國　立　交　通　大　學

運輸科技與管理學系

碩 士 論 文

Waiting Strategies for the Dynamic Dial-A-Ride Problem

動態撥召公車問題等待策略之研究

研 究 生：袁智偉
指導教授：黃家耀　助理教授
　　　　　韓復華　教授

中 華 民 國 九 十 六 年 六 月

Waiting Strategies for the Dynamic Dial-A-Ride Problem

動態撥召公車問題等待策略之研究

研　究　生：袁智偉　　Student : Chi-Wai Yuen

指導教授：黃家耀　　Advisors : Ka-Io Wong

韓復華　　Anthony Fu-Wha Han

國立交通大學

運輸科技與管理學系

碩士論文

A Thesis

Submitted to Department of Transportation Technology and Management

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Transportation Technology and Management

June 2007

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 六 年 六 月

Waiting Strategies for the Dynamic Dial-A-Ride Problem

Student: Chi-Wai Yuen Advisor: Ka-Io Wong

Anthony Fu-Wha Han

Department of Transportation Technology and Management
National Chiao Tung University

Abstract

The Dial-A-Ride problem (DARP) is a problem of providing demand responsive transport that delivers passengers from their specified origins to destinations with desired time windows. To keep a level of services, the operator should deliver the passengers subject to maximum riding time and waiting time constraints. Such problem is similar to the pick-up and delivery problem with time windows in the theme of supply chain management, but considering passenger transportation rather than goods transportation. In a Static Dial-A-Ride Problem, the operator accepts requests before the day of operation, while the Dynamic Dial-A-Ride Problem (DDARP) allows receiving requests throughout the operating period. This study focused on solving DDARP, under different degree of dynamism (*dod*), with minimum operating vehicles and minimum total travel distance.

There are two sub-problems solving DDARP: routing and scheduling is a sub-problem for route constructions which aims to plan the vehicle routes and stops in visiting the requests with several objectives. The Cheapest Insertion (CI) method is used to construct initial routes which can later be improved by exchange method. Scheduling is a sub-problem for designing the arrival and departure time for each stop along the routes. It aims to insert as many real-time requests as possible in each vehicle during the day of operation. Three different strategies, namely, Drive First (DF), Wait First (WF) and Dynamic Wait (DW) are described in this study.

A set of simulation experiments were designed to evaluate the performance of the three waiting strategies under different *dod*. Compared to the results of DF strategy and WF strategy, the DW strategy provides a better solution with requiring less operating vehicles and shorter travel distance. We found that the requirement of extra operating costs serving a fixed number of dynamic requests decreased as more requests were known in-advance. This observation follows the principle of “economy of scale”. An interesting finding in the study was that the system may require less operating vehicles and total travel distance in a fully dynamic environment (*dod* = 100%) as compared to a highly dynamic problem (say *dod* = 60%), under a fixed number of total requests. This is in opposition to our intuition that more information can bring the system to a lower cost, and we name this case a “counter-intuitive” observation.

***Keywords*:** Dynamic Dial-A-Ride Problem, Degree of Dynamism, Waiting Strategies, Drive First, Wait First, Dynamic Wait

動態撥召公車問題等待策略之研究

學生：袁智偉　　　　　　　　　　　　　　指導教授：黃家耀 助理教授

韓復華 教授

國立交通大學運輸科技與管理學系碩士班

摘　要

運輸是供應鏈管理配送貨物不可或缺的一環。尤其在競爭激烈的社會裡，即時及門運輸(door-to-door transportation)需求的服務品質變得愈來愈重視，如何能提供有效完善的配送服務也是未來重要的課題之一。撥召公車具有及門運送的特性，乘客可指定搭乘地點與到達目的地，在指定時間內車輛完成接載與配送的服務。撥召公車問題分為靜態問題與動態問題。靜態問題為派遣中心在車輛出發前接收乘客之起迄需求點及按照其目的選擇指定搭乘時間或到達時間，當發車後即不再接收新的即時需求。動態問題則考慮在營運期間內可隨時接收派車需求，即時需求出現時現有車輛可允許在不違反已排定之服務點前提下服務新需求點，營運結束可計算出當天需求之動態度，即為動態需求數佔總需求數之比例，作為實際不同需求特性下派遣中心對後續路線規劃之探討。本研究以最少營運車輛數及最短總旅行距離為目標求解動態撥召公車問題。

動態撥召公車問題可透過兩部分求解：路線構建及排程建立。路線構建部份在決定各服務點的先後順序，本研究針對靜態問題先使用最省插入法，再透過交換法改善路線服務順序；針對動態問題則因應車輛即時的位置透過即時最省插入法插入動態需求。排班建立部分則在決定各車輛到達與離開各服務點之時間，主要使用三種等待策略：優先行駛策略(DF)、優先等待策略(WF)及動態等待策略(DW)。

本研究以兩個層面之模擬方法分析動態撥召公車問題，第一部分為以固定總需求數下調整動態度以比較三種等待策略之績效表現。可發現在求解動態問題上 DF 策略比 WF 策略需要較少車輛數但較長總旅行距離。本研究提出之 DW 策略則能求解比 DF 策略較短之總旅行距離，使得 DW 策略比 DF 策略及 WF 策略同時較少營運車輛數及較短總旅行距離之效果。第二部分為固定即時需求下，額外成本與已知需求數間的關係。從中發現當已知需求數愈來愈多時，需要服務同數量之動態需求數呈遞減狀況，符合一般經濟規模的原理。但在第一部分中我們卻發現在大約 60%以後的動態問題所需要之營運車輛數及總旅行距離有下降的趨勢，甚至在大型問題上此情況更為明顯，似乎與現實狀況下對"資訊愈早取得愈有價值"之一般認知有所不符，本研究稱此現象為直覺之相反"counter-intuitive"，值得作後續之探討。

關鍵字：動態撥召公車問題、動態度、等待策略、優先行駛策略(DF)、
優先等待策略(WF)、動態等待策略(DW)

Acknowledgements

經過一年多的努力，碩士論文總算順利完成。首先感謝兩位恩師 韓復華教授及 黃家耀助理教授兩年來之悉心指導與教誨，及生活上之關心與照顧，使我在研究所求學階段中獲益良多，師恩浩蕩、永銘在心！

另外在論文口試期間承蒙中華大學 蘇昭銘副教授及大師兄 卓裕仁助理教授不吝指教與斧正，提供諸多寶貴意見使本論文更臻完善，在此衷心感謝。論文審查期間感謝系上 王晉元副教授撥冗閱讀並給予指導，使本論文能更嚴謹充實，學生衷心感佩。

別了，交大，我在台灣求學六年的家！回想大一剛進來交大的第一天，在我腦海中還以為是昨天在發生的事情。在這段時間裡，首先感謝陪我走過這段光陰的好朋友：老魂、啟基，我會一直記得我們三個自稱大一時「十二舍賤人三人組」的事蹟，還有我們常常從清夜走路回來時啟基提出“走得愈快，距離愈短”的茅論，可惜這個理論只有我們三個人知道，不然一定成為新一代的偉人。大馬朋友欣雁、隆哥、F4、添耀、偉權、大姐菁、鳳儀、印尼代表 Rudi 及快要變成外籍新娘的文運、緬甸美女巧巧以及美國大帥哥 Elbert、澳門代表 Marva 姐姐等，很開心有機會跟你們為僑聯做事，也感謝你們一直容忍我的牛脾氣。跟你們一起玩樂的日子怎麼寫也寫不完，其中讓我最深刻倒是巧巧與偉權的餿主意「大風雨當然要去港南吹風」的事蹟、十幾台機車夜遊讓警車以為我們是飆車族的大陣型記錄，每次回想都覺得好好笑，這些開心片段只能在腦海中回味。

感謝詩敏一直在我旁邊默默的給予我支持與鼓勵，使我克服多種困難，包括初次研究所考試的失敗、第二次考試的緊張心情等，以及承受了我很多無情的語氣，沒有你的支持，我相信我的研究不能這麼順利的完成。還有基佬的愛玩性格使我不致於變成只會唸書的蛀書蟲，也感謝你常說要教我泡妹的技巧。特別鳴謝慈幼 F5A 的豬朋狗友們，讓我一直把“We are the best”銘記於心。一起為做“Times”雜誌封面人物而努力吧！

很開心有機會加入了網路實驗室的大家庭，我也很榮幸能夠與學長姐學弟妹打成一遍，首先感謝 Lab 老大朱哥(朱頭威哥)、正妹雷達器俊德、一直叫我小李飛刀的昆諭、平易近人的淑詩、美食“皇后”水餃俐諭、程式達人育廷、臉色紅潤的裕智在學業上及生活上的幫忙，讓我更快速的融入這個實驗室。當了六年同學的昱傑及阿春，一開始忍受我的爛國語。我最記得第一天認識昱傑的時候是上“美國文學”的時候，感覺不是六年前的事；而對你最深刻的是你的音量器壞了，怎麼調聲音都調不大；阿春第一次認識應該是大一系網練習中，你教曉我一句最實用的話“wa 就 and 島”，讓我學會了對別人講的第一句台語實話，終身受用；也會記得你的“鎰”是“金”“益”的“鎰”。台中大美女彥臻，很貼心的你們在台語溝通時還不忘幫我用國語翻譯，雖然不是同步，也讓我非常感動。我不會忘記我們在實驗室“Virtua Tennis”打遍天下無敵手的偉蹟，“E, kekeke…”，雖然只認識你一年多的時間，也有寫不完的回憶。謝謝盛山學長不定時舉辦實驗室聚餐，使我們的感情更加緊密。Poney、仲豪、沂茹在我有需要幫忙的時候你們都不會缺席，非常感謝你們，接下來換你們接棒研究工作，實驗室要靠你們去發揚光大了，加油！另外，大家一定要把我在體檢單驗出女性的身份保密喔，謝謝！^_^

最後，我還要把我的成果獻給無論在金錢上、精神上一直都在支持我的媽媽、四位姊姊、姊夫們、小欣欣、阿姨們及舅舅們，沒有你們，我相信我沒有機會這麼順利的完成我的學業，以後就讓我來為家裡出一份力吧！“I love you all！”

「阿媽，我終於得咗啦！哈哈！」

袁智偉 謹誌

記于 交大網路實驗室 2007.6

Table of Contents

List of Tables

List of Figures

CHAPTER 1
INTRODUCTION

1.1 Background

The Vehicle Routing Problem (VRP) is an important component in logistics and supply chain management. It plans a set of vehicle routes, passing through a number of demand points. Each customer must be assigned exactly once with the aim to minimize the operating cost.

The Pickup and Delivery Problem with Time Windows (PDPTW) consists of planning routes for picking up and delivering goods within a pair of service intervals. Compared to the classic VRP problems, two mathematical constraints must be satisfied in planning routes, which are the pairing constraint and precedence constraint, in PDP problems. The "pairing constraint" states that the paired pickup and delivery locations must be served by the same route. The "precedence constraint" stipulates the pickup locations must be served before the relevant delivery locations. An additional constraint that the requests should be served within a specified time interval is considered as "time constraints". The objective of this problem usually considers in minimizing number of vehicles and the sum of travel distance. An example of PDPTW is the express-mail courier services assign a vehicle to start from a depot routing to pickup and deliver mails and parcels. It usually solves the problem for freight transports, however, in the case of people transportation, a study of the Dial-A-Ride Problem (DARP) is considered as a variant of PDPTW problem.

The DARP is a special case of PDPTW, with focusing on the operational constraints for people transportation, in which adequate service quality should be supplied. For example, the door-to-door transportation service for elderly or handicapped people is classified as a DARP problem. It consists of vehicle routes and schedules for a number of customers who specified origins and destinations with desired time windows, and the operator provides a fleet of identical vehicles starting at the same depot (Cordeau and Laporte 2003b). The purpose of this problem is to plan a set of minimum cost vehicle routes able to accommodate all the customer requests, concerning the service quality that the time of pickup or delivery must be within some pre-specified intervals.

The dial-a-ride service may operate in static or dynamic environment. Static DARP has been widely studied for more than thirty years. In the static cases, the requests are assumed to be known in advance, and therefore it is possible to plan all vehicle routes before operation. In contrast, the Dynamic DARP (DDARP) has been less studied until recently with the progress of the information and communication technologies (ICT). In the dynamic environment, the requests are gradually revealed throughout the day and vehicles are assigned to meet the real-time requests. Practically, the fully Dynamic DARP has rarely existed since some of the requests are often known when vehicles routing plan is started, and therefore the DDARP is usually a partial dynamic problem. To serve real-time requests, the routes can be constructed and modified in real-time (Madsen et al. 1995, Cordeau and Laporte 2003b).

1.2 Problem Definition

The definition of waiting strategy in DDARP is to allocate the waiting time of a vehicle at different stops along the route, which satisfies the time windows of customers. As the known requests are planned to the routes, they are then scheduled by using different strategies to serve the requests. Considering the service time interval for each location, the vehicle may wait at one location for providing services. The vehicle may be able to serve a newly arrived location by efficiently arranging this waiting time periods. In static problems, the operator plans to serve the requests with minimum operating costs; therefore, waiting time is part of fixed operating cost in the problem. On the other hand, waiting time can be an issue to insert real-time requests to the existing vehicles; the waiting strategy is used to allocate the waiting time in order to respond real-time requests for operations.

The efficiency of using different waiting strategies can be displayed with two categories, operating cost and level of services. Operators concern serving requests with minimum operating cost, including minimum number of vehicles, total travel distance and total duration time. When more vehicles are needed to serve the requests, fixed operating cost would also increase. Furthermore, more total travel distance needs much variable cost and finally, more duration time means to pay much labor cost in operations. Meanwhile, customers concern to take the services with minimum waiting time out of the vehicles and minimum riding time inside the vehicles. They desire that the vehicles arrive to the location in minimum waiting time. In addition, they also hope to travel from origin to destination with minimum riding time. These five terms, number of operating vehicles, total travel distance, total duration time, total waiting time of customers and total riding time of customers, can be used to explain the performance of different waiting strategies in which will be described in CHAPTER 5.

1.3 Objectives

This study focuses on solving the DDARP by investigating three waiting strategies of scheduling. For those requests which are known in advance, the route is planned to satisfy the requests with minimum operating costs, which considers the operating vehicles and travel distance. For the dynamic requests, three different strategies, namely, Drive First (DF), Wait First (WF) and Dynamic Wait (DW), are described to model how the operator should response to the real-time requests by keeping the vehicles waiting at specific locations. Simulation and computational tests are used to show the performance of the proposed strategies.

1.4 Structure of the Thesis

In this chapter, we have introduced the background of the DARP problem. The static and dynamic problems were described, and the objectives of this work were also presented.

Chapter 2 provides a literature review of previous studies for this problem. As DARP problem is an NP-hard problem, it is difficult to find an exact solution. In the static problem, a quick and simple heuristic is proposed to find an approximated solution. Since in the dynamic setting of the problem, the operator has to respond in real-time in short duration, the insertion method should be fast enough to serve real-time requests.

Chapter 3 describes the two sub-problems in solving DARP problem in regardless of static and dynamic. In the routing sub-problem, cheapest insertion heuristic (CI) is introduced together with an exchange improvement. In the scheduling sub-problem, the assumptions and execution of three different strategies are discussed.

Chapter 4 explains how different strategies are implemented in solving the DDARP. The real-time requests can be inserted to the existing routes under different dynamic strategies.

Chapter 5 compares the results using the proposed strategies in scheduling decision. A set of DDARP test instances is generated by simulations. The result of the simulation using different dynamic strategies are compared and analyzed.

Chapter 6 summarizes the study and recommends for the further research.

The flow of the study is shown in Figure 1.1.

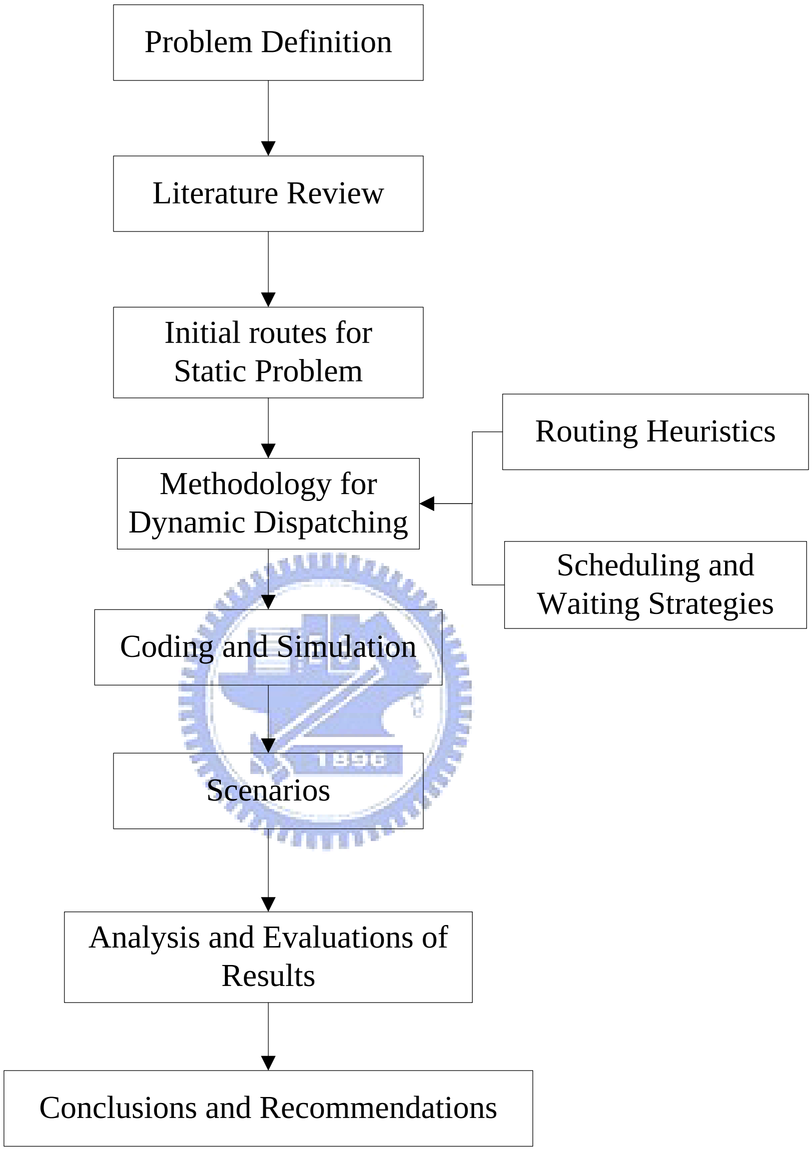

Figure 1.1 Flow of the study

CHAPTER 2
LITERATURE REVIEW

Section 2.1 describes the problem definition and formulation for the DARP problem. In addition, the difference between DARP and PDPTW problem will be explained later. In section 2.2, we review some efficient heuristics methods for solving the static DARP problem. As real-time requests appear during the operating period, the meta-heuristics methods may not be fast enough to find a solution, different routing methods to respond real-time requests will be discussed in section 2.3. Finally, we have made a summary in section 2.4.

2.1 Similarity and Dissimilarity of DARP and PDPTW

The DARP problems will be described in this section. In section 2.1.1, we describe the notation of variables which will be used in this study. In section 2.1.2, the similar mathematical formulation between DARP and PDPTW problems is given. Although the DARP and PDPTW are similar in mathematical formulation, they have their own characteristics. The comparison of two problems will be compared in section 2.1.3. In addition, the time windows setting for the problems will be discussed in section 2.1.4.

2.1.1 Notation of variables

In the DARP problem, the operator is responsible to provide services to customers. Once a customer *i* is going to travel from a pickup location O_i, delivery location D_i. According to the trip purpose, the customer can choose either a desired pickup time (DPT_i), for example, if he or she would like to return home from shopping; or a desired delivery time (DDT_i), for example, if he or she goes to school from home. The time window constraints for each request can then be expanded.

To guarantee the quality of service, this information can be used to generate a set of time window constraints that must be satisfied by the vehicle route. First, the direct ride time (DRT_i) can be calculated between the origin and destination under operating speed without deviation; the maximum ride time (MRT_i) has a equation in terms of DRT_i. Since the operator cannot arrive at the location for each request with a desired time, a maximum wait state for the desired pickup or delivery for each user is assumed to be a constant, WS, that the time for serving the locations are acceptable within the time interval. Finally, the boundary of the pickup time, earliest pickup time (EPT_i) and latest pickup time (LPT_i), and the delivery time, earliest delivery time (EDT_i) and latest delivery time (LDT_i) can be expressed as a function of DPT_i or DDT_i, MRT_i, DRT_i, WS. In the view of considering each request as a paired of pickup and delivery locations, each location j would have a service time interval [a_j, b_j]. When considering pickup location of the request i, [a_j, b_j] refers to the time window of [EPT_i, LPT_i]. However, [a_j, b_j] refers to the time window of [EDT_i, LDT_i] for delivery location of the request i. M denotes as number of operating vehicles serving the total requests in the system. A_j and D_j are the arrival time and departure time of location j for the vehicle. The notation of the set of data is shown in Table 2.1. These notations will facilitate the description of mathematical formulation that follows.

Table 2.1 Notations of input of customer requests

O_i	Origin location of request *i*
D_i	Destination location of request *i*
DPT_i	Desired pickup time of request *i*
DDT_i	Desired delivery time of request *i*
WS	Maximum wait state for pickup or delivery
DRT_i	Direct ride time of request *i*
MRT_i	Maximum ride time of request *i*
EPT_i	Earliest pickup time of request *i*
LPT_i	Latest pickup time of request *i*
EDT_i	Earliest delivery time of request *i*
LDT_i	Latest delivery time of request *i*
a_j	Earliest service time of location *j*
b_j	Latest service time of location *j*
M	Number of operating vehicles
A_j	Arrival time of the vehicle at location *j*
D_j	Departure time of vehicle at location *j*

2.1.2 Mathematical formulation

Since the mathematical formulations between DARP and PDPTW problems are similar, we modified this formulation from a survey of PDPTW by Mitrovic-Minic (1998).

There are some conditions that should be satisfied in a solution of the DARP.

