ABSTRACT

Title of Document: MODEL AND ALGORITHM FOR SOLVING REAL TIME DIAL-A-RIDE PROBLEM

Taehyeong Kim, Doctor of Philosophy, 2011

Directed By: Professor Ali Haghani
Department of Civil and Environmental Engineering

This research studies a static and real-time dial-a-ride problem with time varying travel times, soft time windows, and multiple depots. First, a static DARP model is formulated as a mixed integer programming and in order to validate the model, several random small network problems are solved using commercial optimization package, CPLEX.

Three heuristic algorithms based on sequential insertion, parallel insertion, and clustering first-routing second are proposed to solve static DARP within a reasonable time for implementation in a real-world situation. Also, the results of three heuristic methods are compared with the results obtained from exact solution by CPLEX to validate and evaluate three heuristic algorithms. Computational results show that three heuristic algorithms are superior compared to the exact algorithm in terms of the calculation time as the problem size (in terms of the number of demands) increases. Also among the three heuristic algorithms, the heuristic algorithm based on

sequential insertion is more efficient than other heuristic algorithms that are based on parallel insertion and clustering first-routing second.

For the case study, Maryland Transit Administration (MTA)'s real operation of Dial-a-ride service is introduced and compared with the results of developed heuristic. The objective function values from heuristic based on clustering first-routing second are better than those from MTA's operation for all cases when waiting cost, delay cost, and excess ride cost are not included in the objective function values.

Also, the algorithm for real-time DARP considering dynamic events such as customer no shows, accidents, cancellations, and new requests is developed based on static DARP. The algorithm is tested in a simulation framework. In the simulation test, we compared the results of cases according to degree of gap between expected link speeds and real link speeds. Also for competitive analysis, the results of dynamic case are compared with the results of static case, where all requests are known in advance. The simulation test shows that the heuristic method could save cost as the uncertainty in new requests increases.

MODEL AND ALGORITHM FOR SOLVING REAL TIME DIAL-A-RIDE PROBLEM

By

TAEHYEONG KIM

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor Ali Haghani, Chair
Professor Paul Schonfeld
Professor Gang-Len Chang
Professor Bruce Golden
Professor Sung Lee

Dedication

This work is dedicated to my dear parents

Acknowledgements

I wish to express my sincere appreciation to my advisor, Professor Ali Haghani, for his guidance, encouragement and continuous support during my Ph. D. studies. His broad-range knowledge and insightful critiques provided excellent guidance to this dissertation. It has been a real pleasure and privilege working with him.

I would like to thank my advisory committee members, Professor Paul Schonfeld, Professor Gang-Len Chang, Professor Bruce Golden, and Professor Sung Lee for all their valuable suggestions and comments on this research. Their advice and suggestions throughout the completion of this research were of great value to me. Also, I am grateful to Professor Kyungwhan Kim in Korea for his support and guidance.

I would like to thank Minseok Kim and Myungseob Kim for their help for this dissertation. Also, I am grateful to our group members who have given helpful comments to me. Special thanks go to Masoud Hamedi, Rafael Olarte, Kaveh Farokhi, and Mona Asudegi.

I wish to express my deep gratitude to my family members who have encouraged me with their love and support for the long time. My warmest thanks and gratitude go to my parents. It would be hardly possible to accomplish this work without their sacrifice and tremendous support. Also, I would like to thank my two brothers and two sisters-in-law. My gratitude is extended to my church members who always understand me with their warm hearts and loves. Finally, I would like to thank God.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background

Dial-a-Ride service (also called demand responsive or paratransit) is the most widely available transit service, with 6,700 agencies providing transit service in the United States[1]. Dial-a-Ride service is comprised of passenger cars, vans or small buses operating in response to calls from passengers or their agents to the transit operator. The operator dispatches a vehicle to pick up the passengers and transport them to their destinations. Dial-a-Ride service can be described as shown in Figure 1.1.

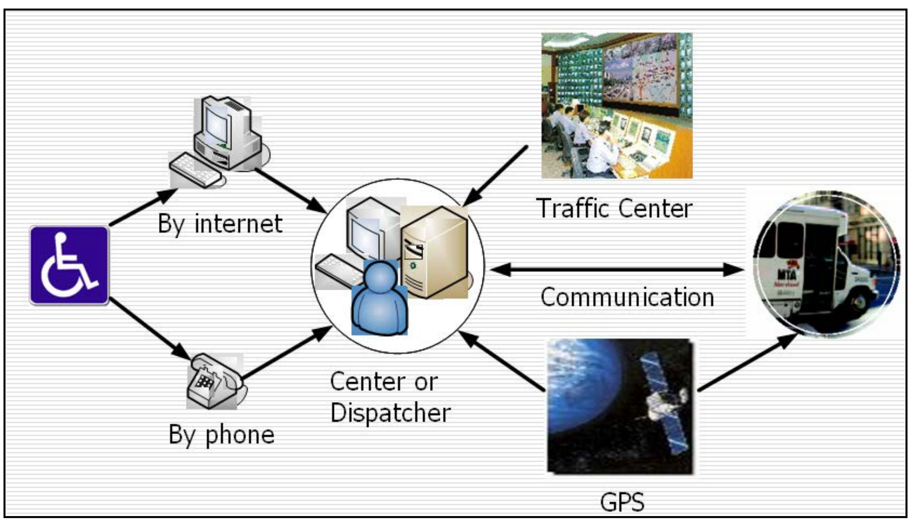


Figure 1.1 Dial-a-Ride Service

Most agencies limit this service to disabled persons, their attendants and

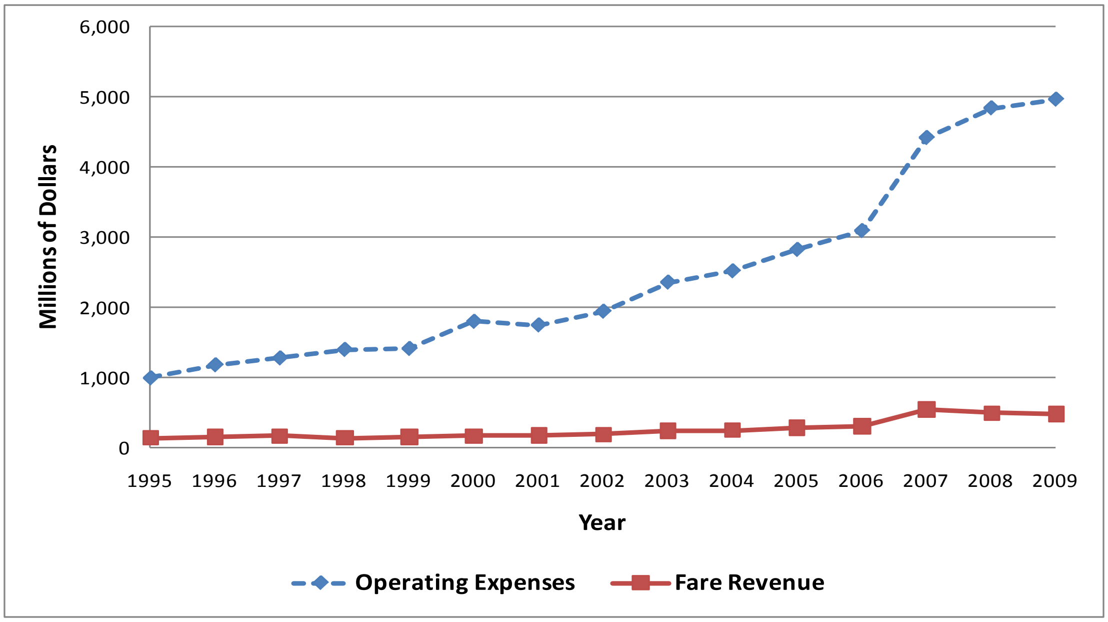
[1] Based on American Public Transportation Association (APTA) database: National total 6,700 agencies (Report year 2009), 2011 Public Transportation Fact Book.

companions, or seniors. There were 54 million people with one or more physical or mental disabilities in the United States in 2008[2], and this number is increasing as the population as a whole is growing older. Americans with Disabilities Act (ADA) of 1990 was passed with the purpose of eliminating discrimination against individuals with disabilities and limited access to transportation that have kept these people from participation in the many aspects of society such as employment, public accommodation, recreation, and health services. This act requires that all transit agencies which operate a fixed route system have to provide paratransit and other special transportation services as a supplement service for individuals with disabilities[3].

Dial-a-ride services are operated by public transit agencies, non-for-profit organizations, and for-profit companies or operators. Unlike regular fixed route transit, dial-a-ride provides shared-ride, door-to-door, or curb-to-curb services with flexible routes and schedules using passenger cars, vans or small buses. Since most true dial-a-ride services in United States are subsidized, the cost to the rider can be very low. Most of the operating expenses are spent for purchased transportation and vehicle operations. Figure 1.2 shows the trends of operating cost and fare revenue for paratransit in the United States between 1995 and 2009. We can see that the gap between operating expenses and fare revenue is steadily increasing every year. In 2009 the total operating cost of paratransit services in the nation exceeded 4.9 billion dollars while 0.48 billion dollars was collected in fares. Among the total operating cost, 2.6 billion dollars was spent for purchased transportation and 1.5 billion dollars

[2] Based on 2008 American Community Survey
[3] By sec. 223, ADA of 1990.

was spent for vehicle operations.



Source: America Public Transportation Association (APTA) database for 2009 (2011 Public Transportation Fact Book)

Figure 1.2 The trends of operating expenses and fare revenue for paratransit

The latest development of advanced information technologies such as automatic vehicle location (AVL), Global Position Systems (GPS), digital telecommunication, computers, and GIS, are making dial-a-ride systems more efficient, productive, and reliable. It is necessary to develop an integrated decision support system that combines these advanced technologies to provide more efficient and effective dial-a-ride routing and scheduling for real world applications. Also a decision support system needs appropriate on-line algorithms for solving these large scale routing and scheduling problem.

1.2 Scope of Dial-a-Ride Problems (DARP)

The Dial-a-Ride Problem (DARP) belongs to the generic class of vehicle routing and scheduling problems and has been extensively studied for several decades. In the DARP, n customers specify the locations of their origin and destination, the desired time of pickup or delivery, and specific type of transportation requirement. The aim of DARP is to design vehicle routes and schedules for those customers such that a specified objective is optimized. The general objective of DARP is to minimize total transportation cost and at the same time minimize user inconvenience under a set of constraints. What makes the DARP different and somewhat more difficult than most other routing problems is that transportation cost and user inconvenience must be weighed against each other when designing a solution (Cordeau, and Laporte, 2003a, 2003b).

Generally, DARP can be described in terms of the nature of the Dial-a-Ride system (Chan, 2004). The nature of Dial-a-Ride systems can be categorized as follows:

1) the pattern of origins to destinations (one-to-many, many-to-many, many-to-one)
2) the type of reservation (advanced, real-time or both)
3) the number of depots (single or multiple)
4) the number of vehicles (single or multiple)
5) the type of requests (pick-up request, drop-off request or both)
6) the treatment of travel time (static or dynamic)

There are variations in the model objectives and constraints. And, these complicate the classification of DARP. Also this makes it difficult to compare algorithms that focus on different solution types.

The objectives can be categorized as follows:

1) Objectives related to service providers:

   minimizing the total vehicle travel time,

   minimizing the number of vehicles used,

   minimizing vehicles' waiting time, and,

   maximizing total number of trips per vehicle

2) Objectives related to customers:

   minimizing customers' excess ride time,

   minimizing customers' waiting time, and,

   minimizing customers' service time deviations

Generally, most DARPs use a general cost function formed to combine several of the above objectives together as objective function. Also most DARPs include several constraints that must be satisfied by each route as follows:

1) round trip: every route starts and ends at the depot
2) coupling: for every request i, the origin/destination pair (i^+, i^-) must belong to the same route
3) precedence: the origin stop i^+ must be visited before the destination stop i^-
4) vehicle capacity: vehicles are limited in capacity by seating and vehicle type at any instant

5) time window: customers usually specify either desired pick-up or drop-off times and must be scheduled to be picked up or dropped off at specific time periods (time windows)
6) route duration: the total duration of each route must not exceed a specified time as a routing and scheduling criterion.
7) maximum ride time: the ride time of any user must not exceed a specified maximum ride time.
8) maximum waiting time: the total waiting time at origin stops must not exceed a specified time when the vehicle is carrying passengers while no waiting at destination stops is allowed.

1.3 Motivation of Research

Generally, all demands are known in advance based on reservations on previous days or subscriptions for regular service in static DARP. Usually this problem needs to be solved before the operations start at the beginning of every day. Before Fu (1999, 2002), all researchers assumed that travel times in an urban traffic environment are fixed and constant. In reality, travel times are subject to change according to the time of the day, the current weather conditions, the time of the year, accidents, events, etc. These variations in travel times may have important effects on the reliability and quality of routing and scheduling. Most studies related to the DARP assume that the service is provided by a single depot that has a fleet of m vehicles to simplify the complicated real-world problem. But, Dial-a-Ride service in many instances is provided from multiple depots especially in widely urbanized areas like metropolitan cities in which dial-a-ride service is provided by two or more private companies.

Wilson et al. (1971), Wilson and Weissberg (1976) and Wilson and Colvin (1977) were among the first to study the DARP with specific interest in developing real-time (dynamic) algorithms for several paratransit systems. Due to high operating costs, most of the dial-a-ride systems turned into reservation-based (static) operation after late 1970s. Recently, due to the advancement of computing and real-time transportation surveillance technologies, real-time DARPs are becoming focus of attention for operations researchers and paratransit operators. There are only a few studies that deal with algorithm development and solution of real-time DARPs considering real-time request.

Also, there is a need to develop fast and appropriate algorithms for solving both static and the real-time DARPs efficiently. Thus, this dissertation research mainly focuses on developing algorithms and solution approaches for both static and real-time DARPs with many origins to many destinations considering time varying travel times, soft time windows, multiple depots, and heterogeneous vehicles. This research extends the works of Fu (1999a, 2002a) and Xiang et al. (2008) by considering a more realistic objective function and constraints for dial-a-ride problem that were ignored in their work, and considering multiple depots.

1.4 Research Objective and Scope

The main objectives of this research can be summarized as follows:

1. Formulate a new model: This research will develop a formulation for a static dial-a-ride model considering time-dependent travel times, soft time windows, multiple depots, and heterogeneous vehicles. The model will be formulated as a mixed integer mathematical program (MIP).

2. Develop a heuristic methodology to solve this problem: The real world DARPs are very difficult to solve exactly within a reasonable computing time. The formulation that is to be developed in 1 above is no exception. Therefore, we will develop a heuristic algorithm for finding reasonable solutions to this problem within a reasonable time in order to be used in a real-time situation.

3. Perform extensive numerical test and sensitivity analysis: We will evaluate the performance of the algorithm developed in 2 above in a variety of test problem instances. We will compare its results with the results of optimal solutions in small test problems. In large problems sensitivity analysis will be performed.

4. Build simulated frameworks and Implement: It is necessary to build simulated frameworks and provide a methodology to evaluate the performance of the developed model and algorithm. We will develop a simulation framework based on real-world data obtained from one of the local transit agencies that provides paratransit service. The simulation framework will mimic the real time travel conditions and generate the demand based on the real time data. This will allow us to evaluate the performance of the system and its applicability in real world operations.

1.5 Organization of the Dissertation

The organization of this dissertation is as follows. Chapter 1 introduces the background and motivation for this research, and the research objectives. Chapter 2 summarizes the literature about models, algorithms and solution approaches for the static and dynamic DARPs. In Chapter 3, we describe the characteristics of our problem and then present the proposed formulation of the static DARP. Chapter 4 presents the developed algorithm for solving static DARP considering time varying travel times, soft time windows, multiple depots, and heterogeneous vehicles. The computational results which are based on several test problems for static DARP are discussed. In Chapter 5, the case study for real world large-scale DARP is discussed and the results of heuristic method are compared with real operation. Also, sensitivity analysis for the parameters of objective function is described. The methodology for solving dynamic DARP considering dynamic demands, cancellation, no-show, accidents, and real time travel time is described in Chapter 6. Also, our developed model and algorithm for Dynamic DARP is implemented and tested in the simulation framework. Finally, the conclusion and further study will be in Chapter 7.

Chapter 2: Literature Review

In this chapter, first we briefly discuss previous research related to the static and the dynamic DARP and then we describe modern heuristics for solving real world DARP. At the end we present a summary and discuss the characteristics of our proposed model. Several surveys on models and algorithms developed for the DARP can be found in Savelsbergh and Solomon (1995), Mitrovic-Minic (1998, 2001), Desaulniers et. al (2002), and Cordeau and Laporte (2003a, 2007).

Many models and algorithms for the DARP have been developed over the last 40 years. Generally, DARP can be categorized into two types based on the nature of the demand. The first type is the static DARP, in which all demands are known in advance based on reservations on previous days or subscriptions for regular service. The objective of routing and scheduling for this problem is to determine the assignment of all demands to the available vehicles and develop the respective routes and schedules for those vehicles. Usually this problem needs to be solved before the operations start at the beginning of every day. It can be performed off-line and thus demands less time. The second type is the real-time (or dynamic) DARP, in which some demands arrive in real-time. The objective in this case is to assign the new demands in real time into the existing routes and schedules of vehicles already in service. In this case, the routes and schedules for the new demands must be found in a very short period of time. Thus, unlike the static DARP, a fast online routing and scheduling algorithm is required to solve this real-time DARP.

The algorithms developed for DARP can be categorized into exact methods, classical heuristic algorithms, and modern heuristic algorithms. Classical heuristics can be classified into construction heuristics and improvement heuristics. Construction procedures build a set of feasible routes starting from the information that define dial-a-ride problem. Improvement procedures start with a set of feasible routes that are found using construction procedures and seek to improve the solution through a sequence of steps. Construction algorithms can be divided into three groups: Decomposition methods, Insertion algorithms, and Clustering-first routing-second. Modern heuristic methods were first designed in the early 1980s to solve complex and difficult combinatorial optimization problems that arise in many practical areas. After 1990s, most research in DARP has focused on development of modern heuristics or metaheuristics such as simulated annealing, tabu search, and Genetic Algorithms (GA).

DARP with time windows is a NP-hard problem (Baugh et al. 1998). In real world, there are sometimes thousands of requests to be served. Due to the complexity of the problem and its large scale, it is impossible to find optimal solutions for this problem within a reasonable computation time. This is the main reason why most of the research in this area is focused on finding fast heuristic algorithms that find good solutions in reasonably short times.

2.1 The Static DARP

The static DARP can be categorized into the single vehicle and multiple vehicle problems according to the number of vehicles which serve customers. If the number of customers in each route remains small, algorithms developed to solve the single vehicle problems can be used as subroutines in multiple vehicle problems.

2.1.1 The Single Vehicle Static DARP

Exact methods for solving dial-a-ride problems were first developed by Psaraftis (1980). Psaraftis (1983) modified an exact dynamic programming algorithm he had developed earlier (Psaraftis, 1980) for the single vehicle many-to-many immediate request Dial-a-Ride problem. In this problem, each customer has specified upper and lower bounds for his pickup and delivery times, i.e., time windows and the objective is to minimize the time needed to provide service to all customers. The major difference between the first (1980) and the second algorithm (1983) is the substitution of backward recursion with forward recursion in order to consider time windows. His interesting innovation is the use of a maximum position shift (MPS). In order to guarantee good service to all customers he has introduced a bound for both the pickup position shift and the delivery position shift.

Sexton and Bodin (1985a, 1985b) investigated the single vehicle dial-a-ride problem in which each customer specifies a desired time for pick-up and drop-off. They developed a heuristic routing and scheduling algorithm based on Benders' decomposition. Decomposition methods are based on the idea of dividing a problem into two phases: routing and scheduling. Their objective was to minimize customer inconvenience. In their model, the objective function is expressed as a linear combination of excess ride time, that is, the time difference between the actual ride time and the direct ride time of a user and the deviation of the user's desired drop-off time from the actual drop-off time. They applied their algorithm to several data sets from Gaithersburg and Baltimore, MD., containing between 7 and 20 users.

For the same problem, Desrosiers et al. (1986) proposed a forward dynamic programming algorithm. Their objective function was to minimize the total distance traveled instead of the total time required to serve all customers. Their algorithm was applied to solve instances containing up to 40 users.

Van Der Bruggen et al. (1993) considered a single depot, a set of customers with known demands, and time windows. For this problem, they developed a local search method based on a variable-depth search which consists of two phases. In the first phase, a feasible route is constructed, and in the second phase the route is improved. In order to escape from local optima, they used a simulated annealing algorithm. They found high quality solutions by simulated annealing. But, their method requires a relatively large computation time. Their objective function was to minimize total route duration. Their method was applied to real data set of Toronto, Canada with request sizes ranging from 5 to 38.

2.1.2 The Multiple Vehicles Static DARP

Jaw et al. (1986) proposed ADARTW (Advanced Dial-A-Ride with Time Window). In this model, time windows are imposed on the pick-up time of inbound requests and on the drop-off time of outbound requests. Also, a maximum ride time is imposed for each user and is expressed as a linear function with respect to the direct ride time of the user. In addition, no waiting at a stop is allowed whenever passengers are aboard the vehicle. Their objective function was to minimize the combination of total customer disutility. They developed one of the first insertion heuristics using sequential insertion for the DARP. It was applied to randomly generated instances containing 250 requests and a real world instance with 2617 requests.

Bodin and Sexton (1986) introduced a cluster-first route-second heuristic for the problem, employing a space time heuristic to form a route for customers in a cluster. The objective was to minimize total customer inconvenience, which is the weighted sum of differences between actual and desired delivery time, and differences between actual and shortest possible ride times.

Fuzzy arithmetic rules and logic were first applied to develop the schedules for DARP by Kikuchi and Donnelly (1992). They introduced fuzziness in the values of two basic input parameters, travel time and the desired time of vehicle stop. Their algorithm was based on two steps: developing the initial route and inserting leftover trips. This algorithm was originated from those developed by Jaw et al. (1986) and Kikuchi and Rhee (1989).

