzzy waiting time $\tilde{w}_i^k$ is a triangular fuzzy number $Tr(w_i^{1k}, w_i^{2k}, w_i^{3k})$, then we can replace the fuzzy objective function in equation [3.2] with the following equation:

$$\min F_2 = \frac{\sum_{i=1}^{n}(w_i^{1k} + 2w_i^{2k} + w_i^{3k})}{4n} \qquad [3.24]$$

Thus, for comparison purpose, the fuzzy aggregate objective function in equation [3.11] can also be transformed into a crisp value:

$$\max F = \frac{[F_1]^{\omega_1}}{[F_2]^{\omega_2}} \qquad [3.25]$$

3.6.6 Fuzzy-ACS Procedure

This section will describe the complete Fuzzy-ACS procedure in pseudo-code form as shown in Figure 3.11.

Procedure Fuzzy-ACS;

Set up Fuzzy-ACS parameter {q_0, ρ, α, β};
Get static case $S = \{1,2,3..\ n\}$ from *event scheduler;*

Set pheromone $\tau_{ij} = Q$, *where i and j* $\in S$;

Create initial solution X_0 using *clustered-insertion* method;

Update pheromone τ_{ij} using equation [3.23], where: *i and j* $\in X_0$;

Set $X_{best} = X_0$; $F_{(Xbest)} = F_{(X)}$;

Set CPU time = 0;

```
While (CPU Time≤maxTime)
    For each group of ants
        List of available nodes, S = {1,2,3 … n};
        While ((S≠ ∅) ∪(a≤fleet size))
            set a = 0;
            For each ant
                Starting node = i;
                Select next node j (j ∈ S) based on equation [3.22];
                Compute the credibility of each ant k to visit node j;
                If (Cr_k ≥ Cr*)
                    next node = j;
                    S = S\{j};
                    Update starting node, i = j;
                Else
                    a = a + 1;
                    continue;
            End For
            Update local pheromone based on equation [3.23];
            Apply random swap local search;
            Calculate fitness value F_(X);
            If (F_(X)≤ F_(Xbest))
                F_(Xbest) = F_(X) ;
                X_best= X ;
            End If
            Update global pheromone based on equation [3.23];
        End While
    End For
    Update CPU Time;
End While
Return the best solution;
```

Figure 3.10 Pseudo-codes of Fuzzy-ACS.

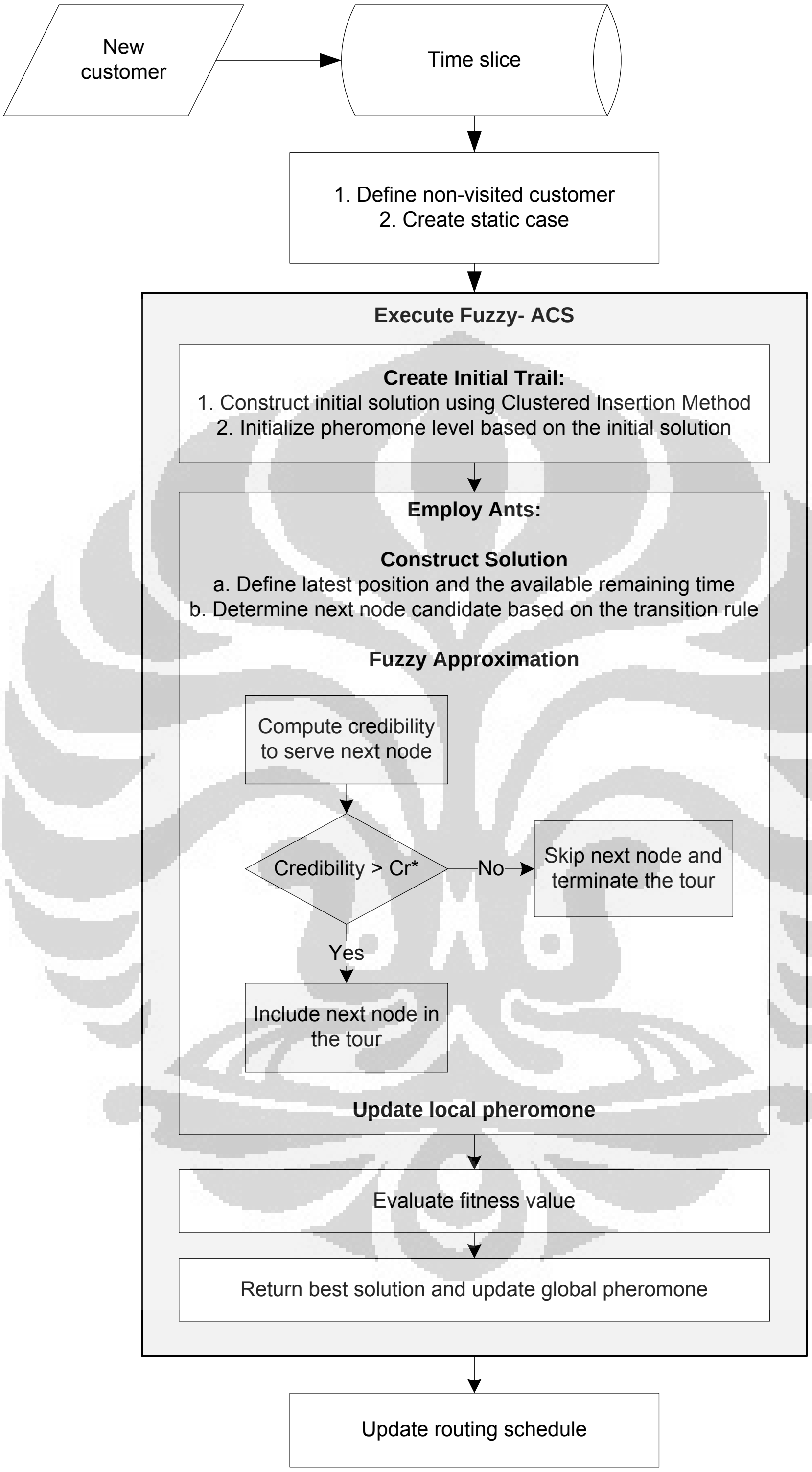

Figure 3.11 Flowchart of the proposed method.

Chapter 4
Computational Experiences

This chapter will perform computational experiment for DVRP model which has been presented in the previous chapter. The purpose of the experiment is to examine the empirical behavior of certain parameters and solution method. The model was coded in C++ language using CodeBlocks 10.05 on a PC with Intel Core i5 processor and 2 GB RAM. The detailed experimental results will be described in the following sections.

4.1 Data and Parameters

The data in this thesis is generated by modifying the datasets from Montemanni et al. (2003). Their datasets are originally from the classic CVRP benchmark instances of Christofides et al. (1979), Taillard (1994) and Fisher et al. (1996). These instances have typical data types which consist of:

1) number of the customers,
2) depot and customer location,
3) customer demand,
4) vehicle capacity, and
5) maximal number of vehicles used.

In order to generate the dynamicity in the problem, Kilby et al. (1998) and Montemanni et al. (2003) add three new data types to the benchmark data sets:

1) arrival time of customers,
2) duration of each visit, and
3) working day period.

These characteristics of datasets are basically different with our problem. It is necessary to modify the data so it can represent the model as described in Chapter 3. The proposed DVRP model employs vehicles to provide on-site service for the customers instead of delivering goods. Thus, customer demand and vehicle capacity in the datasets can be ignored. However, since the actual service time for each customer is unique, it is necessary to modify the visit duration data in the instance. This is done by replacing the visit duration data by customer demand data. Therefore, each customer will have different actual service time.

The other characteristic is that our problem implies limited vehicle number instead of unlimited one. Data type regarding number of vehicles then needs to be added to the instance. Besides, historical data for service time should also be added for the approximation process. Ten data was picked randomly from the actual service time to be assumed as historical service time data. This historical data is used as foundation for designing approximation range for customer service time.

Thus, based on those modifications, the instances now have the following data types:

1) number of the customers,
2) arrival time of customers,
3) depot and customer location,
4) service time duration,
5) working day period, and
6) historical service time.

C50 instance from Montemanni et al. (2003) dynamic datasets is treated as the raw data. The instance has 50 customers which uniformly distributed in a square location. The arrival time of customers are randomly distributed during the working day period (12 hours). Figure 4.1-4.3 illustrates the main characteristics of the problem. Based on that dataset, one can generate many different cases by varying the degree of dynamism in the problem (Larsen et al., 2002). Besides, the predetermined decision in number of vehicles, time limit for optimization, and dispatcher preference index Cr^* will also make the problem setting different. The details of the data can be seen in Appendix A.

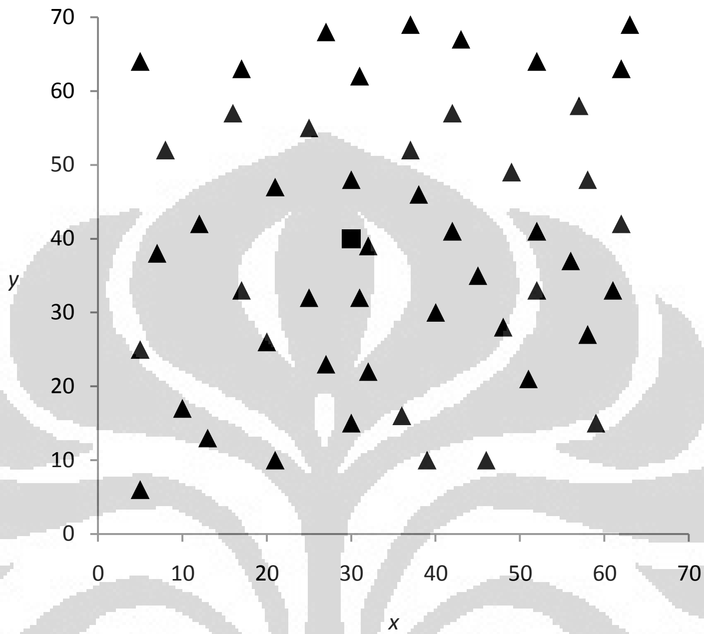

Figure 4.1 Distribution of customer locations in Cartesian diagram.