(a) A pair of pickup and delivery locations for a request is served by the same vehicle (pairing constraint)

(b) For a pair of pickup and delivery locations, the pickup is always served before the delivery (precedence constraint)

(c) A vehicle can be waited at its initial location or at any pickup or delivery location

(d) All of the time windows constraints must be satisfied (hard time window constraint)

(e) The total distance traveled by the operating vehicles is minimized

Assume a complete graph $G = (V, A)$, where $V = \{O_1, O_2, \ldots, O_N, D_1, D_2, \ldots, D_N\}$ and $A = \{(x_i, x_j) : x_i, x_j \in V, i \neq j\}$ in Euclidean distance. Each vertex pair (O_N, D_N) stands for request transportation from origin O_N to destination D_N.

In this problem, the operator serve the requests by using M vehicles, which start from and end at a fixed location, depot. Let N customers are known in advance, the operator should provide services for those customers with minimum operating costs.

For simplicity, we do not discuss the properties for the pickup or delivery locations in this section, but consider on using efficient strategies to generate paths to serve all the locations.

The mathematical formulation of DARP problem is shown below. In the static problem, the operator has a resource of a set of m vehicles, M, with a fixed capacity of C^m. The vehicles start and end at a depot, $d(m)$, to serve N known customers. For a request i asks for service, he or she would have a pair locations, origin O_i and destination D_i. In the views of service locations, the time windows at each location j

must be within [a_j, b_j], which refers to Table 2.1 as the earliest pickup or delivery time and the latest pickup or delivery time according to the properties of the locations, respectively, with a load of l_j units. For each different stop locations, j and k, the direct travel time and travel cost of them are t_{jk} and c_{jk}, respectively. Set P^+ is a set of all pickup locations, when P^- denotes as a set of all delivery locations and $P = P^+ \cup P^-$. Q includes P and the location of the depot. Three types of variables are also used: first, the binary flow variables $X^m_{j,k}$ which will be set to 1 when the vehicle serves from location j to k. Next, T_j is the time when finished services at location j. Also, the variable L_j is the load after location j is serviced.

Min

$$C_m \sum_{j \in P^+} X^m_{d(m),j} + \sum_{m \in M} \sum_{i,j \in Q} c_{j,k} X^m_{j,k} \tag{1}$$

Subject to

$$\sum_{m \in M} \sum_{j,k \in Q} X^m_{j,k} = 1 \qquad j \in P^+ \tag{2}$$

$$\sum_{k \in Q} X^m_{j,k} - \sum_{k \in Q} X^m_{k,j} = 0 \qquad j \in P, m \in M \tag{3}$$

$$\sum_{j \in P^+} X^m_{d(m),j} = 1 \qquad m \in M \tag{4}$$

$$\sum_{j \in P^-} X^m_{j,d(m)} = 1 \qquad m \in M \tag{5}$$

$$\sum_{j \in Q} X^m_{O_i,j} - \sum_{j \in Q} X^m_{j,D_i} = 0 \qquad O_i \in P^+, m \in M \tag{6}$$

$$T^m_{O_i} + t^m_{O_i,D_i} \le T^m_{D_i} \qquad O_i \in P^+, D_i \in P^-, m \in M \tag{7}$$

$$X^m_{j,k}(T^m_j + t^m_{j,k}) \le T^m_k \qquad j,k \in P, m \in M \tag{8}$$

$$X^m_{d(m),j}(T^m_{d(m)} + t^m_{d(m),j}) \le T^m_j \qquad j \in P^+, m \in M \tag{9}$$

$$X^m_{j,d(m)}(T^m_j + t^m_{j,d(m)}) \le T^m_{d(m)} \qquad j \in P^-, m \in M \tag{10}$$

$$a_j \le T^m_j \le b_j \qquad j \in P, m \in M \tag{11}$$

$$a_{d(m)} \le T^m_{d(m)} \le b_{d(m)} \qquad m \in M \tag{12}$$

$$X^m_{j,k}(L^m_j + l_k) = L^m_k \qquad j \in P, k \in P^+, m \in M \tag{13}$$

$$X^m_{j,k}(L^m_j - l_{k-N}) = L^m_k \qquad j \in P, k \in P^-, m \in M \tag{14}$$

$$X^m_{d(m),j}(L^m_{d(m)} + l_j) = L^m_j \qquad j \in P^+, m \in M \tag{15}$$

$$L_{d(m)}^{m} = 0 \qquad m \in M \qquad (16)$$

$$l_j \le L_j^m \le C^m \qquad j \in P^+, m \in M \qquad (17)$$

$$X_{j,k}^{m} \in \{0,1\} \qquad j,k \in N, m \in M \qquad (18)$$

Equation (1) is the objective to minimize total operating cost containing number of operating vehicles and total travel distance. $\sum_{j \in P^+} X_{d(m),j}^{m}$ represents that total number of operating vehicles are planned to depart from the depot, that is, number of operating vehicles required for providing services. $X_{j,k}^{m}$ will be discussed in Equation (18). C_m is the fixed cost of operating vehicles and $C_{j,k}$ is the cost travel from location j to location k; therefore, C_m is much larger than $C_{j,k}$.

The descriptions the constraints are as follow:

Equation (2) to Equation (5) is the flow constraints, Equation (6) is the pairing constraints, Equation (7) is the precedence constraints, Equation (8) to Equation (10) are the compatibility between routes and schedules, Equation (11) and Equation (12) are the time windows constraints, Equation (13) to Equation (15) are the compatibility between routes and capacity of the vehicles and Equation (16) and Equation (17) are the capacity constraints. Descriptions of each equation are as below.

Equation (2) Vehicle m is in the location j and goes to a unique stop location k.

Equation (3) Number of vehicles enter and leave location j.

Equation (4) When the vehicle m starts from the depot, the next location must all be a pickup point.

Equation (5) When the vehicle m is towards the depot, the precedent location should be a delivery point.

Equation (6) Pairing constraints. When O_i is placed to the vehicle m, its relevant destination, D_i should also be existed in the same vehicle.

Equation (7) Precedence constraints. The time of the vehicle m serves the destination D_i must be less than or equal to the time serves at the associated origin Oi plus the direct ride time between O_i and D_i.

Equation (8) When the vehicle m leaves from j and goes to k, the time for that vehicle at k will be large than or equal to the time of the vehicle serves at location j plus the direct ride time between j and k.

Equation (9) When the vehicle m starts from the depot and goes to a pickup point location j, the time for that vehicle at j will be large than or equal to the time of the vehicle starts at the depot plus the direct ride time between $d(m)$ and j.

Equation (10) When the vehicle m finishes serving the last delivery location j and goes back to the depot $d(m)$, the time for that vehicle at j will be large than or equal to the time of the vehicle serves at location j plus the direct ride time between j and $d(m)$.

Equation (11) to (12) The time of the vehicle m arrive to the indicated locations must be in the boundary of $[a_j, b_j]$.

Equation (13) When it is the situation from location j to k which is a pickup point, the total capacity of the vehicle at j must be the total load at location j plus the load at location k.

Equation (14) When the vehicle leaves j and goes to k which is a delivery point, the total capacity of the vehicle at k is the total load at location j minus the load at location k.

Equation (15) When the vehicle starts at the depot $d(m)$ towards a pickup location j, the total capacity of the vehicle at j is the total load at $d(m)$ plus the load at location j.

Equation (16) The total load for the vehicle m when departed from the depot is 0.

Equation (17) The total load in location j must be larger than or equal to the load at location j and less than or equal to the capacity limit of the vehicle m.

Equation (18) When the route passes from j to k using vehicle m, the value $X_{j,k}^{m}$ will be 1, otherwise the value will be 0.

2.1.3 Comparison of the problems

This section compares the differences between the DARP and PDPTW problems. DARP is a generalized PDPTW problem, while their mathematical formulations are similar. Nevertheless, DARP focuses on carrying passengers while PDPTW is for courier services. The comparison between both of the problems is shown in Table 2.2. They are mainly different in zone size, maximum delivery duration, capacity constraints and time windows constraints.

Table 2.2 Comparison between DARP and PDPTW problems

	DARP	PDPTW
Formulations	Mathematically similar (as shown in equations (1) to (18))	
Capacity constraints	Number of seats of vehicles	Size of vehicles
Time window constraints	User can specify either DPT or DDT	User specified DDT
Maximum delivery duration	Specified by the system e.g. MRT is a function of DRT	Specified by the customer e.g. urgent and non-urgent parcels
Service Area	Within a city	With a distance reached in a reasonable working hour (e.g. 8 hrs)

(1) Capacity Constraints

As DARP focuses on carrying passengers, the capacity for the vehicles is the number of seats for passengers. However, PDPTW aims to carry as many freight as possible in a reasonable service time, and the capacity for the vehicles is size of the vehicle that can carry more freights in operating period.

(2) Time Windows Constraints

It is possible in DARP to allow passengers to specify their desired pickup time (DPT) or desired delivery time (DDT) according to their trip purposes, e.g. passengers specify DPT when return home from shopping or specify DDT as they go to see a movie. In contrast, customers normally only specify their desired delivery time for courier services. Generally, customers can ask for the *k*-h delivery services, e.g. 4-h delivery services, the parcels are acceptable to be delivered within the *k* hour period.

(3) Maximum Delivery Duration

In DARP, passengers specify their origin and destination, also their DDT or DPT The operator then expand the maximum delivery duration under their operating speed, i.e. the MRT is a function of DRT.

In PDPTW, when the customers specify the *k*-h delivery services, the *k* period will be the maximum delivery duration for deliveries. The decision for *k* hour durations are usually a consideration of parcels urgency.

(4) Service Area

DARP usually works within a city. Since passengers do not want to take a public vehicle mode for a long time, DARP is rarely successful for long distance transports. Instead, short distance transports have potential to attract passengers to take services, e.g. similar to taxi for door-to-door services but cheaper in price.

PDPTW works with a distance reached in a reasonable working hour. Customers specify the origin and destination for delivery services, the distance between them can be either long or short, the total time window period will also be long or short according to the properties of the parcels, e.g. urgent or non-urgent parcels.

2.1.4 Time windows setting

Owing to the difference time window settings between DARP and PDPTW problems, this section will describe the setting of both problems. In section 2.1.4.1, time windows setting in DARP is illustrated. On the other hand, the time windows setting in PDPTW problem is followed in section 2.1.4.2.

2.1.4.1 Time windows setting in DARP

For the definition of the notations, readers can refer to Table 2.1 in section 2.1.1.1 We adopt the settings by Diana and Dessouky (2004). Customers can choose either a desired pickup time (*DDT*) or a desired delivery time (*DDT*) in this problem and a pair of origin-destination locations with their trip purposes. For example, when customer *i* wants to return home from shopping, he or she would specify a desired pickup time (DPT_i), or the customer wants to present a specific place at a specific time, such as goes to see a movie at a fixed time, he or she would like to arrive to the destination at a specific delivery time, the desired delivery time for user *i* (DDT_i) will be formed. The calculations for the time windows of the request will be described in follow.

The direct ride time (DRT_i) can first be calculated by the direct distance from the origin to the destination of customer *i* under a fixed operating speed. Second, the maximum ride time (MRT_i) for the customer is then assumed to be obtained by equation (19).

$$MRT_i = \max(B + A \times DRT_i, DRT_i + WS) \quad (19)$$

where *A* and *B* are user-specified constants (e.g. *A* = 2, *B* = 20 minutes); therefore, MRT_i is a linear function of DRT_i. *WS* is the maximum waiting state for pickup or delivery locations as introduced in Table 2.1. (e.g. *WS* = 30 minutes). Generally, the MRT_i can be obtained by the equation on the left side; however, MRT_i may be smaller than *WS* in some cases using that equation, for example, the MRT_i is 20 minutes where *WS* is 30 minutes; therefore, the constraint of *WS* becomes meaningless in this situation. To avoid MRT_i becoming smaller than *WS* in some cases, another equation is used where *WS* is a function of MRT_i.

A set of time windows, EPT_i, LPT_i, EDT_i, LDT_i, for customer *i* can be obtained by using the DPT_i or DDT_i, MRT_i and DRT_i. *WS* is maximum waiting state for each request. Figure 2.1 shows the time windows for the user *i* who specified earliest pickup time, where the time windows could be computed as

$$EPT_i = DPT_i \quad (20)$$

$$LPT_i = EPT_i + WS \quad (21)$$

$$EDT_i = EPT_i + DRT_i \tag{22}$$

$$LDT_i = EPT_i + MRT_i \tag{23}$$

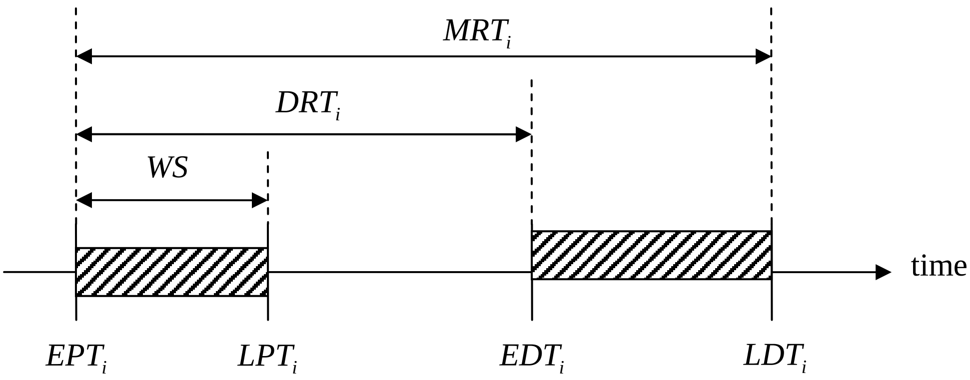

Figure 2.1 Time windows for the DPT-specified requests

Figure 2.2 shows the time windows for the request *i* who specified latest delivery time, in which the time windows could be computed as

$$LDT_i = DDT_i \tag{24}$$

$$EDT_i = LDT_i - WS \tag{25}$$

$$LPT_i = LDT_i - DRT_i \tag{26}$$

$$EPT_i = LDT_i - MRT_i \tag{27}$$

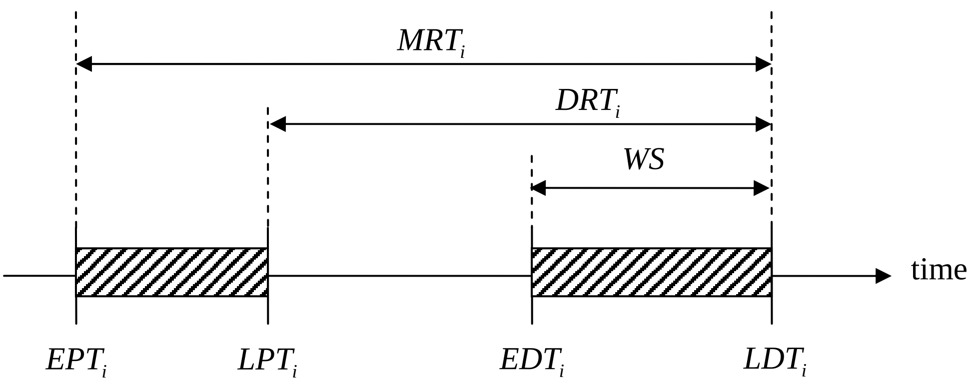

Figure 2.2 Time windows for the DDT-specified requests

2.1.4.2 Time Windows Setting in PDPTW

We choose Mitrovic-Minic and Laporte (2004) as an instance of generating time windows for PDPTW problems. When concerning courier services, operators divide the maximal service time into *k*-hour service type, including 1-h requests, 2-h requests and 4-h requests. Customers first indicate their origins and destinations, as well as the length of duration for their requests.

Figure 2.3 shows the setup method for this scenery for the customer *i*. As a vehicle is assigned to serve the customer *i*, it has to finish the pickup and delivery services within the maximum service time k_i, which is similar to the MRT_i in DARP problem. Again, the DRT_i is obtained by the direct distance and operating speed. Compared to one of the time window settings in DARP problem, the time windows here are only considered to be the relationships between the EPT_i, MRT_i and DRT_i. The EPT_i is the time when the request occurs, the LDT_i will be the EPT_i plus *k*-hour, $LDT_i = EPT_i + k_i$. The EDT_i can be calculated by EPT_i and DRT_i, that is, $EDT_i = EPT_i + DRT_i$. Finally, the LPT_i can be found as $LPT_i = LDT_i - DRT_i$.

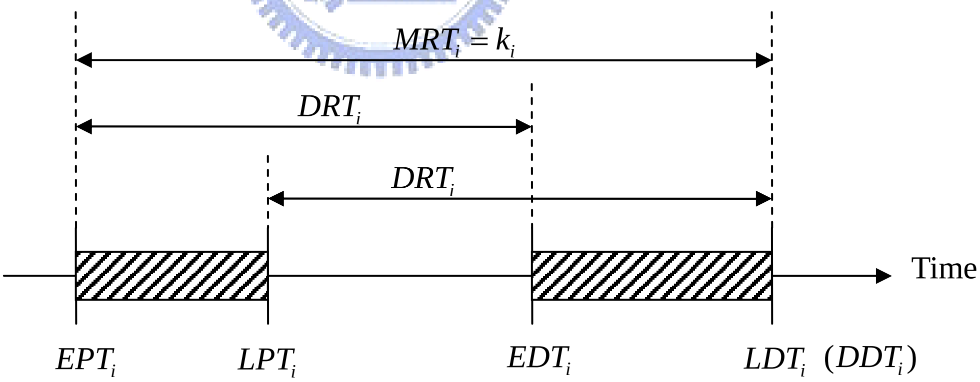

Figure 2.3 An example of setting up time windows in PDPTW problems

2.2 Static Problems

DARP problem is an NP-hard problem and difficult to solve because of its paired constraints and precedence constraints. Psarafits (1983) proposed a dynamic programming approach that could find an exact solution, but the problem size was limited to less than and equal to 9 customers with user-specified time windows on departure and arrival time. In practice, the customer sizes must be larger, it would be hundreds or even thousands requests combined with PDP pairs and also time window constraints, therefore, this problem cannot be solved for an exact solution in polynomial time (Mitrovic-Minic 1998). However, some papers used heuristic methods to find nearly exact solution in short time.

Heuristics for solving the Dial-a-ride problem have been studied for a number of decades. The examples of the heuristics method are cheapest insertion heuristic (Jaw et al. 1986; Madsen et al. 1995), new regret heuristic (Diana and Dessouky 2004) and tabu heuristic (Cordeau and Laporte 2003a; Mitrovic-Minic et al. 2004)

One of the mostly cited references in this area is due to Jaw et al. (1986). They proposed the Advanced Dial-A-Ride with Time Windows (ADARTW) algorithm for the Dial-A-Ride Problem (DARP) of advance-request, with multiple vehicles and service quality constrains. In their algorithm, the time windows for each customer *i* are determined with the customers specified desired pick-up time (DPT_i) or desired delivery time (DDT_i), by calculating the values of earliest pick-up time (EPT_i), latest pick-up time (LPT_i), earliest delivery time (EDT_i), latest delivery time (LDT_i). In their assumptions, the vehicle capacity was limited, and loading and unloading time were specified for picking-up and delivering customers and the vehicles were not allowed to idle when carrying customers. In the algorithm, for a number of customers, the sequence of insertion was labeled with the order of earliest pick-up times EPT_i (i=1,…,N). Starting with a single vehicle fleet size, each customer was inserted to the position which incurred the smallest cost among all feasible insertion positions, concerned the parameters of disutility to the system's customers and operator costs. If a customer was infeasible to be assigned to any of the vehicle fleet, an additional vehicle would be introduced. The algorithm finally terminated when all customers were inserted to the vehicles.

Diana and Dessouky (2004) adopted and modified the time settings by Jaw et al. (1986), and formulated their static DARP with time windows without capacity constraints. They developed a route initialization procedure which exclusively keeps into account the spatial and temporal effects of the demand, and a parallel regret insertion heuristic to improve some degree of flexibility for further insertions. Instead of ranking the requests with a certain criteria, for example, earliest pickup time or latest delivery time as in classic insertion heuristics, the regret insertion builds up an incremental cost matrix for each of the unassigned requests assigning to each of the existing vehicle routes. A regret cost, which is a measure of the potential difficulty if a request is not immediately assigned, is calculated for each request, and the algorithm seeks for the one with the largest regret cost, and inserts it into the existing schedules. The whole procedure is repeated until all requests are inserted.

Cordeau and Laporte (2003a) proposed a tabu search heuristic for the dial-a-ride problem. Customers specified their requests for origins and destinations, also their time window on the arrival time and departure time of their outbound trip and inbound trip respectively. The model aimed to design a set of vehicle routes, with a supplied fleet size, to satisfy all requests with least operating cost. The algorithm started from an initial feasible solution s_0, and when the best solution was found in a neighborhood $N(s_t)$ at iteration t, the new solution was changed to s_t. Besides, the recent visited solutions were declared forbidden for a number of iterations so as to avoid cycling, unless they contributed a new incumbent. When the solution was being searched, the time window and vehicle capacity constraints were allowed to violate. If the current solution was feasible with the constraints, the cost function was re-calculated by dividing the cost parameters, otherwise by multiplying them. Finally, the best feasible solution can be reached by repeating the several iterations.

In the problem of not considering time window constraints, Tseng (1992) proposed three different insertion criteria and eight dispatching headways types, for a case study of Science Park Administration in Hsin-Chu, Taiwan. The three different insertions, namely minimum incremental time, minimum incremental distance and minimum incremental cost, were compared, and the minimum incremental time would be a better heuristic algorithm taking account of operating cost and level of service. In the division of 43 zones in the case study, eight dispatch types were presented, including dispatch a vehicle with certain time intervals, with certain accumulated requests and combined with both the decisions. The Simulation Language for Alternative Modeling (SLAM) was used to evaluate the result. Finally, the combined certain time intervals and accumulated requests with minimum incremental time algorithm led a better service type in the simulation.

Cordeau and Laporte (2003b) surveyed over 30 publications and showed a review on the features and variants of the Dial-a-Ride problem. They summarized the important algorithms which have been published over the last thirty years in static and dynamic problems with single- or multi-vehicle type. They concluded that excellent heuristics were existed to solve static problems, but dynamic problems are rarely studied. Since DARP is focused on carrying people, the level of service is an important index for operation. Combined with the intelligent technologies in order to respond new requests in real-time, the operating service level will be enhanced. Therefore, solving DDARP will be an important issue in practice.