Ioachim et al. (1995) proposed a mini-cluster first, route second approach using column generation to solve a multi-vehicles, door-to-door, handicapped transportation system with time windows. A mini-cluster is a set of geographically and temporally cohesive transportation requests that can feasibly be served by the same vehicle. Specifically, they designed vehicles to simultaneously accommodate three different types of handicapped persons: the ambulatories who use regular seats, those in folding wheel-chairs, and those in non-folding wheel-chairs. Their objective was to minimize total mini-cluster cost, that is, the sum between the total internal travel time and the estimated external travel time (i.e., the average external distance multiplied by the number of the mini-clusters in the solution). Internal travel time is the travel time within a mini-cluster and external travel time is the travel time between one mini-cluster and another. They tested a large scale problem with 2545 requests generated by an operation day in Toronto, Canada.

Toth and Vigo (1997) examined the problem of determining an optimal schedule for a fleet of vehicles used to transport handicapped persons with time windows in an urban area and developed a parallel insertion procedure for their problem. The heuristic tended to produce several short routes. To improve this heuristic procedure they developed a tabu thresholding post-optimization procedure. Their objective function was to minimize the total cost of service. The instances of between 276 and 312 requests were tested based on real data of the city of Bologna, Italy. The results showed significant improvement over previous hand-made solutions.

Baugh et al. (1998) presented a heuristic algorithm for solving multiple vehicles DARP. They used cluster-first route-second strategy and the problem was solved by using simulated annealing for clustering and a modified space-time nearest neighbor heuristic for developing the routes for the clusters. In addition a tabu list was included to improve the performance of the simulated annealing algorithm. The algorithm was tested on instances randomly generated with up to 25 customers. Also, it was tested on a real-life data set containing up to 300 customers that was provided by the Winston Salem Transit Authority (WSTA).

Before Fu (1999b), all researchers assumed that travel times in an urban traffic environment are fixed and constant. He proposed improving paratransit scheduling by considering dynamic and stochastic variations in travel time. He analyzed a specific case having 463 trips in the morning peak time with hypothesized O-D travel time variation pattern and evaluated the performance of schedules such as total travel time, vehicle productivity, number of vehicles, average ride time, average service time deviation, and percentage of violated trips. He found that both dynamic and stochastic variations in travel times had important effects on the quality of the schedules, and an appropriate consideration of these variations in the scheduling process could substantially improve the reliability and productivity of the schedules.

Fu (2002b) developed a DARP model explicitly considering the time varying, stochastic attributes of travel times and an algorithm which is efficient enough for solving large size problems of this type. He used parallel insertion heuristics for his model. His model's objective function was to minimize a weighted sum of the total client inconvenience such as excess ride time and service time deviation, and the cost to the service providers like total vehicle travel time. He introduced a unique travel time model satisfying FIFO (First In First Out) assumption, and tested his model and algorithm on a set of hypothetical instances with 2800 trips.

Cordeau and Laporte (2003) proposed a tabu search heuristic for the static multi-vehicle dial-a-ride problem. Starting from an initial solution s_0, the algorithm moves at iteration t from s_t to the best solution in a neighborhood $N(s_t)$ of s_t. As is common in such algorithms, a continuous diversification mechanism is put in place in order to reduce the likelihood of being trapped in a local optimum. Their model was tested on 200 and 295 requests based on real data.

Diana and Dessouky (2004) presented a new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. This algorithm is a parallel insertion heuristic with regret metric, aimed at improving the myopic behavior that is often the drawback of insertion algorithms. Instead of ranking the requests by certain criteria such as earliest time window or latest time window as in classic insertion heuristics, the regret insertion build up an incremental cost matrix for each of the unassigned requests when assigned to each of the existing vehicle routes. The proposed algorithm was tested on instances of 500 and 1000 requests built from the data of paratransit service in Los Angeles County. The computational results show the effectiveness of their approach in terms of trading-off solution quality and computational times.

A modified parallel insertion heuristic to solve the DARP with multi-dimensional capacity constraints was proposed by Wong and Bell (2006) and the performance of the proposed algorithm was tested in simulation. The objective function of the problem was a weighted combination of the total operating time of the dial-a-ride fleet, the passenger delay (time extra to their direct travel time), and the cost for taxi trips for transporting the requests that are not inserted. A set of hypothetical problems having a total 150 demands are generated and solved.

Xiang et al. (2006) used a local search strategy based on insertion algorithm, a diversification strategy, and an intensification strategy for solving a large scale static DARP under complex constraints. The performance of the heuristic was evaluated by intensive computational tests on some randomly generated instances. With a good initial solution, lager instances up to 2000 requests were solved in less than 10 hours on a popular personal computer.

A grouping genetic algorithm for clustering phase and an insertion mechanism for routing phase to solve the problem of transporting handicapped people in terms of service quality and number of used vehicles were developed by Rekiek et al. (2006). The proposed algorithm was tested on instances of 100 to 164 clients. Melachrinoudis et al. (2007) proposed a heuristic using tabu search with request reinsertions to minimize a linear combination of total vehicle transportation costs and total clients' inconvenience time for client transportation in a health-care organization. Their algorithm was tested on problems with up to 50 requests. Both Rekiek et al. (2006) and Melanchrinoudis et al. (2007) dealt with multi depot case and individual depots (or centers) have their own service areas in their problems.

A rejected-reinsertion heuristic for the static multi-vehicle DARP was proposed by Luo and Schonfeld (2007). A rejected-reinsertion operation is performed each time it is infeasible to insert a new request into the vehicle routes. Each assigned request close to the new request in time frame and geographic location is tentatively removed from its current vehicle and the new request is inserted into the best position in that vehicle route, followed by the reinsertion of the removed request elsewhere in the system. Of all available rejected-reinsertions, the least-cost one is then implemented. The heuristic was tested with their own problems randomly generated and with test problems from Dianna and Dessouky (2004). The proposed heuristic achieves vehicle reductions of up to 17% over the parallel insertion heuristic.

Jorgensen et al. (2007) developed a heuristic using genetic algorithm to construct clusters and space-time nearest neighbor procedure to construct the routes.

The objective of their problem was to minimize a linear combination of transportation time, ride time, excess of maximum ride time, waiting time, time windows violations, work time and excess work time. They solved instances of 24 to 144 requests.

Parragh et al. (2009) introduced a heuristic two-phase solution approach for the dial-a-ride problem with two objectives that the one is to minimize total distances traveled by vehicles and the other is to minimize mean user ride times. Phase one consists of an iterated variable neighborhood search-based heuristic, generating approximate weighted sum solutions. Phase two is a path linking module, computing efficient solutions. Instances of 16 to 96 requests randomly generated were tested. Recently, Parragh et al. (2010) proposed variable neighborhood search-based heuristic, using three classes of neighborhoods such as swap, chain, and zero split neighborhood. Their heuristic was tested on random instances containing between 24 and 144 requests used by Cordeau and Laporte (2003).

A summary of the static DARPs is presented in Table 2.1. This table is based on the study by Cordeau and Laporte (2007).

Table 2.1 Summary of the static DARPs

Authors	Type of Problem	Objective Function	Main constraints	Method (Algorithm)	Size of Problem Solved
Psaraftis (1983)	Single vehicle, Many-to-many, Single depot	Minimize route duration	Hard time windows (pick up and drop off) No Capacity, No precedence	Exact, Dynamic programming (forward recursion)	N≤9
Sexton and Bodin (1985a,b)	Single vehicle, Many-to-Many, Single depot	Minimize total customers' inconvenience	Hard time windows (Upper bounds on pick up and drop off times), Vehicle capacity	Heuristic, Benders's decomposition procedure	7≤N≤20 (real data)
Bodin and Sexton (1986)	Multi-vehicles Many-to-Many, Single-depot	Minimize total customers' inconvenience	Hard time windows (pick up and drop off), Vehicle capacity	Heuristic, Cluster-first route-second	N≤85
Desrosiers et al. (1986)	Single vehicle, Many-to-Many, Single depot	Minimize total distance traveled	Hard time windows (pick up and drop off), Vehicle capacity	Exact, Dynamic programming (forward recursion)	N≤ 40 (real data)
Jaw et al. (1986)	Multi-vehicles, Many-to-Many, Single depot	Minimize the combination of several types of disutility	Hard time windows (pick up or drop off) Vehicle capacity, Max. ride time, Max. time deviation	Heuristic, Insertions	N=250 (simulation), N=2617 (real data)
Desrosiers et al. (1988)	Multi-vehicles, Many-to-Many, Single depot	Minimize total routing cost	Hard Time windows (pick up or drop off)	Heuristic, Mini-clustering and column generation	N≤200

Note: This table is based on Cordeau and Laporte (2007).

Table 2.1 (continued)

Authors	Type of Problem	Objective Function	Main constraints	Method (Algorithm)	Size of Problem Solved
Kikuchi and Donnelly (1992)	Multi-vehicles, Many-to-Many, Single depot	Minimize the disutility and idle time	Hard time window (pickup and drop off), Vehicle capacity, Max. ride time, Max. time deviation	Heuristic, Insertion and Fuzzy logic	$25 \leq N \leq 200$ (hypothetical data)
Van Der Bruggen et al. (1993)	Single vehicle, Many-to-Many, Single depot	Minimize the route duration	Hard time windows (pick up and drop off) Vehicle capacity	Local search method based on a variable-depth search	$5 \leq N \leq 38$ (real data)
Ioachim et al. (1995)	Multi-vehicles, Many-to-Many, Single depot	Minimize total mini-cluster cost (the total internal travel time + estimated external travel time)	Hard time windows (pick up and drop off), Multi-dimensional vehicle capacity (3 types of customers), Max. route duration	Heuristics, Create mini-clusters and group them by column generation.	$N=2545$ (real data)
Toth and Vigo (1997)	Multi-vehicles, Many-to-Many, Single depot	Minimize the total cost of service	Soft time windows (pick up or drop off), Vehicle capacity, Max. travel time	Heuristic, Parallel insertion followed by tabu thresholding	$276 \leq N \leq 312$ (real data)
Baugh et al. (1998)	Multi-vehicles, Many-to-Many, Single depot	Minimize combination of distances traveled by all vehicles, customer inconvenience, and the number of vehicles used	Soft time windows (pick up and drop off), Vehicle capacity	Heuristic, Simulated annealing, Tabu list	$N \leq 25$ (random data), $N \leq 300$ (real data)

Table 2.1 (continued)

Authors	Type of Problem	Objective Function	Main constraints	Method (Algorithm)	Size of Problem Solved
Borndörfer et al. (1999)	Multi-vehicles, Many-to-Many, Single depot	Minimize operational costs (drivers and vehicles)	Hard time windows (pick up and drop off), Vehicle capacity, Max. route duration	Heuristic, Set partitioning using branch-and-cut algorithm	$859 \leq N \leq 1771$ (real data)
Fu (1999a)	Multi-vehicles, Many-to-Many, Single depot, Dynamic and stochastic travel time	Minimize the total disutilities of the Service operator and customers	Hard time windows (pick up and drop off), Vehicle capacity, Vehicle service Time periods, Max. ride time	Heuristic, Insert Algorithm, Artificial neural network for travel time	N=3024 (real data)
Fu (1999b)	Multi-vehicles, Many-to-Many, Single depot, Dynamic and stochastic travel time	Minimize total weighted sum (client inconvenience and the cost to the service providers)	Hard time windows (pick up and drop off), Vehicle capacity, Service time window, Max. ride time	Heuristic, Parallel insertions	N=463 (real data)
Fu (2002b)	Multi-vehicles, Many-to-Many, Single depot, Dynamic and stochastic travel time	Minimize total weighted sum(client inconvenience and the cost to the service providers)	Hard time windows (pick up and drop off), Vehicle capacity, Service time window, Max. ride time	Heuristic, Parallel insertion	N=2800 (simulation data)

Table 2.1 (continued)

Authors	Type of Problem	Objective Function	Main constraints	Method (Algorithm)	Size of Problem Solved
Cordeau and Laporte (2003)	Multi-vehicles, Many-to-Many, Single depot	Minimize the total routing cost of all vehicles	Soft time windows (pick up and drop off), Vehicle capacity, Max. route duration, Max. ride time	Heuristic, Tabu Search	24≤N≤144 (random data) N=200 and 295 (real data)
Diana and Dessouky (2004)	Multi-vehicles, Many-to-Many, Single depot	Minimize a weighted sum of the total distance, the excess ride time, and the total idle times	Hard time windows (pick up and drop off), Vehicle capacity, Max. waiting time, Max. ride time	Heuristic, Parallel regret insertion	N=500 and 1000 (real data)
Wong and Bell (2006)	Multi-vehicles, Many-to-Many, Single depot	Minimize a weighted sum of the total operating time of fleet, the passenger delay, and the cost for taxi trips for transporting uninserted requests	Hard time windows (pick up and drop off), Vehicle capacity, Max. waiting time, Max. ride time	Heuristic, Parallel insertion and local post-optimization	N=150 (random data)
Xing et al. (2006)	Multi-vehicles, Many-to-Many, Single depot	Minimize a weighted sum of the fixed cost, mileage, driving time, waiting time, and service time	Hard time windows (pick up and drop off), Vehicle capacity, Max. waiting time, Max. ride time	Heuristic, Insertion algorithm, a diversification, and an intensification strategy	N<=2000 (random data)
Rekiek et al. (2006)	Multi-vehicles, Many-to-Many, Multi depot	Minimize the number of vehicles	Soft time windows (pick up and drop off), Vehicle capacity Max. ride time	Heuristic, Genetic algorithm	100≤N≤164 (real data)

Table 2.1 (continued)

Authors	Type of Problem	Objective Function	Main constraints	Method (Algorithm)	Size of Problem Solved
Lou and Schonfeld (2007)	Multi-vehicles, Many-to-Many, Single depot	Minimize the number of vehicles that satisfies all demands	Hard time windows (pick up and drop off), Vehicle capacity, Max. ride time	Heuristic, Parallel insertion (rejected-reinsertion)	N=500 and 1000 (real data)
Melachrinoudis et al. (2007)	Multi-vehicles, Many-to-Many, Multi depot	Minimize a linear combination of the transportation cost and user inconvenience	Soft time windows (pick up and drop off), Vehicle capacity	Heuristic, Tabu search	N<= 50 (random data)
Jorgenson et al. (2007)	Multi-vehicles, Many-to-Many, Single depot	Minimize a linear combination of transportation time, ride time, excess of maximum ride time, waiting time, time windows violations, work time and excess work time	Soft time windows (pick up and drop off) Vehicle capacity, Max. ride time	Heuristic, Genetic algorithm and space-time nearest neighbor procedure	$24 \leq N \leq 144$ (random data)
Parrah et al. (2009)	Multi-vehicles, Many-to-Many, Single depot	Minimize total distance traveled and mean user ride time	Soft time windows (pick up and drop off), Vehicle capacity, Max. ride time	Heuristic, Variable neighborhood search and path relinking	$16 \leq N \leq 96$ (random data)
Parrah et al. (2010)	Multi-vehicles, Many-to-Many, Single depot	Minimize total routing cost	Soft time windows (pick up and drop off), Vehicle capacity, Max. ride time, Max. route duration time	Heuristic, Variable neighborhood search	$24 \leq N \leq 144$ (random data)

2.2 Dynamic DARP

Dynamic DARP was first examined by Wilson et al. (1971, 1976, 1977) who developed real-time algorithms for several paratransit systems of Haddonfield, NJ, and Rochester, NY. They developed real time demands, multiple vehicles, many-to-many DARP model with no time windows using provisional assignment heuristic method.

Daganzo (1978) presented a model to evaluate the performance of many-to-many dial-a-bus system, and Stein (1978) developed a probabilistic analysis of the dial-a-ride problem, describing only the basic approach and its motivation at a fundamental level.

Psaraftis (1980) developed an exact optimization procedure to solve the single vehicle, many-to-many, immediate request dial-a-ride problem. He solved the dynamic problem as a sequence of static problems using backward dynamic programming. He used the MPS (Maximum Position Shift) constraint to prevent the possibility that the service of any particular customer would be indefinitely deferred by the algorithm. Only very small instances ($n \leq 9$) could be handled by this algorithm.

Madsen et al. (1995) applied a heuristic (REBUS) to the dynamic DARP with time windows, multiple vehicles capacities, and multiple objectives. This method is based on the insertion heuristic proposed by Jaw et al. (1986). It was applied to a real-life instance containing 300 users based on data of Copenhagen, Denmark.

Fu (1999a) developed a software system to integrate dial-a-ride routing and scheduling principles and practical experience considering travel time variability in urban roadway networks. He used the artificial neural network technique, which allows heuristic estimation of O-D travel times in a dynamic and stochastic fashion. Also he extended the insertion algorithm for this problem. His objective function was to minimize the total disutilities of the service operator and customers. A real scheduling problem from Edmonton, Canada, with a total of 3024 trips and 109 vehicles, was tested.

Teodorovic and Radivojevic (2000) developed a model based on fuzzy logic for dynamic DARP. They assumed that the passengers, dispatcher, and drivers equally have fuzzy notion of the travel times and distances. Thus, the time of travel given approximately can be represented by certain fuzzy sets and numbers. Using fuzzy arithmetic, they calculated waiting times for the vehicles, waiting times for the passengers, and moments of arrival at specific nodes. The values calculated in such a way represent the input data for approximate reasoning algorithms developed. The model developed was tested on 10 numerical examples with 900 requests.

Fu (2002a) proposed an on-line algorithm and a simulation model for solving and evaluating the dynamic DARP. Also, he presented technological components of an advanced paratransit operation system (APOS) and principle component and structure of the simulation system. In this model, time-dependent and stochastic traffic patterns are considered, and the model is able to consider the availability of real-time link travel time data. Finally, a series of simulation experiments were performed to investigate the differences in operational performance between a paratransit system using AVL (Automatic Vehicle Location) and one without AVL on a set of hypothetical cases ranging from 100 to 300 trips. He mentioned that the benefits of AVL due to increased flexibility in dynamic scheduling are highly case-dependent.

Although the GAs are considered to be good approaches for solving routing problems, they have not been explored in many instances of DARP. Uchimura et al. (2002) studied a dial-a-ride service, which operates public taxi as a door-to-door service, provided by small buses and/or vans. A genetic algorithm scheme was applied for optimization and finding the proper solutions for this problem. The model was tested on 10 requests based on real data.

In order to speed up computation time, Attanasio et al. (2004) developed several parallel tabu search heuristics for the dynamic DARP based on a tabu search previously proposed for the static DARP by Cordeau and Laporte (2003). The main ingredients of this procedure are as follows: parallelization strategy, static solution construction, feasibility check, and post-optimization phase. The heuristics were tested on a set of 26 instances; twenty of them were randomly generated, while the remaining six instances were real-life, large scale problems with 200 and 295 trips.

Coslovich et al (2006) proposed a two-phase insertion heuristic based on route perturbations. The first phase is run off-line and aims at creating a feasible neighborhood of the current route. The second phase is run on-board the vehicle every time a new requests occurs and has the purpose of inserting the delivery of the new customer in the current route. A simple insertion procedure allows for quick answers with respect to inclusion or rejection of a new customer. The initial solution is improved by means of local search using 2-opt arc swaps.

Xiang et al. (2008) studied a dynamic and stochastic dial-a-ride problem bearing complex constraints on a time-dependent network. A flexible scheduling scheme was proposed to dynamically cope with different stochastic events, such as the travelling time fluctuation, new requests, absences of customers, vehicle breakdowns, cancellations of requests, traffic jams and so on. When a new event occurs the schedule is re-optimized. This paper used a similar heuristic by Xiang et al. (2006) to solve the dynamic problem. The simulation results of different scenarios with different percentage of dynamic requests reveal that this scheduling scheme can generate high quality schedules and is capable of coping with various stochastic events.

Luo and Schonfeld (2011a) adapted an insertion-based rejected-reinsertion heuristic developed for the multi-vehicle static DARP to solve the dynamic DARP. The main objective was to minimize the number of vehicles used to satisfy all trip requests subject to service quality constraints. They developed two online implementation strategies, called immediate insertion and rolling horizon insertion, coupled with two variations of the insertion heuristic, rejected-reinsertion without and with periodic improvement procedures. The heuristics were tested on the same randomly generated data for the static DARP introduced by Luo and Schonfeld (2007). The online rejected-reinsertion heuristic with periodic improvement achieved the best results.

Also, Luo and Schonfeld (2011b) proposed three performance metamodels using the response surface metamodeling approach for the dynamic many-to-many DARP. The models predict, respectively, the minimum vehicle fleet size requirement, the average passenger time deviation from desired time, and the average passenger ride time ratio. The metamodels are validated using a set of randomly generated data.

A summary of the dynamic DARPs is presented in Table 2.2.

Table 2.2 Summary of dynamic DARPs

Authors	Type of Problem	Objective	Main constraints	Method (Algorithm)	Size of Problem Solved
Wilson et al. (1971)	Multiple vehicles, Many-to-many, Single depot	Minimize combination of user time and vehicle travel time	No time windows, Waiting time, Travel time and total time constraint	Provisional assignment heuristic	N<=250 (per hour) simulation data
Wilson et al. (1976)	Multiple vehicles, Many-to-many, Single depot	Minimize combination of wait time, wait time deviation, travel time, delivery time deviation, system resources	No Time windows, Vehicle capacity	Provisional assignment heuristic	N=40 and 80 (per hour) simulation data
Psaraftis (1980)	Single vehicle, Many-to-Many, Single depot	Minimize combination of route duration, ride time and waiting time	No Time windows, Vehicle capacity, Maximum position shift	Exact dynamic Programming (backward recursion)	N<=9 (numerical example)
Xu (1994)	Single vehicle, Many-to-Many, Single depot	Minimize system time	No time windows, Infinite and unit capacity	DDRP-PART	Numerical example

Note: This table is based on Cordeau and Laporte (2007).