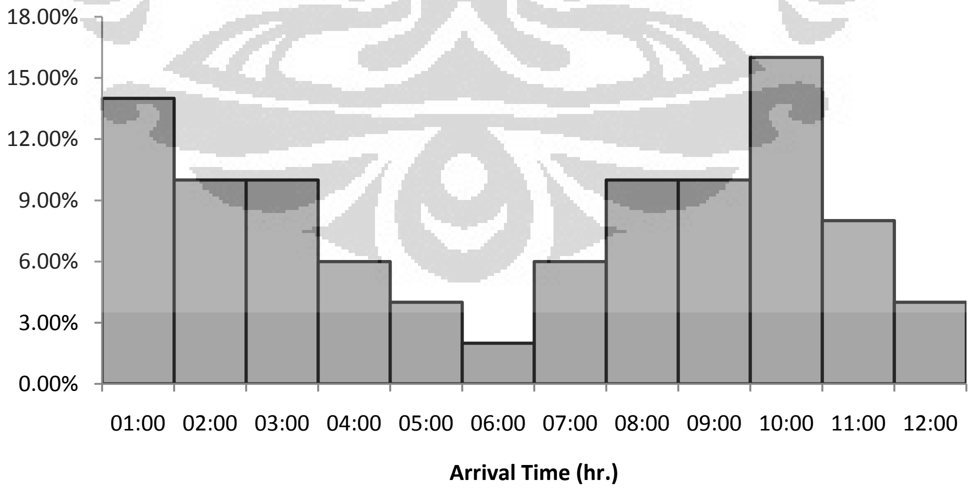

Figure 4.2 Distribution of customer arrival time in full dynamic scenario.

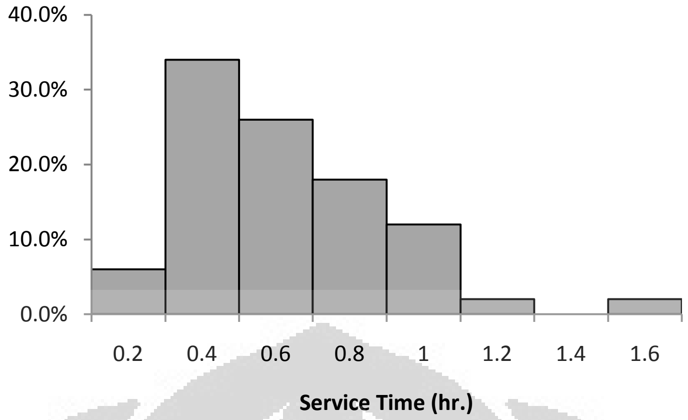

Figure 4.3 Distribution of actual customer service time.

4.2 ACS Parameter Tuning

Each metaheuristic has a set of predetermined parameters which has to be initialized before execution. These parameters will control the balance between exploitation and exploration process in the search space. An appropriate parameter setting will give significant impact on the quality of solution. Unfortunately, there are no universal initial parameters setting to solve all optimization problems (Dobslaw, 2010). Thus, one needs to tune the parameter values for each different type of problems.

In ACS, there are four parameters which need to be tuned:

1) number of ant groups m,
2) pheromone evaporation rate ρ,
3) selection pressure q_0, and
4) relative importance between α and β.

This study applies *factorial experimental design* to tune the parameter values of ACS. Factorial design is an experiment whose design consists of several factors, each with two or more levels. The design will take all possible combinations of those levels across all factors as its experiment unit.

The objective of this experiment is to discover the effects of the multiple parameters on the solution quality. By knowing the effects, the appropriate combination of the parameter values for the algorithm can be determined. The critical values of the parameters can be obtained from the previous researches which used the same method (Adenso-Diaz and Laguna, 2006). The experiment design can be seen in Table 4.1. There are three factors with two levels and one factor with three levels. Hence, $2^3 \times 3^1 = 24$ experiments are required. For every set of experiment, 30 independent replications were performed.

Table 4.1 Experiment design for ACS parameter tuning.

Factor	Level	References
m	3	Montemanniet al. (2003)
	10	Dorigo and Gambardella (1997), Gambardella et al. (1999)
ρ	0.1	Dorigo and Gambardella (1997), Gambardella et al. (1999)
	0.3	
q_0	0.7	
	0.9	Gambardella et al. (1999)
$(\alpha - \beta)$	-1	Dorigo and Gambardella (1997)
	0	Montemanniet al. (2003), Gambardella et al. (1999)
	1	

Since the objective of the experiment is to find the best parameter setting for the algorithm, it is necessary to minimize the existence of any other factors which potentially affect the model response. For this reason, static case (*dod* = 0) was preferred for the experiment to avoid the dynamicity effect. Besides, large number of vehicles is applied in order to make sure that the approximation capability will not affect the model response.

However, the algorithm is originally intended to handle the dynamic problem. Thus, it is assumed if the algorithm can produce good quality solution in a static problem, then it can also produce good quality solution in the dynamic problem. The reason is that it can be considered as a sequence of static cases. The setting of the problem for this experiment is described in Table 4.2.

Table 4.2 Problem setting for ACS parameter tuning.

Problem Setting	Value
*Cr**	0.6
Number of Vehicles	10
Degree of dynamism	0
Number of time slices	12
Computational time limit	10 seconds

The factorial design assumes linear relationship between the parameters. The effect model of the experiment is formulated in the following equation:

$$
\begin{aligned}
y_{ijkl} = {} & \mu_{ACS} + m_i + \rho_j + \left(q_0\right)_k + (\alpha-\beta)_l \\
& + m_i\rho_j + m_i\left(q_0\right)_k + m_i(\alpha-\beta)_l + \rho_j\left(q_0\right)_k + \rho_j(\alpha-\beta)_l + \left(q_0\right)_k(\alpha-\beta)_l \\
& + m_i\rho_j\left(q_0\right)_k + m_i\rho_j(\alpha-\beta)_l + m_i\left(q_0\right)_k(\alpha-\beta)_l \\
& + m_i\rho_j\left(q_0\right)_k(\alpha-\beta)_l + e_{ijkl}
\end{aligned} \qquad [4.1]
$$

In order to find out the significance of the change in parameter value, a statistical test was performed. The testing hypothesis for the experiment is formulated as follows:

$$
\begin{aligned}
& H_0 : Y_i = 0 \\
& H_1 : \text{ at least one } Y_i \neq 0;
\end{aligned} \qquad i = 1, 2, 3 \ldots n \qquad [4.2]
$$

where *Y* represents the factors considered in the experiment and *n* represents the number of combination unit in the experiment design. Analysis of variance (ANOVA) is employed to evaluate the hypothesis. All of the statistical computation was accomplished using MINITAB 14 software.

Figure 4.4 illustrates the ANOVA result. It shows that all of the main parameters in ACS have significant effect on the solution quality (p = 0.000). The interaction plot in Figure 4.5 also shows that combination of high value in m, ρ, and β parameters will give better solution to the model. Therefore, from these results, the best combination of parameter values for ACS is as follows: m = 10, ρ = 0.3, q_0= 0.9, α = 1, and β = 2. The rest of experiments will apply this setting for the ACS parameters.

```
Analysis of Variance for Wait_Time, using Adjusted SS for Tests

Source             DF    Seq SS    Adj SS    Adj MS        F      P
m                   1    458.29    458.29    458.29   110.81  0.000
rho                 1     19.18     19.18     19.18     4.64  0.032
q0                  1   5557.22   5557.22   5557.22  1343.66  0.000
 (A-B)              2     32.56     32.56     16.28     3.94  0.020
m*rho               1      0.94      0.94      0.94     0.23  0.634
m*q0                1      2.46      2.46      2.46     0.59  0.441
m* (A-B)            2     23.69     23.69     11.84     2.86  0.058
rho*q0              1     33.71     33.71     33.71     8.15  0.004
rho* (A-B)          2     38.59     38.59     19.29     4.66  0.010
q0* (A-B)           2     13.85     13.85      6.93     1.67  0.188
m*rho*q0            1      7.65      7.65      7.65     1.85  0.174
m*rho* (A-B)        2     31.49     31.49     15.75     3.81  0.023
m*q0* (A-B)         2      3.50      3.50      1.75     0.42  0.656
rho*q0* (A-B)       2     23.82     23.82     11.91     2.88  0.057
m*rho*q0* (A-B)     2     10.33     10.33      5.17     1.25  0.287
Error             696   2878.58   2878.58      4.14
Total             719   9135.86


S = 2.03369    R-Sq = 68.49%    R-Sq(adj) = 67.45%
```

Figure 4.4 ANOVA result for the experiment.

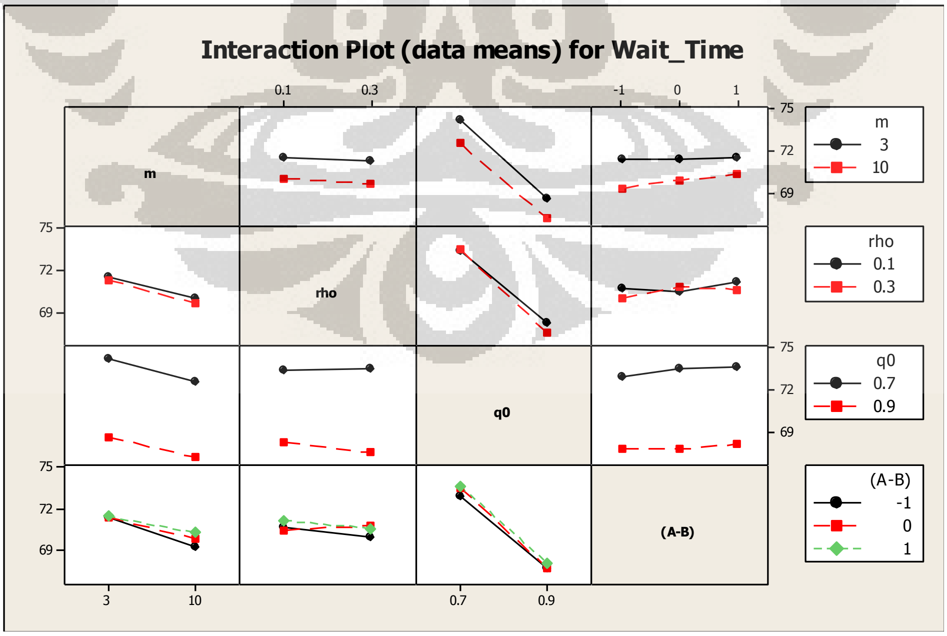

Figure 4.5 Parameters interaction plot.

4.3 Model Performance

4.3.1 Clustered-Insertion Method

Insertion heuristic is a common preferred method for constructing initial solution in DVRP. The reason is because the insertion heuristic is naturally suitable for dynamic environment. It sequentially inserts unrouted customer at the position which generates smallest cost in the tour (see equation [2.11]). Insertion method has been used in Kilby et al. (1998), Gendrau et al. (1999), Montemanni et al. (2003), and Garrido and Riff (2010) to solve DVRP.

In the previous chapter, a modification for insertion heuristics called *clustered-insertion* method has been proposed. It combines insertion method with clustering technique. In order to examine the performance, the proposed method is tested against some instances from Christofides et al. (1979). Since the effectiveness of insertion heuristics is largely influenced by the order of insertion (Kilby et al., 1998), the condition for the experiment is set in two different settings: fixed order and random order. In the fixed order, the requests are inserted sequentially from the farthest to the nearest position of the depot, while in random order, the customers are simply inserted in random sequence. The computational results are shown in Table 4.3.