2.3 Dynamic Problems

DDARP are relatively less studied before the development of information and communication technologies (ICT) recently. There are only a limited number of pioneering studies focused on this problem. Psarafits (1980) formulated a dynamic programming problem in solving single vehicle DARP with minimizing the operating time and passenger dissatisfaction. In their problem, time windows were not considered.

Madsen et al. (1995) proposed an insertion heuristic algorithm, REBUS, to solve a real case of Copenhagen Fire-Fighting Service (CFFS), which is a static dial-a-ride routing and scheduling problem with time windows (DARPTW) with multiple capacities and objectives. The problem considered the multi-dimensional capacity of vehicles, i.e. the seats in the vehicle could be changed to ordinary seats, lying seats, children seats, wheel chair places, bed places according to different conditions. Several performance terms were concerned in the objective, including minimum total driving time, number of vehicles, total waiting time, deviation from promised service and cost. In the insertion procedure of REBUS, jobs were sorted with the difficulty of insertion, such as jobs with narrow time window, long travel time and requirement of spaces. The feasible insertion of the jobs to the vehicle considered several parameters, such as jobs driving time, waiting time, deviation from desired service time and capacity utilization. REBUS is developed based on ADARPTW in Jaw et al. (1986). In the comparison, REBUS algorithm first sorted the unassigned jobs with costs. Therefore, REBUS could handle multiple capacities and objectives than ADARTW; it could also implement in solving the dynamic problem.

Larsen (2000) defined the degree of dynamism (*dod*) to measure the dynamism of dynamic requests. They followed Lund et al. (1996) where *dod* is a ratio of number of dynamic requests to the number of total requests in the network as in (28).

$$dod = \frac{Number\ of\ dynamic\ requests}{Number\ of\ total\ requests} \tag{28}$$

Once the vehicles first satisfy all the static requests, the dynamic requests are inserted to the routes without changing the planned locations. The dynamic requests may not be all satisfied under operating period. To keep the level of operating services, a deadline is usually used to avoid the requests arriving too late and close to the end of the operation.

Mitrovic-Minic et al. (2004) developed a double-horizon heuristics to solve the dynamic PDPTW problem focused on courier services. The heuristics contained short-term and long-term procedures. They first used cheapest insertion procedure to insert the requests in short-term period. In order to make the routes to respond the dynamic requests easier, they also improved the routes by using tabu improvement procedure in long-term.

Another topic for scheduling decision was shown in Mitrovic-Minic and Laporte (2004). Four waiting strategies for dynamic pickup and delivery problems with time windows (PDPTW) were introduced. The strategies were Drive-First (DF), Wait-First (WF), Dynamic Waiting (DW) and Advanced Dynamic Waiting (ADW). They focused on the comparison of the strategies which produced shorter routes for the experiments. DF suggested a vehicle to drive from its current location at the earliest departure time, when WF required it to wait at its current location and left at the latest possible departure time of the next location. In considering the service zone, DW and

ADW were proposed. DW was a combination of DF and WF, it suggested that vehicle drove within each service zone with DF strategy, and use WF strategy when the vehicle finished serving all locations in the zone. However, in contrast to DW, ADW took the longest feasible WF strategy at the end of each service zone. Finally, the strategies were compared with several experiments. In the experiments, the sequence of locations were first determined on each vehicle routes, by using initial cheapest insertion heuristic and tabu search improvement. The arrival and departure time for each of the locations were then solved with one of the waiting strategies. In overall, the ADW resulted in the best results out of the four strategies.

Other newly methods solving Dynamic Problems can be referred to Branke et al. (2005) and Coslovich et al. (2006).

2.4 Summary

This chapter first presented the problems of the DARP and PDPTW. As DARP is a generalization of PDPTW problem, they are in similar mathematical formulation in static problem. However, DARP problems, which focus on carrying passengers, and PDPTW problems, which focus on courier services, are different in problem settings.

The literature reviews for DARP problems are then followed. Since it is a NP-hard problem, the DARP problem containing the paired constraints and precedence constraints is difficult to find an exact solution. In static problems, DARP problem can be solved by heuristics methods, such as cheapest insertion method, tabu-search heuristics and other meta-heuristic methods. In dynamic problems, some ideas solving dynamic PDP paired problems were reviewed. First, an idea of degree of dynamism which will be adopted in our studies was introduced. Another idea using the strategies to solve dynamic PDPTW problem were studied. As the use of efficient strategy solving dynamic PDPTW problem may not be efficient in solving DDARP. Therefore, a new strategy is proposed to solve DDARP.

CHAPTER 3
ROUTING AND SCHEDULING OF DARP

There are two components in solving DDARP: routing and scheduling components (Cordeau and Laporte 2003b). Routing is a sub-problem of route construction where an ordered sequence of locations is planned into vehicle routes. Scheduling consists of using waiting strategies to determine the arrival and departure times for each location *i* along the route.

For the requests that are known before planning, routing component is used to construct the routes with minimum operating costs, and then the vehicles can be scheduled to visit all the locations. In this case the scheduling component does not affect the overall solution, since we are only interested in the sequence of stops on the route.

When the real-time requests arrive as the vehicles are serving the requests on the routes, the routing and scheduling processes will be executed again to put the dynamic requests to the existing routes. However, scheduling with different strategies will be different in the situations of the waiting time, which is an important issue when our objective is to insert as many real-time requests as possible in the future. Therefore, different strategies will be different in result of total number of vehicles and total travel distance when serving the same set of requests.

The routing and scheduling, which are the decision factors that make the appropriate vehicle serving the requests, are discussed in section 3.1 and 3.2. Finally, the summary of this chapter is presented in section 3.3.

3.1 Route Construction and Improvement Methods

Routing is the procedure to decide where the requests are placed in a route. Section 3.1.1, the cheapest insertion (CI) heuristic method is introduced. In addition, section 3.1.2 presents an exchange method to improve the solution by cheapest insertion heuristic method.

3.1.1 Route construction: cheapest insertion heuristic

In this section we will present the insertion heuristic method which is first proposed by Jaw et al. (1986). It is a quick and simple algorithm with inserting requests to routes. On the other hand, it is known that the solution can be improved by using meta-heuristic methods, e.g. tabu heuristic improvement (Cordeau and Laporte 2003a) have been proposed to solve the problem. They provide a better result than insertion heuristic methods with longer computing time. However, as the dynamic requests are revealed in real-time, the requests have to be inserted into the existed routes as soon as possible. Therefore, the meta-heuristic methods may not be a suitable method in solving dynamic problem. The simple cheapest insertion is the suitable method in react to the real-time requests, and therefore, will be considered in this study.

The procedure of the insertion heuristic is shown in Table 3.1. First of all, the operator use one vehicle to serve the requests with M = 1. For each request i who specified his or her O_i, D_i and DPT_i or DDT_i in a set of static problems, the operator calculates DRT_i, MRT_i, EPT_i, LPT_i, EDT_i and LDT_i using the equations (19) to (27). The requests are then sorted with the EPT in ascending order. This sorting can generate better result in insertion method.

For each of the requests that has not been inserted, they are considered to be inserted in sequence into a suitable position without violating the time windows. If there are more than one vehicle that could feasibly serve the request, a comparison between the vehicles is made for minimizing operating cost, and the request is inserted into the best position. If there is no vehicle feasibly serving the request, it will be served by a new vehicle. The total fleet will therefore become $M = M + 1$. In addition, the capacity is usually non-binding in the capacity of 9 passengers in maximum. The insertion heuristics is finally terminated when all the static requests are satisfied.

Table 3.1 Pseudo code for routing procedure

```
SET M = 1
GIVEN O_i, D_i and DPT_i or DDT_i for i ∈ N and user-specified constant A, B and WS
CALCULATE DRT_i and MRT_i by equation (19)
CALCULATE EPT_i, LPT_i, EDT_i and LDT_i using equations (20) to (27)
SORT the requests with the earliest pickup time in ascending order
FOR each request i
    FOR each vehicle m in the fleet size M
        FIND all feasible positions of insertion that do not violate the time windows
        FIND the ( best ) position in minimizing the operating costs
    ENDFOR
    IF there are any ( best ) position is found THEN
        FIND the vehicle m with the minimal additional cost
        INSERT request i to vehicle m
    ELSE
        SET a new vehicle to the fleet size ( M = M + 1 )
        INSERT the request to the new vehicle
    ENDIF
ENDFOR
```

3.1.2 Route improvement: exchange method

To our assumptions, the solution can be improved by using an exchange method only for initial route constructions. We assume that the customers are informed to be served by a fixed vehicle, that is, the customers will be picked up by the vehicles with pre-known vehicle identification numbers. This information can be changed before the vehicles start visiting the requests. Cheapest insertion heuristics provides a quick but simple solution; however, a better solution can be obtained by removing and re-inserting the requests one by one. Similar algorithm can be referred to the “Trip Insertion” of Toth and Vigo (1997). Cheapest insertion with exchange method can be used to improve the routes of static requests, which potentially provide more flexible insertion in responding to real-time requests. The pseudo code for the cheapest insertion with exchange method is shown in Table 3.2.

The idea of the trip insertion can be explained as follows:

1. Remove a pair of locations (i^+, i^-) from a vehicle *j*.
2. Compare the cost to the minimum cost if it is served by another vehicle *m*.
3. Re-insert the request to another vehicle which feasibly serves the request with a smaller cost.

Table 3.2 Pseudo code of the exchange method

```
FOR each request i
    REMOVE from the inserted vehicle j
    FOR each vehicle m in the fleet size M
        FIND all feasible positions of insertion that do not violate the time windows
        FIND the ( best ) position in minimizing the operating costs
    ENDFOR
    IF there are other vehicles that can be feasible to serve the request i THEN
        INSERT request i to vehicle m with the minimal serving cost
    ELSE
        INSERT request i back to vehicle j
    ENDIF
ENDFOR
```

3.2 Scheduling and Waiting Strategies

A scheduling strategy designs the arrival time and departure time of each stop along the routes. A number of feasible solutions which satisfy the time window constraints are presented. In the scheduling, different strategies are included to evaluate the better service of quality, where the strategies affect on passenger waiting times.

The routing decision introduced in previous section plans the vehicles to serve a set of paired pickup and delivery locations. Once the routes are planned, the schedules will be similar to Figure 3.1 (a). The figure shows three locations in one route for an instance. For simplicity, we do not discuss the properties for the pickup or delivery locations in this section, but consider on using efficient strategies to generate paths to serve all the locations. Let the boundary [a_i, b_i] be the time window of a location i, where a_i refers to the earliest serving time and b_i is the latest serving time. Three strategies will be introduced in this section, namely Drive First (DF) strategy, Wait First (WF) strategy and Dynamic Wait (DW) strategy. Mitrovic-Minic and Laporte (2004) also proposed four waiting strategies solving Dynamic Pickup and Delivery Problem with time windows. The DF and WF strategies in this study are similar to their study. However, our proposed DW strategy, in which we concern the improvement of DF strategy, is different from their DW and ADW strategies, in which they concerned dynamic partitioning to form service zones. The planned route serving the locations is demonstrated in Figure 3.1 (b). According to the characteristics of different waiting strategies, operators construct a list of timetable to serve the locations. Therefore, the route of the vehicle will follow the line shows in the figure.

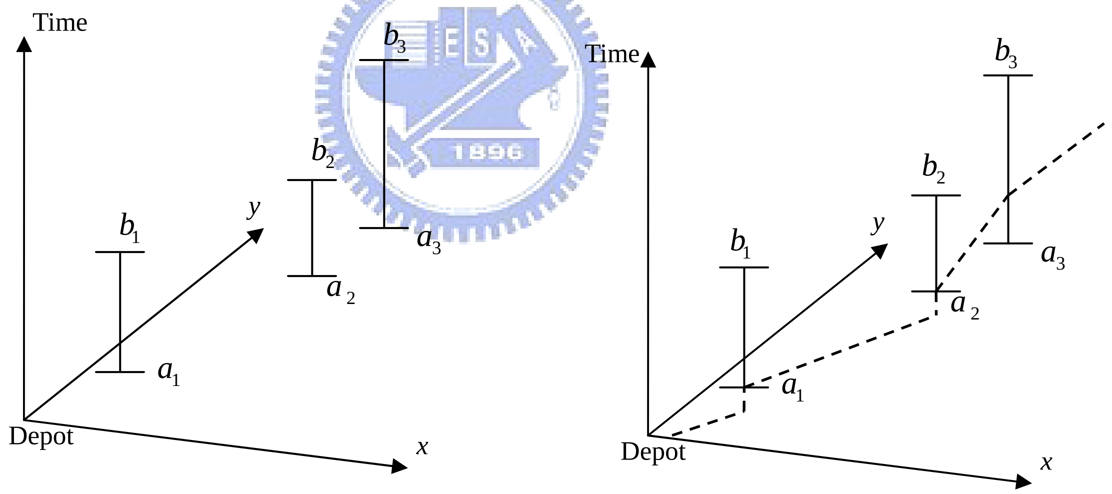

Figure 3.1 (a) Figure 3.1 (b)

Figure 3.1 A route represented in three-dimensional space: (a) locations on the route; (b) trajectory of the vehicle

In the above figures, the three-dimensional diagram is converted into the bases refer to the coordinates of the locations while the vertical axis is the time axis. However, the routes will be intricate when the locations become more. For simplicity, the two-dimensional space, in which the indicated locations represent the bases of the coordinate axes, is used to express the routes instead in three-dimensional space. Therefore, the x-axis in the two-dimensional space giving in Figure 3.2 represents location of requests but does not follow a distance scale.

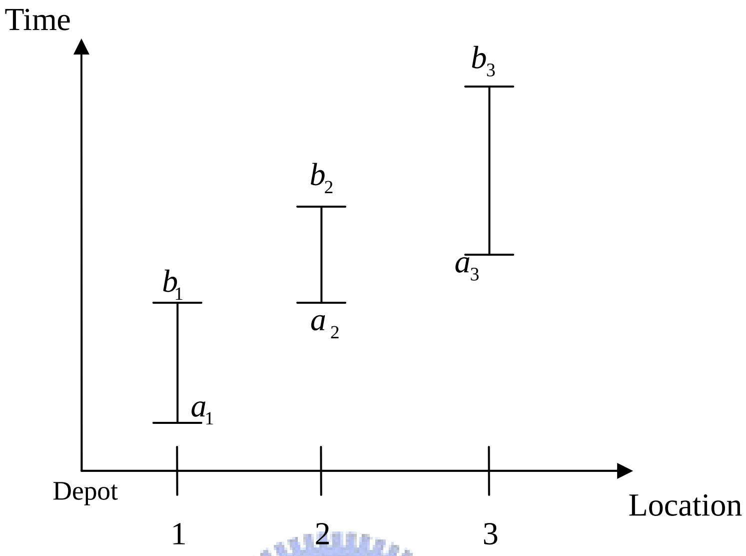

Figure 3.2 Locations on a route in a two-dimensional space

3.2.1 Drive First (DF) strategy

For the Drive First (DF) strategy, the vehicle is driven as soon as it can, where $\underline{A}_i$ is the arrival time and $\underline{D}_i$ is the departure time for DF strategy. Again, $[a_i, b_i]$ is the boundary of time windows that each stop can be served. If the arrival time $\underline{A}_i$ is earlier than the earliest serving time a_i, the vehicle has to wait at the arrived location until the start serving at a_i, then the vehicle departs at time $\underline{D}_i$. Therefore, $\underline{D}_i = a_i$, and waiting time is the difference between $\underline{D}_i$ and $\underline{A}_i$. On the other hand, if $\underline{A}_i$ is later than a_i, the vehicle starts serving and departs immediately at $\underline{D}_i$ with no wait. Figure 3.3 displays the DF strategy for a route of a vehicle. For example, the vehicle arrives at location 1 at $\underline{A}_1$, which is earlier than a_1, the vehicle then wait until start serving at a_1, therefore, the earliest departure time $\underline{D}_1$ will be at a_1, i.e. $\underline{D}_1 = a_1$. Furthermore, the waiting time for the vehicle at location 1 would be $\underline{D}_1 - \underline{A}_1$. When the vehicle arrive at location 3 at $\underline{A}_3$ which is later than a_3, the vehicle serve the location immediately and depart without waiting, therefore $\underline{A}_3 = \underline{D}_3$. The vehicle departs immediately to the depot after serving all the requests.

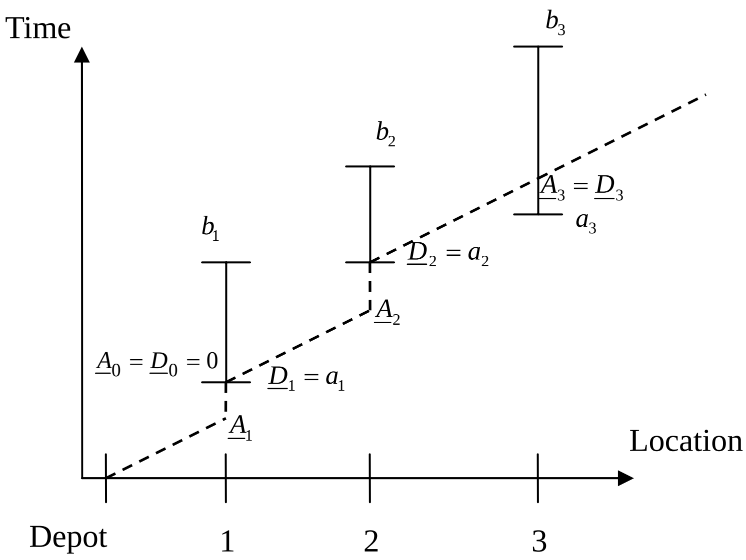

Figure 3.3 Trajectory of a vehicle using Drive First (DF) strategy

(SOURCE: Mitrovic-Minic and Laporte 2004)

Assume that there are N locations along a route. The vehicle, starting at a depot of location 0, has an initial arrival time and departure time of A_0 and D_0, respectively. The pseudo code for DF strategy is shown in Table 3.3. The arrival time and departure time for every location is therefore calculated by the criteria below. WT_i defines the time of wait of the vehicle at location i.

Table 3.3 Pseudo code for Drive First (DF) strategy

```
SET N = number of stops along the route
SET A0 = D0 = 0 for depot
FOR each location i      (for i = 1, 2, …, N)
    Ai = Di-1 + t(i-1), i
    IF Ai is earlier than ai THEN
        Di = ai
        WTi = Di - Ai
    ELSEIF Ai is larger than or equal to ai THEN
        Di = Ai
        WTi = 0
    ENDIF
ENDFOR
```

3.2.2 Wait First (WF) strategy

In the Wait First (WF) strategy, the vehicle stays to wait at its current location as long as it is feasible. $[a_i, b_i]$ is the boundary for serving each stop and the arrival time and departure time for this strategy denote as $\bar{A}_i$ and $\bar{D}_i$, respectively. When the latest arrival time in location i+1 is b_{i+1}, the vehicle waits at location i until the time plus the direct travel time to the location i+1 is at b_{i+1}. Figure 3.4 shows an example for a route using WF strategy. The latest arrival time for location 2 is b_2, the vehicle therefore waits at location 1 and departs at $\bar{D}_1$. $\bar{A}_1$ is then equal to b_1, The waiting time in location 1 is therefore $\bar{D}_1 - \bar{A}_1$.

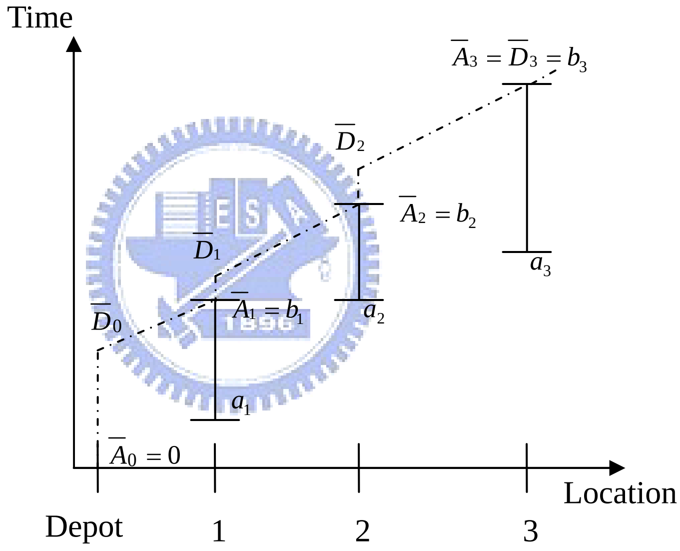

Figure 3.4 Trajectory of a vehicle using Wait First (WF) strategy

(SOURCE: Mitrovic-Minic and Laporte 2004)

The Pseudo code for WF strategy is shown in and Table 3.4. Again, we have N serving locations, where N+1 is an additional indicator for the depot when the vehicle is off duty. When an arrival time for one location is known, the departure time before that location can then be calculated. Let the time of "end of day" be A_{N+1}, the rest of the arrival time and departure time for all locations can be calculated as follows.

Table 3.4 Pseudo code for Wait First (WF) strategy

```
SET N = number of stops along the route
SET A_{N+1} = D_{N+1} = the time of "end of day" at the depot
FOR each location i     (for i = N+1, N-1, …, 1)
    D_{i-1} = A_i - t_{i,(i+1)}
    IF D_{i-1} is later than b_{i-1} THEN
        A_{i-1} = b_{i-1}
        WT_{i-1} = D_{i-1} - A_{i-1}
    ELSEIF D_{i-1} is smaller than or equal to b_{i-1} THEN
        A_{i-1} = D_{i-1}
        WT_{i-1} = 0
    ENDIF
ENDFOR
```

When the dynamic requests are considered, WF requires more number of vehicles but less total travel distance than DF. As WF requires vehicles waiting at the current location as long as possible, especially when the vehicles wait at depot, more requests will be known before the vehicles start routing the locations. Therefore, it has an advantage to improve the planned routes more efficiently than DF strategy. It results in shorter total travel distance than DF. However, WF wastes too much time to wait for the dynamic requests, the new requests may not be inserted to the existing routes because of violating the time windows, more number of vehicles are necessary to serve the same amount of requests than DF (Mitrovic-Minic and Laporte, 2004). On the other hand, DF needs longer travel distance in dynamic problems. When dynamic request reveals where the vehicle waits to serve a location, it is possible to use the waiting time to serve the new request and then go back to the location. Therefore, a detour will be existed in this situation. A strategy of dynamic waiting is proposed here to improve the efficiency of DF and WF for smaller number of vehicles and total travel distance.