Table 2.2 (continued)

Authors	Type of problem	Objective	Main constraints	Method (Algorithm)	Size of Problem Solved
Madsen et al. (1995)	Multi-vehicles, Many-to-Many, Single depot	Multi criteria Objective	Hard time windows (pick up or drop off), Vehicle capacity, Max. route duration, Max. deviation between actual and shortest possible ride times	Heuristic, Vertex insertion	N=300 (real data)
Fu (1999a)	Multi-vehicles, Many-to-Many, Single depot, Dynamic and stochastic travel time	Minimize the total disutilities of the Service operator and customers	Hard time windows (pick up and drop off), Vehicle capacity, Vehicle service Time periods duration, Max. ride time	Heuristic, Insert Algorithm, Artificial neural network for travel time	N=3024 (real data)
Teodorovic and Radivojevic (2000)	Multi-vehicles, Many-to-Many, Single depot	Minimize the total vehicle traveling distances and total waiting time of the vehicles	Hard time windows (pick up and drop off), Vehicle capacity, Max. ride time	Heuristic, Two approximate reasoning algorithms, Fuzzy logic	N=900 (numerical example)
Fu (2002a)	Multi-vehicles, Many-to-Many, Dynamic and stochastic travel time	Minimize a combination of total service time and total disutilities caused to the customer	Soft Time windows pick up and drop off), Vehicle capacity, Vehicle service time, Periods duration, Max. ride time, Seating requirement	Heuristic, Insert Algorithm	N≤300 (per hour) Simulation data

Table 2.2 (continued)

Authors	Type	Objective	Time windows Other constraints	Method (Algorithm)	Size of Problem Solved
Uchimura et al. (2002)	Single vehicle Many-to-Many, Single depot	Minimize total distance of vehicle and onboard Distance of customers	No time windows, No capacity, Coupling and Precedence	Heuristic, Genetic algorithm	N=10 (real data)
Attanasio et al. (2004)	Multiple vehicles, Many-to-Many, Single depot	Minimize routing cost	Soft Time windows (pick up and drop off), Vehicle capacity, Max. route duration, Max. ride time	Heuristic, Parallel tabu search	N=200 and 295 (real data)
Coslovich et al. (2006)	Multiple vehicles, Many-to-Many, Single depot	Minimize the overall inconvenience of the advanced customers	Soft time windows (pick up and drop off), Vehicle capacity, Max. ride time	Heuristic, Insertion algorithm	$20 \leq N \leq 50$ (Random data)
Xiang et al. (2008)	Multi-vehicles, Many-to-Many, Single depot	Minimize a weighted sum of the fixed cost, mileage, driving time, waiting time, service time, overdriving time, overworking time, and delay time	Soft time windows (pick up and drop off), Vehicle capacity, Max. waiting time, Max. ride time	Heuristic, Insertion algorithm, a diversification strategy	Simulation data
Luo and Schonfeld (2011a)	Multiple vehicles, Many-to-Many, Single depot	Minimize the number of vehicles that satisfies all demands	Hard time windows (pick up and drop off), Vehicle capacity, Max. ride time	Heuristic, Parallel insertion (rejected-reinsertion)	Simulation data

2.3 Modern Heuristic Algorithms for Solving DARP

Modern heuristic methods were first designed in the early 1980s to solve complex and difficult combinatorial optimization problems that arise in many practical areas. After 1990s, most research for DARP has focused on development of modern heuristics or metaheuristics such as simulated annealing, tabu search, and GAs. A much smaller number of metaheuristic solution methods have been developed for the static and dynamic DARP. In this section, we briefly discuss simulated annealing, tabu search, and GAs.

2.3.1 Simulated Annealing

Simulated annealing is a versatile heuristic optimization technique based on the analogy between simulating physical annealing process of solids and solving large-scale combinatorial optimization problems. It can be summarized as follows. The algorithm starts off from an arbitrary initial configuration. In each iteration a new configuration is generated. The difference in objective value is compared with an acceptance criterion which accepts all improvements but also admits, in a limited way, deteriorations in cost. The mechanism for accepting worse feasible solutions is a mechanism against getting procedure stuck in local optima. Simulate annealing was used by Van Der Bruggen et al. (1993) and Baugh et al. (1998) to solve their dial-a-ride problems.

2.3.2 Tabu Search

Tabu search is a modern local search technique which examines the neighborhood of a current solution in order to make the next move to the best neighbor. To avoid cycling, solutions that were recently examined are forbidden or tabu for a number of iterations. There are lots of variations among tabu search techniques, because the number of parameters that define the technique can be tuned differently, and the best neighbor can have a completely different meaning, sometimes even allowing a move to an infeasible solution (Glover, 1997).

Tabu search stands out as a very powerful tool for the DARP since it is at the same time highly flexible and efficient. It is now clear that tabu search is capable of consistently generating high quality solutions on a large variety of routing problem. On the negative side the running time of tabu search algorithms can be rather high (Cordeau and Laporte, 2003, Cordeau et al., 2004). Baugh et al. (1998), Cordeau and Laporte (2003), Attanasio et al. (2004), and Melachrinoudis et al. (2007) used tabu search to solve DARP.

2.3.3 Genetic Algorithm (GA)

Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics. GAs are very useful in solving very difficult problems and have received considerable attention in combinatorial problems.

There are two distinguishing characteristics of GAs that separate them from the general optimization technique. The first is that GAs start with an initial set of random feasible solutions, not a single solution. GAs are population-to-population approach, can escape from local optima and are very effective in global search with a multi directional search while conventional optimization techniques based a point-to-point approach have the danger of falling in local optima (Gen and Cheng, 1997). The second characteristic is that GAs can handle any kind of objective function and constraints. It is difficult to apply simple GAs to complex optimization problems. However, with some modification, GAs can solve some particular problems such as Traveling Salesman Problem (TSP) and Time-Dependent Vehicle Routing Problem (TDVRP) (Jung, 2000). Uchimura et al. (2002), Rekiek et al. (2006) and Jorgensen et al. (2007) used GA for their dial-a ride model.

2.3.4 Fuzzy Logic

Fuzzy logic is widely used in intelligent control systems to make inferences about vague rules describing the relation between imprecise, qualitative linguistic estimations of the inputs and outputs of a system. These control rules usually represent the knowledge of an expert. A set of fuzzy rules, describing the control strategy of the operator, forms a fuzzy control algorithm, that is, approximate reasoning algorithm, whereas the linguistic expressions are represented and quantified by fuzzy sets. The main advantage of this approach is the possibility of introducing and using rules from experience, intuition, heuristics, and the fact that a model of the process is not required (Teodorovic and Vukadinovic, 1998). Fuzzy logic was used by Kikuchi and Donnelly (1992) and Teodorovic and Radivojevic (2000) to solve their dial-a-ride models.

2.4 Comparison of Heuristic Algorithms

It is difficult to compare the performance of heuristics. One possible way to compare heuristic performances is to check the problem size a heuristic can solve and the computation time it needs. But this comparison is unfair because we have to consider the rapid development of computer hardware and the diversity of computer systems available to different research groups.

Another difficulty in comparing heuristic performances is that most research was motivated by different real world problems. Thus, there are not many papers on the same problem. Problem type (single and multi-vehicles, single and multi-depot, heterogeneous and homogeneous fleet, soft and hard time windows, etc.), objective function, and constraints can be different. Also, size of service area, road network, and treatment of travel times can be different. This makes it hard to compare the heuristic methods.

2.5 Summary

In this chapter, previous literature related to the static and dynamic DARP were discussed. The single-vehicle DARP and multiple-vehicle DARP were presented and several algorithms and solution methods developed for both the static and dynamic DARP were described.

Exact solutions to the DARP have been limited to small problems and heuristic algorithms have been developed for large problem. Insertion heuristics are used widely because they are quite fast while metaheuristics need more running time than insertion heuristic. A much smaller number of metaheuristic solution methods have been developed for the static and dynamic DARP.

Most studies related to the DARP assume that the service is provided by a single depot that has a fleet of m vehicles to simplify the complicated real-world problem. But, in real world, there may be several depots, especially in widely urbanized areas like metropolitan cities, and two or more private companies serve whole areas together.

Very few research studies deal with real-time demand and time-dependent travel time simultaneously. Fu (1999a, 2002a) and Xiang et al. (2008) are the only authors who have attempted to deal with these issues in solving DARP. These are the most advanced research in dynamic dial-a-ride problem; however there is a potential for extending their models for application in real world as follows:

1) Our model has a comprehensive objective function combining service provider's cost and customers' cost, and complex constraints in order to explain and reflect real world operation more reasonably than other models. Also, the mathematical formulation of this model is proposed.

2) Most studies related to the DARP assume that the service is provided by a single depot that has a fleet of *m* vehicles to simplify the complicated real-world problem. But, Dial-a-Ride service in many instances is provided from multiple depots especially in widely urbanized areas like metropolitan cities in which dial-a-ride service is provided by two or more private companies.

3) Most of research proposed a maximum allowable deviation value for pick-up and drop-off time using hard time windows. There could possibly be a scenario where the time windows are hard, and the demands are such that there is no feasible solution where no time window is violated. But, we can get a larger feasible solution set by loosening time constraints using soft time windows. Thus, more feasible options are available for the algorithm when building the schedules and the system can be solved more efficiently.

4) In static DARP, the routing and scheduling are done considering time varying travel times in each link to reflect real situation and increase efficiency of service. Of course, travel times in each link are updated at every time interval in dynamic DARP.

5) Our model is capable of solving large real world problem and the results of model are compared with those of operation in real world.

As noted earlier, it is very difficult to solve dynamic DARP using exact algorithms to obtain an optimal solution within a reasonable computation time. Therefore, fast heuristic algorithms must be developed which can find a reasonable solution to this problem within a reasonable time, so that it can be used in a real-time situation. This dissertation research mainly focuses on developing such algorithms and solution approaches for real-time multi-depots, many-to-many DARP with time-dependent travel times, soft time windows, and heterogeneous vehicles. The proposed model has the following characteristics:

1) Our model can adjust the number of vehicles to minimize the total cost for serving the required demands. Thus, our model can explain and reflect real world operation more reasonably.

2) We can get more reasonable and flexible solution using soft time window which are allowed to be violated subject to penalties.

3) Our model can deal with the time-dependent travel times and demands for services that arise in real time simultaneously for multi-depot dial-a-ride problem.

4) The heuristic solution method can solve large problems with acceptable gaps in reasonable solution time.

Chapter 3: Problem Description and Formulation

3.1 Problem Description

This research is focused on real-time dial-a-ride problem with many-to-many, time-dependent travel times, multi-depots, and heterogeneous vehicles. This is a Dynamic Vehicle Routing Problem (DVRP) with time windows and its real world applications are pick-up and drop-off services for disabled persons or paratransit service. In this problem we consider accommodating two types of demands. Some demands are known in advance since the customers have the ability to make reservations for service on a particular day in advance. Other demands are not known a priori and arrive during the service period, for example 6 AM to 6 PM. We also consider time varying travel times because link travel speeds are not fixed during the service period and fluctuate. Each customer has a pick-up and drop-off time window. Like a real-world situations we consider more than one depot, from where vehicles of a fleet can start operating.

3.1.1 Definitions and Examples

We define the following terms to be used throughout this document.

Definition 1 *A demand* is a request of a customer. Each demand has both a pick-up and a drop-off location as well as time windows for pick-up and drop-off.

Definition 2 *A pick-up node* is the pick-up location of a demand.

Definition 3 *A drop-off node* is the drop-off location of a demand.

Definition 4 *A route* is a sequence of pick-up and drop-off nodes assigned to one vehicle.

Definition 5 *Excess ride time* is the time a customer spends in the vehicle, in addition to the time it takes to travel directly from his or her pick-up node to his or her drop-off node. That is, the total ride time minus the direct ride time is the excess ride time.

Definition 6 *Route duration* is the time it takes for the vehicle to leave the depot, service all customers on its route and return to the depot again.

Figure 3.1 through 3.4 present an example of real-time DARP considered in this research. Let's assume that there are two depots and three vehicles are available at each depot and there are 8 demands (16 nodes) at initial time T_0 as shown in Figure 3.1.

i^+ : the pick-up node of demand i , i^- : the drop-off node of demand i

Figure 3.1 Demand information at initial time T_0

Figure 3.2 shows an initial routing plan that uses four vehicles by considering travel time, vehicle capacity, and time windows of demand nodes based on the demands known in advance. According to this plan, the route of vehicle 1 is depot1$\rightarrow 2^+ \rightarrow 1^+ \rightarrow 2^- \rightarrow 1^- \rightarrow$ depot1, the route of vehicle 2 is depot1$\rightarrow 3^+ \rightarrow 4^+ \rightarrow 3^- \rightarrow 4^- \rightarrow$ depot1, the route of vehicle 4 is depot2$\rightarrow 6^+ \rightarrow 5^+ \rightarrow 5^- \rightarrow 6^- \rightarrow$ depot2, and the route of vehicle 5 is depot2$\rightarrow 7^+ \rightarrow 7^- \rightarrow 8^+ \rightarrow 8^- \rightarrow$ depot2.

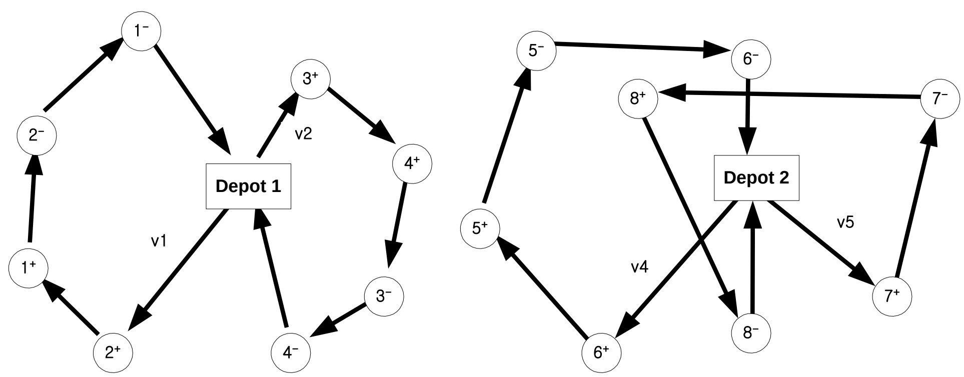


Figure 3.2 Initial routing plan at time T_0

Let us assume that while vehicle 1 is approaching 2^+, the pickup node of demand 2, vehicle 2 is approaching 4^+, the pickup node of demand 4 after visiting 3^+, the pickup node of demand 3, vehicle 4 is approaching 6^+, the pickup node of demand 6, and vehicle 5 is approaching 7^-, drop-off node of demand 7. At this time interval we receive information about newly arrived demands at nodes 9^+, 9^-, 10^+, 10^-, 11^+, 11^-, 12^+,12^-, and new travel times between all pairs of nodes based on current traffic condition. Figure 3.3 shows this. In this figure, shadowed circles represent new demands, and the dotted lines show the originally planned routes for each vehicle at initial time T_0. Based on the information about the new demands, new travel times, and the current locations of vehicles, we modify routes by inserting new demands to good locations of the already generated routes. The starting points of the vehicles en route are the nodes at which the vehicles are currently located or the ones into which they are headed.

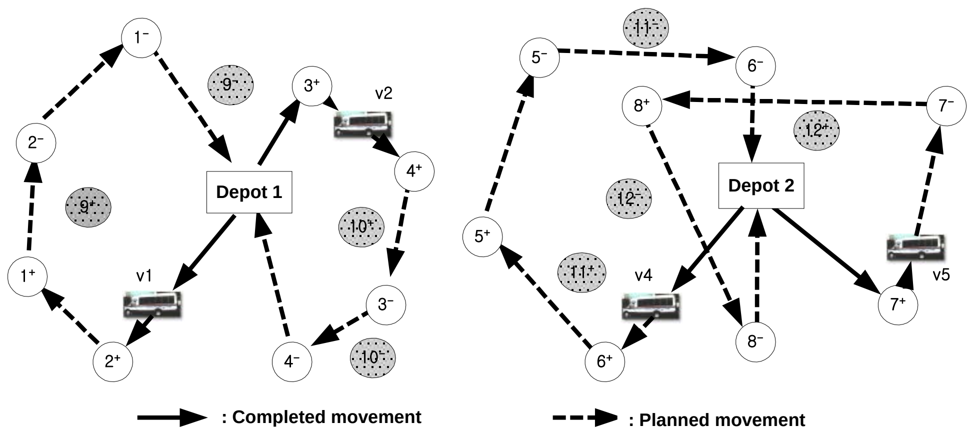


Figure 3.3 The newly arrived demands at time interval T_n

The results of the route adjustment are shown in Figure 3.4. The new routes for the vehicles are as follows:

Vehicle 1: The present location (v1) $\rightarrow 2^+ \rightarrow 1^+ \rightarrow 9^+ \rightarrow 2^- \rightarrow 1^- \rightarrow 9^- \rightarrow$ depot1

Vehicle 2: The present location (v2) $\rightarrow 4^+ \rightarrow 10^+ \rightarrow 3^- \rightarrow 10^- \rightarrow 4^- \rightarrow$ depot1

Vehicle 4: The present location (v4) $\rightarrow 6^+ \rightarrow 11^+ \rightarrow 5^+ \rightarrow 5^- \rightarrow 11^- \rightarrow 6^- \rightarrow$ depot2

Vehicle 5: The present location (v5) $\rightarrow 7^- \rightarrow 12^+ \rightarrow 8^+ \rightarrow 12^- \rightarrow 8^- \rightarrow$ depot2

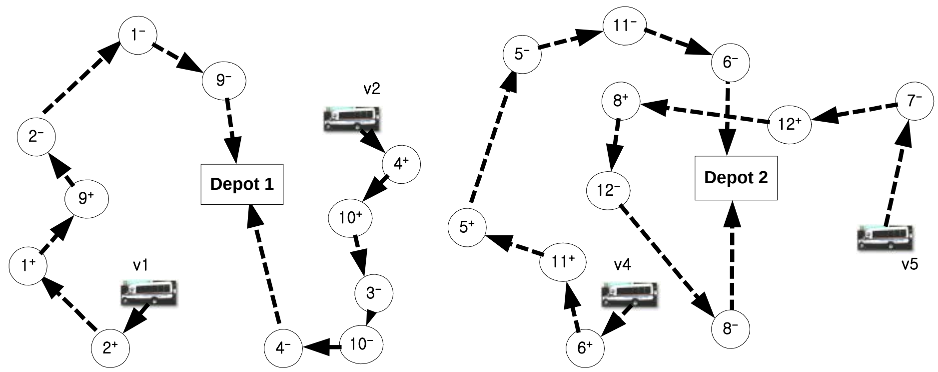


Figure 3.4 The new routing plan at time interval T_n

3.1.2 Assumptions and Limitations

The characteristics of the problem can be described in terms of the nature of demands, travel time, routing plan, number of depots, vehicle capacity, time windows, and real time information.

(1) Demands

Some demands are known in advance because of customers' reservations before the trip day. Other demands (real-time demands) are not known and arrive during the service period. Real time demands can arrive at any time between 6 A.M. and 6 P.M. After a demand is accepted, it can be canceled. But, a demand reservation cancellation must be made a minimum of 1 hour before the scheduled pick-up time for that demand. If a cancellation for some demand is received close to its pick-up time, the system will not be able to adjust the route to avoid the excess travel time to get to that demand. Thus, if we know of a cancellation in advance, we can save travel times. In case of the MTA (Maryland Transit Administration) paratransit service, a minimum of two hours advanced notice is required for cancellation.

Every demand has a demand request time, pick-up node, drop-off node, and load (ambulatory, wheelchair, and transferable wheelchair passengers). A customer is picked up from the pick-up node and transported to the drop-off node by the same vehicle. There are no customer priorities, and passengers cannot be transferred between vehicles.

(2) Travel Times

Travel times are subject to change according to the time of the day. Travel times from one location to another are not necessarily the same in both directions. We assume that in static situation, we have link flow speeds within each time interval (10minutes) which are based on historical data in network. In real time situation, link flow speeds on the network within each time interval is available through various surveillance mechanisms in real time. If there is no real-time data available, average travel speeds based on historical data can be used.

Given link flow speeds we can calculate the expected travel time between origin and destination at starting time using a time dependent shortest path algorithm. Calculating time dependent shortest path needs much more computation and memory than the general shortest path problem. For one-to-one time dependent shortest path algorithm, we extended one-to-all Dijkstra's algorithm with double buckets used to get the shortest paths in static networks by Cherkasssky et al. (1993) and Zhan

(1997). Also, for holding the FIFO property, flow speed model used by Sung et al. (2000) is adopted for this problem.

(3) Routing Plan

First the initial routing plan is developed based on the demands which are known in advance. This initial routing plan will then be modified periodically incorporating newly arrived demands and cancellations if any, and the new link travel time information. Based on the information about the new demands, new travel times, and the current locations of vehicles, we plan new routes by inserting new demands to good locations of the already generated routes. Any time a new routing plan is developed, two types of demands are being considered. The old demands are the demands that are already assigned to the vehicles in the previous route planning process, and the new demands are the demands requested after that. We allow reassignment of the demands that are not yet picked up to other vehicles when we are planning new routes. It is reasonable to expect that this will increase productivity and efficiency.

We assume that there is a maximum route duration, *u*, which cannot be exceeded by any vehicle. Also, the ride time for a customer who is picked up at pick-up node *i* cannot exceed his or her maximum ride time, r_i. The maximum ride time for customer *i*, r_i can be calculated by using a linear function of the direct ride time, $R_{i,n+i}$ between pick-up node, *i* and drop-off node, *n*+*i* as follows (Jaw et al., 1986).

$$r_i(\min) = \beta(\min) + 1.5 \times R_{i,n+i}(\min)$$

In our model, $R_{i,n+i}$ is replaced by $R^t_{i,n+i}$, that is, the direct ride time between pick-up node, i and drop-off node, $n+i$ at time t based on the time-dependent travel times.

(4) Depots

We assume that there are multiple-depots. The number of available vehicles at a depot and locations of depots are known. When a vehicle completes its service, it has to return to the depot to which the vehicle belongs. Relocating of vehicles between depots is not allowed. This is consistent with paratransit operations of Baltimore area, MD. Maryland Transit Authority (MTA) and two private transit companies operate paratransit services in this area and each company has its own depot.