Table 4.3 Insertion method comparison result.

Instance	Best Known Solution	Clustered Insertion Method			Deviation (%)
		Order	Average	Best	
c50	524.61	Fixed	573.65	573.65	8.55%
	524.61	Random	697.32	595.7	11.93%
c75	835.26	Fixed	995.832	995.83	16.12%
	835.26	Random	1166.98	1066.58	21.69%
c100	826.14	Fixed	942.29	942.29	12.33%
	826.14	Random	1111.81	1018.08	18.85%
c150	1028.42	Fixed	1298.6	1298.6	20.81%
	1028.42	Random	1633.24	1500.22	31.45%
c200	1291.29	Fixed	1650.75	1650.75	21.78%
	1291.29	Random	2085.78	1885.36	31.51%

The computational results in table 4.13 indicate that the proposed clustered-insertion method is able to give good initial solution. The best results of the proposed method gives an average deviation 15.92% to the benchmark's best known results. The results also show that the proposed method performs better in a fixed order where the customer is inserted from the farthest to the nearest position of the depot.

4.3.2 Fuzzy-ACS

In this section, Fuzzy-ACS is tested against dataset in section 4.1. The basic setting for simulation is shown in table 4.4. We compared two implementation of Fuzzy ACS with different constructive heuristics to create the initial trail. The first one used clustered-insertion method while the second one used Kilby et al.'s insertion method without post-optimization procedure.

Table 4.4 Problem setting for Fuzzy ACS experiments.

Problem Setting	Value
*Cr**	0.6
Number of Vehicles	5
Number of time slices	12
Computational time limit	10 seconds

Two types of experiments were also created to examine the algorithm performance. The first experiment was executed in static environment (dod=0) with 50 customers. It observed the algorithm performance in each of the time step where a static case is created. Since the computational time in DVRP is highly restricted, a specified time was considered as the stopping criterion for the algorithm. In this experiment, each of algorithms were run for 10 seconds and replicated for 30 times. The best results were used to draw the evolution of the solution obtained in optimization process as shown in figure 4.6.

Figure 4.6 shows that Fuzzy ACS with clustered insertion method was able to give better result than Kilby et al.'s method. Since the algorithms were run in limited computational time, initial solution will play important role in the optimization process.

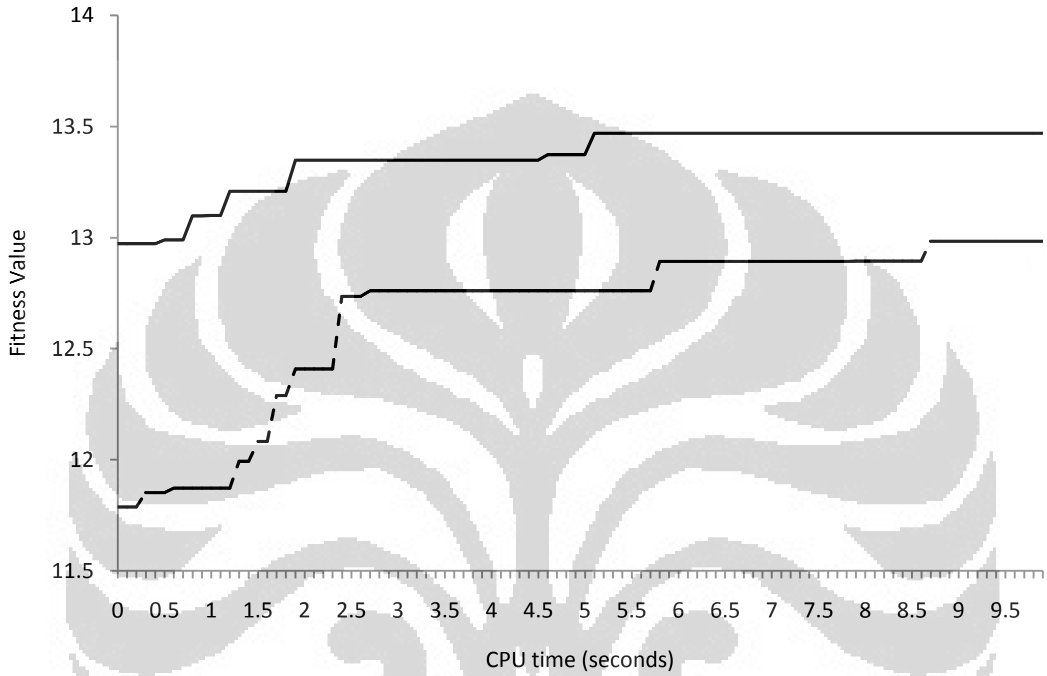

Figure 4.6 Algorithm performances in static instance.

The second experiment is aimed to examine the algorithm performance in different degree of dynamism (dod). We run each of the algorithms in five different dod setting. Every experiment unit was run for 30 replications. The result summary can be seen in table 4.5. It shows that the combination of Fuzzy-ACS and clustered insertion method in dynamic instances is also superior to the combination of Kilby et al.'s method. This result gives evidence that clustered insertion method can give good initial trail for ACS either in static or dynamic instances.

Table 4.5 Solution comparison in dynamic instances

dod	Result	Clustered Insertion method + Fuzzy ACS				Kilby et al.'s method + Fuzzy-ACS			
		Customers Served (F1)	Avg. Waiting Time (F2)	Failed Orders	$\frac{[F_1]^1}{[F_2]^1}$	Customers Served (F1)	Avg. Waiting Time (F2)	Failed Orders	$\frac{[F_1]^1}{[F_2]^1}$
0	Average	50	4.01	0	**12.47**	50	4.04	0	12.38
	Best	50	3.83	0	**13.48**	50	3.88	0	12.89
0.25	Average	50	3.97	0	**12.59**	50	3.92	0	12.76
	Best	50	3.81	0	**13.12**	50	3.83	0	13.04
0.5	Average	50	2.89	0	**17.32**	50	2.89	0	17.3
	Best	50	2.81	0	**17.78**	50	2.82	0	17.73
0.75	Average	48.3	2.26	0	**21.48**	48.8	2.31	0	21.23
	Best	50	2.1	0	**23.81**	49	2.1	0	23.36
1	Average	37.53	1.52	2	**24.67**	35.3	1.55	2	22.83
	Best	38	1.44	2	**26.31**	36	1.47	2	24.45

4.4 Sensitivity Analysis

This section provides sensitivity analysis of the parameter in the problem. It includes number of time slices, degree of dynamism, and credibility preference index. The purpose is to examine the behavior of the model under different parameter setting.

4.4.1 Number of Time Slices

Number of time slice will determine how frequent the optimization is executed during the working day. In this experiment, different numbers of time slices are simulated in order to examine its effects on the performance. The basic setting of the problem for the experiment is listed in Table 4.6. Each experiment was run for 30 replications. The experiment results are summarized in Table 4.7.

Table 4.6 Problem setting for *time slices* experiments.

Problem Setting	Value
*Cr**	0.6
Number of Vehicles	5
Degree of dynamism	0.5
Computational time limit	10 seconds

Table 4.7 Summary of computational results for *time slices* experiments.

Number of time slices	6	12	24	48
Number of customers served	50	50	50	50
Averagewaiting Time (hr.)	2.96	2.88	2.85	2.84
Standard deviation (hr.)	0.06	0.04	0.04	0.05

ANOVA was performed to examine the experiment results. The testing hypothesis is as follows:

$$\begin{aligned} &H_0 : y_1 = y_2 = y_3 = y_4 = 0 \\ &H_1 : \text{ at least one } y_i \neq 0 \quad i = 1,2,3,4 \end{aligned} \qquad [4.3]$$

The ANOVA result in Figure 4.7 indicates that there is a significant difference between the responses in the experiment (p = 0.000). From this result, we can conclude that the decision in number of time slices will affect the solution quality. Therefore, it needs to be tuned carefully. This result also confirms the previous research conclusion in Montemanni et al. (2003). The researchers stated that the careful tuning of time slice number can lead to better result. If the number of time slices is too small, then the optimization process cannot take the advantage of new information. Therefore, the rest of experiments will use n_{ts}= 12 as the basic setting. This setting will make the optimization process run every 1 hour in the actual time. During this period, the system will aggregate all the received information and create a static instance for the scheduling purpose.

```
One-way ANOVA: Wait_Time versus Nts

Source   DF      SS     MS      F      P
Nts       3  216.59  72.20  38.49  0.000
Error   116  217.58   1.88
Total   119  434.17
```

Figure 4.7 ANOVA result for number of time slices experiments.

4.4.2 Degree of Dynamism

Degree of dynamism (*dod*) is a criterion to measure dynamicity in a problem. It is defined as a fraction of dynamic request to the total of request in a day (see equation [2.8]). The more dynamic the requests, the more complex is the problem to be solved. Thus, this section will analyze the effect of *dod* on the solution quality.

The current experiment sets the problem into five different scenarios of *dod*. Each scenario has the same workload but with different customer distribution (see Appendix B for details). In static scenario (*dod* = 0), all orders are known in the beginning of the day. In partially, dynamic scenario (e.g. *dod* = 0.25, 0.5, 0.75), some of the orders are known from the beginning of the day, while some others are coming randomly during a fraction of working day. The basic setting for the experiment is shown in Table 4.8. Each experiment was run for 30 replications. The experimental result is summarized in Table 4.9.

Table 4.8 Problem setting for *dod* effect experiments.

Problem Setting	Value
*Cr**	0.6
Number of Vehicles	5
Number of time slices	12
Computational time limit	10 seconds

Table 4.9 reveals that the more dynamic the problem, the worse is the performance in terms of average travel time and number of customer served. The algorithm performs worst at of full dynamic case. In that case, only a fraction of total requests can be served. Besides, some promised customers also fail to be served. These results confirm the previous research conclusion in Kilby et al. (1998) and Larsen et al. (2002), which stated that increasing the level of dynamicity resulted in a linear increase in route cost. In addition, the experimental result also illustrates that the average waiting time decreases as the dynamicity of the problem increases. The reason is because, in dynamic environment, the dispatcher can take the advantage of late customer arrival to shorten the average waiting time of the customers. From this result, it can be concluded that our proposed method is suitable for the dynamic problem since it can properly take the advantage of the new information in a dynamic problem.