3.2.3 Dynamic Wait (DW) strategy

For the reason that DF has a potential of less number of vehicles but longer total travel distance than WF in solving DDARP, an idea of Dynamic Waiting (DW) strategy which tries to combine the benefits of these two strategies is considered. The DW strategy is developed based on DF, since DF requires less number of vehicles than WF in dynamic from our numerical test that will be presented in CHAPTER 5. When the route is scheduled by DF, the arrival time, departure time and waiting time are obtained in every location. Avoided the chance of detour, the vehicle should not wait to serve the location, but could be wait after serving the location. Figure 3.5 shows the sketch for DW strategy. Compare to the DF strategy in Figure 3.3, location 2 has a time interval for the vehicle to wait until location 2 starts to be served at a_2. If a real-time request appears nearby this vehicle, it is allowed to serve the real-time request and then goes back to serve the request at location 2. A detour will therefore be generated. In order to eliminate the detour state, the time of wait in location 2 is planned to shift to location 1, the vehicle does not wait to serve location 2. Since the idle time is located at the served location 1, the vehicle will not go back to location 1 as real-time request appears because it has already been served. The difference between DW and WF is that DW departs from the served location when the direct time to the next location can be started serving where WF departs from the served location when the direct time to the location can be ended serving. DW is therefore similar in total number of vehicles as DF but in less total travel distance.

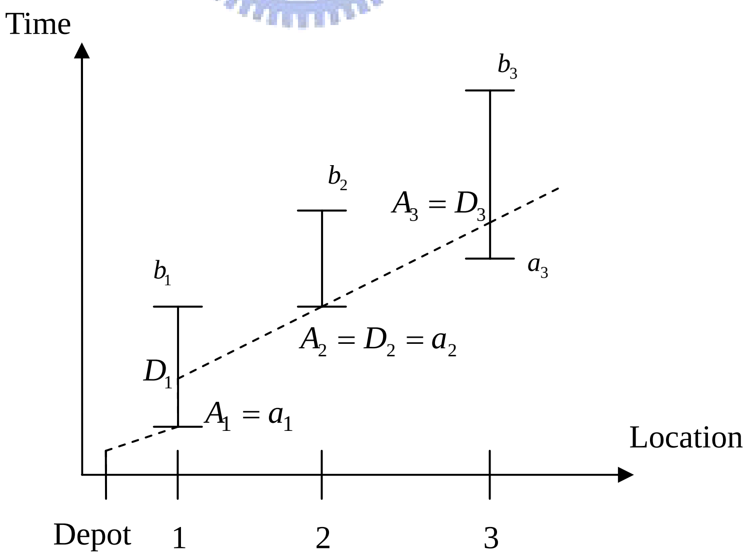

Figure 3.5 Trajectory of a vehicle using Dynamic Waiting (DW) strategy

Table 3.5 is a pseudo code for the DW strategy. The notations for the DW strategy are A_i, D_i and WT_i, representing arrival time, departure time and waiting time of location *i*, respectively.

Figure 3.6 shows the paths which are produced by using DF, WF and DW strategies. The two extreme strategies, DF and WF, construct a service interval, noted that any path suggested by a strategy within this interval is feasible. The dash line at the bottom of the interval stands for DF strategy when the dotted line at the top of it stands for WF strategy. Finally, the path with solid line in the interval is constructed by DW strategy.

Table 3.5 Pseudo code for Dynamic Wait (DW) strategy

```
FOR each location i        (for i = 1, 2, …, N+1)
    CALCULATE  A_i, D_i, WT_i  (underlined) from DF strategy
    IF  WT_i (underlined) > 0  THEN
        D_{i-1} = D_{i-1} (underlined) + WT_i (underlined)
        A_i = D_i (underlined)
        WT_{i-1} = WT_i (underlined)
    ELSE
        D_{i-1} = D_{i-1} (underlined)
        A_i = A_i (underlined)
        WT_{i-1} = WT_{i-1} (underlined)
    ENDIF
ENDFOR
```

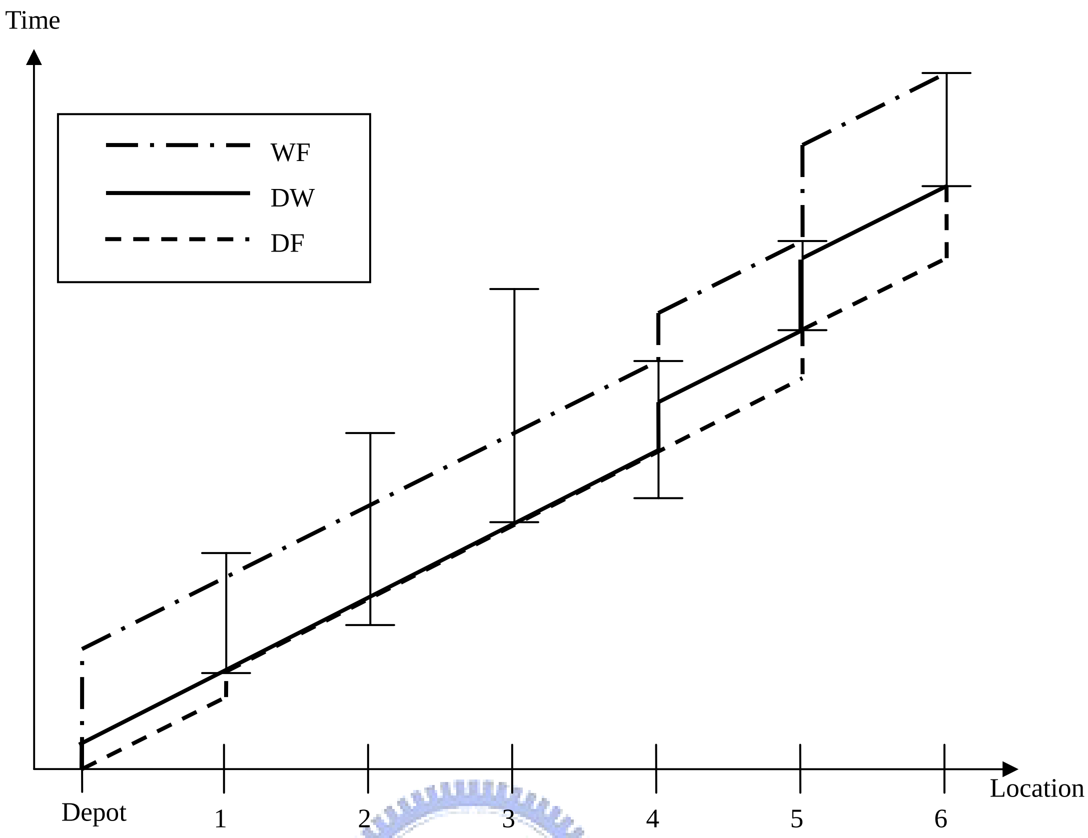

Figure 3.6　Trajectory of a vehicle using the three waiting strategies

3.3 Summary

This chapter showed how to solve DARP problems by two components, routing and scheduling. Routing is a decision placing the requests in suitable routes with several objectives. We present cheapest insertion method which is a quick and simple algorithm to insert the requests in the routes. Since we assume that the customers are informed to be served by a fixed vehicle, an exchange method is used to improve the solution only in static problem.

Scheduling considers waiting strategies to design the arrival time and departure time of each stop along the routes. The characteristics of three waiting strategies, Drive First (DF), Wait First (WF) and Dynamic Wait (DW) are introduced. Although their performances to serve the static requests are the same, they are different in respond to real-time requests. As DF assigns the vehicles driving to the service location as soon as possible, it results in more total travel distance because of detouring but less number of vehicles requirement. On the other hand, WF assigns the vehicle waiting at the service location as soon as possible, it results in more number of vehicles because of long waiting time at the locations but less total travel distance. In the objectives of minimizing number of vehicles and total travel distance, we propose DW strategy as an improvement of the DF strategy. Here we only show the rule and assumptions of the three waiting strategies. The issues of how to implement them in solving dynamic requests by dynamic dispatching will be discussed in the next chapter.

CHAPTER 4
DYNAMIC DISPATCHING USING DIFFERENT WAITING STRATEGIES

We focus on solving the dynamic problems in this section. In the static case in which some requests are known in advance, the operator plans routes with minimum operating cost to satisfy the known requests. In dynamic problem, however, idle time is the important issue to insert real-time requests to the existed routes. The purpose of the waiting strategies is to allocate the waiting time in response to the real-time requests. In section 4.1, the assumptions for the system's operating mode are introduced. Three cases for the vehicles states when dynamic requests arrive will be explained in section 4.2 and 4.3. Section 4.2 considers the situation that real time requests appear when vehicle is idling at one point, where it is either waiting to serve or waiting at the served corresponding location. Moreover, another case considering real time requests appear when the vehicle is busy on traveling to the next location will be presented in section 4.3. At last, we will summarize this chapter in section 4.4.

4.1 Dynamic Structure of the Problems

The operating mode as follows describes how the operation responses the real-time requests by dispatching vehicles using the waiting strategies in CHAPTER 3.

1. The requests which are known in advance are scheduled to a set of initial planned routes before the day of operation. The requests contain the origins and destinations of the passengers, with their specific pickup times or delivery times.
2. A dispatching office accepts the real-time requests. The time that a real-time request is received is the earliest time a vehicle can pickup the request.
3. The dispatch inserts the real-time requests received into the existing vehicle routes.
4. The drivers have a list of locations to visit with time windows but the list can be updated over time for real-time requests.
5. A new vehicle will be assigned to serve the real-time request which cannot be

satisfied by the existing operating vehicles. Meanwhile, the new vehicle becomes one of the operating vehicles.

6. The degree of dynamism (*dod*) can be obtained by the total number of static and dynamic requests throughout the day of operation. This can be used to estimate the demand distribution for operation in the future.

Figure 4.1 presents the structure for solving DDARP. First, the structure in the block on the left side shows the construction of initial routes with the known requests under the objectives of minimum number of vehicles and total travel distance. Second, scheduling is used to design the service time at every location with different waiting strategies. The structure in the block on the right side indicates how to respond the real-time requests with the existing operating vehicles. The new real-time requests are inserted to the existing routes for minimum number of vehicles and total travel distance. Scheduling is used to modify the route immediately when one request is inserted.

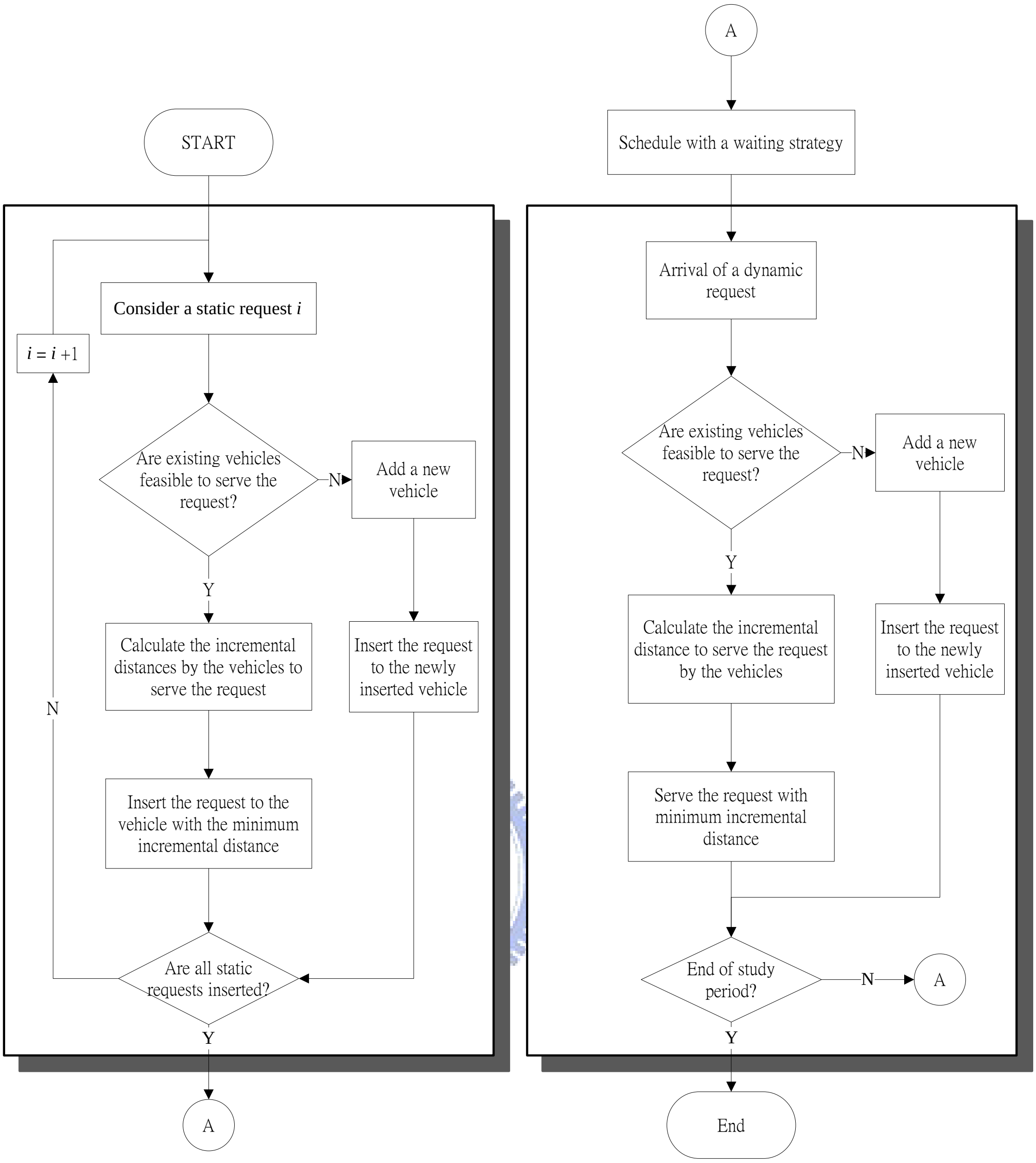

Figure 4.1 Operation of DARP in dynamic

When a dynamic request is received, vehicles will either be busy for traveling to the next location on the route or idling in a location. Table 4.1 shows the states for the vehicles in dynamic problem. Case 1 represents real-time requests appear when the vehicle idles at a non-serve or served location. Case 2 considers real-time requests appear when the vehicle is busy for traveling.

Case 1 is further divided into two parts, (a) before-service vehicles and (b) after-service vehicles. As in DF strategy, the vehicles are driven as soon as possible. They always arrive to a location earlier than or in the boundary of the time windows and then depart immediately; therefore, they hardly wait after serving locations. In WF strategy, the vehicles wait at locations as long as they are feasible. They always depart from their current locations in the boundary or later than the time windows; therefore, they hardly arrive at a location before providing services. In addition, DW strategy is an improved strategy that does not assign vehicles waiting at a location to provide services; therefore, DW will not occur in case 1 (a) too.

Case 2 shows that a dynamic request appears when the vehicles are busy on traveling to the next locations. In this situation, the three strategies may happen to meet the real-time requests. This situation will be described in section 4.3.

Table 4.1 Possible state of the vehicles under different strategies

State of the vehicles / Waiting strategies	Case 1 Vehicle Idling		Case 2 Busy Vehicles
	(a) Before-service vehicles	(b) After-service vehicles	
DF	✓		✓
WF		✓	✓
DW		✓	✓

In the process of new request insertion, we just compare the pickup time of the new request, the property of precedence constraint, to the location in the existed route.

Table 4.2 provides the pseudo code to check the insert position for the dynamic request. Since the precedence constraint exists in DARP problem, we first the insert position only for the pickup point. For the request reveal in real-time before the vehicle idles to provide service in one location, the pseudo code can refer to the codes in case 1. On the other hand, when the vehicle which is busy for traveling to one location meets the real-time request, the pseudo code can be seen in case 2.

Table 4.2　Pseudo code to check the insert position for the dynamic request

```
GIVEN the data for a paired dynamic location j+, j-
CHECK the insertion with the pickup location j+
FOR each vehicle m when a request is received at time
    IF the vehicle m idles to serve a location THEN
        CHOOSE case 1 (a)
    ELSEIF the vehicle m idles at the served location THEN
        CHOOSE case 1 (b)
    ELSE
        CHOOSE case 2
    ENDIF
ENDFOR
```

4.2 Dynamic Dispatching of Idle Vehicles

Vehicle idles in one location can be divided into two types: before-service vehicles and after-service vehicles, the difference between them will be presented in section 4.2.1 and 4.2.2.

4.2.1 Before-service vehicles

Vehicles may make detours serving new requests in this situation. A new request *j*, which is concerned by an idling vehicle waiting to serve a location *i*, is checked the feasibility of insertion. If *j* is possibly inserted to the route, the vehicle first arrives to *j* and goes back to *i*. As Figure 4.2 shows, *j* could be inserted if the travel time of serving is less than the period of serving new request. The route will then be updated in real-time as shown in Figure 4.3, where solid line is the new route and dotted line represents the original route. A detour is existed in location *i*, where the vehicle arrives to *i* more than once, yet serving *i* once, also.

An example of one route is updated in real time in two-dimensional space as shown in Figure 4.4. The shape in grey color and a logo of the vehicle represent the position of the vehicle; it should follow the path in the dotted lines where the solid lines indicate the finished path for the vehicle. In addition, we use minutes for the time scale. Figure 4.4 (a) displays the planned route when starts serving the known requests ($t = 0$). When the vehicle arrives at one location but it cannot be served at the moment, the vehicle should wait until the earliest service time for the request. Figure 4.4 (b) gives an example in this situation. The vehicle arrives at the location 1^+, having a time windows from t = 30 to t = 60, at t = 10; therefore, the vehicle has to wait for 20 minutes until 1^+ can be served at t = 30. Meanwhile, a new paired request, indicating as 4^+ and 4^- for pickup and delivery location, respectively, appears, the vehicle is allowed to serve 4^+ and then goes back to serve 1^+. Therefore, the path is immediately modified as Figure 4.4 (c). The vehicle arrives at location 4^+ at t = 20 and then drives to serve location 1^+ at t = 30 shown as Figure 4.4 (d). It results in using the idle time of 20 minutes at location 1^+ to serving a new location 4^+, but more travel distance is required because of the states for the detours. DF strategy may meet in this case as it has potential to assign the vehicle to the location before serving time. However, WF and DW will be shown in case 1 (b) indeed.

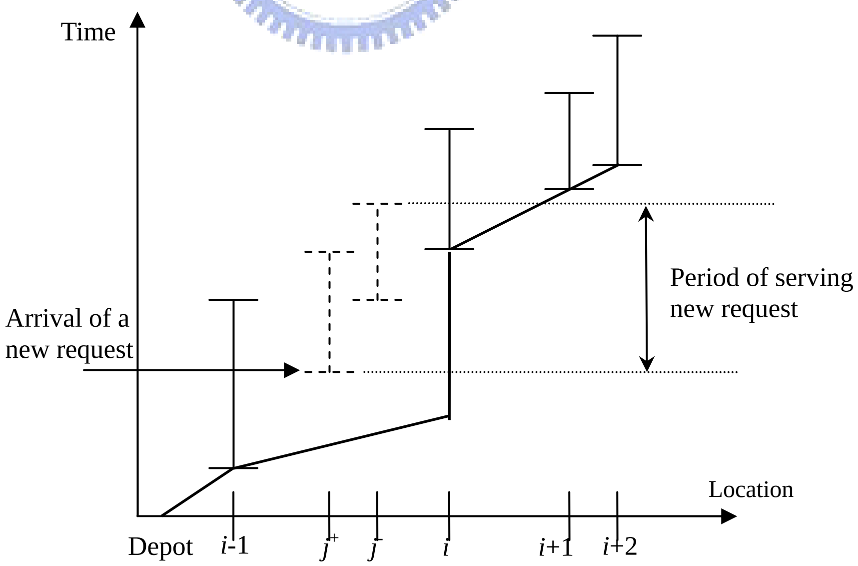

Figure 4.2 Arrival of new request in case 1 (a)

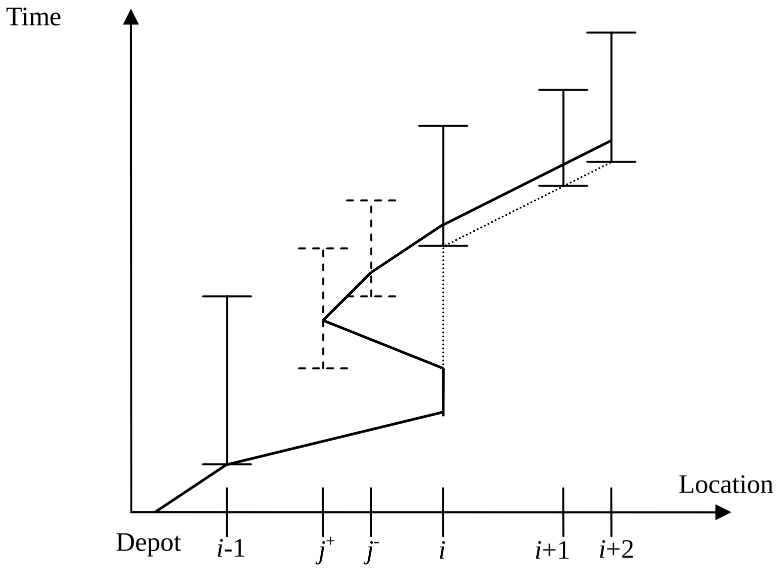

Figure 4.3 Updated route in case 1 (a)

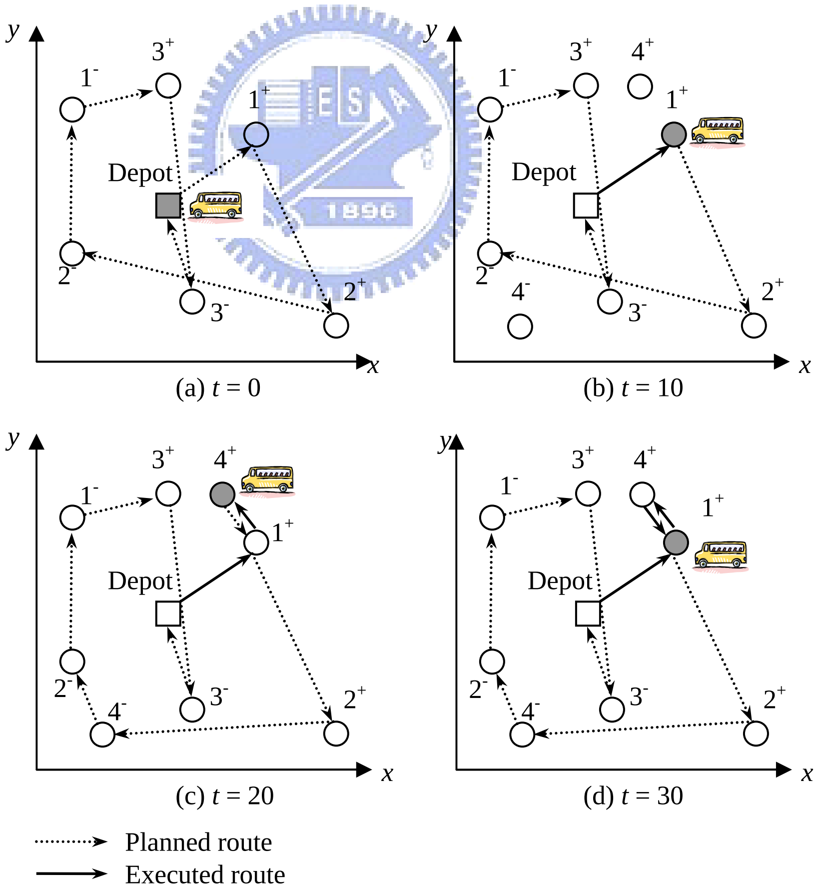

Figure 4.4 Updated route for case 1 (a) in two-dimensional space

Table 4.3 gives the pseudo code in this case. As the vehicle waiting to serve the location i meets the real-time request j, it is possible to serve the new request within the latest service time of i. First, the total travel time from i to j^+ and then j^+ to i and the service interval from the reveal time of j, EPT_{j+}, to the latest service time, b_i. If the maximum detour is larger than the total travel time, a dummy is set for indicating the vehicle has arrived to location i but has not served yet. Meanwhile, the paired locations are concerned to be inserted before the location i.