(5) Vehicles

Each vehicle has its own capacity, and the vehicles are not homogeneous. Since most dial-a-ride services are provided to handicapped patients who are transported to and from hospitals and medical facilities, it is important to distinguish the patient types. Three different types of handicapped patients are considered in this research as follows: the ambulatories who use regular seats, those who use wheelchairs, and those who use transferable wheelchairs. Thus, we consider three types of vehicles. One is designed to carry only ambulatory passengers, another is designed to carry ambulatory passengers and passengers using wheelchairs, and the other is designed to carry ambulatory passengers and passengers using transferable wheelchairs. This is consistent with the industry norms. If there is no available vehicle to accept demands at certain time in service period, those demands are serviced by taxi.

We assume that the capacities of all vehicles are known. We minimize the total number of vehicles used for services because the fixed cost of using additional vehicles, in real world, is much higher than the routing costs. In this research, vehicles have the same fixed unit costs, regardless of the vehicle's type. This is not a restrictive assumption and can be relaxed very easily.

(6) Time windows

Each customer who uses the service has a time window for pick-up and a time window for drop-off. The time window at demand node *i* is denoted by $\left(a_i, b_i\right)$, where a_i is the earliest allowable arrival time and b_i is the latest allowable arrival time. In this research, we consider soft time windows and allow vehicles to arrive at the pick-up and drop-off nodes before or after the time interval that is designated for service. The early and late arrivals at a location are penalized so that the violation of time windows is kept to a minimum. The early arrival penalty incurs when a vehicle arrives before the earliest allowable arrival time, a_i, and the delay penalty incurs when a vehicle arrives after the latest allowable arrival time, b_i. This is more realistic than hard time windows and consistent with real world operations.

While it is possible that time windows can be negotiated between the customers and the scheduler, in practice, fixed time windows are normally adopted. Time windows are set based on the customer's desired pick-up or drop-off time as follows.

1) time window at pick-up node

$a_i =$ the desired pick-up time of customer i

$b_i =$ the desired pick-up time of cutomer $i + \alpha$ minutes

2) time window at drop-off node

$a_i =$ the desired drop-off time of cutomer i- α minutes

$b_i =$ the desired drop-off time of cutomer i

If the vehicle arrives at the pick-up or drop-off node before the earliest allowable arrival time, it has to wait for servicing customers at the node. But, it is not allowed to wait for servicing customers at the node if there is any customer on board.

(7) The real time information

We assume that there is a real-time communication system between the vehicles and the control center (or the main depot). The control center has information about the location of all vehicles, and status of old demands and newly arrived ones. The control center also receives real time traffic condition, link travel speed and incidents information in real time from a traffic management center. Given this information, at each time interval, dynamic shortest paths between nodes in the network can be calculated.

<u>3.2 Problem Formulation</u>

In this section we provide a mathematical formulation for static multi-depot DARP with time windows as a mixed integer programming problem. The objective of the formulation is to minimize the total cost that consists of the service provider's cost and the customers' inconvenience cost. The service provider's cost includes fixed costs of used vehicles, the routing costs, and vehicle waiting cost, while the customers' inconvenience cost includes customers' excess ride time cost and delayed service cost. This formulation is used to generate the initial routing plan based on the booked requests and the expected travel times between demands at initial time. It will be modified at fixed time intervals if there is any event (new demands or cancellations) arrived within previous time interval or to accommodate the real time traffic and any accident.

3.2.1 Notation and Variables

The data sets, constraints, and decision variables used in this model formulation are defined follows.

(1) Data Sets

$D(m,i,j)$: Demand Set, where m=identity number of demand, i=pick-up node, and j=drop-off node

I: the set of demand indentification numbers{set of m in $D(m,i,j)$}

P: the set of pick-up nodes {set of i in $D(m,i,j)$} $= \{1,2,3,...,n\}$

B: the set of drop-off nodes {set of j in $D(m,i,j)$} $= \{n+1, n+2, n+3, \dots, 2n\}$

ϕ: total pick-up and drop-off node set $(P \cup B)$

$\delta(k,l,i,j)$: Vehicle information set, where k=vehicle number, l=depot number, i=starting node, j=ending node

V : vehicles set {set of k in $\delta(k,l,i,j)$}

H : the set of depots {set of l in $\delta(k,l,i,j)$}

S : the set of starting nodes of all vehicles {set of i in $\delta(k,l,i,j)$}

E : the set of ending nodes of all vehicles {set of j in $\delta(k,l,i,j)$}

K : total vehicles

η : the union of the set of all demand nodes and the set of all starting nodes ($\phi \cup S$)

N : the set of all nodes ($\phi \cup S \cup E$)

N_s : the set of all nodes except the starting nodes ($N - S$)

D_s : the set of all demand nodes except the starting nodes ($\phi - S$)

(2) Constants

f_c : the fixed cost for a vehicle

P_e : the excess time penalty caused by excess ride time of customer

P_w : the waiting penalty caused by early arrival at each demand node

P_d : the delay penalty caused by late arrival at each demand node

R_c : the traveling cost per unit time

t : time counter

α : the starting time of the service period

ω : the end time of the service period

M : a large positive number

R_{ij}^{t} : time dependent shortest travel time from demand node i to j at starting time t

a_i : the earliest allowable arrival time at demand node i

b_i : the latest allowable arrival time at demand node i

s_i : dwell time needed at demand node i for boarding or alighting

$w_{\max}$: maximum acceptable waiting time at each demand node

$d_{\max}$: maximum acceptable delay time at each demand node

u : maximum route duration for all vehicles

r_i : maximum ride time for passenger with pick-up node i

$(q_i^a, q_i^{wc}, q_i^{wt})$: load of ambulatory passengers, wheelchair passengers, and transferable wheelchair passengers at demand node i (if demand node i is a pick-up service, then q_i is a positive value, and if demand node i is a drop-off service, than q_i is a negative value.)

$(C_k^a, C_k^{wc}, C_k^{wt})$: capacity of vehicle k

(3) Decision Variables

$$x_{ij}^{kt} = \begin{cases} 1, \text{ if vehicle } k \text{ departs from node } i \text{ to node } j \text{ at time } t \\ 0, \text{otherwise} \end{cases}$$

w_i : waiting time at demand node i (desired arrival time at demand node i - actual arrival time at demand node i)

d_i : delayed time at demand node i (actual arrival time at demand node i - desired arrival time at demand node i)

$(Q_{ik}^a, Q_{ik}^{wc}, Q_{ik}^{wt})$: acutal load of vehicle k when departing demand node i (ambulatory seats, wheelchair seats, transferable wheelchair seats)

$$y_{ik} = \begin{cases} 1, \text{ if vehicle } k \text{ arrives with passengers on board at node } i \\ 0, \text{otherwise} \end{cases}$$

3.2.2 Objective Function

The objective of this problem is to minimize the total cost composed of the service provider's cost and the customers' inconvenience cost.

First, we try to minimize the service provider's cost including the fixed costs for the used vehicles, the routing costs, and vehicle waiting cost. We minimize the number of used vehicles within the total available number of vehicles. Whenever vehicles arrive at the demand node earlier than the desired time, a penalty is incurred for waiting.

The total fixed costs are as follows:

Total fixed cost = (fixed cost/vehicle) × total number of vehicle

$$= f_c \times \sum_{k \in V} \sum_{i \in S} \sum_{j \in N_s} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} \tag{1}$$

The routing costs can be written as follows:

Routing cost = (traveling cost/min) × total traveling time (min)

$$= R_c \times \sum_{i \in \tau} \sum_{j \in N_s} \sum_{k \in V} \sum_{t=\alpha}^{\omega} \left(R_{ij}^{t} \times x_{ij}^{kt} \right) \tag{2}$$

Vehicle waiting cost

= (waiting penalty/min) $\times$ total vehicle waiting time (min)

$$= P_w \times \sum_{i \in \phi} w_i \tag{3}$$

Second, we minimize the user inconvenience cost including customers' excess ride time cost and delayed service cost. Customers' excess ride time is used as a proxy for bad customer service. Whenever the service is delayed, a penalty is incurred for delay.

Customers’ excess ride time cost

= (excess time penalty/min) $\times$ total customers excess ride time (min)

$$= p_e \times \sum_{i\in P}\left(\sum_{k\in V}\sum_{l\in N_s}\sum_{t=\alpha}^{\omega}(t-s_{n+i})x_{n+i,l}^{kt} - \sum_{k\in V}\sum_{j\in\phi}\sum_{t=\alpha}^{\omega} t x_{ij}^{kt} - \sum_{t=\alpha}^{\omega}\left(R_{i,n+i}^{t} \times \sum_{k\in V}\sum_{j\in\phi} x_{ij}^{kt}\right)\right) \qquad (4)$$

Delayed service cost

= (delay penalty/min) $\times$ total delayed time (min)

$$= P_d \times \sum_{i\in\phi} d_i \qquad (5)$$

Finally, the overall objective function is as follows:

$$\begin{aligned} Min \quad & f_c \times \sum_{k\in V}\sum_{i\in S}\sum_{j\in N_s}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} + R_c \times \sum_{i\in\tau}\sum_{j\in N_s}\sum_{k\in V}\sum_{t=\alpha}^{\omega}\left(R_{ij}^{t}\times x_{ij}^{kt}\right) + P_w \times \sum_{i\in\phi} w_i \\ & + p_e \times \sum_{i\in P}\left(\sum_{k\in V}\sum_{l\in N_s}\sum_{t=\alpha}^{\omega}(t-s_{n+i})x_{n+i,l}^{kt} - \sum_{k\in V}\sum_{j\in\phi}\sum_{t=\alpha}^{\omega} t x_{ij}^{kt} - \sum_{t=\alpha}^{\omega}\left(R_{i,n+i}^{t}\times\sum_{k\in V}\sum_{j\in\phi}x_{ij}^{kt}\right)\right) \\ & + P_d \times \sum_{i\in\phi} d_i \end{aligned} \qquad (6)$$

3.2.3 Constraints

The constraints in this model can be divided into five groups: Depot, capacity, precedence and coupling, routing, and time window constraints.

(1) Depot constraints

The depot constraints require that unused vehicles start and end in the depot to which they belong.

$$\sum_{i\in S}\sum_{j\in P}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} \le 1 \qquad k \in V \qquad (7)$$

$$\sum_{j\in E}\sum_{i\in B}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} \le 1 \qquad k \in V \tag{8}$$

Every vehicle has to return to the depot before the end of the service period.

$$\sum_{k\in V}\sum_{i\in B}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}\left(t+R_{ij}^{t}\right)\right) \le \omega \qquad j \in E \tag{9}$$

(2) Capacity constraints

Each vehicle has its own capacity ($C_k^a, C_k^{wc}, C_k^{wt}$) for ambulatory passengers, wheelchair passengers, and transferable wheelchair passengers. These capacities cannot be exceeded at any time.

$$Q_{ik}^{a} \le \sum_{j\in\phi}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} C_k^{a} \qquad k \in V, i \in \phi \tag{10}$$

$$Q_{ik}^{wc} \le \sum_{j\in\phi}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} C_k^{wc} \qquad k \in V, i \in \phi \tag{11}$$

$$Q_{ik}^{wt} \le \sum_{j\in\phi}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} C_k^{wt} \qquad k \in V, i \in \phi \tag{12}$$

If pick-up demand node j is visited after visiting demand node i, then the carried load by a vehicle at demand node j is the carried load by the vehicle at demand node i plus the load of the demand node j. When node j is a drop-off node, the value of Q_{jk}^{a} at demand node j is less than that of Q_{ik}^{q} because the q_j^a has a negative value.

$$Q_{ik}^{a} + q_j^{a} - Q_{jk}^{a} - M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \le 0 \qquad k \in V, i \in \eta, j \in N_s, i \ne j \tag{13}$$

$$Q_{ik}^{a} + q_j^{a} - Q_{jk}^{a} + M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \ge 0 \qquad k \in V, i \in \eta, j \in N_s, i \ne j \tag{14}$$

$$Q_{ik}^{wc} + q_j^{wc} - Q_{jk}^{wc} - M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \le 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j \tag{15}$$

$$Q_{ik}^{wc} + q_j^{wc} - Q_{jk}^{wc} + M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \ge 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j \tag{16}$$

$$Q_{ik}^{wt} + q_j^{wt} - Q_{jk}^{wt} - M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \le 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j \tag{17}$$

$$Q_{ik}^{wt} + q_j^{wt} - Q_{jk}^{wt} + M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \ge 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j \tag{18}$$

(3) Precedence and coupling constraints

The precedence and coupling constraints represent the requirement that each customer must first be picked up at node *i* and then dropped off at node *n+i* by the same vehicle *k*. Each demand node is visited exactly once during a day. These constraints are represented by three equations.

$$\sum_{k \in V} \sum_{j \in \phi} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} = 1 \qquad i \in P \tag{19}$$

$$\sum_{j \in \phi} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} - \sum_{l \in \phi} \sum_{t=\alpha}^{\omega} x_{l,n+i}^{kt} = 0 \qquad k \in V, i \in P \tag{20}$$

$$\sum_{l \in \phi} \sum_{t=\alpha}^{\omega} \left(x_{l,n+i}^{kt} (t + R_{l,n+i}^{t}) \right) - \sum_{j \in \phi} \sum_{t=\alpha}^{\omega} \left(x_{ij}^{kt} (t + R_{ij}^{t}) \right) \ge 0 \quad k \in V,\ i \in P \tag{21}$$

(4) Routing constraints

When a vehicle arrives at a node which is not a depot, it has to travel to either another demand node or a depot (route continuity).

$$\sum_{\substack{l \in N_s \\ j \ne l}} \sum_{t=\alpha}^{\omega} x_{jl}^{kt} - \sum_{\substack{i \in \eta \\ i \ne j}} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} = 0 \qquad k \in V,\ j \in D_s \tag{22}$$

If demand node j is visited after visiting demand node i, then the arrival time at demand node j must be equal to the sum of the departure time at demand node i and the travel time, R_{ij}, from demand node i to demand node j.

$$\sum_{\substack{l\in N_s \\ j\neq l}}\sum_{t=\alpha}^{\omega} tx_{jl}^{kt} - s_j - w_j - \sum_{\substack{i\in\eta \\ i\neq j}}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}(t+R_{ij}^{t})\right) + M\left(1-\sum_{\substack{i\in\eta \\ i\neq j}}\sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \geq 0 \quad k\in V,\ j\in\phi \tag{23}$$

$$\sum_{\substack{l\in N_s \\ j\neq l}}\sum_{t=\alpha}^{\omega} tx_{jl}^{kt} - s_j - w_j - \sum_{\substack{i\in\eta \\ i\neq j}}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}(t+R_{ij}^{t})\right) - M\left(1-\sum_{\substack{i\in\eta \\ i\neq j}}\sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \leq 0 \quad k\in V,\ j\in\phi \tag{24}$$

The route duration for each vehicle cannot exceed the maximum route duration.

$$\sum_{l\in B}\sum_{m\in E}\sum_{t=\alpha}^{\omega}\left(x_{lm}^{kt}\left(t+R_{lm}^{t}\right)\right) - \sum_{i\in S}\sum_{j\in P}\sum_{t=\alpha}^{\omega}\left(tx_{ij}^{kt}\right) \leq u \qquad k\in V \tag{25}$$

The ride time for a customer who is picked up at node i cannot exceed the maximum ride time, r_i.

$$\sum_{k\in V}\sum_{l\in N_s}\sum_{t=\alpha}^{\omega} tx_{n+i,l}^{kt} - \sum_{k\in V}\sum_{j\in\phi}\sum_{t=\alpha}^{\omega} tx_{ij}^{kt} \leq r_i \qquad i\in P \tag{26}$$

(5) Time window constraints

The waiting time at a demand node j is the gap between the earliest arrival time and the actual arrival time at the demand node j.

$$w_j = Max\left(0, a_j - \sum_{k\in V}\sum_{i\in\eta}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}(t+R_{ij}^{t})\right)\right) \qquad j\in\phi \tag{27}$$

The waiting time cannot exceed the maximum acceptable waiting time, $w_{\max}$.

$$0 \leq w_j \leq w_{\max} \qquad j\in\phi \tag{28}$$

The delay time at a demand node *j* is the gap between the latest arrival time and the actual arrival time at the demand node *j*.

$$d_j = Max\left(0, \sum_{k\in V}\sum_{i\in \eta}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}(t+R_{ij}^{t})\right)-b_j\right) \qquad j\in\phi \tag{29}$$

The delay time cannot exceed the maximum acceptable delay time, $d_{\max}$.

$$0 \le d_j \le d_{\max} \qquad j\in\phi \tag{30}$$

If there is any customer on board, it is not allowed to wait for servicing customers at the node.

$$M\left(1-y_{ik}\right) \ge w_i \qquad k\in V,\ i\in\phi \tag{31}$$

$$My_{ik} \ge Q_{ik}^{a} - q_i^{a} + Q_{ik}^{wc} - q_i^{wc} + Q_{ik}^{wt} - q_i^{wt} \qquad k\in V,\ i\in P \tag{32}$$

$$M(y_{ik}-1) \le Q_{ik}^{a} - q_i^{a} + Q_{ik}^{wc} - q_i^{wc} + Q_{ik}^{wt} - q_i^{wt} - 1 \qquad k\in V,\ i\in P \tag{33}$$

$$My_{ik} \ge Q_{ik}^{a} + Q_{ik}^{wc} + Q_{ik}^{wt} \qquad k\in V,\ i\in B \tag{34}$$

$$M(y_{ik}-1) \le Q_{ik}^{a} + Q_{ik}^{wc} + Q_{ik}^{wt} - 1 \qquad k\in V,\ i\in B \tag{35}$$

3.3 Summary

We proposed a mixed integer programming formulation for the static multi-depot DARP considering time varying travel times, soft time windows and heterogeneous vehicles. The objective of the formulation is to minimize the total cost that consists of the service provider’s cost and the customers’ inconvenience cost. The service provider’s cost includes fixed costs of used vehicles, the routing costs, and vehicle waiting cost, while the customers’ inconvenience cost includes customers’ excess ride time cost and delayed service cost. The constraints in this model consist of five groups: Depot, capacity, precedence and coupling, routing, and time window constraints.

The overall formulation is summarized as follows.

$$
\begin{aligned}
Min \quad & f_c \times \sum_{k\in V}\sum_{i\in S}\sum_{j\in N_s}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} + R_c \times \sum_{i\in\tau}\sum_{j\in N_s}\sum_{k\in V}\sum_{t=\alpha}^{\omega}\left(R_{ij}^{t} \times x_{ij}^{kt}\right) + P_w \times \sum_{i\in\phi} w_i \\
& + p_e \times \sum_{i\in P}\left(\sum_{k\in V}\sum_{l\in N_s}\sum_{t=\alpha}^{\omega}(t-s_i)x_{n+i,l}^{kt} - \sum_{k\in V}\sum_{j\in\phi}\sum_{t=\alpha}^{\omega} t x_{ij}^{kt} - \sum_{t=\alpha}^{\omega}\left(R_{i,n+i}^{t} \times \sum_{k\in V}\sum_{j\in\phi} x_{ij}^{kt}\right)\right) \\
& + P_d \times \sum_{i\in\phi} d_i
\end{aligned}
$$

Subject to

$$\sum_{i\in S}\sum_{j\in P}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} \le 1 \qquad k \in V$$

$$\sum_{j\in E}\sum_{i\in B}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} \le 1 \qquad k \in V$$

$$\sum_{k\in V}\sum_{i\in B}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}\left(t + R_{ij}^{t}\right)\right) \le \omega \qquad j \in E$$

$$Q_{ik}^{a} \le \sum_{j\in\phi}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} C_k^{a} \qquad k \in V, i \in \phi$$

$$Q_{ik}^{wc} \le \sum_{j\in\phi}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} C_k^{wc} \qquad k \in V, i \in \phi$$

$$Q_{ik}^{wt} \le \sum_{j\in\phi}\sum_{t=\alpha}^{\omega} x_{ij}^{kt} C_k^{wt} \qquad k \in V, i \in \phi$$

$$Q_{ik}^{a} + q_j^{a} - Q_{jk}^{a} - M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \le 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j$$

$$Q_{ik}^{a} + q_j^{a} - Q_{jk}^{a} + M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \ge 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j$$

$$Q_{ik}^{wc} + q_j^{wc} - Q_{jk}^{wc} - M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \le 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j$$

$$Q_{ik}^{wc} + q_j^{wc} - Q_{jk}^{wc} + M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \ge 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j$$

$$Q_{ik}^{wt} + q_j^{wt} - Q_{jk}^{wt} - M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \le 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j$$

$$Q_{ik}^{wt} + q_j^{wt} - Q_{jk}^{wt} + M \times \left(1 - \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\right) \ge 0 \quad k \in V, i \in \eta, j \in N_s, i \ne j$$

$$\sum_{k \in V} \sum_{j \in \phi} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} = 1 \qquad i \in P$$

$$\sum_{j \in \phi} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} - \sum_{l \in \phi} \sum_{t=\alpha}^{\omega} x_{l,i+1}^{kt} = 0 \qquad k \in V, i \in P$$

$$\sum_{l \in \phi} \sum_{t=\alpha}^{\omega} \left(x_{l,n+i}^{kt} (t + R_{l,n+i}^{t}) \right) - \sum_{j \in \phi} \sum_{t=\alpha}^{\omega} \left(x_{ij}^{kt} (t + R_{ij}^{t}) \right) \ge 0 \quad k \in V,\ i \in P$$

$$\sum_{\substack{l \in N_s \\ j \ne l}} \sum_{t=\alpha}^{\omega} x_{jl}^{kt} - \sum_{\substack{i \in \eta \\ i \ne j}} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} = 0 \qquad k \in V,\ j \in D_s$$

$$\sum_{\substack{l \in N_s \\ j \ne l}} \sum_{t=\alpha}^{\omega} t x_{jl}^{kt} - s_j - w_j - \sum_{\substack{i \in \eta \\ i \ne j}} \sum_{t=\alpha}^{\omega} \left(x_{ij}^{kt} (t + R_{ij}^{t}) \right) + M \left(1 - \sum_{\substack{i \in \eta \\ i \ne j}} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} \right) \ge 0 \quad k \in V,\ j \in \phi$$

$$\sum_{\substack{l \in N_s \\ j \ne l}} \sum_{t=\alpha}^{\omega} t x_{jl}^{kt} - s_j - w_j - \sum_{\substack{i \in \eta \\ i \ne j}} \sum_{t=\alpha}^{\omega} \left(x_{ij}^{kt} (t + R_{ij}^{t}) \right) - M \left(1 - \sum_{\substack{i \in \eta \\ i \ne j}} \sum_{t=\alpha}^{\omega} x_{ij}^{kt} \right) \le 0 \quad k \in V,\ j \in \phi$$

$$\sum_{l \in B} \sum_{m \in E} \sum_{t=\alpha}^{\omega} \left(x_{lm}^{kt} \left(t + R_{lm}^{t} \right) \right) - \sum_{i \in S} \sum_{j \in P} \sum_{t=\alpha}^{\omega} \left(t x_{ij}^{kt} \right) \le u \qquad k \in V$$

$$\sum_{k \in V} \sum_{l \in N_s} \sum_{t=\alpha}^{\omega} t x_{n+i,l}^{kt} - \sum_{k \in V} \sum_{j \in \phi} \sum_{t=\alpha}^{\omega} t x_{ij}^{kt} \le r_i \qquad i \in P$$

$$w_j = Max\left(0, a_j - \sum_{k\in V}\sum_{i\in\eta}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}(t+R_{ij}^{t})\right)\right) \qquad j\in\phi$$

$$0 \le w_j \le w_{\max} \qquad j\in\phi$$

$$d_j = Max\left(0, \sum_{k\in V}\sum_{i\in\eta}\sum_{t=\alpha}^{\omega}\left(x_{ij}^{kt}(t+R_{ij}^{t})\right) - b_j\right) \qquad j\in\phi$$

$$0 \le d_j \le d_{\max} \qquad j\in\phi$$

$$M\left(1-y_{ik}\right) \ge w_i \qquad k\in K,\ i\in\phi$$

$$My_{ik} \ge Q_{ik}^{a} - q_i^{a} + Q_{ik}^{wc} - q_i^{wc} + Q_{ik}^{wt} - q_i^{wt} \qquad k\in V,\ i\in P$$

$$M(y_{ik}-1) \le Q_{ik}^{a} - q_i^{a} + Q_{ik}^{wc} - q_i^{wc} + Q_{ik}^{wt} - q_i^{wt} - 1 \qquad k\in V,\ i\in P$$

$$My_{ik} \ge Q_{ik}^{a} + Q_{ik}^{wc} + Q_{ik}^{wt} \qquad k\in V,\ i\in B$$

$$M(y_{ik}-1) \le Q_{ik}^{a} + Q_{ik}^{wc} + Q_{ik}^{wt} - 1 \qquad k\in V,\ i\in B$$

Chapter 4: Heuristic Algorithms for Static DARP and Computational Results

In this paper, the approach for solving this model can be divided into two phases including a construction phase and an improvement phase. In the first phase, feasible routes are constructed and in the second phase the routes are improved.