Table 4.9 Summary of computational result in *dod* effect experiments.

dod	Result	Customers Served	Fail Orders	Avg. Waiting Time (Hr.)	Avg. Travel Time (Hr.)
0	Average	50	0	4.01	0.328
	Best	50	0	3.83	0.334
0.25	Average	50	0	3.97	0.343
	Best	50	0	3.81	0.334
0.50	Average	50	0	2.88	0.391
	Best	50	0	2.81	0.392
0.75	Average	48.3	0	2.25	0.550
	Best	50	0	2.09	0.562
1	Average	37.53	2	1.52	0.592
	Best	38	2	1.44	0.568

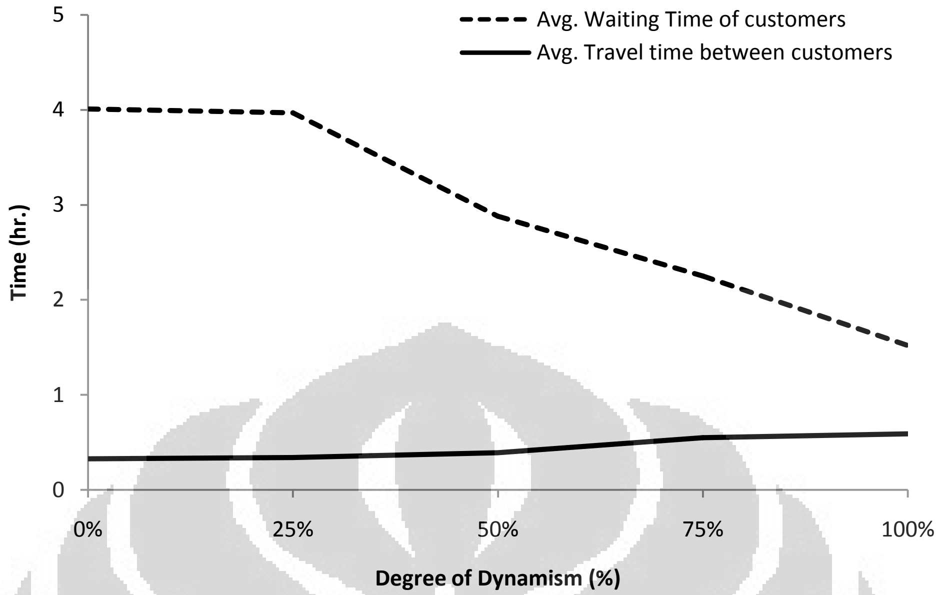

Figure 4.8 Performance behaviors for different degrees of dynamism.

Since higher dynamicity will worsen the system performance, the dispatcher could also consider for applying cut-off time strategy like in Montemanni et al. (2003). This strategy will give a limit to the system such that it will not accept any other requests after a certain time of period. Therefore, the dispatcher will not have to take a risky decision in the critical time. The other strategy is the dispatcher could also use higher preference index in the scheduling process. It will make the system more likely to avoid the risk. Therefore, any request which has a higher risk of failure will be rejected from the schedule.

4.4.3 Credibility Preference Index

Credibility preference index *Cr** is a subjective parameter value between 0 and 1 which indicates the dispatcher behavior toward the risk. As stated in the previous chapter, low value of *Cr** means that dispatcher is a risk taker which prefers to utilizing the remaining time as much as possible even though there is a chance of failure. High value of *Cr** means that the dispatcher prefers to avoiding the risk even though there is still an available time to be utilized.

Intuitively, the dispatcher will have low *Cr** value when they believe that their capacity is more than enough to handle all of the requests in the day. On the other hand, the dispatcher will have high *Cr** when they think that their capacity is not large enough to handle all the dynamic requests. From here, we can see a correlation between dispatcher preference index and capacity.

To examine the correlation between the preference index and capacity, an experiment which involves both of factors is conducted. The capacity in the problem is determined by the number of vehicles. The more vehicles used in the day, the more capacity is the company has. Table 4.10 illustrates the experimental design. Each experiment was run 30 replications. Table 4.11 presents the summarized result, while ANOVA results are shown in Figure 4.9.

Table 4.10 Design for *Cr** experiments.

Factors	Level			
	1	**2**	**3**	**4**
Number of Vehicles	3	4	5	6
Cr*	0.3	0.6	0.9	-

Table 4.11 Summary of computational result in *Cr** experiments.

Number of Vehicles	Credibility Preference Index								
	0.3			**0.6**			**0.9**		
	Customers Served (F_1)	Avg. Waiting Time (F_2)	$\frac{[F_1]^1}{[F_2]^1}$	Customers Served (F_1)	Avg. Waiting Time(F_2)	$\frac{[F_1]^1}{[F_2]^1}$	Customers Served (F_1)	Avg. Waiting Time(F_2)	$\frac{[F_1]^1}{[F_2]^1}$
3	37.83	3.89	9.73	33.80	3.20	10.57	30.00	2.66	11.28
4	49.40	3.80	13.00	45.07	3.33	13.54	40.53	2.87	14.11
5	50.00	2.88	17.38	50.00	2.88	17.37	48.97	2.77	17.67
6	50.00	2.25	22.26	50.00	2.26	22.12	50.00	2.25	22.27

From the ANOVA result in Figure 4.9, we can see that number of vehicles greatly affects the quality of the solution (p =0.000). The more vehicles, the better are the solution quality. The reason is because when the dispatcher has a larger capacity; he/she can serve more customer requests in the day and hence increase the solution quality.

Analysis of Variance for Y, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Num_Vehicle	2	25415.3	25415.3	12707.7	6289.17	0.000
Cr*	2	519.3	519.3	259.7	128.51	0.000
Num_Vehicle*Cr*	4	149.3	149.3	37.3	18.47	0.000
Error	261	527.4	527.4	2.0		
Total	269	26611.3				

S = 1.42147   R-Sq = 98.02%   R-Sq(adj) = 97.96%

Figure 4.9 ANOVA result for preference index experiments.

The main effects plots illustrated in Figure 4.10 shows that the solution quality has positive correlation with the preference index. It increases linearly along the *Cr** value. However, higher *Cr** value will actually make the system more sensitive to the risk. It tends to reject any request which potentially violate the capacity constraint. Consequently, the dispatcher will serve less number of requests during the day. This should worsen the solution quality. However, since fewer requests mean fewer queues, it also decreases the average waiting time of the customers which means the solution quality is increased.

From those arguments, we can see that there is a tradeoff between the objectives in the problem. The relative weights in the aggregate objective function (ω_1 and ω_2) will play important role to evaluate whether the solution is better or worse under a specified preference index. Since the relative weights are subjective parameters, it is not easy to determine the values. It will depend on the decision maker preference – how much increase in waiting time is worth to be sacrificed for gaining one more request served in the day

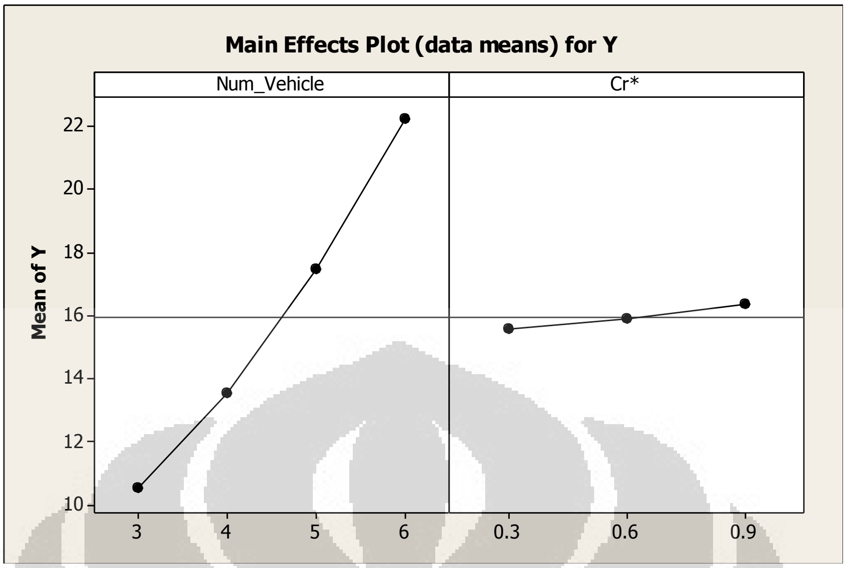

Figure 4.10 Main Effects plot in preference index experiments.

.The aggregate objective function in this study was designed in simple non-linear form with equal degree of importance (see equation [3.11]). The experiment result shows that this function gives better solution when the value of *Cr** is higher. Therefore, in this case, we can conclude that higher *Cr** will give better solution quality.

The ANOVA result also shows that there is a significance interaction between *Cr** value and preference index. The interaction plot as shown in Figure 4.11 reveals as the capacity increases, the difference in preference index will be less significance. Thus, the preference in *Cr** value is only significant when the capacity is relatively tight. This result has not been examined in the previous researches since in most VRPs the researchers assumed that the dispatcher can always afford the required capacity. However, there are some cases where the dispatcher only can provide a certain capacity like what we consider in this thesis. Here, in dynamic case, the full information regarding total request in a day is not available at the beginning of the period. Therefore, the dispatcher only can rely on the available capacity he/she had. That makes the preliminary decision in number of vehicles and preference index be significant in determining the overall result.

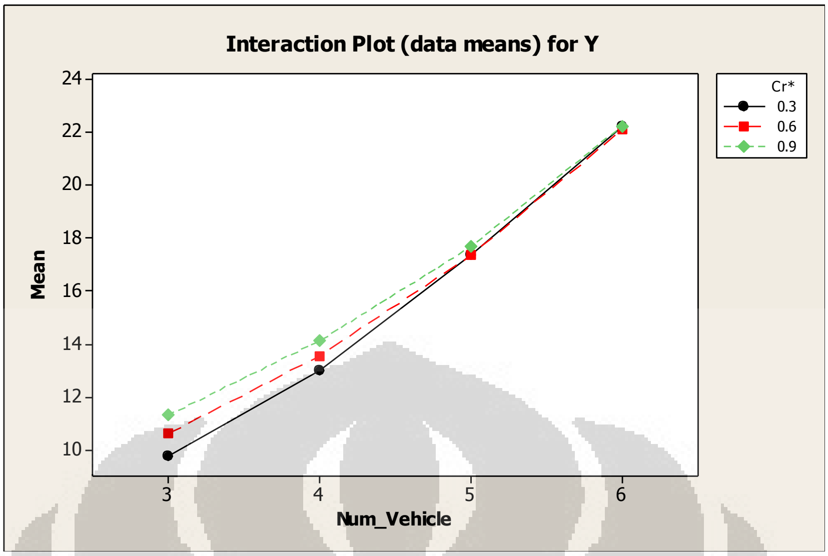

Figure 4.11 Interaction Plot between Cr* and Number of Vehicle.

4.5 Graphical Representation of Optimization Results

This section will show the graphical result of the tour built by Fuzzy-ACS in the end of the working day. The graphic will represent the best solution for five different cases which have been shown in Table 4.9.

4.5.1 Case 1 (dod = 0)

Table 4.12 Best solution for case 1 (dod = 0).