Table 4.3 Pseudo code for case 1 (a)

```
GIVEN the data for a paired dynamic request j+, j-
FOR the vehicle m waits to serve the location i
    CALCULATE the travel time TT_{i,j} from i to j+ and j+ to i
    CALCULATE the maximum detour* MD_i = (b_i − EPT_{j+})
    IF MD_i ≥ TT_{i,j} THEN
        i becomes a dummy node and unserved at this moment
        DT_dummy = EPT_{j+}
        INSERT j+ to the route before i
        FIND a suitable position for j- after j+
    ENDIF
ENDFOR
```

* maximum detour is the maximum remaining time that a vehicle can detour to another location

4.2.2 After-service vehicles

New requests may not be served by the existed vehicles in this situation. When a new request j is concerned by an idling vehicle which is waiting at the served location i, the possible insertion of j is checked after i. If j is possibly inserted to the route, the vehicle waits until the direct ride time to location i+1, including j. As Figure 4.5 shows, j could be served if the travel time of serving is less than the period of serving new request. The route will then be modified in real-time as shown in Figure 4.6.

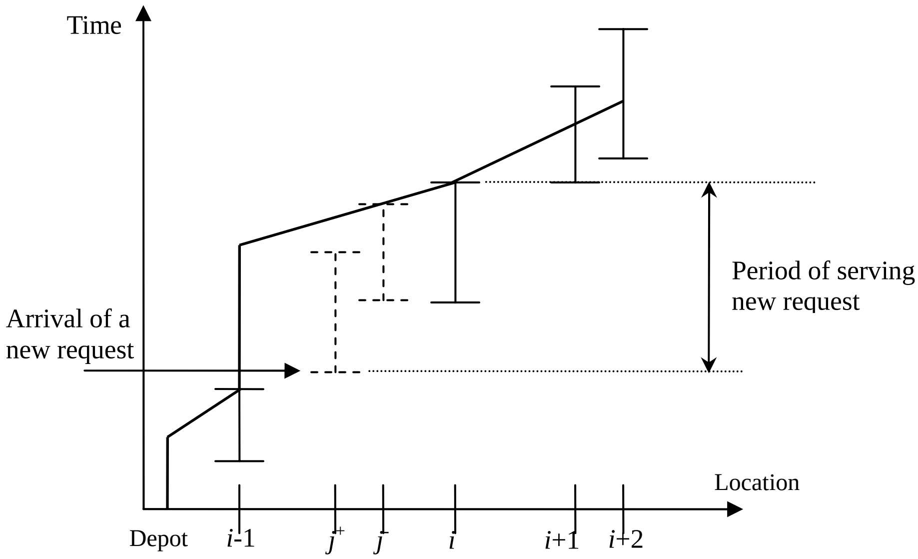

Figure 4.5 Arrival of new request in case 1 (b)

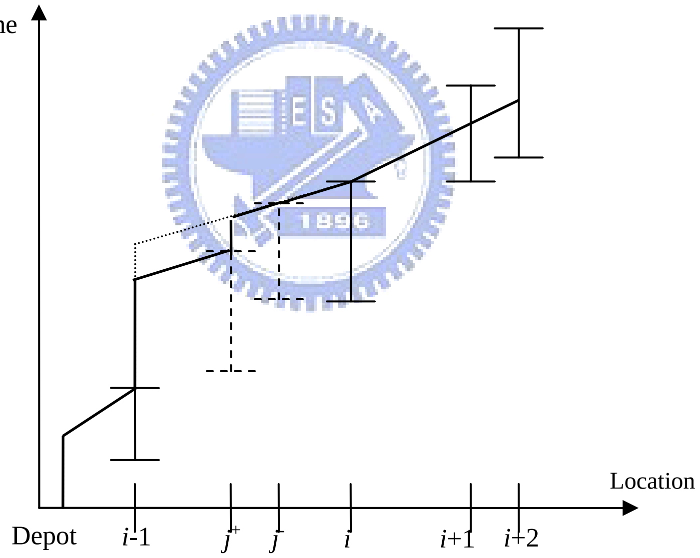

Figure 4.6 Updated route in case 1 (b)

Figure 4.7 shows the same example for one route in two-dimensional space. At t = 0, the route with different strategies is the same, as displayed in Figure 4.7 (a). However, the vehicle waits at the depot in Figure 4.7 (b) at t = 10, where the vehicle waits at location 1^+ in Figure 4.7 (b). At the same time, paired request 4 appear to be served, the vehicle drives to serve the new request, just as Figure 4.7 (c) shows and then drives to the planned location 1^+ in Figure 4.7 (d). The result in the state of non-detouring contributes in shorter total travel distance. But the new request may not be satisfied in this situation as the vehicle spends lots of time on waiting. Therefore, a new vehicle may be assigned to serve the request. WF and DW strategies may occur in this event.

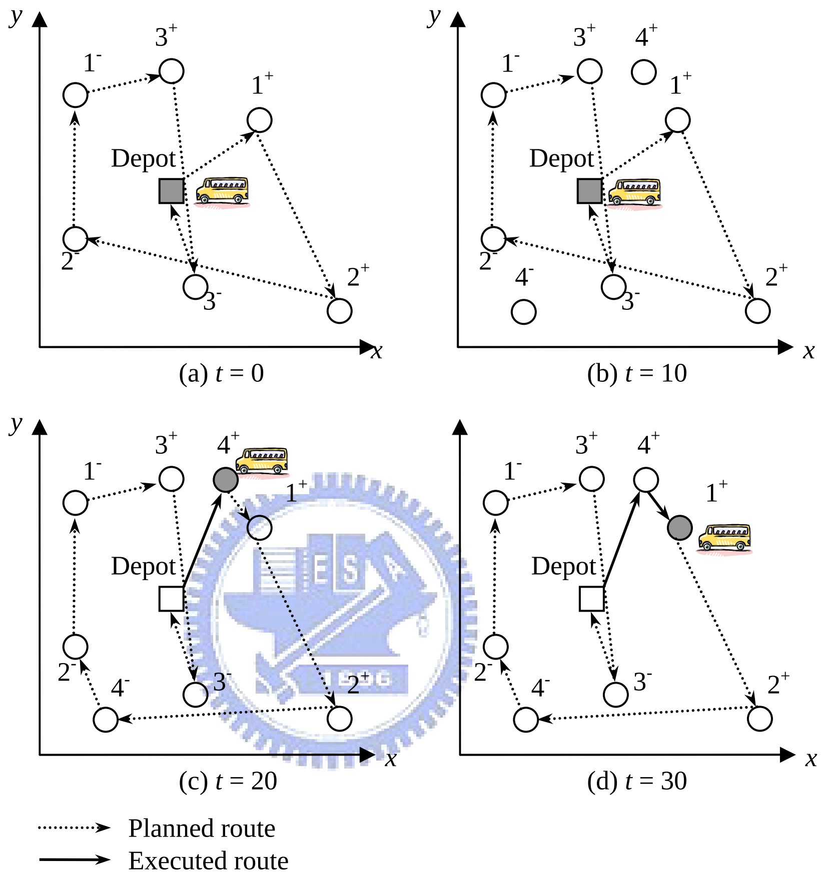

Figure 4.7 Updated route for case 1 (b) in two-dimensional space

Table 4.4 gives the pseudo code for this event. As the vehicle waiting at the served location i meets the real-time request j, it is possible to serve the new request within the service time to the location $i+1$. First, the total travel time from i to j^+ and then j^+ to $i+1$ and the sum of the difference between departure time of location i and the maximum detour from the reveal time of j, EPT_{j+}, and the direct distance from location i to location $i+1$. If the service interval is larger than the total travel time, the paired locations are concerned to be inserted before the location $i+1$.

Table 4.4 Pseudo code for case 1 (b)

```
GIVEN the data for a paired dynamic request j+, j-
FOR the vehicle m waits at the served location i
    CALCULATE the travel time TT_{i,j+} + TT_{j+,i+1} from i to j+ and j+ to i+1
    CALCULATE the maximum detour* MD_i = (DT_i - EPT_{j+}) + DRT_{i,i+1}
    IF MD_i >= TT_{i,j+} + TT_{j+,i+1} THEN
        INSERT j+ to the route between i and i+1
        FIND a suitable position for j- after j+
    ENDIF
ENDFOR
```

* maximum detour is the maximum remaining time that a vehicle can detour to another location

4.3 Dynamic Dispatching of Busy Vehicles

This study only considers the real-time insertion when the vehicle stops at one location. Therefore, two steps checking the possibility of inserting new requests to the running vehicle is required. First, the vehicle arrives to the next planned location *i*. Next, the possibility of inserting new request is checked. When the vehicle idles arriving to *i*, the similar method to insert new request is shown in section 4.2. Otherwise, the vehicle considers the new insertion after serving *i* as shown in Figure 4.8. If the new request can be served in the period, the vehicle then serves the new request and goes to *i*+1 just as the case in Figure 4.9 presents.

The path in two-dimensional space in this case is shown in Figure 4.10. The vehicle starts to follow the path at the depot at $t = 0$ in Figure 4.10 (a). When new request “4” appears at $t = 25$ where the vehicle is on the way to the location 1^+ shown in Figure 4.10 (b), the vehicle first arrive at the planned location 1^+ and then check the feasibility to serve the real-time location 4^+, the path is then immediately modified as shown in Figure 4.10 (c). Since the location 1^+ has already served, the vehicle will not go back to it; therefore, a detour will not exist in this case. The modified route can be seen in Figure 4.10 (d).

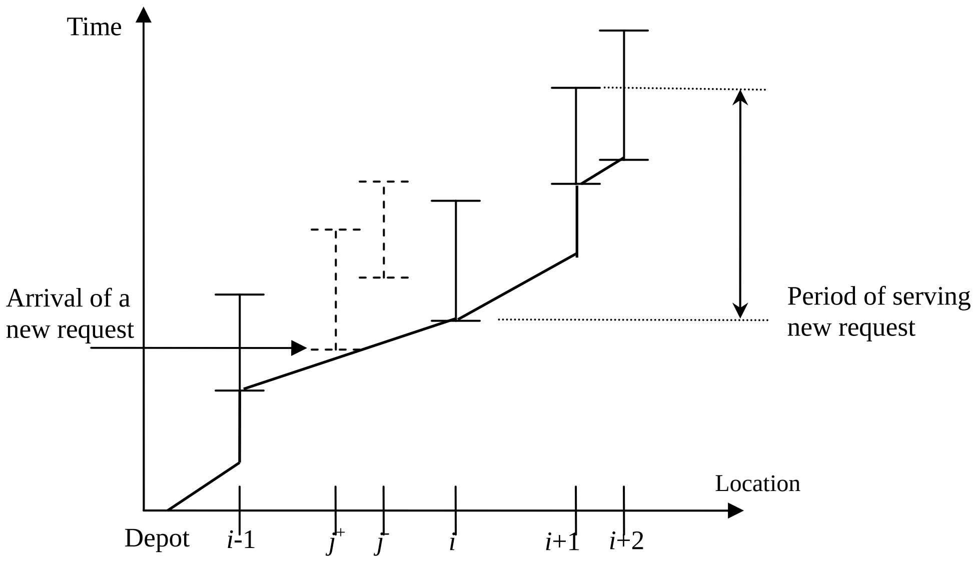

Figure 4.8 Arrival of new request in case 2

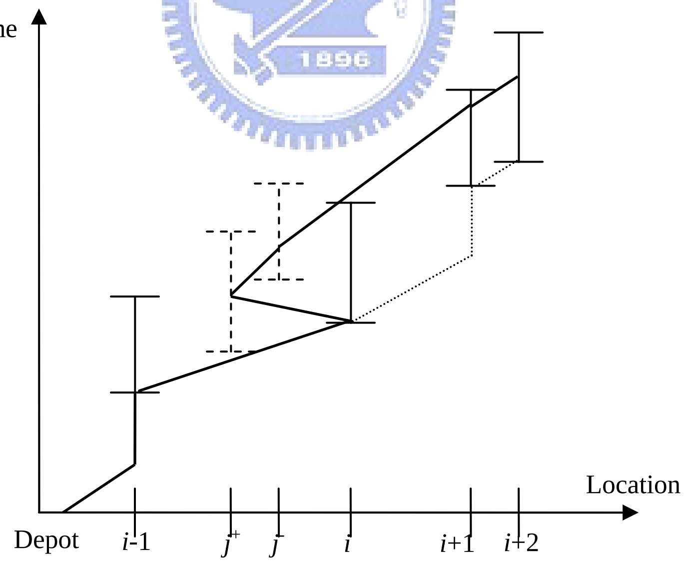

Figure 4.9 Updated route for case 2

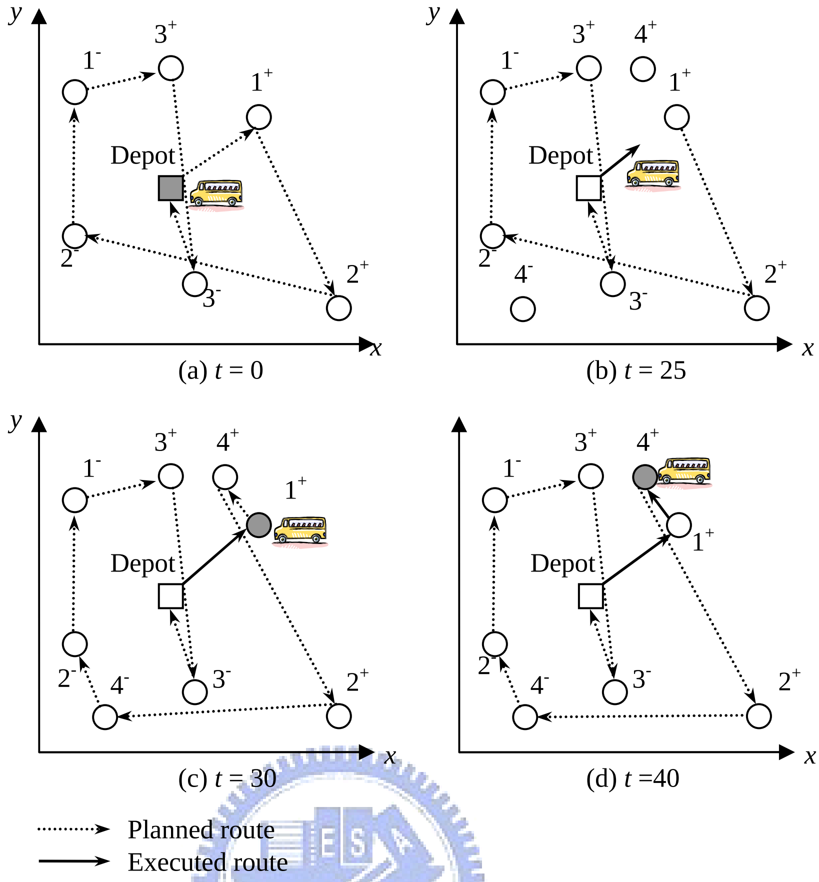

Figure 4.10 Updated route for case 2 in two-dimensional space

Table 4.5 shows the pseudo code for case 2. As the vehicle which is busy on traveling to the location *i* meets the real-time request *j*, the request will be concerned when the vehicle arrives to *i*. According to the concerned strategy, the calculation in this case is similar to that in case 1. If DF is concerned, similar calculation can be refer to case 1 (a). On the other hand, WF and DW can be solved with similar criterion as case 1 (b).

Table 4.5 Pseudo code for case 2

```
GIVEN the data for a paired dynamic request j+, j-
FOR the vehicle m is traveling to location i
    IF the waiting strategy is DF THEN
        CONSIDER inserting j+ before i, which becomes a dummy node, or
        after i (similar to case 1 (a)).
    ELSEIF the waiting strategy is WF or DW THEN
        CONSIDER inserting j+ between i and i+1 (similar to case 1 (b)).
    ENDIF
ENDFOR
```

4.4 Summary

This chapter presented the solution strategies to solve the DDARP. The operating mode of this system was described: the known requests are planned to serve before the vehicles start providing services. Next, the dispatch offices are available to receive real-time requests. Then, the service path for the vehicles can be modified in real-time to respond the dynamic requests. The structure was presented to solve DDARP.

Because it is assumed that the vehicles can update their routes (or their next destinations) only when the vehicles are waiting or reach to the next stop, three cases are considered to insert new requests, including waiting of before-service vehicles, waiting of after-service vehicles and busy on traveling vehicles . DF does not meet the situation of the vehicle waiting at the after-service location while DW and WF do not meet the situation of that at before-service location. Finally, three strategies may meet the case of the appearance of real-time requests where the vehicle is busy on traveling to the next location, the vehicle is assigned to arrive at the next location and then check the feasibility to serve the new request, the path of the vehicle will be modified consequently.

CHAPTER 5
SIMULATION ANALYSIS AND RESULTS

To our knowledge, no test instances are available in DDARP. We test the strategies by generating 30 instances with different size of requests. The scenarios will be introduced section 5.1. Next, the performance of the strategies to solve this problem can be compared in system and operation aspects in section 5.2 and 5.3, respectively. Finally, the summary will be shown in section 5.4.

5.1 Scenarios of Random Incidents

In the simulations, we just consider the customers would only specify their desired pickup time (*DPT*) and a paired of origin-destination locations. The service period is 480 minutes and the service area is 20 km×20 km with vehicle speed of 30 km/h traveling in Euclidean distance. The constants of *A*, *B* and *WS* are 2, 30 minutes and 30 minutes respectively to compute the time windows by Equations (19) to (23). The requests are generated with the Poisson process, where the parameter of the arrival rate can be obtained by

$$\lambda = \frac{Number\ of\ requests}{Service\ period} \tag{29}$$

For each problem, we set a number of static requests, N_s, and dynamic requests, N_d, for the simulation. The arrival rate for static requests, λ_s, and dynamic requests, λ_d, can be calculated with Equations (30) and (31) respectively. Furthermore, the total number of requests in the whole study is the sum of the number of static and dynamic requests shown in Equation (32). Finally, the degree of dynamism can be defined as the percentage of the number of dynamic requests to the total requests received by the Equation (33).

$$\lambda_s\,(calls/\,min) = \frac{N_s\,(\,calls\,)}{480\,(\,min\,)} \tag{30}$$

$$\lambda_d\,(calls/min) = \frac{N_d\,(calls)}{480\,(min)} \tag{31}$$

$$N_{Total}\,(calls)\; = N_s\,(calls) + N_d\,(calls) \tag{32}$$

$$dod\,(\%) = \frac{N_d\,(calls)}{N_s\,(calls) + N_d\,(calls)} \times 100\% \tag{33}$$

Since the arrival of requests follow Poisson Process, the arrival time between each customer has the exponential distribution. The locations of the requests are uniformly distributed inside the service area. The paired pickup and delivery locations are independently and identically distributed. However, the time window constraints depend on the locations between the paired pickup and delivery point, that is, the time window will be longer when the pickup and delivery points are further away.

Table 5.1 shows the performance under different set of problems using DF strategy as instances. The indicators N_s, N_d and *dod* of the problems represent the number of static requests, the number of dynamic requests and degree of dynamism. Example 1 illustrates the state of fully static problem with no dynamic request. Besides, example 2 shows that 50 % of dynamic requests exist in the system. Example 3 presents the problem with fully dynamic requests.

The performance can be seen in through operating cost and service quality. In the part of operating cost, the labeled "*m*" depicts the requirement for the number of vehicles. In addition, total travel distance and total duration time are also considered. In the environment of service quality, the difference between the earliest pickup time and the exact pickup time stands for the waiting time for the customers who join the system. Furthermore, the ride time represents the time for all customers who take the vehicle from the origin to the destination.