4.1 Heuristic Algorithms

The three different heuristic algorithms including a heuristic based on sequential insertion (HSI), a heuristic based on parallel insertion (HPI), and a heuristic based on clustering first-routing second (HCR) are proposed. These are different according to the way to construct feasible routes.

The framework of this heuristic is depicted in Figure 4.1.

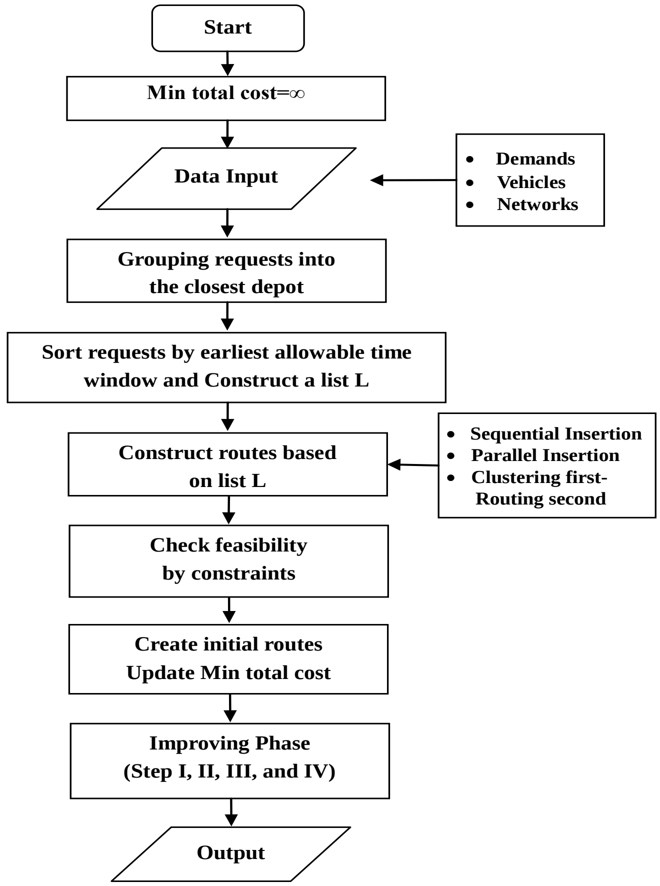


Figure 4.1 The framework of the heuristic for static DARP

4.1.1 Construction

Before construction phase, all demands are grouped into the closest depot and sorted by earliest allowable time window.

In construction phase, we build a set of feasible routes for each depot starting from the information that define dial-a-ride problem by three different heuristic methods such as sequential insertion, parallel insertion, and clustering first-routing second.

The starting time of a route is determined as follows:

$$et_{1st} - T(depot, 1st)$$

where,

$et_i: earliest\ allowable\ time\ window\ of\ demand\ i,$

$1st: first\ pickup\ node\ of\ a\ route,$

$T(depot, 1st): travle\ time\ for\ depot\ to\ 1st$

This time should not be less than start time of service. It is clear that the earlier to start, the better to avoid violating the time window constraint (Xiang et al. 2006).

Also, feasible routes are made by checking capacity, time window, and route duration constraints at construction phase. When assigning a vehicle to the route, the method of randomly assigning vehicles is used. First, vehicles at depots are listed by the configuration (only ambulatory, ambulatory and wheelchair, or ambulatory and transferable wheelchair). After checking the closeness of first demand location of the route to depots and the type (ambulatory, wheel chair, or transferrable wheelchair) of first demand of the route, the vehicle is randomly chosen from the list and assigned to the route.

(1) Sequential Insertion

The procedure is similar to the one proposed in Jaw et al. (1986). The algorithm starts by sorting demands in increasing order of their pickup times and inserts one demand at a time into one vehicle's schedule. The used algorithm in this research is described as follows:

1. Sort demands by earliest time window of them and create list *L*.
2. Construct conflict table *C*. If demands *i* and *j* can't be serviced in one trip in the worst case, they are marked as conflicting.

3. Cluster demands according to the list L and the conflict table *C*. Unvisited requests are clustered into different groups.
4. Construct a feasible route by sequentially extracting as many demands as possible from one route.
5. After extracting one route, the remaining requests in the route are regrouped into a new route.
6. Feasible routes are continuously made until there is no demand left.

(2) Parallel Insertion

The procedure is similar to the one proposed by Toth and Vigo (1997). First a small set of empty routes is initialized, and then iteratively unscheduled demands are inserted into the existing route which has cheapest insertion cost for those demands. The algorithm is described as follows:

1. Sort demands by their earliest time window and create list *L*.
2. Construct conflict table *C*. If demand i and j can't be serviced in one trip in the worst case, they are marked as conflicting.
3. Initialize a set of empty routes, *M*.
4. Then each demand is processed in the list in sequence as follows and assigned to a vehicle until the list of demands is exhausted.

   For each demand i ($i = 1, 2, \ldots, N$),

   4.1: For each route j ($j = 1, 2, \ldots, M$)

a) Find all the feasible insertion sequences in which demand i can be inserted into the route *j*. If it is infeasible to assign demand *i* to route *j*, examine the next route *j +1*, and restart Step 4.1; Otherwise

b) Find the insertion of demand *i* into the route j that results in minimum additional cost. Call this additional cost C_j.

4.2: If it is infeasible to insert demand *i* into any route j, then make a new route, *M+1* and insert demand *i* into that route. Otherwise, assign *i* to the route *j** which has a minimum additional cost for all *j* (*j* = 1,2, …, *M*).

Additional cost is calculated as follows:

$$additional\_cost = AfterInsertionValue(j) - BeforeInsertionValue(j)$$

where,

$AfterInsertionValue(j)$: $the\ cost\ of\ route\ j\ after\ insertion\ of\ demand\ i$

$BeforeInsertionValue(j)$: $the\ cost\ of\ route\ j\ before\ \ insertion\ of\ demand\ i$

(3) Clustering first-Routing second

By this approach, first clusters are made and a vehicle is assigned to each cluster. Then, customers in the group are assigned to each cluster. The method to make clusters for customers is described as follows:

1. Calculate urgency index value for each customer *i*

$$U_i = \alpha \times \frac{d1_i}{max\ d1} - \beta \times \frac{ET_i}{max\ ET} + \gamma \times \frac{d2_i}{max\ d2}$$

where,

$\mathrm{d1_i}$: Euclidean distance between pickup and dropoff node of a customer i

$\mathrm{d2_i}$: Euclidean distance between the depot and pickup nodee of a customer i

$\mathrm{ET_i}$: Earliest allowable arrival time of customer i

2. Sort customers by index value and make a list.
3. Choose seed (starting customer) from the list for each cluster. The customers are chosen orderly from the list so as to create a set of clusters equal to the minimum number of routes that have been set.
4. Assign vehicles to seeds. Vehicles are randomly chosen but they should accept the type of the assigned customer.

After making clusters, the method to add unassigned customers to clusters is as follows:

1. Add other customers in the group to clusters one by one according to geographical closeness until every customer belongs to one of the clusters
2. For unassigned customer i,

   2.1 Calculate Euclidean distance between unassigned customer i and last added customer *j* in each cluster.

$$d_{ij} = distacne(pickup\ node\ of\ i\ and\ pickup\ node\ of\ j) + distance(drop\ node\ of\ i\ and\ drop\ node\ of\ j)$$

   2.2 Compare the distances of clusters

   2.3 Finally add customer *i* to the cluster which has the shortest distance.

After adding all customers to the clusters, check feasibility of routes by constraints. If a customer violates any constraints, and then the violating customer is moved to next route. If there is no route to accept the violating customer, then a new route is made.

4.1.2 Improvements

After obtaining a feasible solution in the construction phase, the solution is improved through 4 steps of improvements.

(1) Step I: Acceptable waiting and delay time

While the violation of time window is allowed in this model, we can adjust the service quality by the maximum acceptable waiting and delay time in this phase. After constructing phase, in initial solution, there may be bad routes in which some demands have unreasonable waiting or delay time. In this phase, all routes are improved to satisfy maximum acceptable waiting and delay time. It is allowed for a demand on a route to be moved into another route that starts from a different depot.

The method is described as follows:

1. After checking waiting and delay time at each demand, choose demands that have larger values than maximum acceptable waiting and delay time that is set in advance.
2. Remove those demands from the original route and insert them into another route that does not conflict with them.
3. If there is no route that can accept them, make a new route.

(2) Step II: Remove one insert one

For the local improvement procedure to explore the local region, several trip operators such as remove two insert one, remove one insert one, and exchange can be applied. Among these operators, remove one insert one is frequently used in the local search. Also, in this research, remove one insert one is applied at this phase. This trip insertion removes a given trip from a route and then inserts it into the best position of another route. For example, using this method, customer 3(demand node 5 and 6) is inserted in route 1 which produces a smaller additional insertion cost, as shown in Figure 4.2.

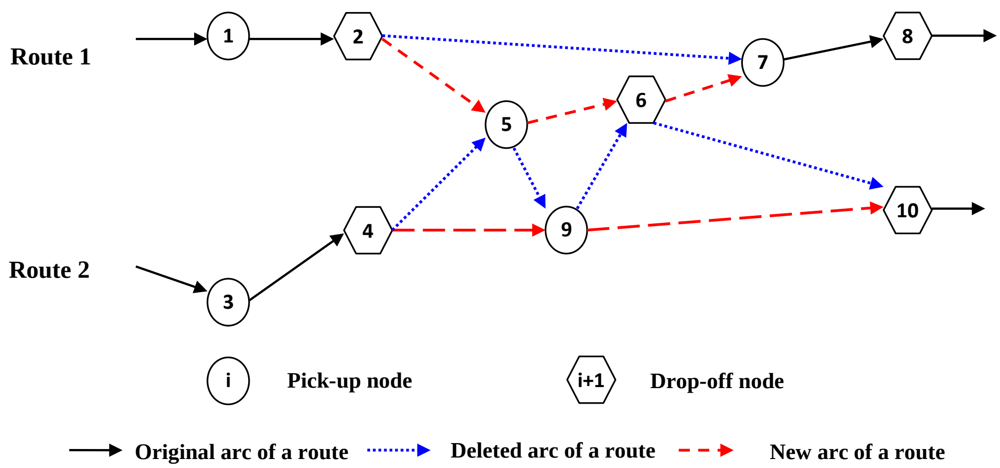


Figure 4.2 Illustration of the remove one insertion one method

The method is described in detail as follows:

1. At a route (original route), choose one pair (pickup and drop off nodes), search all possible routes (target routes) that can accept it, and insert it into each target route. Find best position in each target route.

2. Calculate the Saving Cost (SC) for the original and each target route. Among all target routes, choose the target route that has the best saving cost (it should be positive), then update the original and target route and minimum total cost.

$$SC(i, j) = f(i) + f(j) - f(i') - f(j')$$

where,

$f(i)$: the cost of a route i,
i : the original route before moving,
j : the target route before moving,
i' : the original route after moving,
j' : the target route after moving

3. If inserting one pair into other routes fails or best saving cost is negative, the pair is inserted back into its original route.
4. Repeat steps 1-3 for other pairs in the original route.
5. Repeat steps 1-4 for other routes.

Through this step, most of all routes having much waiting and delay times can be improved and waiting and delay cost in the objective function can be reduced. It is allowed for a demand on a route to be moved into another route that starts from a different depot.

(3) Step III: Combining vehicles

After improving step II, there may be some routes which have a few demands. These routes can be combined into other routes that can accept customers from them without violating constraints. Combining vehicles is necessary to reduce the objective function since the fixed cost accounts for a significant portion of the total cost. It is allowed for a vehicle to be combined into another vehicle that starts from a different depot.

The procedure of improving step III is described as follows:

1. Choose a route that has fewer demands than $min\_demands$ guaranteeing that each route has at least some demands.
2. Find other routes which the chosen route can be combined into and calculate saving cost for each case.
3. Combine the chosen route into the route which has maximum saving cost.

(4) Step IV: Adjusting vehicle starting time

The starting time of a route is determined in a simple way as to set the departure time from the depot as the earliest arrival time at the first pickup node minus the travel time between the depot and that node. Still, there is a possibility to reduce the waiting time and the total route duration by adjusting vehicle starting time at the depot. The method proposed by Xing et al. (2006) is used at this step.

At this step, we check the waiting time at a demand node in each route. If there is any waiting time at a demand node in a route, the starting time of the vehicle serving that route is adjusted using marginal time as follows:

For each route i (i = 1, 2, …, M),

1. Check the waiting time at each demand node in route i.
2. If there is any waiting time at a demand node in route i, then, calculate marginal time at each node in route i. The marginal time at a node is defined as the

maximum delay in arrival at the current node that does not cause violation of the time windows at the following nodes. Otherwise, go to next route $i+1$ and repeat step 1.

3. Adjust vehicle starting time as earliest arrival time at first pickup node minus the travel time between the depot and that node plus the marginal time at first node in route i.

4.1.3 Time-dependent shortest path algorithm

As it is mentioned earlier, travel times are subject to change according to the time of the day. We assume that in static situation, we have link flow speeds within each time interval (10minutes) which is based on historical data in network. In real time situation, link flow speeds on the network within each time interval is available through various surveillance mechanisms in real time. If there is no real-time data available, average travel speeds based on historical data can be used.

Given link flow speeds we can calculate the expected travel time between origin and destination at starting time using a time dependent shortest path algorithm. Calculating time dependent shortest path needs much more computation and memory than the general shortest path problem. For one-to-one time dependent shortest path algorithm, we extended one-to-all Dijkstra's algorithm with double buckets used to get the shortest paths in static networks by Cherkasssky et al. (1993) and Zhan (1997). Also, for holding the FIFO property, flow speed model used by Sung et al. (2000) is adopted for this problem.

The time dependent shortest path algorithm used for this problem is as follows:

```
algorithm time dependent shortest path algorithm
begin
    /* initialization */
    d(j) := ∞ for each node j ∈ N;
    d(s) := 0;
    pred(s) := 0;
    INIT_BHEAP(source);

    /* main loop */
    while(1) do
    begin

        EXTRACT_MIN(i);
        if(i = NULL) break;

        for each (i, j) ∈ A(i) do
        begin
            value = ArrivalTime(d(i), (i, j));
            if(value < d(j)) then
                TIME_TO_POS(value, pos_new);
                if (NODE_IN_BHEAP(j)) then
                    if(ins = (pos_old != pos_new)) then
                        REMOVE_FROM_BHEAP(j, pos_old);
                else ins:=1;
                if (ins)then INSERT_TO_BHEAP(j, pos_new);
                d(j) := value
                pred(j) := i;
                if(j = ending) then break;
        end
    end
end
```

INIT_BHEAP(source): Create an empty double heap
EXTRACT_MIN(node): Find and return a minimum value of node
REMOVE_FROM_BHEAP(node, pos): Delete a value of node on pos^{th} label in heap
INSERT_TO_BHEAP(node, pos): Insert a new value of node on pos^{th} label in heap
TIME_TO_POS(travel_time, pos): Find the location in heap for new travel time

In this time dependent shortest path algorithm, arrival times of node are calculated by arrival time function as follows:

Function ArrivalTime(d(i), (i, j))
$temp: = d(i) * 1.0/Scale\_Factor$;
$IntervalNumber: = (temp - FirstIntervalStart) / IntervalLength$;
$Res\_length: = lengh\ of\ (I, j)$;
$temp\_speed := Link\_Speed\_Function(IntervalNumber, (i, j))$;
$Res\_length := Res\_length - \frac{temp\ \_speed}{60} * ((IntervalNumber + 1) * IntervalLength - temp)$;
while$(Res\_length > 0)$ **do**
begin
$IntervalNumber := IntervalNumber + 1$;
$temp\_speed := Link\_Speed\_Function(IntervalNumber, (i, j))$;
$Res\_length := Res\_length - \frac{temp\ \_speed}{60} * \left((IntervalNumber + 1) * IntervalLength - IntervalNumber * IntervalLength\right)$;
end
$arrival\_time := (IntervalNumber + 1) * IntervalLength + \frac{Res\_length}{temp\_speed} * 60$;
Retrun *arrival_time;*

4.2 Lower Bound

In this section the approach to find a lower bound is presented. The original formulation is reformulated with new variables and constraint using LP relaxation. Then, the new mixed integer programming problem is solved. For finding the lower bound the method used by Jung (2000) for pickup and delivery problem is modified for DARP.

When the original problem without integer relaxation is solved, the largest problems that can be solved within a reasonable computational time are problems with 5 demands and 10 time intervals. In the lower bound solution procedure, we try to solve larger problems, although the results are not the exact solutions.

4.2.1 Procedure

The strategy of the lower bound solution procedure is to find a way that minimizes the number of integer variables. The simplest way to minimize the number of integer variable is LP relaxation. It is necessary that new variables and constraints are added to relaxed formulation in order to provide a good lower bound since the objective function of relaxed formulation is too low compared to the known optimal value for very small problems when the original formulation is relaxed without any changes and the problem is solved.