Result	Value
Customer Served	50
Avg. Waiting Time	3.83Hr
Route schedule	0- 46- 12- 5- 49- 10- 30- 39- 33- 45- 15- 40- 0
	0- 27- 1- 22- 31- 28- 3- 20- 35- 36- 43- 0
	0- 47- 4- 17- 37- 44- 42- 19- 41- 13- 18- 0
	0- 32- 11- 38- 9- 50- 34- 21- 29- 2- 16- 0
	0- 6- 48- 8- 26- 7- 23- 24- 14- 25- 0

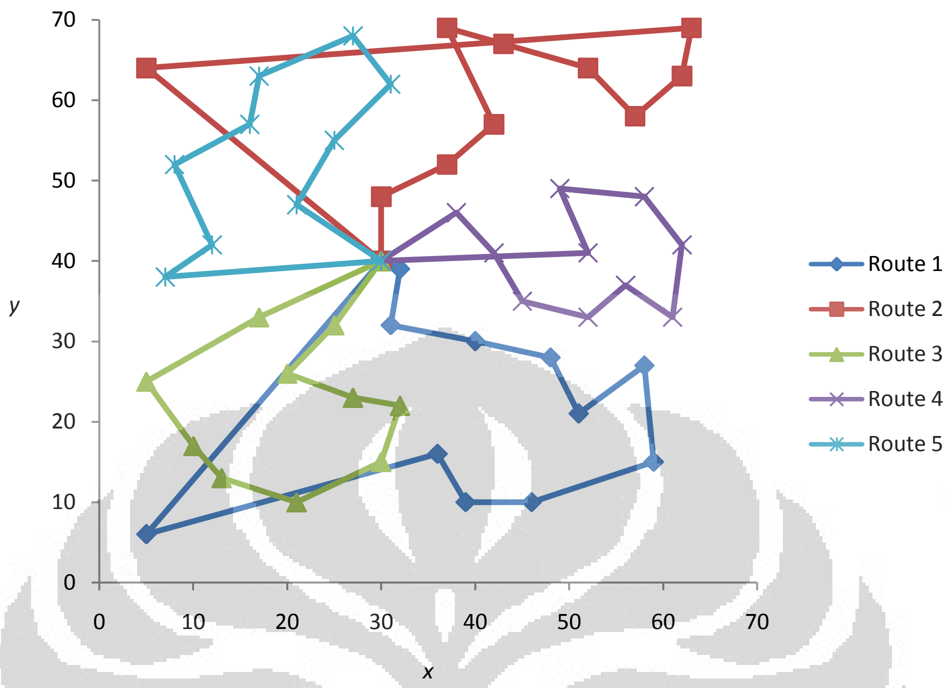

Figure 4.12 Graphical result for case 1.

4.5.2 Case 2 (dod = 0.25)

Table 4.13 Best solution for case 2 (dod = 0.25).

Result	**Value**
Customer Served	50
Avg. Waiting Time	3.81Hr
Route schedule	0- 12- 17- 37- 15- 45- 44- 42- 19- 41- 13- 40- 0
	0- 27- 32- 2- 29- 21- 50- 34- 30- 10- 39- 33- 0
	0- 6- 24- 23- 7- 48- 14- 25- 18- 43- 0
	0- 11- 16- 9- 38- 49- 5- 46- 47- 4- 0
	0- 1- 22- 8- 26- 31- 28- 3- 20- 35- 36- 0

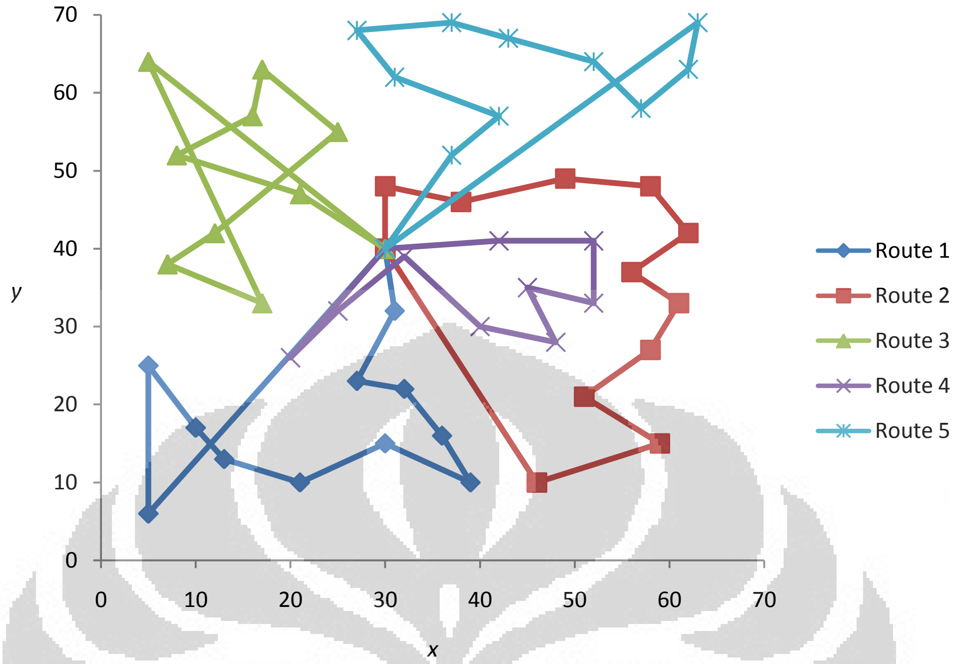

Figure 4.13 Graphical result for case 2.

4.5.3 Case 3 (dod = 0. 5)

Table 4.14 Best solution for case 3 (dod = 0.5).

Result	**Value**
Customer Served	50
Avg. Waiting Time	2.81 Hr.
Route schedule	0- 11- 16- 2- 29- 21- 38- 37- 44- 42- 41- 40- 0
	0- 12- 5- 9- 34- 30- 10- 39- 33- 45- 49- 0
	0- 1- 22- 8- 26- 31- 28- 3- 20- 35- 36- 50- 0
	0- 6- 14- 25- 24- 23- 7- 48- 27- 43- 0
	0- 4- 17- 15- 19- 13- 18- 47- 46- 32- 0

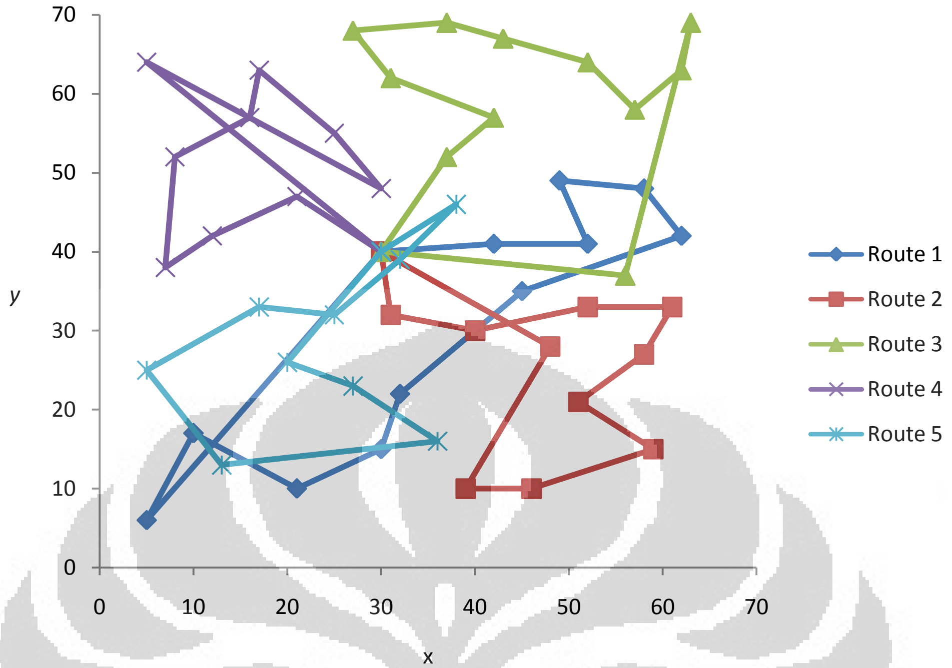

Figure 4.14 Graphical result for case 3.

4.5.4 Case 4 (dod = 0. 75)

Table 4.15 Best solution for case 4 (dod = 0.75).

Result	**Value**
Customer Served	50
Avg. Waiting Time	2.10 Hr.
Route schedule	0- 12- 17- 15- 19- 13- 4- 33- 45- 44- 42- 40- 0
	0- 6- 14- 18- 25- 27- 32- 46- 37- 47- 41- 0
	0- 5- 10- 9- 21- 22- 28- 35- 36- 38- 50- 0
	0- 1- 11- 16- 2- 20- 3- 29- 34- 30- 39- 49- 0
	0- 18- 7- 23- 24- 26- 31- 43- 48- 0

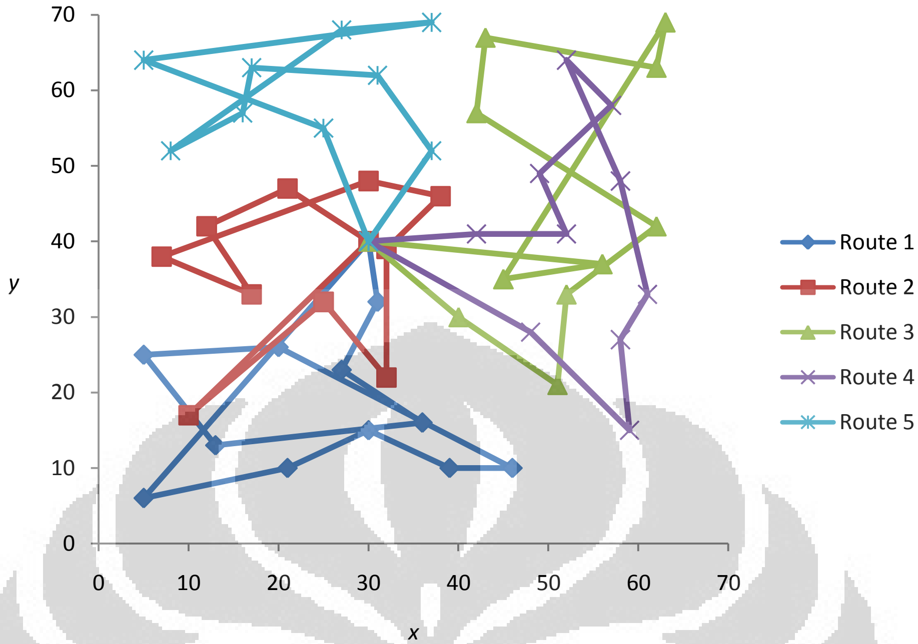

Figure 4.15 Graphical result for case 4.

4.5.5 Case 5 (dod = 1)

Table 4.16 Best solution for case 5 (dod = 1).