Table 5.1 The performance under different set of problems using DF strategy

Example	Problems			Performance				
				Operating cost			Service quality	
	N_s (calls)	N_d (calls)	*dod* (%)	*m* (veh)	Total Distance (veh-km)	Total Duration (veh-min)	Total Customer Wait time (min)	Total Customer Ride time (min)
1	100	0	0	6	1120.07	2854.73	1118.94	3427.30
2	50	50	50	8	1363.51	3905.79	1397.16	3419.11
3	0	100	100	8	1486.86	3891.79	1983.41	2917.32

Figure 5.1 illustrates a trajectory serving the locations with the three strategies in the case of *dod* = 50%. To be exact, this figure represents a calculation of a sample trajectory of a vehicle with the DF, DW and WF strategies using the final solution of the DF strategy for the whole run. This is because, as we will show later, the final solution of each strategy is quite different to each other with the sequence of locations and therefore it is not able to be shown and compared on the same figure. The dash line with triangular legend represents the path of WF strategy; the dotted line with diamond legend represents the path of DF strategy, and the solid line with square legend represents the path of DW strategy. As can be seen, DW strategy stays in the boundary between DF and WF strategies.

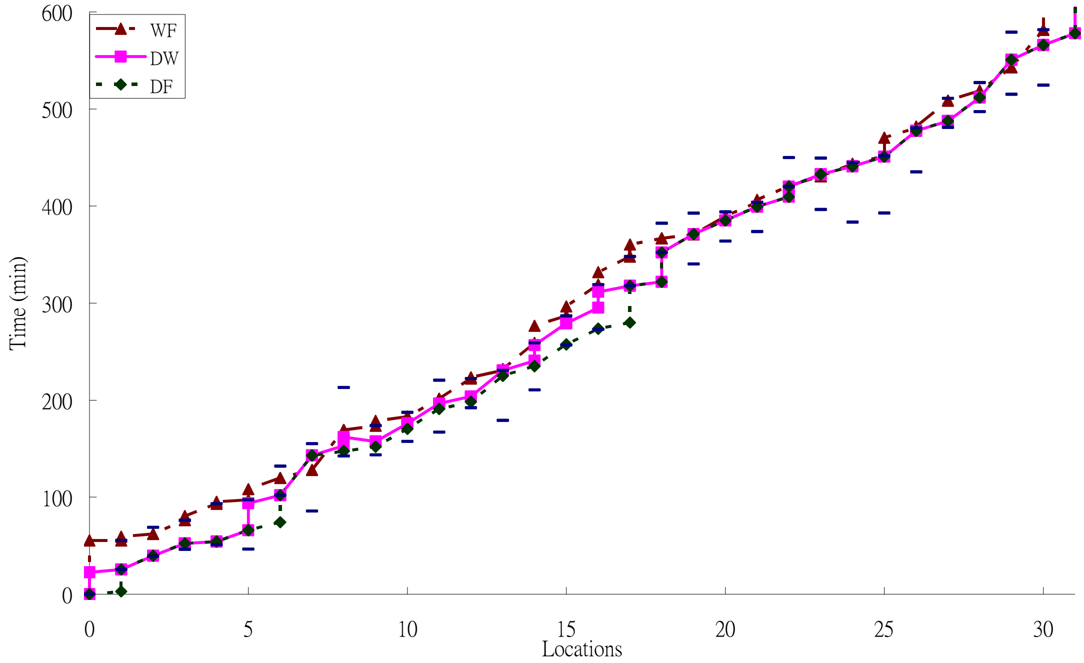

Figure 5.1 A trajectory of a vehicle with the three strategies in the example

5.2 System Performance Measure

In this section, we have fixed the total number of requests where *dod* is used to vary the ratio of number of static requests to that of dynamic requests in the system observing

1. The performance between DF, WF and DW strategies
2. The characteristics of dod

Five indices are used to compare the performance of the strategies, namely number of operating vehicles, total travel distance, total duration time, total waiting time of customers and total riding time of customers. In operation, number of vehicles represents a fixed operating cost, while the total distance traveled is a variable cost which is calculated as Equation (34). In addition, total duration time is a variable cost related to the pay of drivers.

$$Total\ distance\ traveled = \sum_{m=1}^{M} (480\ min - \sum_{i=1}^{\substack{No.of\ stops\\ of\ veh\ m}} WT_i) \times average\ speed\ of\ vehicles \quad (34)$$

$$Total\ duration\ time = \sum_{m=1}^{M} (AT_{end\ at\ depot\ of\ veh\ m} - AT_{start\ at\ depot\ of\ veh\ m}) \quad (35)$$

In the aspect of customers, they concern to take the services with minimum waiting time and minimum ride time. They desire the operating vehicle arrive to the location with minimum out of vehicle waiting time. The waiting time of a request *i* can be calculated as the time deviation between the earliest service time (a_{i+}) and the arrival time of the vehicle (AT_{i+}). Therefore, the total waiting time for all the requests can be computed as Equation (36), where *i* is the set of number of requests *N*.

$$Total\ Customer\ Waiting\ Time = \sum_{i^+=1}^{N} max(AT_{i^+} - a_{i^+}, 0) \quad (36)$$

Another service of quality for customers is the ride time in the vehicle from origin to destination, which can be computed by

$$Total\ Customer\ Ride\ Time = \sum_{i=1}^{N} (AT_{i^-} - AT_{i^+}) \quad (37)$$

The performances introduced above will be used to measure the efficient using the three strategies in the following section. In addition, the number of operating vehicles and total travel distance will be discussed separately with other performances as the reason of the objectives in this study. However, other performances will be shown as reference materials for the three strategies.

5.3 Result Analysis

In this section we will show our simulation results and observations. A comparison between the three strategies with the performances is demonstrated in section 5.3.1. We also find that the characteristics as number of dynamic requests increases in the system, the observation will be described in section 5.3.2. Another observation in section 5.3.3 shows that the economy of scale will be existed to serve the same amount of dynamic requests with varied number of static requests.

5.3.1 Performance of waiting strategies

We have generated and tested the results for the three strategies with 100, 300, 500 and 1000 requests for different *dod* values. To take account of the randomness in the problem generation, we run 30 instances for each case and present the averaged results. The results are shown in Table 5.2 to Table 5.5. We can see that WF needs more vehicles than others; DW requires similar number of vehicles than DF. On the other hand, we observe that WF needs shorter total travel distance than other strategies, but the advantage of it does not exist in high intensity of the demand; for example, the total travel distance with WF strategy is smaller than that with DF strategy in the problem size of 100 requests, but the result of WF strategy is longer than the result of DF strategy in the problem size of 1000 requests. Because of the difference of the requirement between WF and other strategies become larger, the total travel distance with WF will be longer. Furthermore, DW strategy gives better total travel distance compared to DF strategy in all problem size.

We also perform *t*-test to check if the results of two strategies are significantly different. A "**" represents the results for two strategies are significantly different in level of significance of $\alpha = 5\%$ whereas a "*" is in $\alpha = 10\%$. The *t*-values for different size of problems are depicts in Table 5.6 where Table 5.6 (a) is for $N = 100$ requests, Table 5.6 (b) is for $N = 300$ requests, Table 5.6 (c) for $N = 500$ requests and Table 5.6 (d) for $N = 1000$ requests. The standard deviation for the three strategies in different size of problems can be seen in Appendix A.

Table 5.2 Number of vehicles and total travel distance for 100 requests with the three strategies

N=100	DF		WF		DW	
dod(%)	*M*	Distance	*M* (% vs DF)	Distance (% vs DF)	*M* (% vs DF)	Distance (% vs DF)
0	6.83	1130.80	6.83 (0.00)	1130.80 (0.00)	6.83 (0.00)	1130.80 (0.00)
10	7.47	1205.77	8.07 (8.03) **	1188.66 (-1.42) **	7.40 (-0.94)	1190.83 (-1.24) **
20	7.63	1277.18	8.40 (10.09) **	1245.09 (-2.51) **	7.70 (0.92)	1264.01 (-1.03) **
30	8.20	1373.95	9.27 (13.05) **	1291.17 (-6.02) **	8.13 (-0.85)	1346.44 (-2.00) **
40	8.23	1408.84	9.90 (20.29) **	1323.76 (-6.04) **	8.17 (-0.73)	1361.65 (-3.35) **
50	8.60	1463.05	10.07 (17.09) **	1330.31 (-9.07) **	8.70 (-1.16)	1428.63 (-2.35) **
60	8.70	1488.08	10.53 (21.03) **	1367.46 (-8.11) **	8.63 (-0.80)	1456.39 (-2.13) **
70	8.50	1532.66	10.73 (26.24) **	1388.47 (-9.41) **	8.60 (1.18)	1494.73 (-2.47) **
80	8.67	1531.72	11.03 (27.22) **	1373.32 (-10.34) **	8.53 (-1.61)	1502.26 (-1.92) **
90	8.73	1556.73	10.80 (23.71) **	1367.86 (-12.13) **	8.53 (-2.29)	1525.28 (-2.02) **
100	8.60	1538.55	11.03 (28.26) **	1339.02 (-12.97) **	8.60 (0.00)	1539.06 (0.03)

** significantly different between two different strategies in $\alpha = 5\%$
* significantly different between two different strategies in $\alpha = 10\%$

Table 5.3 Number of vehicles and total travel distance for 300 requests with the three strategies

N=300	DF		WF		DW	
dod(%)	*M*	Distance	*M* (% vs DF)	Distance (% vs DF)	*M* (% vs DF)	Distance (% vs DF)
0	13.83	2588.06	13.83 (0.00)	2588.06 (0.00)	13.83 (0.00)	2588.06 (0.00)
10	14.87	2852.83	16.33 (9.82) **	2834.03 (-0.66)	14.90 (0.20)	2817.63 (-1.23) **
20	15.60	3063.42	18.00 (15.38) **	2998.18 (-2.13) **	15.47 (-0.83)	2998.98 (-2.10) **
30	16.33	3165.84	18.77 (14.94) **	3092.21 (-2.33) **	16.20 (-0.80)	3089.02 (-2.43) **
40	16.63	3276.98	19.83 (19.24) **	3175.58 (-3.09) **	16.83 (1.20)	3201.70 (-2.30) **
50	17.07	3351.30	20.33 (19.10) **	3278.41 (-2.17) **	16.97 (-0.59)	3285.55 (-1.96) **
60	17.30	3422.94	21.00 (21.39) **	3354.77 (-1.99) **	17.23 (-0.40)	3335.89 (-2.54) **
70	17.27	3418.82	21.27 (23.16) **	3317.48 (-2.96) **	16.77 (-2.90) **	3323.59 (-2.79) **
80	17.00	3402.50	21.23 (24.88) **	3337.40 (-1.91) **	16.97 (-0.18)	3341.67 (-1.79) **
90	16.47	3369.63	21.40 (29.93) **	3309.62 (-1.78) **	16.70 (1.40)	3324.13 (-1.35) **
100	16.30	3285.34	21.17 (29.88) **	3224.57 (-1.85) **	16.23 (-0.43)	3242.25 (-1.31) **

** significantly different between two different strategies in $\alpha = 5\%$
* significantly different between two different strategies in $\alpha = 10\%$

Table 5.4 Number of vehicles and total travel distance for 500 requests with the three strategies

N=500	DF		WF				DW			
dod(%)	*M*	Distance	*M* (% vs DF)		Distance (% vs DF)		*M* (% vs DF)		Distance (% vs DF)	
0	18.77	3784.43	18.77	(0.00)	3784.43	(0.00)	18.77	(0.00)	3784.43	(0.00)
10	21.00	4197.51	22.97	(9.38)**	4192.04	(-0.13)	21.00	(0.00)	4152.03	(-1.08)**
20	21.83	4476.77	24.87	(13.93)**	4409.73	(-1.50)**	21.73	(-0.46)	4412.61	(-1.43)**
30	22.30	4632.59	25.87	(16.01)**	4582.21	(-1.09)**	22.23	(-0.31)	4554.84	(-1.68)**
40	23.00	4774.60	27.13	(17.96)**	4754.86	(-0.41)	22.87	(-0.57)	4689.26	(-1.79)**
50	22.63	4877.18	27.17	(20.06)**	4854.57	(-0.46)	22.80	(0.75)	4812.00	(-1.34)**
60	22.87	4911.93	28.47	(24.49)**	5012.85	(2.05)**	22.70	(-0.74)	4861.65	(-1.02)**
70	22.97	4969.06	28.63	(24.64)**	5005.46	(0.73)**	23.20	(1.00)	4940.45	(-0.58)*
80	22.93	4966.97	28.73	(25.29)**	5025.66	(1.18)**	22.53	(-1.74)	4895.37	(-1.44)**
90	23.03	4869.43	28.67	(24.49)**	4951.60	(1.69)**	22.50	(-2.30)*	4825.07	(-0.91)*
100	21.97	4705.46	27.80	(26.54)**	4802.33	(2.06)**	21.97	(0.00)	4704.89	(-0.01)

** significantly different between two different strategies in $\alpha = 5\%$
* significantly different between two different strategies in $\alpha = 10\%$

The results of number of vehicles and total travel distance as *dod* increases are shown in Figure 5.2 and Figure 5.3. The dash line in diamond legend, the solid line in square legend and the dotted line in triangle legend represent the performance of WF, DW and DF strategies respectively. The trend for the number of vehicles is increasing when *dod* increases. We also observed that the performance slightly decreases when *dod* increases from about 60% to 100% which will be discussed later in section 5.3.2.

Table 5.5 Number of vehicles and total travel distance for 1000 requests with the three strategies

N=1000	DF		WF				DW			
dod(%)	*M*	Distance	*M* (% vs DF)		Distance (% vs DF)		*M* (% vs DF)		Distance (% vs DF)	
0	30.80	6366.98	30.80	(0.00)	6366.98	(0.00)	30.80	(0.00)	6366.98	(0.00)
10	32.77	7021.39	35.93	(9.64)**	7035.06	(0.19)	32.60	(-0.52)	6937.32	(-1.20)**
20	33.73	7368.94	38.30	(13.55)**	7435.71	(0.91)**	33.67	(-0.18)	7265.97	(-1.40)**
30	34.27	7674.07	40.13	(17.10)**	7803.05	(1.68)**	34.30	(0.09)	7585.74	(-1.15)**
40	35.37	7914.23	41.70	(17.90)**	8108.80	(2.46)**	35.30	(-0.20)	7817.05	(-1.23)**
50	35.30	7977.75	42.97	(21.73)**	8264.52	(3.59)**	35.17	(-0.37)	7894.65	(-1.04)**
60	36.00	8072.53	43.53	(20.92)**	8406.07	(4.13)**	36.10	(0.28)	8000.17	(-0.90)**
70	36.30	8100.61	45.07	(24.16)**	8554.40	(5.60)**	36.10	(-0.55)	8017.16	(-1.03)**
80	35.73	8117.92	44.50	(24.55)**	8625.83	(6.26)**	35.70	(-0.08)	8039.51	(-0.97)**
90	35.50	7995.72	44.97	(26.68)**	8597.63	(7.53)**	35.50	(0.00)	7992.03	(-0.05)
100	34.13	7719.98	43.90	(28.63)**	8255.08	(6.93)**	34.17	(0.12)	7715.50	(-0.06)

** significantly different between two different strategies in $\alpha = 5\%$
* significantly different between two different strategies in $\alpha = 10\%$

Table 5.6 *t*-values for WF and DW strategies compare to DF strategies in different size of problems: (a) 100 requests, (b) 300 requests, (c) 500 requests, (d) 1000 requests

(a) 100 requests

N=100	WF vs DF		DW vs DF	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$
0	0.000	0.000	0.000	0.000
10	3.844	-2.372	-0.812	-4.590
20	5.139	-4.097	0.701	-2.778
30	7.059	-10.082	-0.528	-4.765
40	8.601	-9.089	-0.494	-7.531
50	9.805	-17.445	0.769	-4.361
60	9.845	-11.485	-0.465	-3.942
70	14.969	-9.215	0.551	-3.706
80	13.971	-15.135	-0.812	-3.250
90	11.986	-14.508	-1.185	-2.695
100	9.162	-16.052	0.000	0.000

(b) 300 requests

N=300	WF vs DF		DW vs DF	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$
0	0.000	0.000	0.000	0.000
10	7.478	-2.049	0.372	-4.799
20	10.770	-6.526	-0.812	-7.538
30	14.250	-5.235	-1.000	-7.516
40	13.243	-6.458	1.140	-5.518
50	14.227	-4.933	-0.474	-5.792
60	14.810	-4.324	-0.441	-5.899
70	11.798	-5.452	-1.881	-7.015
80	17.093	-3.605	-0.141	-2.865
90	17.929	-2.885	0.865	-2.553
100	21.773	-3.116	-1.000	-5.974

(c) 500 requests

N=500	WF vs DF		DW vs DF	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$
0	0.000	0.000	0.000	0.000
10	8.651	-0.476	0.000	-6.767
20	14.715	-5.210	-0.722	-6.733
30	13.890	-3.148	-0.403	-6.428
40	15.550	-1.064	-0.571	-5.858
50	20.281	-1.153	0.796	-3.255
60	24.042	5.302	-0.681	-3.280
70	20.785	2.077	0.942	-1.551
80	19.780	2.784	-1.235	-3.909
90	16.874	3.589	-1.464	-1.604
100	19.979	4.069	0.000	-1.000

(d) 1000 requests

N=1000	WF vs DF		DW vs DF	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$
0	0.000	0.000	0.000	0.000
10	14.738	1.217	-1.223	-8.858
20	17.195	3.054	-0.348	-6.408
30	22.071	6.142	0.138	-4.211
40	19.000	8.420	-0.297	-4.265
50	23.242	11.724	-0.559	-3.697
60	23.492	9.489	0.324	-3.742
70	31.047	16.576	-0.769	-3.359
80	33.009	18.882	-0.120	-3.350
90	21.974	18.338	0.000	-0.143
100	28.064	19.925	1.000	-1.034

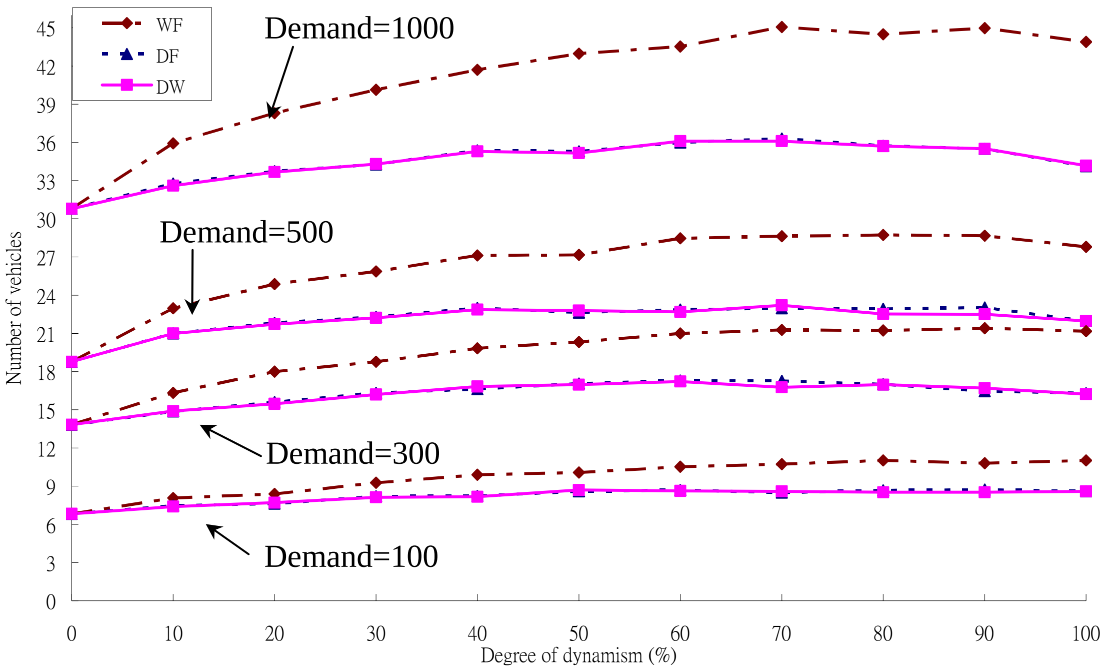

Figure 5.2 Number of vehicles against *dod* for different number of requests

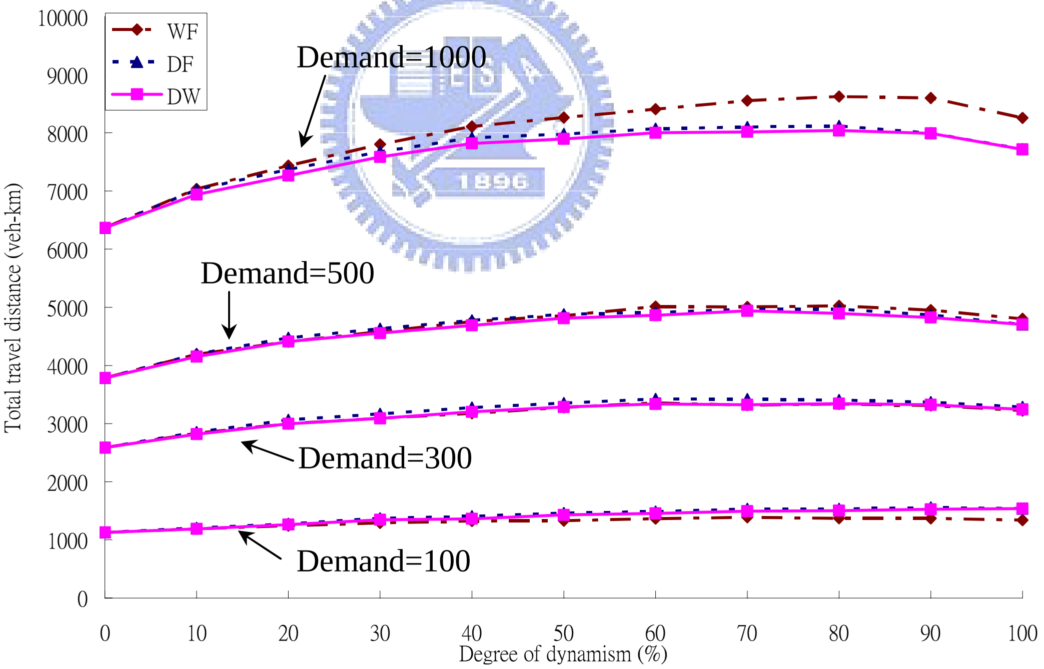

Figure 5.3 Total travel distance against *dod* for different number of requests

The performance of the three waiting strategies for different number of requests is displayed in Table 5.7 to Table 5.10. First, the total duration time contains the total service time and idle time of the vehicles. Next, the service quality is also considered in the performance which contains total waiting time and total ride time for the customers where the scales for them are "veh-min" for total duration, "min" for the waiting time and ride time for customers. We can see that as dod increases, the total duration will also increase. In addition, customer out of vehicle waiting time will also increase but the ride time will decrease as the less detouring by the vehicle when serving more dynamic requests.