In the original formulation, there are two kinds of binary variables. These are x_{ij}^{kt} and y_{ik}. For relaxation, x_{ij}^{kt} and y_{ik} are changed to general integer variables. And, new variables S_{ij}^{kt} , Z_i^t , and V_i^k are added as follows:

S_{ij}^{kt} = 1 if vehicle *k* starts from depot *i* to demand *j* at time *t*,

= 0 otherwise

Z_i^t = 1 if vehicle *k* departs from node *i* at time *t*,

= 0 otherwise

V_i^k = 1 if demand *i* is serviced by vehicle *k*,

= 0 otherwise

New constraints (36) and (37) are added. Constraints (36) states that new variable S_{ij}^{kt} is equivalent to x_{ij}^{kt} when *i* belongs to the set of starting depot.

$$S_{ij}^{kt} = x_{ij}^{kt} \qquad i \in S, j \in \Phi \qquad (36)$$

$$\sum_{i\epsilon S} \sum_{j\epsilon \Phi} \sum_{t\epsilon \alpha}^{\omega} \sum_{k\in V} S_{ij}^{kt} \geq 1 \qquad (37)$$

Constraint (7) in the original formulation is replaced as expression (38).

$$\sum_{i\epsilon S} \sum_{j\epsilon P} \sum_{t=\alpha}^{w} x_{ij}^{kt} \leq 1 \qquad k \in V \qquad (7)$$

$$\sum_{i\epsilon S} \sum_{j\epsilon P} \sum_{t=\alpha}^{w} S_{ij}^{kt} \leq 1 \qquad k \in V \qquad (38)$$

Also new constraint (39), (40), and (41) are added.

$$V_i^k \;=\; \sum_{\mathrm{j}\in \mathrm{N_s}} \sum_{\mathrm{t}=\alpha}^{\mathrm{w}} x_{ij}^{kt} \qquad i \in \Phi, k \in V \tag{39}$$

Constraint (40) states that the new variable Z_i^t is the sum of all connection from demand node *i* to any demand node at time *t*.

$$Z_i^t = \sum_{k\in V} \sum_{j\in N_s} x_{ij}^{kt} \qquad i \in \phi,\ t \in T \tag{40}$$

$$\sum_{t=\alpha}^{\omega} Z_i^t = 1 \qquad i \in \phi \tag{41}$$

Also constraint (27) for waiting penalty and constraint (29) for delay penalty in the original formulation can be rewritten as expression (42) and (43) using Z_i^t.

$$w_j = Max\left(0, a_j - \sum_{k\in V} \sum_{i\in\eta} \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\ (t + R_{ij}^t)\right) \qquad \mathrm{j} \in \phi \tag{27}$$

$$d_j = Max\left(0, \sum_{k\in V} \sum_{i\in\eta} \sum_{t=\alpha}^{\omega} x_{ij}^{kt}\ \left(t + R_{ij}^t\right) - b_j\right) \qquad \mathrm{j} \in \phi \tag{29}$$

$$w_i = Max(0, a_i - \sum_{t=\alpha}^{\omega} tZ_i^t - s_i) \qquad \mathrm{j} \in \phi \tag{42}$$

$$d_i = Max(0, \sum_{t=\alpha}^{\omega} tZ_i^t - s_i - b_i) \qquad \mathrm{j} \in \phi \tag{43}$$

The fixed cost part of the objective function in the original formulation is modified using S_{ij}^{kt}. The underlined part in expression (44) shows the modified part as follows:

$$\begin{aligned} Min \quad & f_c \times \underline{\sum_{k\in V}\sum_{i\in S}\sum_{j\in P}\sum_{t=\alpha}^{\omega} S_{ij}^{kt}} + R_c \times \sum_{i\in\tau}\sum_{j\in N_s}\sum_{k\in V}\sum_{t=\alpha}^{\omega}\left(R_{ij}^t \times x_{ij}^{kt}\right) + P_w \times \sum_{i\in\phi} w_i \\ & + p_e \times \sum_{i\in P}\left(\sum_{k\in V}\sum_{l\in N_s}\sum_{t=\alpha}^{\omega}(t - s_{n+i})x_{n+i,l}^{kt} - \sum_{k\in V}\sum_{j\in\phi}\sum_{t=\alpha}^{\omega} tx_{ij}^{kt} - \sum_{t=\alpha}^{\omega}\left(R_{i,n+i}^t \times \sum_{k\in V}\sum_{j\in\phi} x_{ij}^{kt}\right)\right) \\ & + P_d \times \sum_{i\in\phi} d_i \end{aligned} \tag{44}$$

4.3 Computational Study I

In this section, first, our model is validated through solving a set of small test problems by an exact method using a commercial package, CPLEX. Also, the results of the exact method and lower bound solutions are compared with the results of the heuristic algorithms that are developed for the model in this research. Second, the results of three algorithms based on clustering first- routing second, sequential insertion, and parallel insertion are compared based on problem instances which have 30, 50, and 100 customers respectively. Also the performances of three heuristic algorithms are analyzed in this section.

The heuristic algorithms were coded in C++. All computations were carried out on a machine with 2.0GHZ Intel Core 2 Duo CPU and 3GB memory in Windows XP environment.

4.3.1 Test problems I for validating the model and heuristic algorithms

(1) The characteristic of problem instances

The exact method can solve problems with a few customers that have to be serviced with a few vehicles. We assume that the service area is 20 miles by 20 miles and there are two depots. The location of depot 1 is (7, 10) and depot 2 is (13, 10). The demands are generated at random over the service area. There are 3, 4, and 5 customers with 10 and 15 time intervals respectively. Each combination of number of demand nodes and the time intervals has three cases of examples. Interval length is 6 minutes. For the 10 time interval case, time period is from 9 am to 10am. For the 15 time interval case, time period is from 9 am to 10:30 am.

(2) Parameter settings

The duration of time window is 12 minutes, maximum route duration for 10 time intervals is 60 minutes and for 15 time interval is 90 minutes, maximum acceptable waiting and delay time is 30 minutes, the fixed cost for used vehicle is $10,000/vehicle, the travel cost is $1/minute, the penalty cost for waiting time is $0.5/minutes, the penalty cost for delay time is $0.5/minute, and the penalty cost for customers' excess ride time is $0.5/minute.

4.3.2 Computational results I

In this section, the results from exact solution method, lower bound solution method, and three heuristic methods are presented. The gaps between the exact solution (E), lower bound solution (LB), and three heuristics solutions are calculated as follows:

Total cost gap between HSI and exact solution = (HSI – E)/E*100

Total cost gap between HPI and exact solution = (HPI – E)/E*100

Total cost gap between HCR and exact solution = (HCR – E)/E*100

Total cost gap between HSI and LB solution = (HSI – LB)/LB*100

Total cost gap between HPI and LB solution = (HPI – LB)/LB*100

Total cost gap between HCR and LB solution = (HCR – LB)/LB*100

Total cost gap between LB and exact solution = (E – LB)/LB*100

Calculation time ratio between HSI and exact solution =E/HSI

Calculation time ratio between HPI and exact solution =E/HPI

Calculation time ratio between HCR and exact solution =E/HCR

Calculation time ratio between HSI and LB solution =LB/HSI

Calculation time ratio between HPI and LB solution =LB/HPI

Calculation time ratio between HCR and LB solution =LB/HCR

Calculation time ratio between LB and exact solution =E/LB

Each combination of the number of customers and the time intervals has three cases of examples. In results, the average value of the three examples for each combination is calculated. Table 4.1 and 4.2 show the comparison of the calculation times for the exact method, the lower bound, and the three heuristic algorithms. As the number of customers exceeds 3 with service period of 10 time intervals, the calculation time of exact method increases exponentially and becomes unreasonable. The largest DARP problem size that could be solved in a reasonable time by exact method was 5 customers with service period of 10 time intervals. For most of the cases, the three heuristic algorithms solved the problems within less than 0.2 second while the exact method could not solve the problem which has 5 customers and 15 time intervals. For example, in case of 5customers and 10 time intervals, HPI solved the problem within 0.12 seconds, HSI solved the problem within 0.16 seconds and HCR solved the problem within 0.17 seconds while the exact method solved the problem within about 139 minutes.

To get the exact solution, we spent about 5091 times as much time as required for the HCR solution for the 5 customers and 10 time intervals. For small problems, the three heuristics algorithms solved the test problems faster than the lower bound and the exact method with almost the same objective function values. Figure 4.3 shows the comparison of HPI and the exact solutions.

Table 4.1 The comparison of results for calculation times (I)

Number of Customers	Number of Time interval	Calculation Times (seconds)				
		E	LB	HCR	HSI	HPI
3	10	8.08	6.31	0.16	0.15	0.14
3	15	62.12	9.94	0.16	0.14	0.14
4	10	56.30	63.30	0.17	0.17	0.15
4	15	1410.43	138.37	0.12	0.12	0.11
5	10	867.18	3563.34	0.17	0.16	0.12
5	15	-	4717.96	0.14	0.13	0.12

E: Exact solution
LB: Lower bound solution
HCR: Heuristic algorithm based on Clustering first-Routing second
HSI: Heuristic algorithm based on Sequential Insertion
HPI: Heuristic algorithm based on Parallel Insertion

Table 4.2 The comparison of results for calculation times (II)

Number of Customers	Number of Time interval	Calculation time ratio			
		E&LB	HCR&E	HSI.&E	HPI&E
3	10	1.28	50.74	53.35	56.43
3	15	6.25	426.6	473.42	500.66
4	10	0.89	330.27	340.18	372.77
4	15	10.19	11328.77	11574.97	12991.39
5	10	0.24	5091.08	5410.86	7115.81
5	15	-	-	-	-

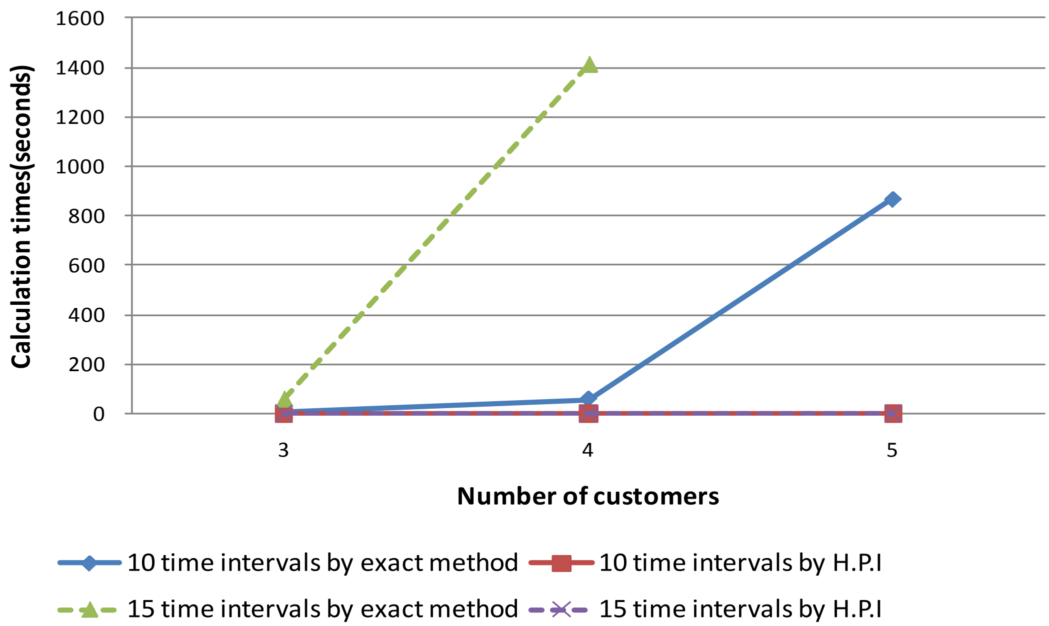


Figure 4.3 The comparison of the exact and HPI heuristic

Table 4.3, 4.4, and 4.5 show the comparison of the objective function values for the exact method, the lower bound, and the three heuristic algorithms. The gaps of the objective function value between the exact method and the three heuristic algorithms are less than 0.006%. For example, the gap range of the objective function value between the exact method and HSI is 0.001% to 0.006% and the gap range of the objective function value between the exact method and HPI is 0.001% to 0.003%. The gaps of the objective function value between the lower bound and the three heuristic algorithms are less than 0.014%. For example, the gap range of the objective function value between the lower bound and HSI is 0.007% to 0.014% and the gap range of the objective function value between the lower bound and HPI is 0.007% to 0.014%. The gap range of the objective functions value between the exact solution and the lower bound is 0.006% to 0.013%.

Table 4.3 The comparison of results for objective functions (I)

Number of Customers	Number of Time interval	Objective Functions				
		E	LB	HCR	HSI	HPI
3	10	30015.3	30013.0	30015.7	30017.0	30015.7
3	15	30019.2	30016.2	30020.0	30019.7	30019.7
4	10	40023.3	40018.2	40023.7	40023.7	40023.7
4	15	36689.0	36686.3	36690.7	36690.0	36690.0
5	10	50028.3	50025.3	50028.8	50028.8	50028.8
5	15	-	40024.2	40027.2	40027.5	40027.2

Table 4.4 The comparison of results for objective functions (II)

Number of Customers	Number of Time interval	Total cost gap			
		E&LB	HCR&E	HSI&E	HPI&E
3	10	0.008	0.001	0.006	0.001
3	15	0.010	0.003	0.002	0.002
4	10	0.013	0.001	0.001	0.001
4	15	0.007	0.005	0.003	0.003
5	10	0.006	0.001	0.001	0.001
5	15	-	-	-	-

Table 4.5 The comparison of results for objective functions (III)

Number of Customers	Number of Time interval	Total cost gap		
		HCR&LB	HSI&LB	HPI&LB
3	10	0.009	0.013	0.009
3	15	0.013	0.012	0.012
4	10	0.014	0.014	0.014
4	15	0.012	0.010	0.010
5	10	0.007	0.007	0.007
5	15	0.008	0.008	0.008

Among the three heuristic algorithms, the heuristic algorithm based on parallel insertion has the best performance based on calculation times and objective function values. For objective function values, HPI, HSI and HCR almost have the same objective function values within less than 0.004% of gap for all cases and HPI has a little better objective function value than HSI and HCR. For calculation times, there are subtle differences in the three heuristic algorithms and HPI solved the problems a little faster than HSI and HCR. We can conclude that the proposed heuristic algorithms work well in static DARP. They produce good results within a reasonable calculation time.

4.4 Computational Study II

4.4.1 Test problems II for analyzing the performance of heuristic algorithms

In order to test larger size problems which are similar to real service, several test problems were generated at random. The three heuristic algorithms were applied to these problems and their performances were compared.

(1) The characteristic of problem instances

We assume that the service area is 20 miles by 20 miles and there are two depots. The location of depot 1 is (7, 10) and depot 2 is (13, 10). The demands are generated at random over the service area. The problem sizes are 30, 50 and 100 customers, respectively. Time periods are from 6 am to 6pm. There are 72 time intervals and each time interval is 10 minutes.

(2) Parameter settings

The width of time window is 30 minutes, Maximum route duration is 720 minutes, Maximum acceptable waiting and delay is 30, 20, and 10 minutes, the fixed cost for used vehicle is $500/vehicle, the travel cost is $1/minute, the penalty cost for waiting time is $0.5/minute, the penalty cost for delay time is $0.5/minute, and the penalty cost for customers' excess ride time is $0.5/minute. The service times at demand node are 2 minutes for a regular passenger and 6 minutes for a passenger using wheelchair.

(3) Link Flow Speeds

In test problems, it is assumed that there are three classes of roads in network. First one is highways on which speed limit is 60 mph. Second one is major roads on which speed limit is 40 mph. The last one is minor roads on which speed limit is 30 mph. Each link belongs to one of these classes. Also it is assumed that link flow speeds of highways and major roads except for minor road are varied according to time interval.

4.4.2 Computational result II

As the sizes of the problems increase, the calculation times for solving the problems increase exponentially. In most cases, the problems are solved within 2 minutes using the heuristic algorithms. The calculation times, objective functions and the problem sizes are described in Tables 4.6 and 4.7.

Table 4.6 The computational results of test problems II: (a) calculation times

Customers	MaxWD	Vehicles at each depot	Calculation times (ms)				
			HSI (a)	HPI (b)	HCR (c)	Savings ((b-a) /b*100)	Savings ((c-a) /c*100)
30	30	15	2782	4184	3428	33.5%	18.87%
30	20	15	3138	3666	3173	14.4%	1.1%
30	10	15	2972	3446	2915	13.8%	-2.0%
50	30	30	7063	8460	8673	16.5%	18.6%
50	20	30	7904	8771	7927	9.9%	0.3%
50	10	30	8036	7590	7993	-5.9%	-0.5%
100	30	99	51647	68469	89682	24.6%	42.4%
100	20	99	52661	58768	72605	10.4%	27.5%
100	10	99	51204	55670	78515	8.0%	34.8%

MaxWD: Maximum allowable waiting and delay time

Table 4.7 The computational results of test problems II: (b) objective functions

Customers	MaxWD	Vehicles at each depot	Objective function values				
			HSI (a)	HPI (b)	HCR (c)	Savings ((b-a) /b*100)	Savings ((c-a) /c*100)
30	30	15	3651	6011	5021	39.3%	27.3%
30	20	15	5149	5542	5043	7.1%	-2.1%
30	10	15	5595	6074	6043	7.9%	7.4%
50	30	30	5686	6301	8088	9.8%	29.7%
50	20	30	8016	6934	7650	-15.6%	-4.8%
50	10	30	8506	7958	8661	-6.9%	1.8%
100	30	99	12979	15825	17704	18.0%	26.7%
100	20	99	14947	16359	16578	8.6%	9.8%
100	10	99	16041	15948	17625	-0.6%	9.0%

As mentioned before, maximum allowable waiting and delay (MaxWD) is considered in this research and the behaviors of the proposed three heuristics are evaluated under three different scenarios in which MaxWD is 30, 20, and 10 minutes, respectively in Tables 4.6 and 4.7. HSI performs better than HCR and HPI based on calculation times for most cases. For example, in case of 100 customers, HSI solved the problem 42.4% faster than HCR for MaxWD of 30 minutes, 27.5% for MaxWD of 20 minutes, and 34.8% for MaxWD of 10 minutes, respectively. Even in worst cases, the difference between the calculation times of HSI and HCR is less than -2.0%. Also, it is shown that the difference between the calculation times of HSI and HCR and the difference between the calculation times of HSI and HPI are decreasing as MaxWD decreases from 30 to 10 minutes for all cases of 30, 50 and 100 customers.

Also considering the objective function values, HSI is better than HPI and HCR in most cases. For example, in case of 100 customers, the solution of HSI is 26.7% better than that of HCR for MaxWD of 30 minutes, 9.8% for MaxWD of 20 minutes, and 9.0% for MaxWD of 10 minutes, respectively. Even in worst cases, the difference between the objective function values of HSI and HCR is less than -5%.

Also the performances of dial-a-ride service by the three heuristic algorithms are shown in Tables 4.8, 4.9, and 4.10 based on routing and scheduling of the results.

Table 4.8 The performances of three heuristic algorithms: (a) 30 customer case

	MaxWD = 30			MaxWD = 20			MaxWD = 10		
	HSI	HPI	HCR	HSI	HPI	HCR	HSI	HPI	HCR
Objective function value	3651	6011	5021	5149	5542	5043	5595	6074	6043
Used vehicle	5	10	8	8	9	8	9	10	10
Serviced customers	30	30	30	30	30	30	30	30	30
Ave. Serviced customers per vehicle	6.0	3.0	3.8	3.8	3.3	3.8	3.3	3.0	3.0
The number of serviced ambulatory	48	48	48	48	48	48	48	48	48
The number of serviced wheelchair	10	10	10	10	10	10	10	10	10
Ave. travel times per vehicle (min)	207.4	93.1	118.1	130.8	108.3	116.0	115.0	101.0	98.4
Ave. travel times per customer (min)	34.6	31.0	31.5	34.9	32.5	30.9	34.5	33.7	32.8
Ave. waiting times per vehicle (min)	15.2	1.3	7.1	3.5	2.6	5.5	2.1	1.2	0.0
Ave. delay times per customer (min)	3.1	2.5	0.1	1.0	3.0	3.3	1.6	0.8	1.2
Ave. excess ride times per customer (min)	2.0	2.40	3.03	2.9	0.7	2.9	1.8	3.0	2.7

MaxWD: Maximum allowable waiting and delay time

Table 4.9 The performances of three heuristic algorithms: (b) 50 customer case

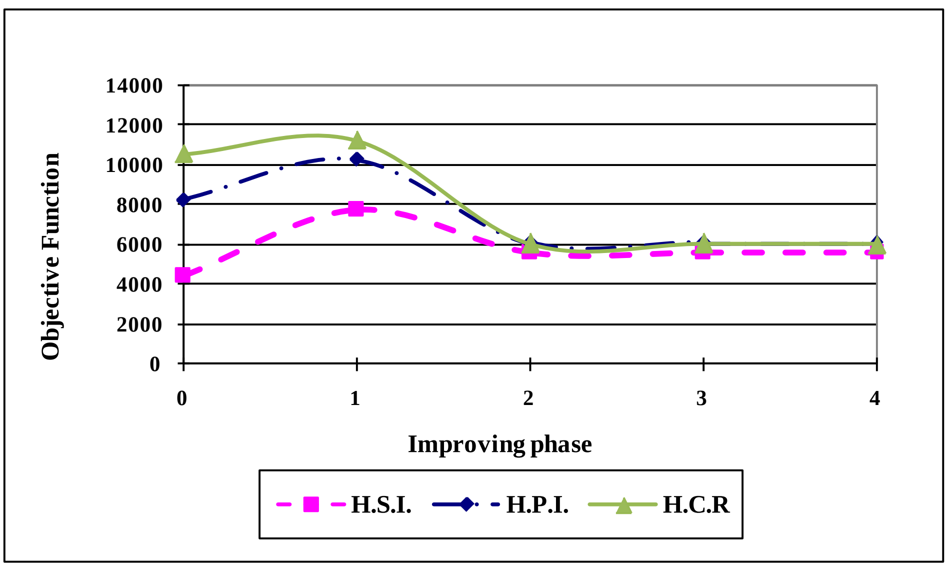
	MaxWD = 30			MaxWD = 20			MaxWD = 10		
	HSI	HPI	HCR	HSI	HPI	HCR	HSI	HPI	HCR
Objective function value	5686	6301	8088	8016	6934	7650	8506	7958	8661
Used vehicle	7	9	13	12	10	12	13	12	14
Serviced customers	50	50	50	50	50	50	50	50	50
Ave. Serviced customers per vehicle	7.1	5.6	3.9	4.2	5.0	4.2	3.9	4.2	3.6
The number of serviced ambulatory	73	73	73	73	73	73	73	73	73
The number of serviced wheelchair	26	26	26	26	26	26	26	26	26
Ave. travel times per vehicle (min)	278.7	175.8	111.5	154.4	178.0	128.0	147.7	153.4	113.4
Ave. travel times per customer (min)	39.0	31.6	29.0	37.1	35.6	30.7	38.4	36.8	31.8
Ave. waiting times per vehicle (min)	15.7	7.7	5.1	2.8	2.6	0.9	0.2	2.2	0.0
Ave. delay times per customer (min)	3.4	3.7	0.9	3.2	2.4	1.8	1.0	1.30	0.8
Ave. excess ride times per customer (min)	3.8	3.7	3.4	2.7	3.3	2.6	2.4	2.9	2.1

Table 4.10 The performances of three heuristic algorithms: (c) 100 customer case

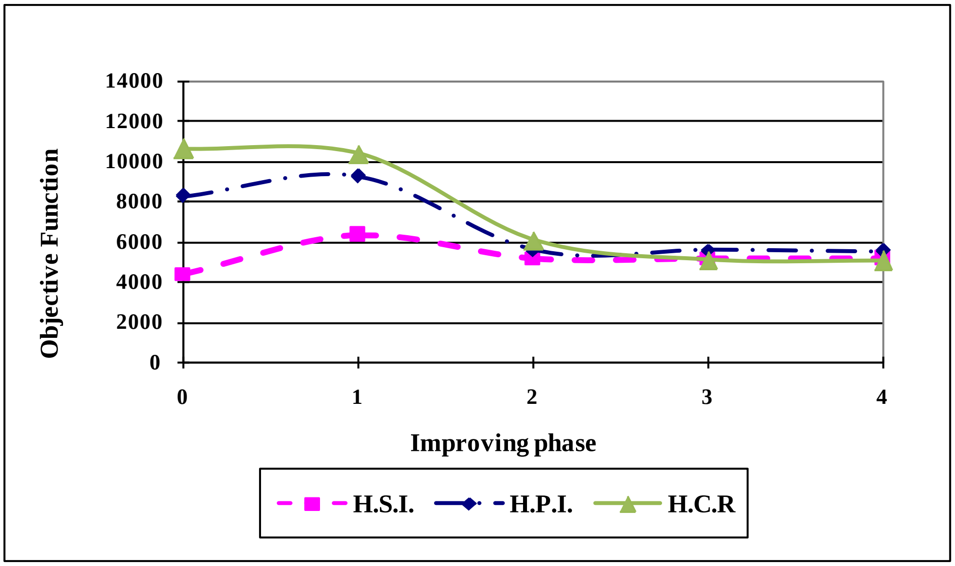
	MaxWD = 30			MaxWD = 20			MaxWD = 10		
	HSI	HPI	HCR	HSI	HPI	HCR	HSI	HPI	HCR
Objective function value	12979	15825	17704	14947	16359	16578	16041	15948	17625
Used vehicle	19	25	29	23	26	27	25	25	29
Serviced customers	100	100	100	100	100	100	100	100	100
Ave. Serviced customers per vehicle	5.3	4.0	3.5	4.4	3.9	3.7	4.0	4.0	3.5
The number of serviced ambulatory	157	157	157	157	157	157	157	157	157
The number of serviced wheelchair	50	50	50	50	50	50	50	50	50
Ave. travel times per vehicle (min)	170.6	122.0	102.3	141.8	118.9	104.3	135.5	130.5	102.8
Ave. travel times per customer (min)	32.4	30.5	29.7	32.6	30.9	28.2	33.9	32.6	29.8
Ave. waiting times per vehicle (min)	4.3	2.1	1.7	1.0	2.6	2.6	1.8	1.0	0.3
Ave. delay times per customer (min)	1.5	1.8	1.1	1.1	1.5	1.4	0.6	0.8	0.7
Ave. excess ride times per customer (min)	2.4	3.2	5.6	2.4	3.2	3.1	2.1	2.6	2.0

For most cases, the number of used vehicles by the solution of HPI and HCR is larger than that which is obtained by the solution of HSI. For 30 customer cases and 30 minutes of MaxWD, 10 vehicles are used in the solution of HPI while 5 vehicles are used in the solution by HSI. Also, average travel times per vehicle and average travel times per customers by the solution of HSI are larger than those by the solution of HPI and HCR for all 30, 50 and 100 customer cases.