Result	**Value**
Customer Served	38
Avg. Waiting Time	1.44 Hr.
Route schedule	0- 6- 7- 8- 14- 23- 24- 27- 35- 0 0- 1- 3- 11- 20- 22- 26- 31- 28- 36- 0 0- 5- 10- 15- 19- 25- 32- 0 0- 4- 12- 17- 13- 18- 33- 0 0- 2- 9- 16- 21- 29- 30- 34- 0

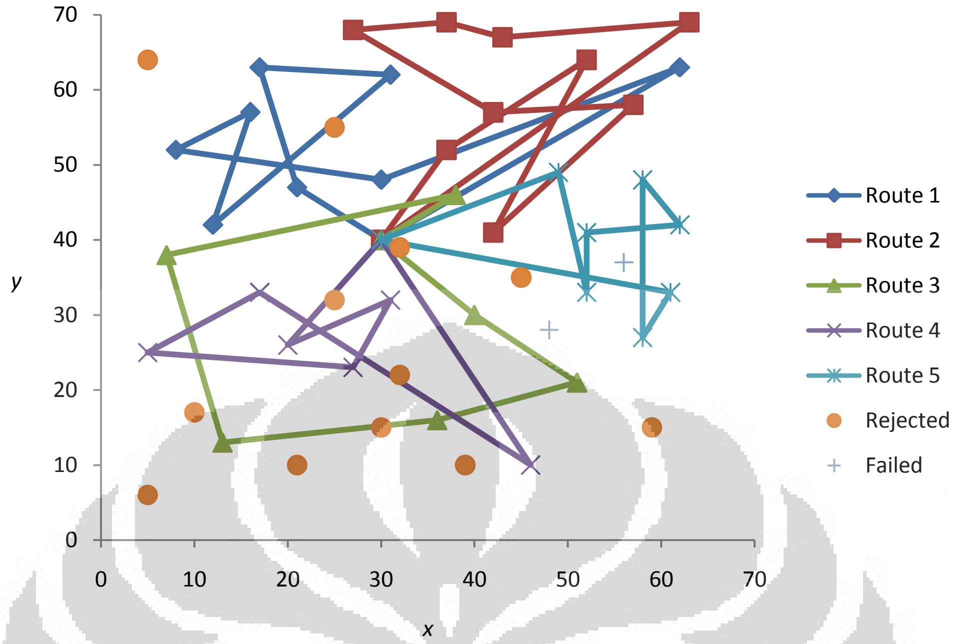

Figure 4.16 Graphical result for case 5.

Chapter 5
Conclusions and Future Study

5.1 Conclusions

This study presents an application of Fuzzy-ACS algorithm to solve dynamic VRP with limited number of vehicles and uncertainty in service time. The results of the study can be summarized as follows:

(1) *Fuzzy chance constrained programming* (CCP) with *credibility measurement* can be used as approximation method for dealing with uncertainty.
(2) The computational results showed that the hybridization of Fuzzy-CCP and *Ant Colony System* (ACS) can produce efficient algorithm which is able to provide high quality solution for dynamic optimization problem with uncertainty.
(3) The proposed *clustered-insertion* method can provide better initial trail for Fuzzy-ACS than Kilby et al.'s insertion method.
(4) In a multi-objectives problem, the design of the objective function will play important role to evaluate the solution objectively.
(5) *Number of time slices* has significant effect on the solution quality. Therefore, it needs to be tuned carefully.
(6) *Degree of dynamicity* (dod) significantly affects the model performance. The more dynamic the problem, the worse is the performance.
(7) In order to improve the performance in full-dynamic case, one can adopt time cut-off strategy to limit the dynamicity in critical period.
(8) *Number of vehicles* has a strongly positive relationship to the quality solution.
(9) *Credibility preference index* only has significant effect when the available capacity is relatively low.

5.2 Contributions

The contributions of this research can be summarized as follows:

(1) This study proposes a new variant of DVRP which provide on-site service with limited number of vehicles and uncertainty in service time
(2) An adaptive fuzzy approximation method has been designed to optimize the available resources and deal with the uncertainty.
(3) A proposed algorithm called Fuzzy-ACS has been developed to solve the specified problem.

5.3 Further Research

The results have shown that the proposed Fuzzy-ACS algorithm can effectively solve dynamic VRP with uncertainty. However, the current model still can be improved or extended in a number of ways:

(1) One can verify the performance of the proposed algorithm by using real case data.
(2) The current model assumes that the customer would wait until the end of the day to be served. One can extends this model into more realistic by considering time windows or adding another uncertainty variable (e.g. travel time) into the model.
(3) In this study, we use single aggregate objective function for handling the multi-objective problem. This method is basically subjective and needs predetermined relative weight to evaluate the result. Thus, one can improve the method by applying more advance multi-objective programming technique e.g. *pareto optimal solution* or *Multi-Objective ACS* to achieve more objective result.
(4) One can also develop the designing process of the adaptive fuzzy set in order to improve the approximation performance.

References

Adenso-Diaz, B. Laguna, M., “Fine tuning of the algorithms using fractional experimental designs and local search,” *Operation Research*, Vol.54, No.1, pp. 99-114, 2006.

Bertsimas, J.D. and Ryzin, G.V., “A stochastic and dynamic vehicle routing problem in the Euclidian plane,” *Operations Research*, Vol.39, No.4, pp. 601-615, 1990.

Bertsimas, J.D. and Ryzin, G.V., “Stochastic and dynamic vehicle routing with general demand and inter-arrival time distributions,” *Advances in Applied Probability*, Vol.25, No.4 pp. 947-978, 1993.

Bianchi, L., “Notes on dynamic vehicle routing: the state of the art,” Technical Report IDSIA-05-01, 2000.

Brito, J., Martinez FJ., Moreno AJ., Verdegay, JL., “Fuzzy approach for vehicle routing problems with fuzzy travel time.”*Fuzzy System (FUZZ).* IEEE International Conference, pp.1-8, 2010.

Brito, J., Moreno, AJ.,Verdegay, J.L., “Fuzzy optimization in vehicle routing problems,” IFSA-EUSFLAT, 2009,

Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds), “Combinatorial Optimization,” John Wiley, Chichester, 1979.

Colorni, A., Dorigo, M., Maniezzo, V., Trubian, M., “Ant system for job-shop scheduling,” *Belgian Journal of Operation Research, Statistics, and Computer Science,* Vol 34, No.1, pp. 39–53, 1994.

Dantzig, G., Ramser J.H., “The truck dispatching problem,” *Management Science*, Vol. 6, No. 1, pp. 80-91, 1959.

Dantzig, G., Fulkerson, R., Johnson, S., “Solution of a large-scale travelling salesman problems,” *Journal of the Operations Research Society of America*, Vol. 2, No. 4, pp. 393-410, 1954.

Dorigo, M., “Optimization, learning and natural Algorithms,” *Ph.D.Thesis*, Politecnico di Milano, Italy, 1992.

Dorigo, M., Gambardella L.M., “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation,* Vol. 1, No. 1, pp.53–66, 1997.

Dorigo, M., Gambardella, L.M., “Ant colonies for the traveling salesman problem,” TR/IRIDIA/1996-3, UniversitéLibre de Bruxelles, Belgium, 1996.

Dror, M., Laporte, G., Trudeau, P., “Vehicle routing with stochastic demands: properties and solution frameworks,” *Transportation Sciences*, Vol.23, pp. 166-176, 1989.

Erbao, C., Mingyong, L., “A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands,” *Journal of Computational and Applied Mathematics*, Vol. 231, pp. 302-310, 2009.

Erbao, C., Mingyong, L., “The open vehicle routing problem with fuzzy system,” *Expert System with Applications,* Vol. 37, pp. 2405-2411, 2010.

Fisher, M., Jaikumar, R., Wassenhove, L., “A generalized assignment heuristic for vehicle routing,” *Networks,* Vol. 11, pp. 109–124, 1981.

Fleischmann, B., Gnutzmann, S., Sandvoß, E., “Dynamic vehicle routing based on online traffic information,” *Transportation Science,* Vol. 38, No. 4, pp. 420– 433, 2004.

Gambardella L.M., Taillard E., Dorigo M., “Ant colonies for the quadratic Assignment problem,”*Journal of the Operational Research Society*, Vol. 50, pp. 167-176, 1999.

Garrido, P. and Riff, M.C., “DVRP: a hard dynamic combinatorial optimization problem tackled by an evolutionary hyperheuristic,” *Journal of Heuristics,* Vol. 16, pp.795-834, 2010.

Gendrau, M., Hertz, A. Laporte, G., “A tabu search heuristic for the vehicle routing problem,” *Management Science,* Vol. 40, pp.1276-1290, 1994.

Gendrau, M., Guertin, F. Potvin, J.Y., Taillard, E. “Parallel tabu search for real-time vehicle routing and dispatching,” *Transportation Science,* Vol. 33, No.4, pp. 381–390, 1999.

Ghiani, G., Guerriero, F., Laporte., G., Musmanno., R., “Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies,” *European Journal of Operational Research,* Vol. 151, pp. 1-11, 2003.

Gupta, R., Singh, B., Pandey, D, "Fuzzy vehicle routing problem with uncertainty in service time," *International Journal Contemporary Mathematics Sciences*, Vol.5, No.11, pp. 497-507, 2010.

Gupta, R., Singh, B., Pandey, D., "Multi-objective fuzzy vehicle routing problem: a case study," *International Journal Contemporary Mathematics Sciences*, Vol.5, No.29, pp. 1439-1454, Vol. 2010.

Hanshar, F.T., Ombuki-Berman, B.M., "Dynamic vehicle routing using genetic algorithms," *Applied Intelligence*, Vol. 27, No. 1, pp. 89-99, 2007.

Harrera, F., Verdegay, J.L., "Three models of fuzzy integer linear programming," *European Journal of Operational Research*, Vol. 83, pp. 581-593, 1995.

Hong, L., Xu, M., "A model of MDVRPTW with fuzzy travel time and time dependent and its solution," *Fifth International Conference on Fuzzy System and Knowledge Discovery*, pp. 473-478, 2008.

Kilby, P., Prosser, P., Shaw. P., "Dynamic VRPs: A study of scenarios," Report APES-06-1998.

Kuo, RJ., Chiu, CY, Lin YJ. "Integration of fuzzy theory and ant algorithm for vehicle routing problem with time window," *Fuzzy Information, 2004. Processing NAFIPS '04, IEEE Annual Meeting of the* , Vol.2, pp. 925- 930, 2004.

Larsen, A., "The dynamic vehicle routing problem," *Ph.D thesis*, Technical University of Denmark, 2001.

Larsen, A., Madsen, O., Solomon, M., "Partially dynamic vehicle routing – models and algorithms," *Journal of the Operational Research Society*, 53, pp.637-646, 2002.