Three strategies seem to have similar results in the static problem. First, the total duration with the three strategies can be obtained by equation (35), as we define that the interval of duration of every vehicle is the difference between the time of start at the depot and end at the depot. Moreover, WF has the same start operating time at the depot with DF, and when WF finishes serving the last location, it assigns to leave immediately to depot. No dynamic requests appears in the static problem, therefore, the total vehicle duration for the three strategies remain the same. Next, the customer waiting time and riding time of WF are normally higher than those of DF and DW strategies. As DW strategy is an improved from the DF strategy, the waiting time of the vehicle at one location is shifted to its previous location, and therefore, the total waiting time and the total riding time using DF and DW strategies are the same.

Compared to the three strategies, WF needs more operating costs but provides lower level of services than DF and DW strategies. It has larger total operation time, longer customer waiting time and ride time than DF and DW. Furthermore, the vehicle duration, customer waiting and ride time for DF and DW are similar.

Table 5.7 Vehicle duration, customer waiting and ride time for 100 requests

N=100	DF			WF			DW		
dod(%)	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time
0	3385.46	960.29	3678.21	3385.46	1626.65	3804.04	3385.46	960.29	3678.21
10	3669.03	1000.73	3645.14	4054.34	1833.76	3905.66	3619.39	1008.27	3639.16
20	3777.91	1106.41	3538.83	4346.03	1936.35	3934.86	3798.75	1091.00	3559.43
30	4077.51	1228.50	3485.01	4713.20	2011.64	3932.95	4052.71	1215.08	3483.29
40	4068.96	1302.57	3442.36	5030.75	2086.46	3974.80	3999.50	1265.15	3444.90
50	4133.21	1397.39	3344.53	5004.16	2186.18	3941.77	4161.88	1389.60	3338.54
60	4274.09	1486.85	3348.69	5324.54	2285.81	3956.87	4236.10	1469.59	3342.23
70	4288.64	1615.40	3232.23	5539.40	2359.36	3943.24	4312.65	1586.84	3267.44
80	4270.92	1730.63	3215.72	5527.90	2448.23	3929.96	4247.61	1704.30	3208.68
90	4410.06	1819.44	3095.68	5578.84	2498.48	3964.98	4340.25	1791.37	3161.41
100	4284.60	1955.66	3112.88	5664.04	2559.11	4013.22	4286.10	1954.86	3113.68

Table 5.8 Vehicle duration, customer waiting and ride time for 300 requests

N=300	DF			WF			DW		
dod(%)	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time
0	6812.23	3313.58	11657.21	6812.23	4641.60	11791.43	6812.23	3313.58	11657.21
10	7443.74	3569.30	11392.10	8316.50	5180.55	11939.50	7448.31	3533.13	11416.31
20	7785.59	3893.29	11216.06	9156.53	5464.16	11950.36	7724.48	3835.83	11201.21
30	8019.78	4078.18	11101.58	9454.80	5695.08	11968.21	7922.87	4019.36	11106.70
40	8274.06	4285.58	10964.40	10103.17	5955.90	11921.20	8291.78	4235.72	10964.18
50	8406.91	4593.08	10970.61	10265.59	6177.67	11997.47	8408.42	4508.75	10925.56
60	8550.04	4846.17	10773.77	10559.36	6393.14	11860.24	8481.91	4744.32	10808.62
70	8509.03	5420.91	10594.63	10672.30	6618.08	11873.86	8353.04	4943.82	10704.40
80	8459.10	5206.69	10718.38	10796.78	6784.58	12003.23	8475.18	5151.83	10748.27
90	8348.14	5471.37	10624.23	10872.68	7010.07	11916.70	8393.43	5376.47	10677.70
100	8231.01	5744.50	10758.03	10746.18	7228.05	12005.57	8133.57	5704.16	10718.70

Table 5.9 Vehicle duration, customer waiting and ride time for 500 requests

N=500	DF			WF			DW		
dod(%)	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time
0	9627.81	5805.53	19584.38	9627.81	7862.91	19726.92	9627.81	5805.53	19584.38
10	10479.04	6171.13	19298.27	11782.29	8566.87	20046.60	10470.06	6145.40	19306.14
20	10985.50	6672.30	18918.36	12817.80	9007.63	19899.91	10928.53	6587.84	18948.58
30	11223.78	7045.19	18861.73	13206.48	9346.20	20005.19	11164.22	6988.57	18881.12
40	11572.29	7424.04	18707.90	13939.19	9666.32	20029.37	11452.84	7329.29	18685.01
50	11651.29	7798.69	18504.34	14195.39	10041.67	19801.86	11695.06	7782.96	18517.64
60	11721.42	8118.39	18541.72	14907.66	10390.31	19857.01	11668.91	8015.61	18546.15
70	11828.30	8411.69	18490.87	14954.48	10678.86	19972.94	11954.01	8414.56	18497.71
80	11870.92	8904.18	18367.05	15009.53	11021.74	19823.82	11613.07	8756.88	18339.17
90	11753.06	9081.33	18364.41	14862.83	11281.58	19907.62	11528.14	9080.14	18364.75
100	11418.99	9569.72	18495.38	14626.03	11683.44	19974.57	11423.02	9513.04	18493.91

Table 5.10 Vehicle duration, customer waiting and ride time for 1000 requests

N=1000	DF			WF			DW		
dod(%)	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time	Vehicle duration	Customer wait time	Customer ride time
0	15821.52	12019.59	39611.31	15821.52	15615.30	39730.34	15821.52	12019.59	39611.31
10	16834.71	12830.39	39137.42	18847.00	16932.49	40280.31	16725.80	12765.29	39102.62
20	17319.02	13710.60	38868.19	20068.68	17634.29	40200.74	17253.67	13598.07	38868.57
30	17840.05	14412.40	38525.93	21160.87	18270.39	40012.33	17727.09	14316.31	38552.07
40	18262.24	15145.56	38377.46	21871.72	18858.37	39939.30	18152.29	15073.24	38399.75
50	18419.43	15871.04	38134.23	22465.75	19463.91	39756.71	18308.03	15657.80	38090.22
60	18588.34	16264.38	38025.36	22735.58	20062.61	39676.12	18523.54	16207.72	37930.87
70	18615.89	16851.14	38021.93	23453.51	20608.16	39727.24	18545.85	16796.81	38105.07
80	18637.52	17463.00	38071.89	23472.51	21192.42	39745.11	18580.80	17346.96	38116.37
90	18577.81	18174.34	37982.98	23549.11	21893.19	39614.90	18490.44	18049.63	38058.18
100	17993.54	18793.83	38289.72	23110.37	22477.68	39842.21	18001.65	18765.64	38300.59

Figure 5.4 shows the total duration against *dod* with three strategies. As WF needs more number of vehicles, the total duration time to serve the same amount of requests will be more than those of DF and DW. Another result also shows that the curve also slightly decreases when *dod* increase from 60% to 100%.

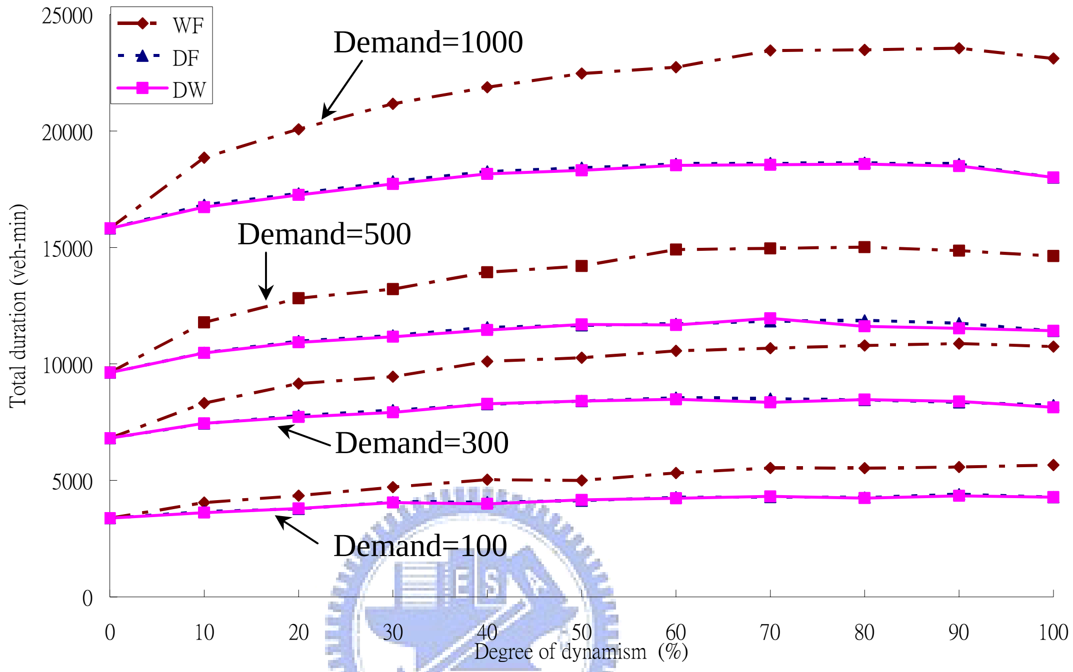

Figure 5.4 Total duration time against *dod*

In the view of level of services, the total waiting time and the total ride time are shown in Figure 5.5 and Figure 5.6. Figure 5.5 indicates the total waiting time for the requests. As *dod* increases, the time of wait for customers also increases, that is, as more real-time customers appear, the time of wait for them also increases.

Figure 5.6 depicts the relationship between total travel time for customers and *dod*. Since less advance requests exist in strongly dynamism problems, vehicles can visit the requests in smaller detour state. Therefore, customers can be served with less riding time, the total ride time for customers thus decrease as *dod* increases. Compared to the total ride time in fully static and dynamic problems, operators could plan efficient routes with minimum operating costs in fully static problems while they plan serve the requests immediately with less detour state in fully dynamic problems, customers need more ride time not providing real time services. It results in higher ride time for customers in fully static problem than fully dynamic problems.

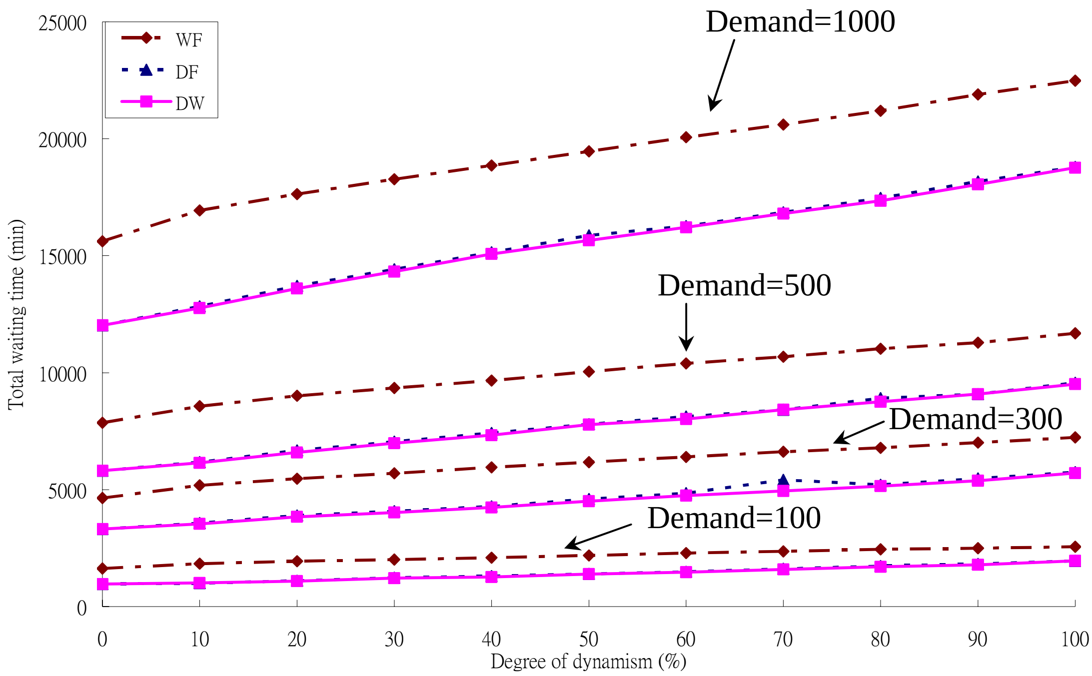

Figure 5.5 Total waiting time for customers against *dod*

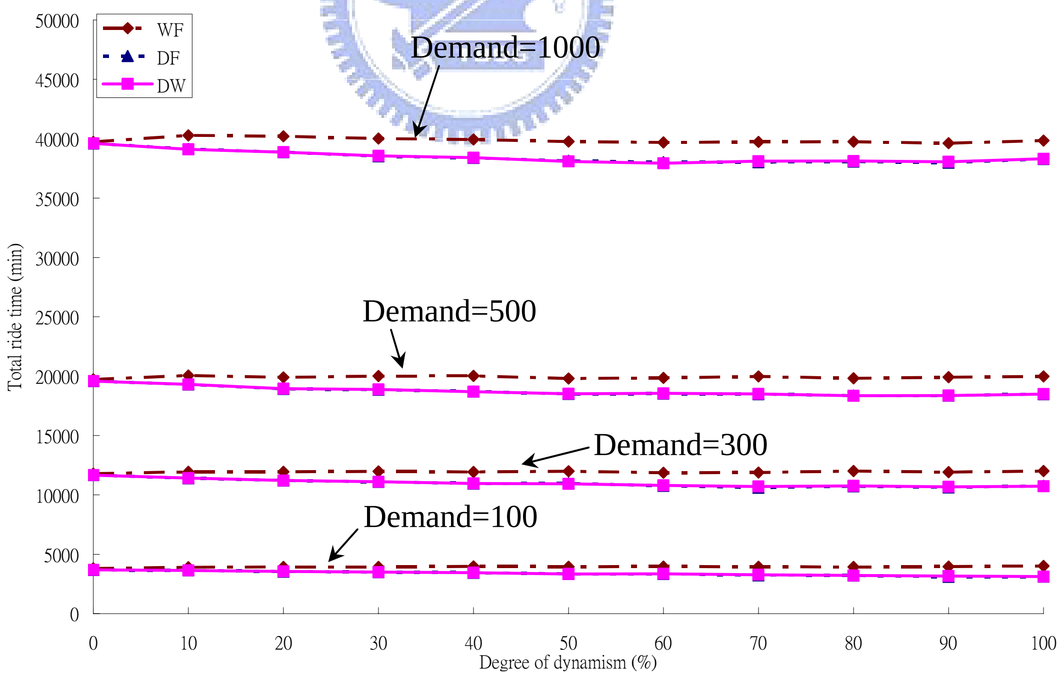

Figure 5.6 Total ride time for customers against *dod*

From the results above we make our conclusions as follow:

(1) WF strategy requires more vehicles but shorter travel distance as compared to the other strategies. But this advantage of shorter total travel distance will no longer occur at high demand levels as it requires larger number of operating vehicles than other strategies. The total travel distance may be therefore longer than the others.

(2) WF strategy is also not a better strategy in the view of other performances. It contributes longer total duration time and level of services for customers than DF strategy and DW strategy. Customers also need to wait at their pickup point until the maximum wait state and take the maximum ride time in the vehicles.

(3) DF strategy seems to be a better strategy compared to WF strategy. However, it may meet detouring to serve real-time requests and eventually longer travel distance than WF strategy. This can be improved in our proposed strategy.

(4) DW strategy requires similar number of vehicles but less total travel distance than DF strategy. This is because it does not need detouring when serving dynamic requests. In the objectives we concerned, DW provides better results than DF and WF strategies.

5.3.2 Characteristics of *dod*

A "counter-intuitive" observation is found in Figure 5.2 and Figure 5.3 in which the performances seem to be worse when *dod* increases from about 60% to 100%. Similar observations have been found in Larsen (2000). The relevant *t*-values for two different dod in the same problem size are illustrated in Table 5.11 in which Table 5.11 (a) is for N = 100 requests, Table 5.11 (b) is for N = 300 requests, Table 5.11 (c) for N = 500 requests and Table 5.11 (d) for N = 1000 requests. For example, the *dod* of "60 vs 70" in N = 100 requests describe that the *t*-value of the performance between *dod* = 60% and *dod* = 70% for the three strategies in problem size of 100 requests. The values with "**" refer to the results for two different *dod* are significantly different in level of significance of α = 5% while "*" is in α = 10%.

The values show that the difference of performances is more explicit when *dod* increases from 90% to 100%. One reason may be described as follow. In the system of 90% dynamism, some dynamic requests may not be served by the nearest vehicle with the planned service location whereas in fully dynamic system, no planned routes are existed before start serving the requests. Therefore, 90% dynamic systems may be more operating vehicles and total travel distance than 100% dynamic systems to serve the same amount of total requests.

From the results above we make our conclusions as follow:

(1) The number of vehicles and total distance traveled are higher at the *dod* level of about 90% compared to the case of 100%. For the same number of requests with different *dod* levels, the resources required (i.e. the number of vehicles and distance traveled) may be higher at *dod* of 90% as compared to 100%. That is even happening at *dod* of 60% for high demand levels. Although wider flexibility exists in operation, the operating vehicle cannot meet the dynamic requests nearby as it has already occupied by some advance requests around 90% dynamic problems. Therefore, the results would be more operating costs than 100% dynamic problems which does not have planned routes.

(2) This "counter-intuitive" observation seems to be more obvious in larger size of problems. From the tests above, the differences between 90% and 100% dynamics are not much significant in problem size of 100 requests. As number of requests becomes larger, the difference becomes more significant. The results are significantly different in total travel distance between two *dod* in N = 300 requests. Furthermore, the tests show that both number of operating vehicles and total travel distance are significantly different between 90% and 100% dynamic when N = 500 requests and N = 1000 requests.

Table 5.11 *t*-values of number of vehicles and total travel distance at different *dod* for different problem size: (a) 100 requests, (b) 300 requests, (c) 500 requests, (d) 1000 requests

(a)

N=100	DF		WF		DW	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$	t_M	$t_{distance}$
60 vs 70	-0.716	2.261**	0.654	1.477**	-0.103	2.194**
70 vs 80	0.636	-0.048	1.082	-1.037	-0.274	0.408
80 vs 90	0.240	1.361*	-0.790	-0.365	0.000	1.117
90 vs 100	-0.483	-0.936	0.801	-1.937**	0.253	0.646

(b)

N=300	DF		WF		DW	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$	t_M	$t_{distance}$
60 vs 70	-0.109	-0.159	0.727	-1.817**	-1.770**	-0.516
70 vs 80	-0.997	-0.610	-0.117	0.884	0.667	0.791
80 vs 90	-1.903**	-1.269	0.510	-1.193	-0.834	-0.875
90 vs 100	-0.561	-3.592**	-0.655	-3.752**	-1.403*	-4.347**

(c)

N=500	DF		WF		DW	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$	t_M	$t_{distance}$
60 vs 70	0.300	2.030**	0.445	-0.230	1.618*	2.411**
70 vs 80	-0.111	-0.075	0.266	0.593	-2.179**	-1.404*
80 vs 90	0.247	-3.439**	-0.148	-2.095**	-0.091	-2.269**
90 vs 100	-2.786**	-5.822**	-2.348**	-4.606**	-1.575*	-4.123**

(d)

N=1000	DF		WF		DW	
dod(%)	t_M	$t_{distance}$	t_M	$t_{distance}$	t_M	$t_{distance}$
60 vs 70	0.638	0.683	2.923**	3.816**	0.000	0.472
70 vs 80	-1.321*	0.441	-1.042	2.141**	-1.120	0.650
80 vs 90	-0.469	-3.584**	0.877	-0.760	-0.450	-1.313*
90 vs 100	-2.858**	-8.244**	-1.945**	-9.791**	-2.797**	-7.201**

** significantly different between two different *dod* in $\alpha = 5\%$

* significantly different between two different *dod* in $\alpha = 10\%$

5.3.3 Impact of operation scale

Another analysis in this problem can be presented in respond to a fixed amount of dynamic requests. We assume 100 dynamic requests will be revealed in real-time where the operators obtain variable static requests to plan the routes in minimum costs.

For instance, if number of static requests is 0, the total number of requests in this problem will be 0+100 =100. In addition, the degree of dynamism in this problem will be calculated as 100 % dynamism. Table 5.12 illustrates varying in number of static requests in the change of degree of dynamism.

Table 5.12 Problem sizes in the testing scenarios of operation scale

N_s	N_d	N_{total}	*dod* (%)
0	100	100	100
11	100	111	90
25	100	125	80
43	100	143	70
67	100	167	60
100	100	200	50
150	100	250	40
233	100	333	30
400	100	500	20

We have tested each of the case for 30 instances. As the same number of dynamic requests appear in the system, a fixed cost will be different when number of requests are known in advance is varying. Therefore, the extra operating cost is used to observe the relationship between high demand of known requests and the operating cost if the dynamic requests are the same. Table 5.13 demonstrates the results for extra number of vehicles and extra total travel distance against the different size of known requests to serve the same amount of dynamic requests. Again, WF needs more number of vehicles but shorter total travel distance while DF needs more travel distance but less number of vehicles. Although the number of vehicles contributed by DW is similar to DF, the total travel distance is shorter. The curves of the performances are shown in Figure 5.7 and Figure 5.8 which are extra number of vehicles and extra total travel distance against different size of static requests, respectively.

Another observation from the graphs can be seen when number of static requests increases, the needs for the extra number of vehicles and extra total travel distance will decrease to respond the same amount of dynamic requests. Since more known requests are known in advance, operators need more vehicles to serve the known requests. As a fixed number of dynamic requests reveals, the operators can assign the existing vehicles to serve the requests, more existing vehicles results in less extra vehicles requirement. Therefore, the principle of economy of scale can be seen in this case.