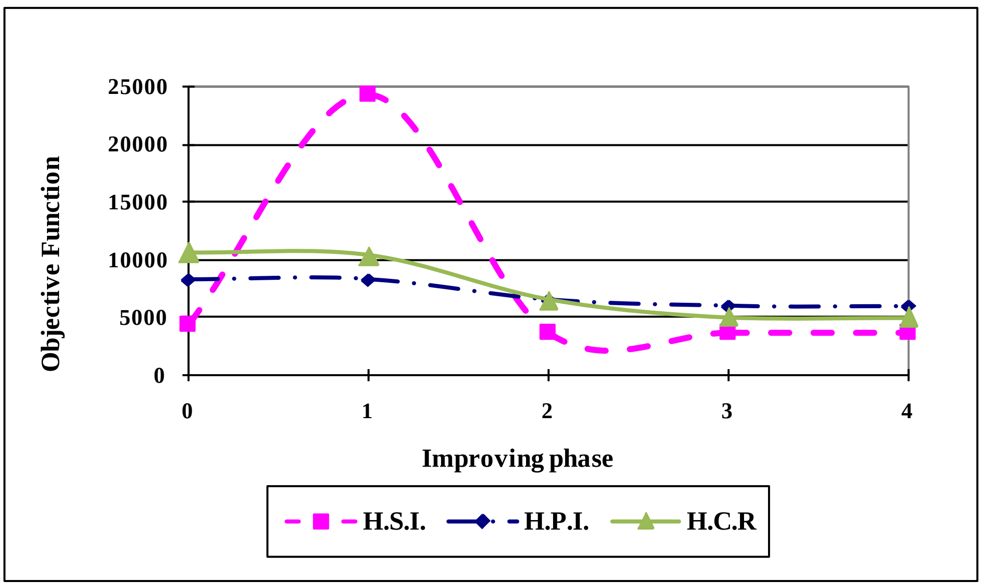
In this research, initial solution is improved through 4 steps of improvement. As seen in Figures 4.4, 4.5 and 4.6, most improvements in the objective function values for these problems are made through improvement steps I and II and objective functions converge as expected.



(a) MaxWD 10

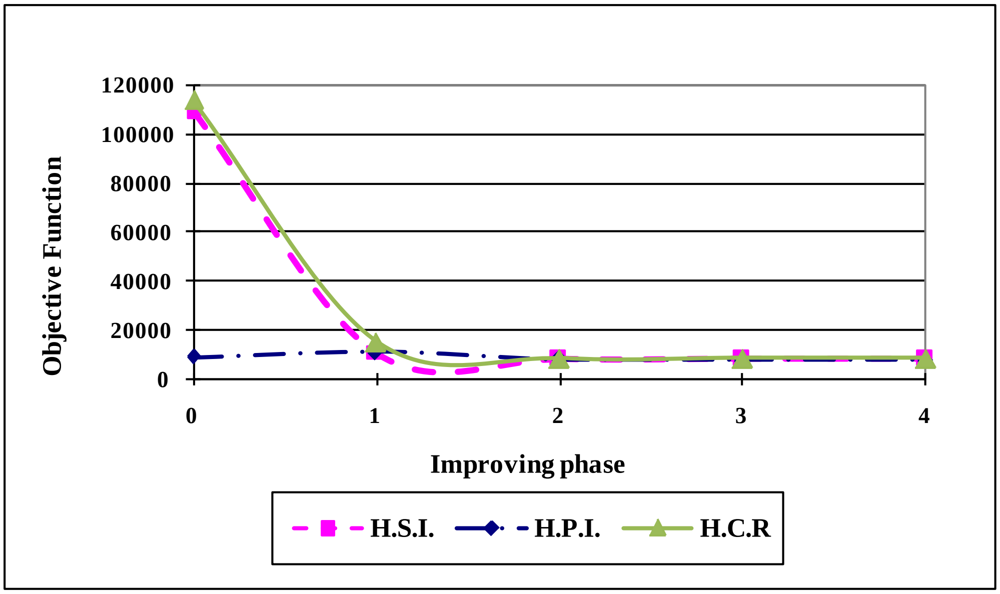


(b) MaxWD 20

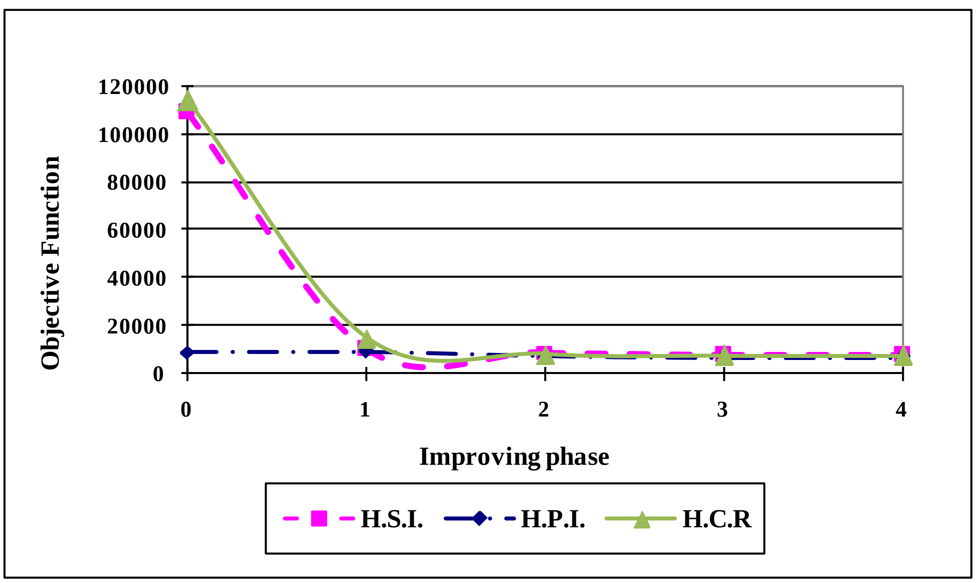


(c) MaxWD 30

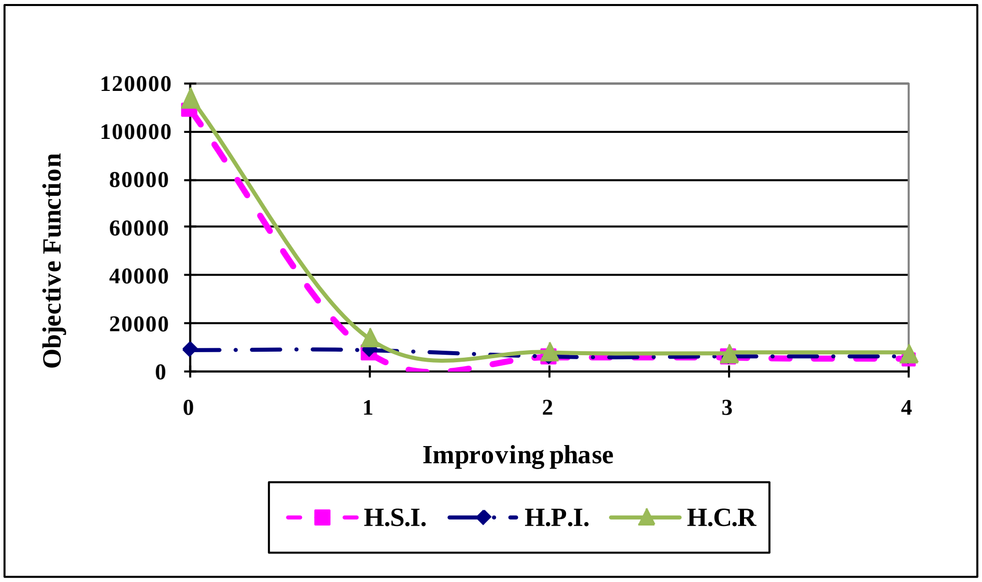
Figure 4.4 Convergence of objective function by improvement phase for 30 customers case



(a) MaxWD 10

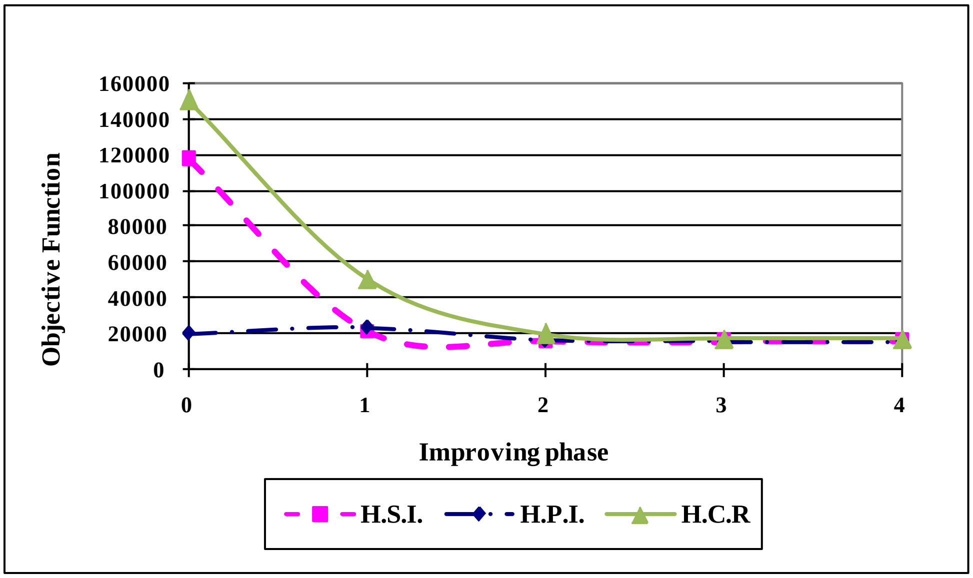


(b) MaxWD 20

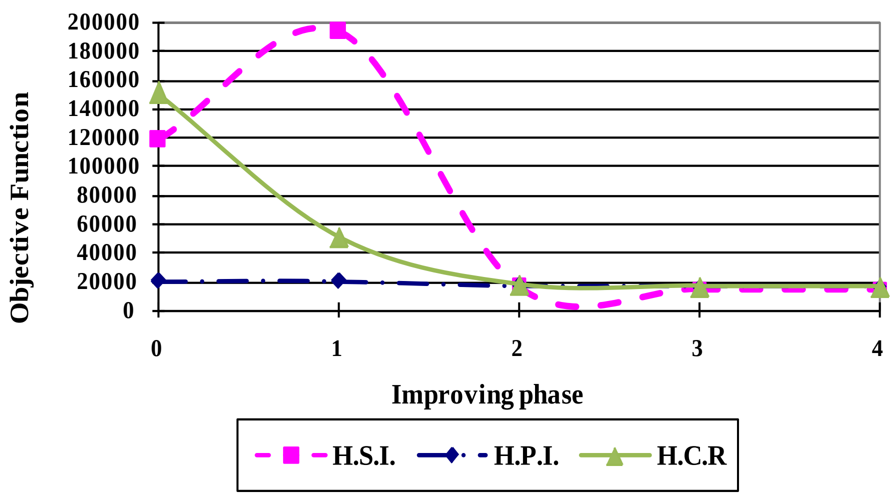


(c) MaxWD 30

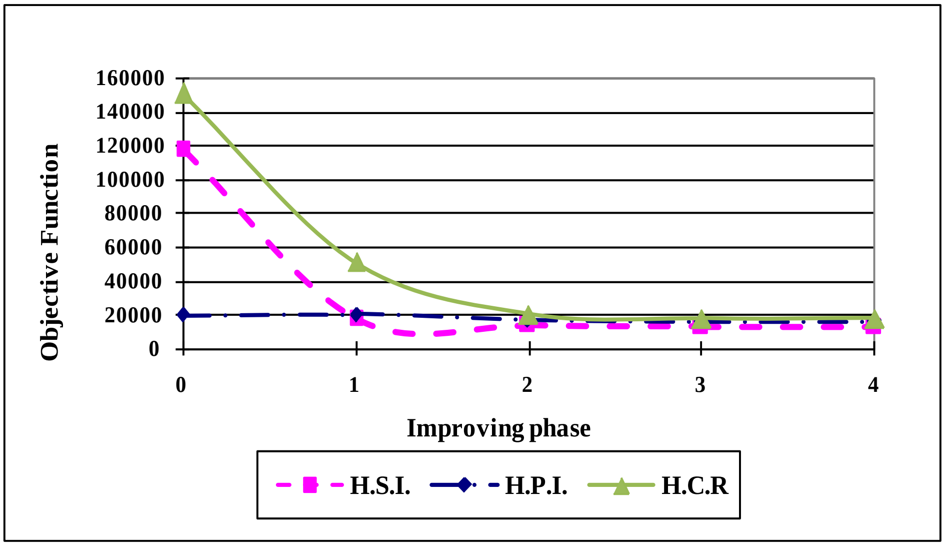
Figure 4.5 Convergence of objective function by improvement phase for 50 customers case



(a) MaxWD 10



(b) MaxWD 20



(c) MaxWD 30

Figure 4.6 Convergence of objective function by improvement phase for 100 customers case

4.5 Summary

In this chapter, heuristic algorithms were developed for static DARP model. The approach for solving this model can be divided into two phases including a construction phase and an improvement phase. In construction phase, we build a set of feasible routes for each depot starting from the information that define dial-a-ride problem by three different heuristic methods namely a sequential insertion(HSI), a parallel insertion(HPI), and a clustering first-routing second(HCR). After obtaining a feasible solution in the construction phase, the solution is improved through 4 steps of improvements, step 1 (acceptable waiting and delay), step 2 (removing one inserting one), step 3 (combining vehicles), and step 4 (adjusting vehicle starting time).

Our model was validated through solving a set of small test problems by an exact method using a commercial package, CPLEX. Also, the results of the exact method and a lower bound method were compared with the results of the heuristic algorithms which are developed for the model in this research. For small problems, the proposed heuristic algorithms solved the test problems faster than exact method with almost the same objective function values.

The results of the three algorithms based on clustering first- routing second, sequential insertion, and parallel insertion were compared based on problem instances that have 30, 50, and 100 customers respectively. HSI performs better than HCR and HPI based on the calculation times for most cases. Also considering the objective function values, HSI is better than HPI and HCR for most cases.

Chapter 5: Case Study for Large-Scale Static DARP

In this chapter, a case study for real world large-scale static DARP is presented. For the case study, Maryland Transit Administration (MTA)'s real operation of Dial-a-ride service is introduced and compared with the results of developed heuristic.

5.1 Dial a-Ride Service by MTA

5.1.1 Overview

The real world data for the case study is provided by the Maryland Transit Administration (MTA) in Baltimore, Maryland. The MTA has the primary responsibility for providing specialized demand-responsive transit (paratransit) services for people with disabilities who are not able to use the fixed-route public transportation in Baltimore City and Baltimore and Anne Arundel counties within three-quarters (3/4) of a mile of any MTA fixed-route service as shown in Figure 5.1.

Figure 5.1 Service areas (left) and the locations of three depots (right)

In the summer of 2004, MTA introduced a new centralized computer-based system for scheduling of rides and daily assignment of passengers to service routes. Actual service is provided through a combination of MTA-owned, two private companies, MV and Yellow Transportation-owned vehicles, all of whom receive instructions from the computer-supported central dispatcher at MTA.

Unless they are canceled in advance, reserved demands are initially scheduled at 5PM, on the day before the scheduled date. In case there are still unscheduled demands, the MTA uses so-called protection routes to serve them. Demands that still cannot be accommodated in protection routes are assigned to Yellow's taxi. Demands are categorized into ambulatory, wheelchair, and transferable wheelchair in terms of the type of space they require in vehicles.

The width of time window is 30 minutes. A minimum of two hours advanced notice is required for cancellation. A no show refers to a scheduled rider not showing up without prior notice.

Table 5.1 shows route assignments for the three providers, MTA, Yellow Transportation and MV. Once MTA's routes are all scheduled, Yellow's 100's and 200's routes are scheduled. And then, MV's 300's and 400's are scheduled. Grouping route name is convenient to identify the provider for a certain route in daily operations, but it may be unreasonable for using resources efficiently and reducing total cost.

Table 5.1 Route assignments for the providers by MTA's operating

Route Name	Provider	Feature
001-040	MTA	Regular Routes
100-161	Yellow	Regular Routes
200-271	Yellow	Regular Routes
300-367	MV	Regular Routes
400-452	MV	Regular Routes
501-504	Yellow	Protection Routes
601-604	MV	Protection Routes
9999	Yellow	Taxi

5.1.2 Dial-a-ride operating data

Corrected data between September 20 and October 1 in 2004 from MTA is available. The one day operating data on September 24, 2004 is extracted from Trapeze data base of MTA. This day is not a holiday to avoid confounding effects due to holidays or other special days. Also, it was a clear day to exclude the effect of weather on service performance. Booked, scheduled, performed, vehicle information and x-y coordinate for customers were extracted from MTA SQL data. On that date, a total 4,726 demand of reserved demands were scheduled excluding 804 demands of that were cancelled in advance. Each demand had a request time, time window, demand location and space type such as ambulatory, wheelchair, and transferable wheelchair. As shown in Table 5.2, there were 2,113 ambulatory passengers, 506 wheelchair passengers, and 33 transferable wheelchair passengers.

Table 5.2 Space type of customers

Space type	Ambulatory	Wheelchair	Transferable wheelchair
The number of passengers	2,113	506	33

The fleet consisted of 103 vehicles of Yellow transportation, 56 vehicles of MTA and 103 vehicles of MV. And there were several types of vehicles by configuration such as ambulatory space, wheelchair space and transferable wheelchair space.

Figure 5.2 shows the distributions of request times of the total 4,726 demands. Most of requests were concentrated on day time, specifically 7am to 10am and 2pm to 5pm.