Lau, HC., Sim, M., Teo, KW., "Vehicle routing problem with time windows and a limited number of vehicles," *European Journal of Operational Research*, 148, pp.559-569. 2003.

Liu, B., "Uncertainty Theory," Springer-Verlag, Berlin, 2004.

Liu, B., "A survey of credibility theory," *Fuzzy Optimization and Decision Making*, 5(4), pp. 387-408, 2006.

Meisel, S., "Anticipatory optimization for dynamic decision making," 1st Edition, XIV, 182, 2011.

Montemanni, R., Gambardella L.M., Rizzoli, A.E., Donati A.V., "Ant colony system for a dynamic vehicle routing problem," *Journal of Combinatorial Optimization*, Vol.10, pp.327–343, 2005.

Nanry, WP., Barnes, W.B., "Solving the pickup and delivery problem with time windows using reactive tabu search," *Transportation Research* part B, Vol. 34, pp. 107-121, 2000.

Pankratz, G., "Dynamic vehicle routing by means of a genetic algorithm," *International Journal of Physical Distribution & Logistics Management*, Vol. 35, No.5, pp. 362-383, 2005.

Pillac, V., Gendreau, M., Gueret, C., Medaglia., A., "A review of dynamic vehicle routing problems," CIRRELT-2011-62, 2011.

Powell, W.B., Carvalho, T.A., Godfrey, G.A., Simao, H.P., "Dynamic fleet management as a logistics queuing network," *Annals of Operation Research*, Vol. 61. pp. 165-188., 1995.

Psaraftis, H.N., "Dynamic vehicle routing: status and prospect," *Annals of Operations Research*, Vol. 61, No.1, pp. 143-164, 1988.

Psaraftis, H.N., "An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows," *Transportation Science*, Vol. 17, No.3., pp. 351-357, 1983.

Resende, M.G.C., Ribeiro, C.C., "Greedy Randomized Adaptive search procedures," in F. Glover and G. Kochenberger (eds.), *Handbook of Metaheuristics*, pp. 219–249, 2003.

Rizolli, A.E, Montemanni, R., Lucibello, E., Gambardella, L.M., "Ant colony optimization for real world vehicle routing problems," *Swarm Intelligence*, Vol. 1, pp. 135-151, 2007.

Taillard, E., "Parallel iterative search methods for vehicle-routing problems," *Networks*, Vol. 3, No. 8, pp. 661–673, 1994.

Teodorovic, D., Lucic, P., "The fuzzy ant system for the vehicle routing problem when demand at nodes is uncertain," *International Journal on Artificial Intelligence Tools*, Vol. 16, No.5, pp. 751-770, 2007.

Teodorovic, D., Pavkovic, G., “A simulated annealing technique approach to the vehicle routing problem in the case of stochastic,” *Transportation Planning Technology,* Vol. 16, pp. 261-273, 1992.

Teodorovic, D., Pavkovic, G., “The fuzzy set theory approach to the vehicle routing problem when demand at nodes is uncertain,” *Fuzzy Sets and System,* Vol. 82, pp. 307-317, 1995.

Toth, P., Vigo, D., “An exact algorithm for the vehicle routing problem with backhaul,” *Transportation Science*, Vol. 31, No. 4, pp. 372-385, 1997.

Toth, P., Vigo, D., “An overview of vehicle routing problem,” In P. Toth and D. Vigo (Eds), *The Vehicle Routing Problem,* Philadelphia: Society for Industrial and applied mathematics, 2002.

Yang, J., Jaillet, P., Mahmassani, H., “Real-time multivehicle truckload pick-up and delivery problems,” *Transportation Science,* Vol. 38, pp. 135-148, 2004.

Zadeh, L.A., “Fuzzy sets,” *Information and Control,* Vol. 8, pp.338-353, 1965.

Zeimpekis, V.S., “Design and evaluation of a real time fleet management system for dynamic incident handling in urban freight distribution,” *Ph.D thesis,* Athena University of Economics and Business, 2007.

Zheng, Y., Liu, B., “Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm,” *Applied Mathematics and Computation*, Vol. 176, pp. 673-683, 2006.

Appendix A
Basic Dataset

In this study, we generate a new dataset by modifying c50 dynamic instance from Montemanni et al. (2003).The modified instance has following data types:

1) number of the customers
2) arrival time of customers,
3) depot and customer location,
4) service time duration,
5) working day period
6) service time historical data.

Table A1 Basic data description.

Data	Detail
Distance type	Euclidian
Number of Depots	1
Depot Location (x,y)	(30, 40)
Number of Customers	50
Working day period	12 hours
Historical data of service time	{0.58, 0.24, 0.79, 0.79, 0.51, 0.62, 0.51, 0.48, 0.55, 0.27}

Table A2 Customer location, arrival time, and service duration.

Customer	Location Coordinate (x,y)	Arrival Time (00:00+)	Service Duration (hr)	Customer	Location Coordinate (x,y)	Arrival Time (00:00+)	Service Duration (hr)
1	(37, 52)	0.03	0.24	26	(27, 68)	6.97	0.24
2	(49, 49)	0.14	1.03	27	(30, 48)	7.42	0.51
3	(52, 64)	0.31	0.55	28	(43, 67)	7.66	0.48
4	(20, 26)	0.41	0.31	29	(58, 48)	7.66	0.21
5	(40, 30)	0.41	0.72	30	(58, 27)	7.73	0.65
6	(21, 47)	0.55	0.51	31	(37, 69)	7.93	0.38
7	(17, 63)	0.72	0.65	32	(38, 46)	8.24	0.41
8	(31, 62)	1.16	0.79	33	(46, 10)	8.27	0.79
9	(52, 33)	1.44	0.38	34	(61, 33)	8.38	0.89
10	(51, 21)	1.50	0.17	35	(62, 63)	8.58	0.58
11	(42, 41)	1.74	0.65	36	(63, 69)	8.72	0.21
12	(31, 32)	1.98	0.99	37	(32, 22)	9.64	0.31
13	(5, 25)	2.15	0.79	38	(45, 35)	9.68	0.51
14	(12, 42)	2.70	0.72	39	(59, 15)	9.88	0.48
15	(36, 16)	2.74	0.34	40	(5, 6)	9.91	0.24
16	(52, 41)	2.74	0.51	41	(10, 17)	10.05	0.92
17	(27, 23)	2.94	0.10	42	(21, 10)	10.05	0.44
18	(17, 33)	3.42	1.40	43	(5, 64)	10.12	0.38
19	(13, 13)	3.56	0.31	44	(30, 15)	10.19	0.55
20	(57, 58)	3.97	0.96	45	(39, 10)	10.32	0.34
21	(62, 42)	4.38	0.27	46	(32, 39)	10.43	0.17
22	(42, 57)	4.72	0.27	47	(25, 32)	10.67	0.85
23	(16, 57)	5.37	0.55	48	(25, 55)	10.74	0.58
24	(8, 52)	6.29	0.34	49	(48, 28)	11.62	0.62
25	(7, 38)	6.46	0.96	50	(56, 37)	11.79	0.34

Appendix B

Fuzzy-Simulated Annealing

B.1 Pseudo-code

ProcedureFuzzy-SA;

Set up Fuzzy-SA parameter {T_0, *maxIter*};
Get static case $S = \{1,2,3.. n\}$ from e*vent scheduler;*

Create initial solution X_0using *clustered-insertion* method;

Set $X_{best} = X_0; F_{(Xbest)} = F_{(X)}$;

Set CPU time = 0;

While (*CPU Time* ≤ *maxTime*)

$$T = \frac{T_0}{maxTime} \times (maxTime - CPU\ time)$$

Iter = 0;

While*(iter ≤ maxIter)*

While*(non_feasible ≤ max_trial)*

Generate new solution *Y* from *X* by *random insertion operation*;

Compute credibility of each route*;*

If ($Cr_k \geq Cr^*$)

Continue;

End If

End While

End While

If($\Delta = F_{(Y)} - F_{(X)} \leq 0$)

Let $X = Y$;

Else

Generate r = random(0,1);

If(r<$\exp(-\Delta/T)$)

Let $X = Y$;

End If

End If

Iter = Iter + 1;

If ($F_{(X)} \leq F_{(Xbest)}$)

$F_{(Xbest)} = F_{(X)}$;

$X_{best} = X$;

End If

Apply random swap local search procedure;

Update *CPU Time*;

End While

Return the best solution;

Figure B1 Pseudo-codes of Fuzzy-SA.

B.2 Parameter Tuning

Simulated Annealing has two parameters that need to be tuned, that is: initial temperature (T_0) and number of iteration (*maxIter*). In order to figure out the appropriate parameter value, we perform statistical experiment. The experiment design can be seen in table C1. We take 30 replications for each of combination of parameters. Afterwards, we use ANOVA to analyze the experiment result.

Table B1 Experiment design for SA parameter tuning.

Parameter	Low	High
Initial Tempearture(T_0)	500	1000
Number of Iteration (*maxIter*)	100	200

Analysis of Variance for WaitTime, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Blocks	29	297.29	297.29	10.25	0.54	0.968
To	1	149.99	149.99	149.99	7.94	0.006
maxIter	1	2.78	2.78	2.78	0.15	0.702
To*maxIter	1	109.98	109.98	109.98	5.82	0.018
Error	87	1643.85	1643.85	18.89		
Total	119	2203.88				

Figure B2 ANOVA result for Fuzzy-SA parameter tuning experiment.

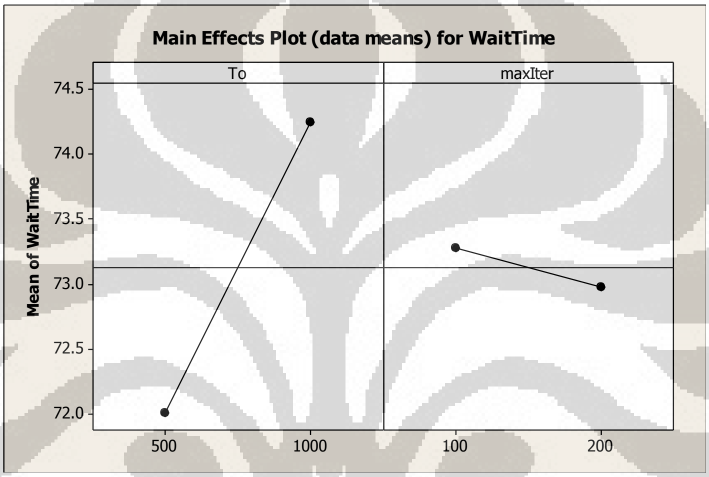

Figure B3 Main Effect plot for Fuzzy-SA parameter tuning experiment.