Table 5.13 Extra number of vehicles and extra total travel distance for 100 dynamic requests at different operation scale

N_{total}	DF		WF		DW	
	M	Distance	*M (% vs DF)*	Distance (% vs DF)	*M (% vs DF)*	Distance (% vs DF)
100	8.60	1538.55	11.03 (28.29)	1339.02 (-12.97)	8.60 (0.00)	1539.06 (0.03)
111	7.17	1490.55	9.60 (33.95)	1306.34 (-12.36)	6.90 (-3.72)	1477.00 (-0.91)
125	6.70	1443.74	9.47 (41.29)	1295.52 (-10.27)	6.93 (3.48)	1410.99 (-2.27)
143	6.67	1453.08	9.10 (36.50)	1270.23 (-12.58)	6.60 (-1.00)	1404.75 (-3.33)
167	6.47	1389.41	8.90 (37.63)	1261.94 (-9.17)	6.43 (-0.52)	1345.64 (-3.15)
200	6.03	1369.61	9.20 (52.49)	1265.08 (-7.63)	6.03 (0.00)	1307.87 (-4.51)
250	5.83	1349.52	8.83 (51.43)	1248.20 (-7.51)	6.17 (5.71)	1308.71 (-3.02)
333	5.67	1279.59	8.43 (48.82)	1209.67 (-5.46)	5.70 (0.59)	1222.98 (-4.42)
500	5.43	1262.72	8.47 (55.83)	1195.68 (-5.31)	5.33 (-1.84)	1198.56 (-5.08)

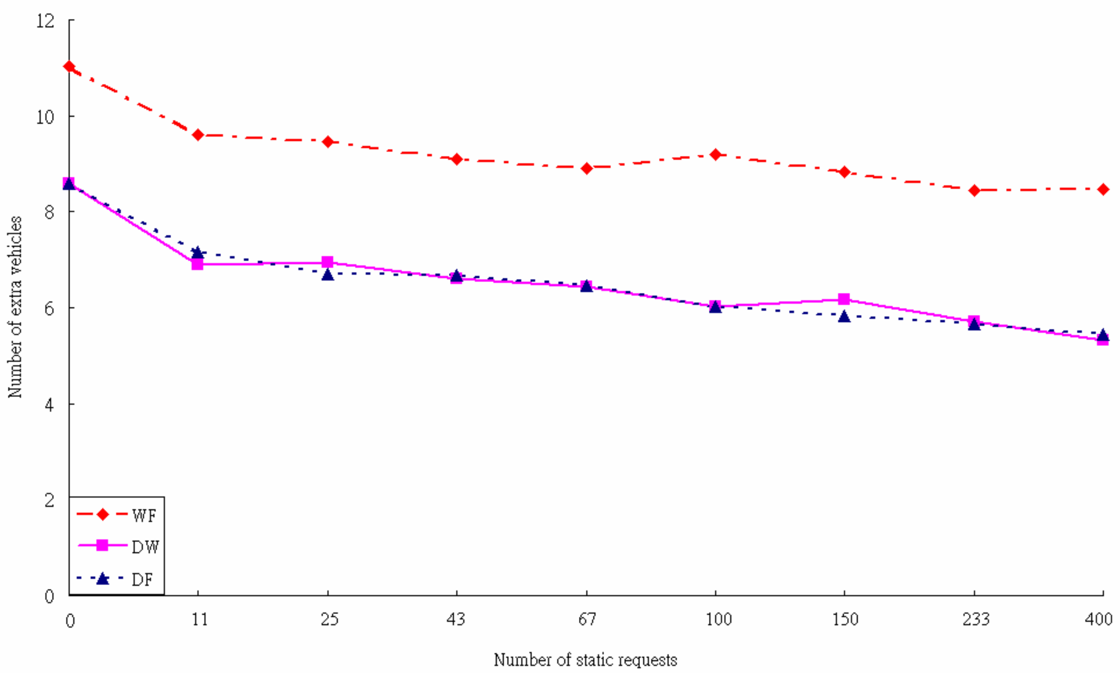

Figure 5.7 Extra number of vehicles in operation for 100 dynamic requests

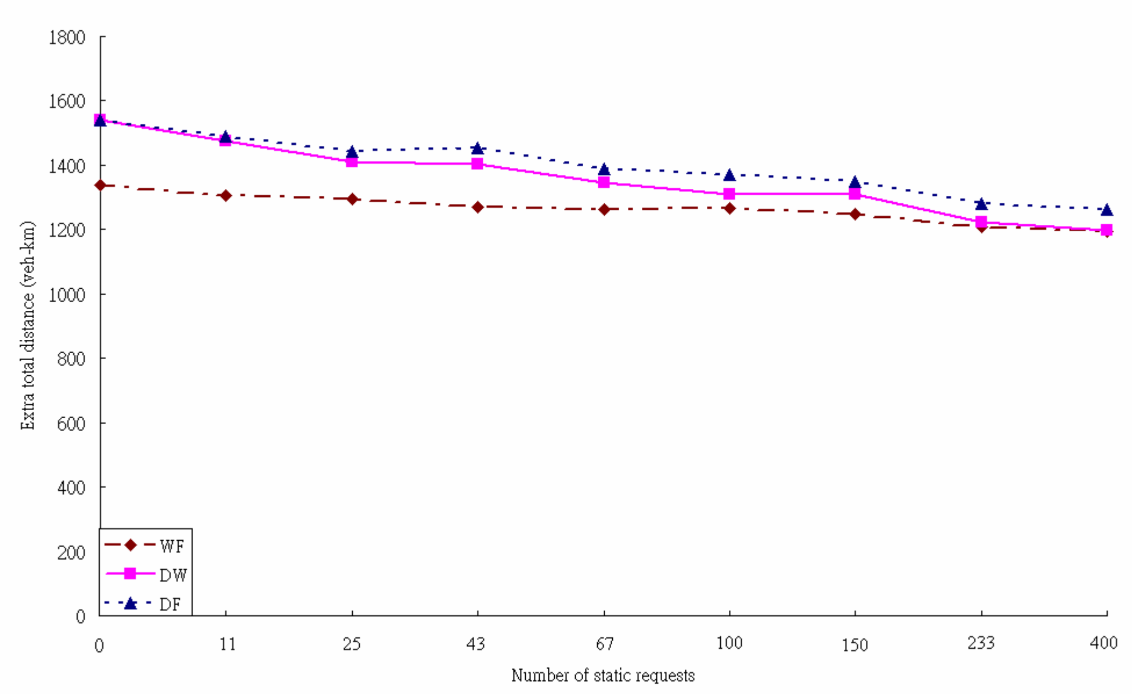

Figure 5.8 Extra total travel distance in operation for 100 dynamic requests

From the results above we conclude that the needs for the extra number of vehicles and extra total travel distance will decrease to respond the same amount of dynamic requests as known requests increase. It results in following the principle of economics of scale to serve the dynamic requests.

5.4 Summary

We simulate 30 instances for the size of 100, 300, 500 and 1000 requests to compare the strategies to investigate the system behavior. WF needs more number of vehicles but shorter total travel distance while DF requires less number of vehicles but longer total travel distance. DW improves to give less number of vehicles and shorter total travel distance compare to both of the strategies.

We found that the operating costs slightly decrease from about 60% dynamic problem to 100% dynamic problem. One reason maybe explained as follow. In the system of highly dynamic problems, some dynamic requests may not be served by the nearest vehicle with the planned service location whereas in fully dynamic system, no planned routes are existed before start serving the requests. Therefore, the operating costs for highly dynamic problems may be more than those for fully dynamic problems.

Another view of operation aspect is used to check the performance of varying the number of static requests in respond to a fixed amount of dynamic requests. The results can be seen in the state of economy of scale because more existing vehicles can be served the new requests; it results in less extra vehicles and total travel distance required to serve the same amount of dynamic requests. The curve for extra vehicles and total travel distance are thus decreasing as number of static request is increasing.

The findings in our studies are therefore summarized as follow:

(1) DW give less number of vehicles and shorter total travel distance compare to DF and WF strategies.

(2) A "counter-intuitive" observation is found since about 60% *dod* problems needs more operating costs than 100% *dod* problems.

(3) This counter-intuitive observation seems to be more obvious as in larger size of problems.

(4) The need for extra need of operating costs to serve the same amount of dynamic requests is decreasing as the number of requests in advance is increasing, therefore, more information known in-advance is helpful for reducing extra operating costs.

CHAPTER 6
CONCLUSIONS AND RECOMMENDATIONS

The conclusions and recommendations for this work are provided in this chapter. Section 6.1 summarizes this study. Some researches which we did not take into account in this thesis are given in section 6.2 for further research opportunities.

6.1 Conclusions

In CHAPTER 1 we have introduced that Dial-a-ride problem (DARP) is a variant of pickup and delivery problem with time windows (PDPTW). They share a similar mathematical formulation but in different problem setting, therefore, a good solution solving PDPTW problem may not be a good solution in DARP problem. Dynamic problems allow serving real time requests while static problems do not. The known requests are first considered to construct initial routes; however, the routes can be modified to serve new requests in real-time. The results in higher operating cost in dynamic problems than in static problems. In this study, we consider minimum number of vehicles and minimum total travel distance to serve the known and real-time requests.

CHAPTER 2 presented the literature reviews for the DARP problems. First, the problem definitions for this problem are described, and the difference between DARP and PDPTW problem. Since DARP problem is a NP-hard problem, heuristics can be used to solve this kind of problems. As static DARP problem has been well studied for a number of decades, experts used different methods to solve this problem. An example of solving the static DARP problem is using cheapest insertion method by Jaw et al. (1986). However, limited number of pioneering studies focused on DDARP. A study of using waiting strategies to solve dynamic PDPTW problem was proposed by Mitrovic-Minic and Laporte (2004). Owing to the different problem setting between DARP problem and PDPTW problem, using an efficient strategy solving PDPTW problem may not also be an efficient strategy solving DARP problem. Therefore, we developed three waiting strategies to solve DDARP.

In CHAPTER 3 we have introduced the concepts of the three waiting strategies. Two components, routing and scheduling, are required to solve DDARP. In routing, the cheapest insertion algorithm is considered to construct an initial solution. The solution is later improved by an exchange method. In scheduling, the three waiting strategies, drive first (DF), wait first (WF) and dynamic wait (DW), are used to arrange the waiting time of the vehicles at each stop on the route. DF requires the vehicles to drive as soon as possible whereas WF waits at one location until the latest time to arrive at the next location directly, and DW is the combination of the two strategies in less number of vehicles and shorter total travel distance. the DW strategy gives an idea of eliminating the state of detour in DF strategy. DF contributes smaller number of vehicles and longer total travel distance than WF in dynamic problem. DW gives a better solution than other strategies.

In CHAPTER 4 we have shown the implementation of the three waiting strategies to solve the dynamic problem. Three cases of vehicles states may happen when a real-time request arrives. In the first case, the request reveals as the vehicle idles before serving a planned location. If real-time request is feasible to be served immediately, the vehicle will first go to the new location and then go back to the original planned location. This results in longer travel distance since the state of detouring. In the second situation, the request is received when the vehicle idles after serving the planned location. If real-time request is feasible to be served, the request will be inserted between the served location and the next planned location. This results in shorter travel distance because detour is not needed. In the third case, the request is received when the vehicle is busy on traveling to the next planned location, the real-time request will be considered when the vehicle arrives to the planned location. In addition, the three strategies may meet in this situation.

In CHAPTER 5 we simulate a set of problems to test the three waiting strategies. In addition, five measures in the performance can be seen in the comparison. WF needs more vehicles but less total travel distance than DF and DW. Moreover, WF provides longer total duration time, it results in higher operating cost among three of the strategies. Although the results of DF and DW are similar, DW results in slightly small number of vehicles than DF. The better result in DW maybe helps in reducing high operating cost in practice.

The operating costs slightly decrease when *dod* increases from about 60% to 100% in our observations. We suggest that when less number of requests is known in advance, operators can treat the known requests as real-time requests. Therefore, the routes can be more flexible to serve the remaining real-time requests, but the level of service for customers may increase.

The system follows the principles of economy of scale. As we have tested the results at different *dod* level for fixed number of dynamic requests but varying the number of known requests. We found that in the case of high demand of static requests, less extra vehicles and extra total distance are needed to serve the same amount of dynamic request, therefore, more information known in-advance is helpful for reducing extra operating costs.

6.2 Recommendations for Further Research

Some ideas that are not concerned in this study can be developed in further research.

1. Use of meta-heurstics methods
   We have used exchange method to improve the initial solution. The use of meta-heuristics methods to improve the initial solution may be considered, and it may be helpful for finding a better solution in solving DDARP in the future.

2. Use the waiting strategy to solve Dynamic VRPTW problems
   We have just concerned the pickup location of real-time requests to a suitable position at the beginning and then put the corresponding delivery location to a suitable position after its pickup location. The current methodology may be more useful in solving dynamic VRPTW problems.

3. Ideas of other dynamic dispatching strategies

We update the planned routes in real-time into three cases. However, they are not flexible when real-time requests arrive. Ideas of other dynamic dispatching strategies can be faster in respond to the real-time requests. In addition, if a vehicle idles without carrying passengers, operators can reposition the idle vehicle to region of high demand intensity in order to reduce the time of response.

REFERENCES

Branke, J., Middendorf, M., Noeth, G. and Dessouky, M., (2005), "Waiting Strategies for Dynamic Vehicle Routing," *Transportation Science*, Vol. 39, pp. 298－312.

Chou, S. J., (2004), "A Study of Vehicle Routing Problem with Time Windows Using Clustering and Tabu Search Algorithms," M.S. Thesis (in Chinese), Department of Information Management Chaoyang University of Technology, Taiwan.

Cordeau, J. F. and Laporte, G., (2003a), "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," *Transportation Research* Part B, Vol. 37, pp. 579 – 594.

Cordeau, J. F. and Laporte, G., (2003b), "The dial-a-ride problem (DARP): Variants, modeling issues and algorithms," *4OR: A Quarterly Journal of Operations Research*, pp. 89 – 101.

Coslovich, L., Pesenti, R., Ukovich, W., (2006), "A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem," *European Journal of Operational Research*, Vol. 175, pp. 1605 – 1615.

Diana, M. and Dessouky, M. M., (2004), "A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows," *Transportation Research* Part B, Vol. 38, pp. 539 – 557.

Hsin, M. H., (2005), "The Vehicle Routing Problem for Dial-a-ride System – A Case of Fu-Kang Bus in Taipei," M.S. Thesis (in Chinese) Institute of Transportation and Communication Management Science, College of Management, National Cheng Kung University, Taiwan.

Jaw, J., Odoni, A. R., Psarafits, H. N. and Wilson, N. H. M., (1986) "A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows," *Transportation Research* Part B, Vol. 20, pp. 243 – 257.

Larsen, A., (2000), "The dynamic routing problem", Ph. D. dissertation, Technical University of Denmark, Kongens, Lyngby, Denmark.

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

Li, H. and Lim, A., (2001) "A Metaheuristic for the Pickup and Delivery Problem with Time Windows," Tools with Artificial Intelligence, Proceedings of the 13th International Conference.

Lu, P. Y., (2002), "Solving the Dynamic Pickup and Delivery Problem with Soft Time Windows using Tabu Search," M.S. Thesis (in Chinese), Department of Industrial Engineering Chung Yuan Christian University, Taiwan.

Madsen, O. B. G., Ravn, H. F. and Rygaard, J. M., (1995), "A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives," *Annals of Operations Research*, Vol. 60, pp. 193 – 208.

Mitrovic-Minic, S., (1998), "Pickup and Delivery Problem with Time Windows: A Survey," technical report SFU CMPT 1998-12.

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

Mitrovic-Minic, S., Krishnamurti, R. and Laporte, G., (2004), "Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows," *Transportation Research* Part B, Vol. 38, pp. 669 – 685.

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

Psarafits, H.N. (1980), "A dynamic programming solution to the single vehicle many-to-many immediate request dial a ride problem," *Transportation Science*, Vol. 14, pp. 131- 154.

TOP – A website for PDPTW test instances

http://www.top.sintef.no/vrp/benchmarks.html (2007/1/10)

Tseng, J.J., (1992), "A Study on Static Dial-A-Ride Routing Problems," M.S. Thesis (in Chinese), Institute of Traffic and Transportation, College of Management, National Chiao Tung University, Taiwan.

Toth, P. and Vigo, D., (1997), "Heuristic Algorithms for the Handicapped Persons Transportation Problem", *Transportation Science*, Vol.31, pp. 60 – 71

APPENDIX A
STANDARD DEVIATION FOR THE THREE STRATEGIES IN THE SIMULATIONS

Table A.1 shows the standard deviation for the three strategies in different size of problems in Section 5.2. Table A.1 (a) is for the case of 100 requests, Table A.1 (b) for 300 requests, Table A.1 (c) is for 500 requests and Table A.1 (d) for 1000 requests. The tables show the standard deviation for each of the waiting strategy according to different rate of *dod*. The standard deviation of the comparisons of WF and DW strategies to the based DF strategy are shown, the values are used to calculate the *t*-values in section 5.2.

Table A.1 Standard deviation for the three strategies in different size of problems: (a) 100 requests, (b) 300 requests, (c) 500 requests, (d) 1000 requests

(a) 100 requests

N=100	DF		WF		DW		WF vs DF		DW vs DF	
dod(%)	S_m	$S_{Distance}$	S_m	$S_{Distance}$	S_m	$S_{Distance}$	S_m	$S_{Distance}$	S_m	$S_{Distance}$
0	0.699	48.144	0.699	48.144	0.699	48.144	0.000	0.000	0.000	0.000
10	0.776	66.508	0.828	58.544	0.814	64.206	0.855	39.513	0.450	17.822
20	0.890	72.599	0.968	65.379	0.794	75.663	0.817	42.903	0.521	25.971
30	0.925	62.158	1.048	41.227	0.860	62.177	0.828	44.974	0.691	31.625
40	0.935	74.380	1.029	60.158	1.020	66.842	1.061	51.269	0.740	34.318
50	0.724	54.195	0.868	50.860	0.988	58.261	0.819	41.676	0.712	43.216
60	1.088	77.429	1.252	51.189	1.217	71.673	1.020	57.522	0.785	44.038
70	1.075	75.278	1.112	58.757	1.037	63.435	0.817	85.708	0.995	56.059
80	0.994	76.374	1.033	54.367	0.937	78.775	0.928	57.326	0.900	49.649
90	0.944	65.528	1.215	61.200	1.008	80.914	0.944	71.304	0.925	63.909
100	1.133	83.872	0.999	53.884	1.133	84.297	1.455	68.082	0.000	2.773

(b) 300 requests

N=300	DF		WF		DW		WF vs DF		DW vs DF	
dod(%)	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$
0	0.986	64.328	0.986	64.328	0.986	64.328	0.000	0.000	0.000	0.000
10	1.042	101.642	1.028	81.556	1.062	103.529	1.074	50.246	0.490	40.183
20	1.133	88.265	1.083	82.700	1.074	81.661	1.221	54.763	0.900	46.820
30	0.922	117.861	0.858	102.061	0.847	96.223	0.935	77.030	0.730	55.974
40	1.066	92.593	1.177	93.445	1.392	94.954	1.324	85.997	0.961	74.722
50	1.363	110.730	1.422	90.037	0.928	102.023	1.258	80.942	1.155	62.188
60	0.988	101.746	1.390	72.922	0.728	87.290	1.368	86.349	0.828	80.830
70	1.143	98.979	1.484	85.525	1.223	97.116	1.857	101.816	1.456	74.356
80	0.947	108.086	1.135	89.061	1.098	79.075	1.357	98.914	1.299	116.315
90	1.196	91.947	1.429	91.336	1.393	76.245	1.507	113.928	1.478	97.634
100	1.149	89.789	1.289	84.078	1.194	69.518	1.224	106.819	0.365	39.513

(c) 500 requests

N=500	DF		WF		DW		WF vs DF		DW vs DF	
dod(%)	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$
0	1.305	68.512	1.305	68.512	1.305	68.512	0.000	0.000	0.000	0.000
10	1.203	125.005	1.299	97.110	1.232	131.341	1.245	62.924	0.587	36.810
20	1.117	96.036	1.306	99.943	1.081	100.778	1.129	70.474	0.759	52.191
30	0.915	86.784	1.252	85.188	1.135	102.498	1.406	87.681	0.907	66.247
40	1.531	103.888	1.833	111.254	1.479	116.086	1.456	101.608	1.279	79.794
50	1.129	134.471	1.177	115.160	1.095	108.323	1.224	107.441	1.147	109.693
60	1.306	108.170	1.525	119.497	1.264	122.839	1.276	104.255	1.341	83.958
70	1.273	109.816	1.245	128.916	1.126	130.203	1.493	95.989	1.357	100.991
80	1.507	104.843	1.639	134.729	1.252	118.162	1.606	115.474	1.773	100.313
90	1.629	114.631	1.493	139.082	1.306	121.795	1.829	125.427	1.995	151.481
100	1.299	103.242	1.375	110.301	1.299	103.232	1.599	130.393	0.000	3.127

(d) 1000 requests

N=1000	DF		WF		DW		WF vs DF		DW vs DF	
dod(%)	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$	S_M	$S_{Distance}$
0	1.243	120.096	1.243	120.096	1.243	120.096	0.000	0.000	0.000	0.000
10	1.305	131.885	1.507	108.576	1.354	135.026	1.177	61.497	0.747	51.986
20	1.461	147.206	1.745	123.395	1.583	157.890	1.455	119.753	1.048	88.022
30	1.617	131.619	1.717	123.417	1.932	166.635	1.456	115.024	1.326	114.895
40	1.377	170.024	1.878	177.649	1.343	179.313	1.826	126.569	1.230	124.789
50	1.915	166.261	1.884	138.695	1.416	147.091	1.807	133.973	1.306	123.121
60	2.051	158.680	1.756	168.098	1.709	139.024	1.756	192.523	1.689	105.902
70	1.557	159.786	2.273	130.641	1.423	139.804	1.547	149.946	1.424	136.074
80	1.760	143.934	1.925	127.796	1.343	126.277	1.455	147.335	1.520	128.198
90	2.080	118.981	2.189	158.133	2.030	152.699	2.360	179.782	1.554	141.724
100	1.592	139.297	2.057	108.233	1.642	144.641	1.906	147.096	0.183	23.707