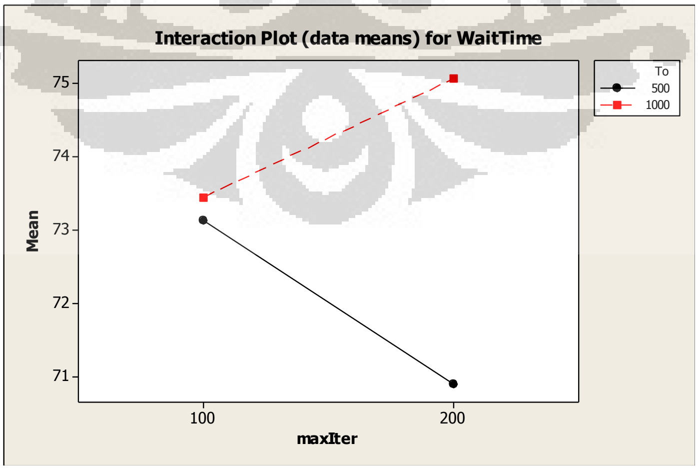

Figure B4 Interaction Effect plot for Fuzzy-SA parameter tuning experiment.

From ANOVA results we can see that initial temperature T_0 has significant effect to the model response. We also can see that there is significant interaction between initial temperature and iteration number. Thus, for the comparison purpose in chapter 4, we will set SA parameter as follow: T_0= 500, *maxIter* = 200.

Appendix C
Problem Variants

This appendix shows the arrival time distribution of four different instances based on its *degree of dynamism* (dod). The distributions are differentiated by simply cutting the horizon time into a fraction based on the *dod* specification.

Table C1 Customer location, arrival time, and service duration.

Customer	Arrival Time (00:00+)			
	dod = 1	dod = 0.75	dod = 0.5	dod = 0.25
1	0.03	0.00	0.00	0.00
2	0.14	0.00	0.00	0.00
3	0.31	0.00	0.00	0.00
4	0.41	0.00	0.00	0.00
5	0.41	0.00	0.00	0.00
6	0.55	0.00	0.00	0.00
7	0.72	0.00	0.00	0.00
8	1.16	0.00	0.00	0.00
9	1.44	0.00	0.00	0.00
10	1.50	0.00	0.00	0.00
11	1.74	0.00	0.00	0.00
12	1.98	0.00	0.00	0.00
13	2.15	0.00	0.00	0.00
14	2.70	0.00	0.00	0.00
15	2.74	0.00	0.00	0.00
16	2.74	0.00	0.00	0.00
17	2.94	0.00	0.00	0.00
18	3.42	0.42	0.00	0.00
19	3.56	0.56	0.00	0.00
20	3.97	0.97	0.00	0.00
21	4.38	1.38	0.00	0.00
22	4.72	1.72	0.00	0.00
23	5.37	2.37	0.00	0.00
24	6.29	3.29	0.29	0.00
25	6.46	3.46	0.46	0.00
26	6.97	3.97	0.97	0.00
27	7.42	4.42	1.42	0.00
28	7.66	4.66	1.66	0.00
29	7.66	4.66	1.66	0.00
30	7.73	4.73	1.73	0.00
31	7.93	4.93	1.93	0.00
32	8.24	5.24	2.24	0.00
33	8.27	5.27	2.27	0.00
34	8.38	5.38	2.38	0.00
35	8.58	5.58	2.58	0.00
36	8.72	5.72	2.72	0.00
37	9.64	6.64	3.64	0.64
38	9.68	6.68	3.68	0.68
39	9.88	6.88	3.88	0.88
40	9.91	6.91	3.91	0.91
41	10.05	7.05	4.05	1.05
42	10.05	7.05	4.05	1.05
43	10.12	7.12	4.12	1.12
44	10.19	7.19	4.19	1.19
45	10.32	7.32	4.32	1.32
46	10.43	7.43	4.43	1.43
47	10.67	7.67	4.67	1.67
48	10.74	7.74	4.74	1.74
49	11.62	8.62	5.62	2.62
50	11.79	8.79	5.79	2.79

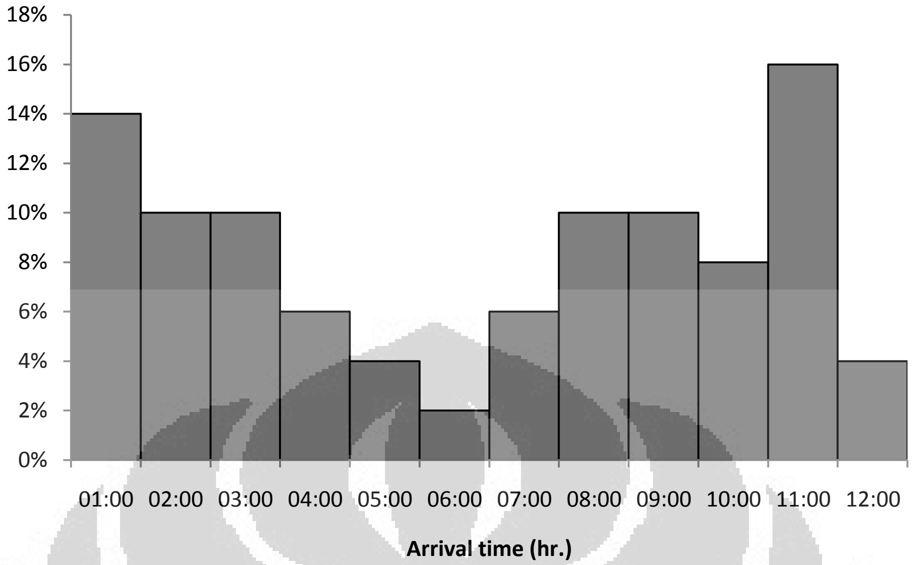

Figure C1 Distribution of customer arrival time when dod = 1.

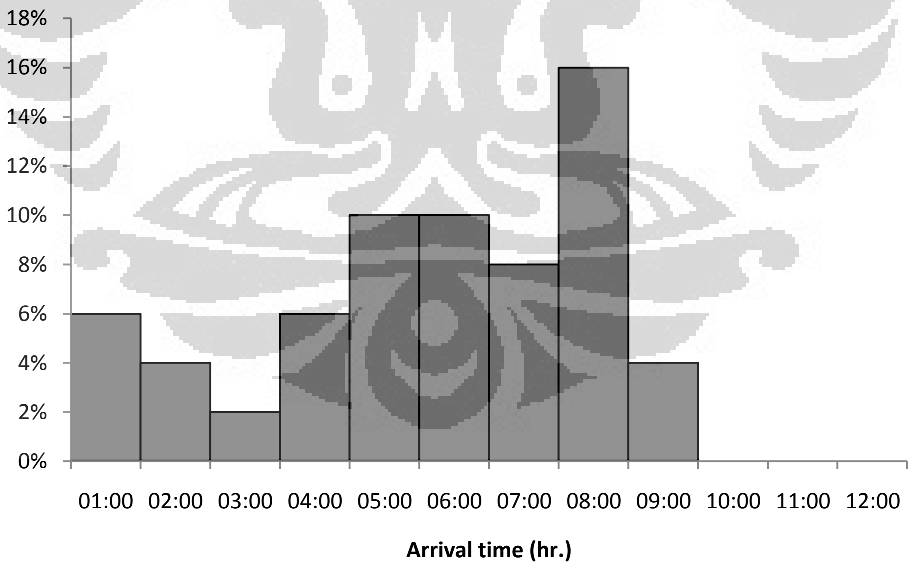

Figure C2 Distribution of customer arrival time when dod = 0.75.

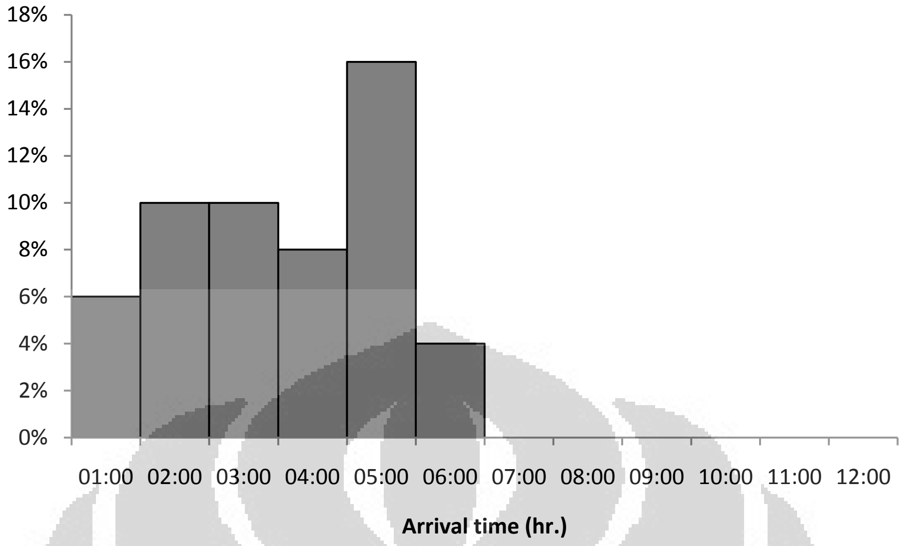

Figure C3 Distribution of customer arrival time when dod = 0.5.

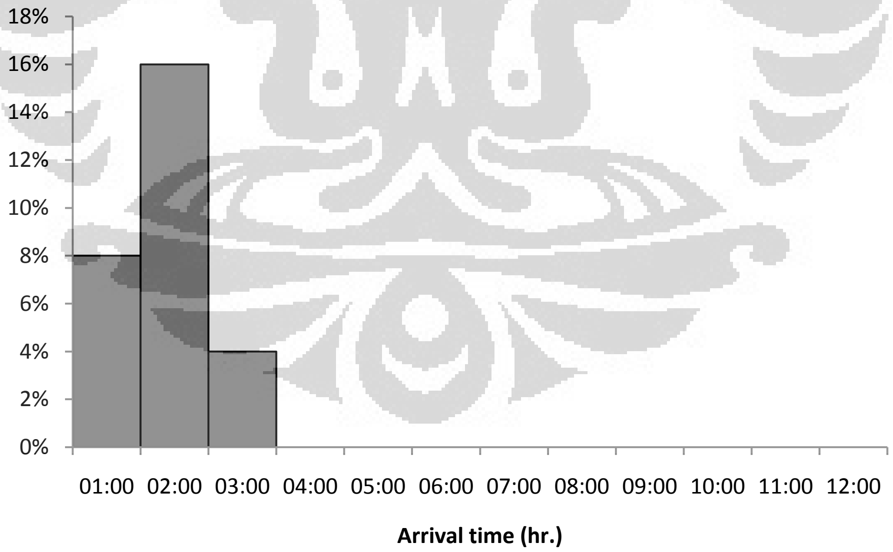

Figure C4 Distribution of customer arrival time when dod = 0.25.