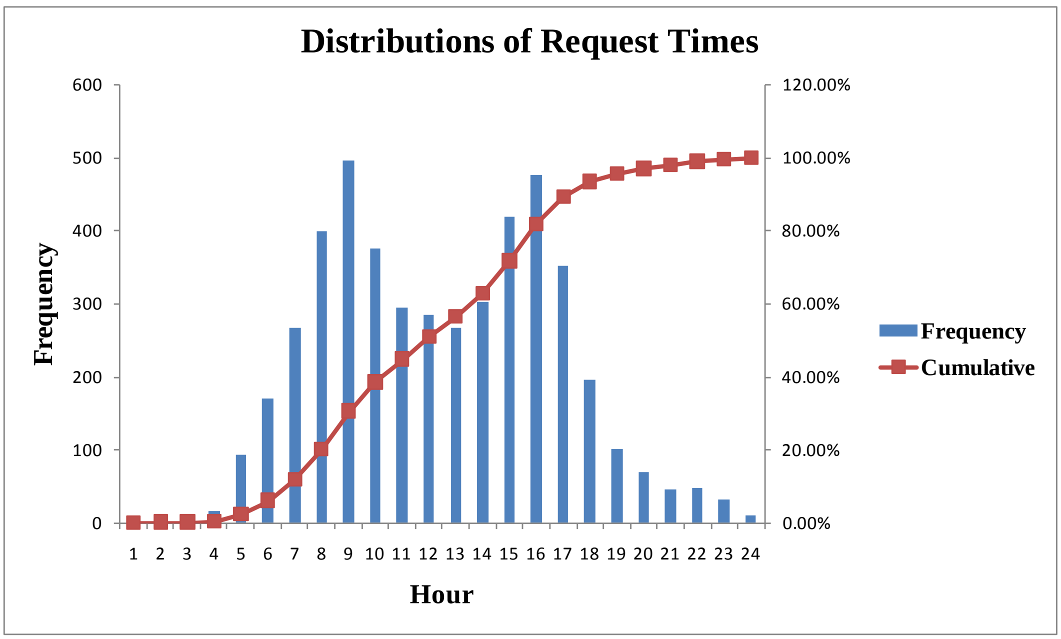


Figure 5.2 The distributions of request times

The time windows of demands were set based on the request times as follows:

For pickup demand,

$$Earliest\ allowable\ time\ window = request\ time\ of\ demand$$

$$Latest\ allowable\ time\ window = request\ time\ of\ demand + 30min$$

For drop off demand,

$$Earliest\ allowable\ time\ window = request\ time\ of\ demand - 30min$$

$$Latest\ allowable\ time\ window = request\ time\ of\ demand$$

Real demand locations are regarded as the closest intersection nodes in network as shown in Figure 5.3. In this Figure, the symbols of disabled are real demand locations and dots are the closest nodes from disabled.

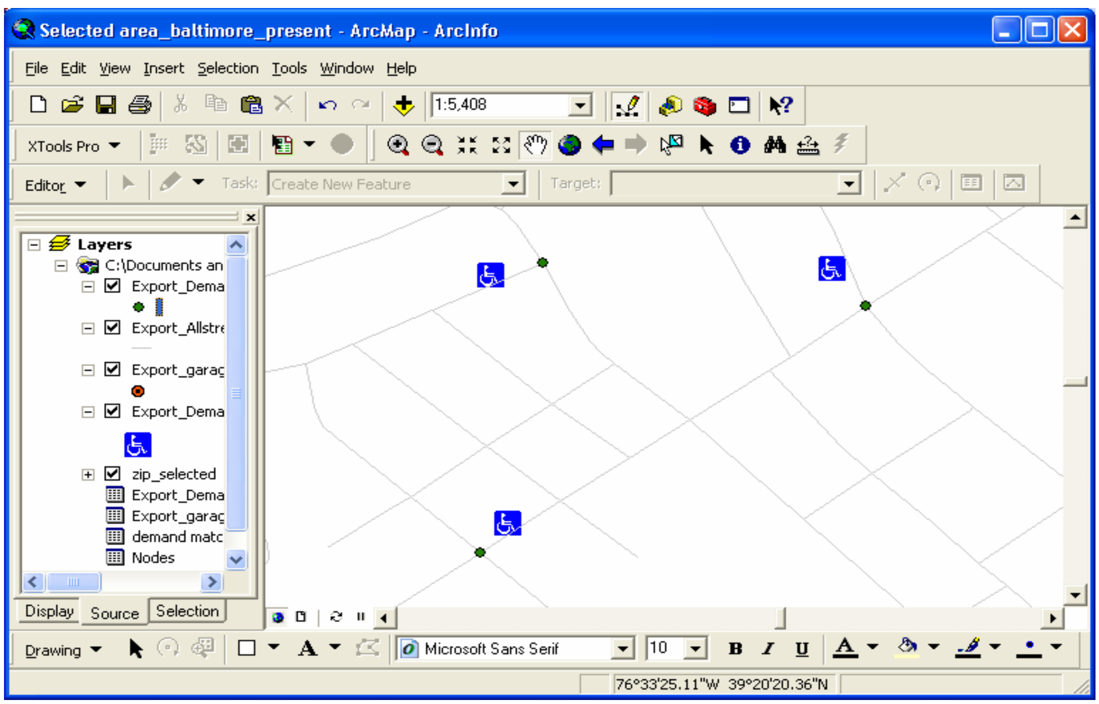


Figure 5.3 Matching demand locations and nodes

5.1.3 Network data

The network data for service areas were obtained from ArcLogistics Data as shown in Figure 5.4. Also, network connectivity was checked. Network connectivity is necessary for calculating time dependent shortest paths.

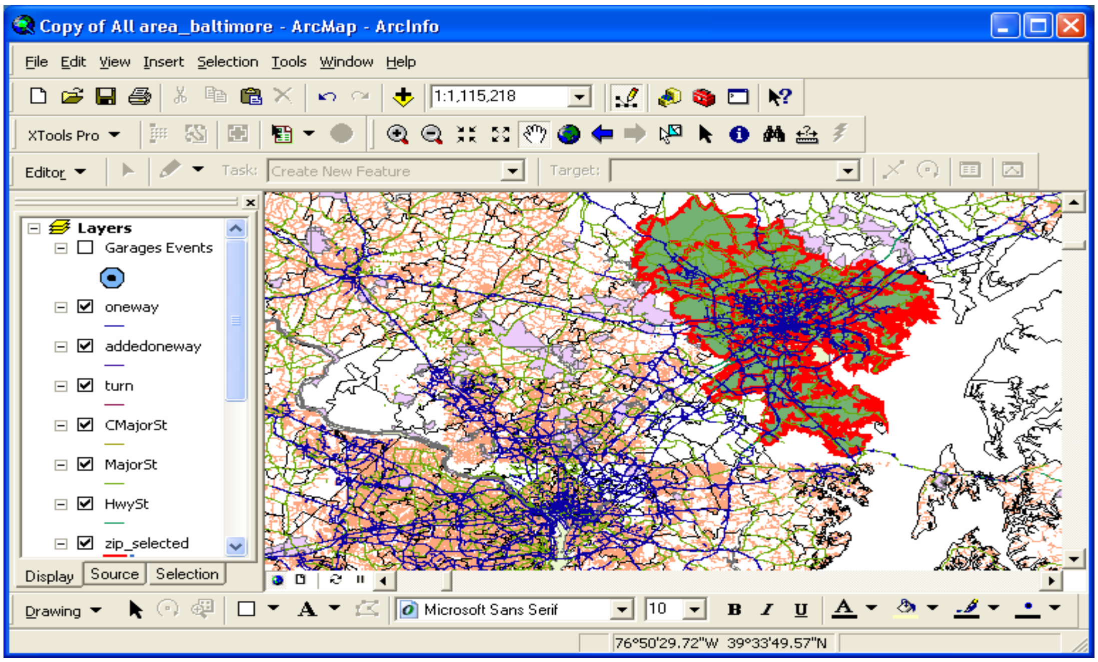


Figure 5.4 Service area map from ArcLogistics Data

As shown in Table 5.3, there are a total of 63,356 nodes, 8,446 links and 142,483 of directional arcs in network.

Table 5.3 Network size

Nodes	Links	Arcs
63,356	8,446	142,483

There are 3 different types of links according to their function. Each link has its own design speed. It is assumed that each link has different link speed according to the link type as shown in Table 5.4. This link speed is based on speed limit of the link.

Table 5.4 Link speed and Road factor by link type

Link type	Speed limit (mile/hour)	Uncongested travel time (min)	Road factor at each interval
Minor Street	30	=Length/30*60	1(always)
Major Street	40	=Length/40*60	-
Highway	60	=Length/60*60	-

Link speed is varying according to the time of day. And, travel time of link i at time interval t is calculated as follows:

$$\begin{aligned} &Travel\ time\ of\ link\ i\ at\ time\ interval\ t(min) \\ &\quad = Length \times 60/(link\ speed\ of\ link\ k \\ &\quad \times Road\ factor\ of\ link\ i\ at\ time\ interval\ t) \end{aligned}$$

For Minor Streets, there is no variation of link speed as road factor is always 1. Figure 5.5 shows the variation of road factor for Highways and Major Streets according to the time of the day.

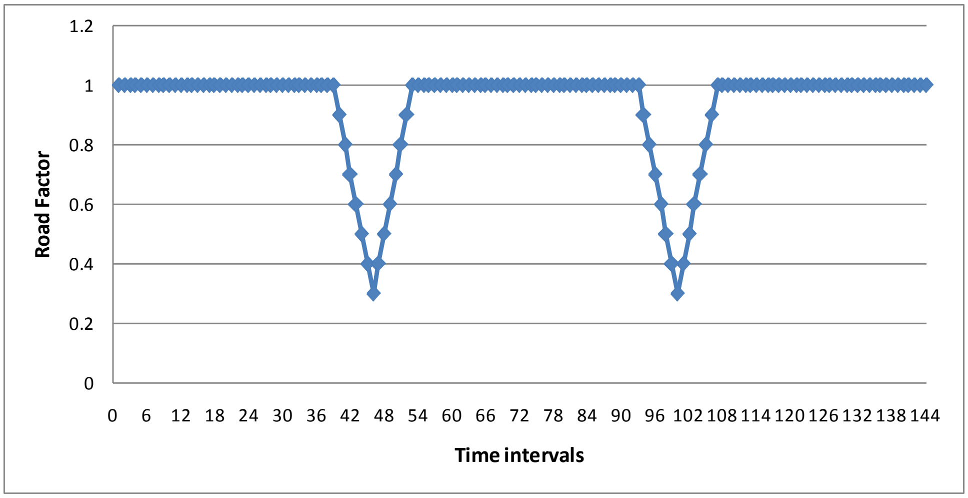


Figure 5.5 Road factor of Highway and Major Street

5.2 Implementation of the algorithm in the case study

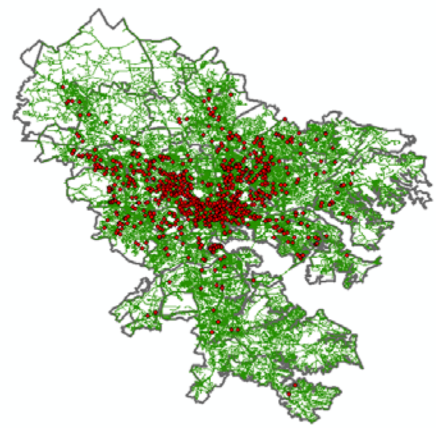
5.2.1 Decomposition of problem

It is necessary to decompose whole problem into subproblems to solve it in reasonable time. The idea of decomposing the problem comes from Figure 5.2. There are two peak periods for reqeust time of demands. Most demands are concentrated during the periods of 7 AM to10 AM and 2 PM to 5 PM. Therefore, the 5 time slots are decided upon based on the distributions of request times of demands (Table 5.5). The demands are grouped into these time slots as follows. If a latest allowalble time window of a pickup demand is within a certain time slot, then that pickup demand and the dropoff demand which is related to that pickup demand are grouped into the same time slot. Figure 5.6 shows the spatial distribution of demands based on the time slots. We can see that demands are mainly located in the center of Baltimore city.

Table 5.5 Distribution of demands by time slots

Time slots	Hour	The number of demands
1	0am – 7am	516
2	7am – 10am	1,320
3	10am – 2pm	1,100
4	2pm – 5pm	1,352
5	5pm - 0am	438
Total		4,726

(a) Time slot1 (0am-7am)



(b) Time slot 2 (7am-10am)

(c) Time slot 3 (10am-2pm)

(d) Time slot 4 (2pm-5pm)

(e) Time slot 5 (5pm-0am)

Figure 5.6 Spatial distributions of demands based on time slots

5.2.2 Procedure of HCR for decomposed problem

In this section, we describe how the proposed HCR in Chapter 4 is modified and implemented to the real large-scale problem. The procedure of HCR for this case study is as follows:

1. Start and T (Time slot) = 0.
2. Group demands into time slots.
3. Sort demands in each group.
4. T=T+1;
5. Construct initial solution based on the solution made on previous time slot, T-1.
6. Improve solution through improving step I, II, III, and IV.
7. If T>5, stop. Otherwise, update and check routes, vehicles, and demands. Go to step 4.

(1) Rate_Ins

Originally, in improving steps II and IV, when inserting a demand from a route into another route we need to check all other routes which can accept it and find a target route which has a minimum additional cost. As problem sizes becoming larger, there are many target routes to be checked and the procedure needs much time to find a good one. Therefore, it is modified to check just randomly selected target routs from the set of all routes using Rate_Ins, the ratio of selected target routes to all available routes for insertion.

(2) Rate_Com

Also, in Improving step III, when combining a route into another route it need to check other whole routes and find a minimum target route. In order to save calculation time, it is modified for just randomly selected target routs of all available routes to be checked using Rate_Com, the rate of selected target routes to all available routes for combining. Of course, solution deterioration by these modifications using Rate_Ins and Rate_Com should be accepted to some degree. Rate_Ins and Rate_Com will be analyzed in section 5.4.

5.3 Parameter settings

The width of time window is 30 minutes, Maximum route duration is 540 minutes, Maximum acceptable waiting and delay is 30, 20, and 10 minutes, the fixed cost for used vehicle is $200/vehicle, the travel cost is $1/minute, the penalty cost for waiting time is $0.5/minute, the penalty cost for delay time is $0.5/minute, and the penalty cost for customers' excess ride time is $0.5/minute. The service times at demand node are 2 minutes for a regular passenger, 4 minutes for a passenger using transferable wheelchair, and 6 minutes for a passenger using wheelchair, respectively.

The heuristic algorithms were coded in C++. All computations were carried out on a machine with 2.93GHZ Intel Core i7 CPU and 8GB memory in Windows 7 environment.

5.4 Computation results

In this section, first we analyze Rate_Ins and Rate_Com and set the value of Rate_Ins and Rate_Com to solve the problem within reasonable time without quality deterioration. Next, we schedule and route 4726 demands based on three cases, MaxWD 30, 20, and 10 minutes, respectively. And, then we compare our results with MTA's real operation.

5.4.1 Rate_Ins and Rate_Com

In this section, Rate_Ins and Rate_Com which are introduced in previous section are analyzed. To test the effect of Rate_Ins and Rate_Com, we change the Rate_Ins and Rate_Com ranges as shown in Table 5.6

Table 5.6 Rate_Ins and Rate_Com ranges

Problems	Rate_Ins	Rate_Com
HCR (1)	0.1	0.5
HCR (2)	0.2	0.5
HCR (3)	0.5	0.5

Table 5.7 and Figure 5.7 show the result of 3 cases using different Rate_Ins and Rate_Com. HCR (1) using Rate_Ins, 0.1 and Rate_Com, 0.5 had a good solution within reasonable time, 264 minutes. HCR (1) saved calculation times by 30.16% compared to HCR (2) and 60.42% compared to HCR (3), respectively. As it is expected, the objective function value of HCR (1) is deteriorated by 0.02% compared to HCR (2) and 0.02% compared to HCR (3) and it is within an acceptable degree. Therefore, the values of 0.1 for Rate_Ins and 0.5 for Rate_Com are used for all computations.

Table 5.7 Computational Result for Rate_Ins and Rate_Com

Problems	Rate_Ins	Rate_Com	Objective functions	Calculation times (min)
HCR (1)	0.1	0.5	95759	264
HCR (2)	0.2	0.5	94113	378
HCR (3)	0.5	0.5	93722	667

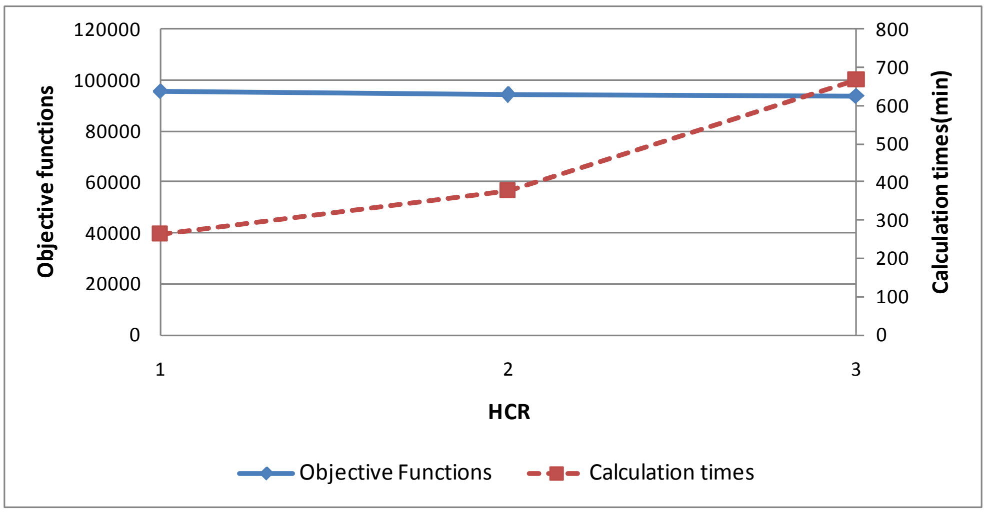


Figure 5.7 Objective functions and calculation times by Rate_Ins and Rate_Com

5.4.2 Preliminary results of the three heuristics

Based on reserved demands excluding the cancelations in advance, a total of 4,726 demands are scheduled and routed using the three heuristics HCR, HSI, and HPI for the large-scale DARP. In this section, the results of three heuristics are compared with each other and the best one of the three heuristics is chosen for further study. For this preliminary test, acceptable waiting and delay time is 30 minutes.

Table 5.8 shows the objective function values and calculation times for the three heuristics. The last two columns in Table 5.5 are savings of cost and calculation time by HCR compared to HSI and HPI.

Table 5.8 The costs and calculation times for three heuristics

Cases	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost	Cal. Time (min)	Saving of cost (%)	Saving of cal. Time (%)
HCR	95759	48400	41035	1194	669	4461	264	-	-
HSI	92161	46600	39564	927	728	4342	375	-3.9%	29.6%
HPI	110548	66800	38494	1010	715	3529	436	13.4%	39.4%

From the objective function perspective, HSI is better than the two other heuristics, HCR and HPI. HSI can reduce the number of routes while HPI has more routes than others. The gap of objective function values of HCR and HSI is not large (-3.9%). From the calculation time perspective, HCR solves the problem faster than HSI and HPI by 29.6% and 39.4%, respectively. HCR is a good heuristic for large-scale DARP from both the objective function value and calculation time perspective.

After comparing the three heuristics, HCR is selected for solving the large-scale DARP. HCR is applied to three cases, MaxWD 10, 20, and 30 minutes. Table 5.9 and Figure 5.8 show the results of the three cases.

We can see that total cost and the fixed cost decrease as maximum acceptable waiting and delay increases. More routes are made to satisfy tight constraints of waiting and delay times. For example, when MaxWD is 30 minutes, 242 routes are made and 308 routes are made when MaxWD is 10 minutes.

As MaxWD increases, more time is needed to complete scheduling and routing. For example, 212 minutes is spent for calculation when MaxWD is 10 minutes, 222 minutes when MaxWD is 20 minutes, and 264 minutes when MaxWD is 30 minutes. We can see that less time is spent for computation as MaxWD is tighter.

Table 5.9 The cost and calculation times for three cases

Problems	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost	Cal. Time (min)
MaxWD 10	108830	61600	41185	363	1047	4635	212
MaxWD 20	99962	53200	40889	640	407	4826	222
MaxWD 30	95759	48400	41035	1194	669	4461	264

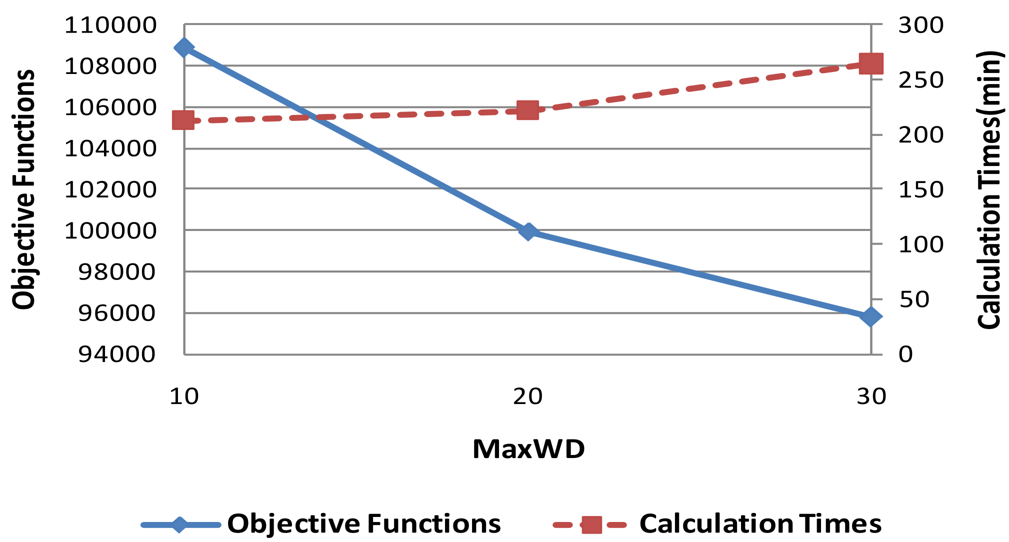


Figure 5.8 The objective functions and calculation times for three cases

Table 5.10 shows that the performances of scheduling and routing for the three cases, MaxWD 10, 20, and 30 minutes, respectively. Average scheduled customers per vehicle increase as MaxWD increases. For example, average scheduled customers per vehicle are 7.67 for MaxWD 10 minutes, 8.88 for MaxWD 20 minutes, and 9.76 for MaxWD 30 minutes. We can see that more customers are serviced by a vehicle as the degree of violation of waiting and delay is allowed to be larger. Also, the average travel times per vehicle increase as MaxWD increases. For example, the average travel times per vehicle are 133.72 minutes for MaxWD 10 minutes, 153.72

minutes for MaxWD 20 minutes, and 169.57 minutes for MaxWD 30 minutes. As the level of acceptable waiting and delay increase, more customers are serviced by a vehicle and therefore travel time of the vehicle increases. As MaxWD increases, average waiting times also increase. For example, average waiting times per vehicle is 2.35 minutes for MaxWD 10 minutes, 4.81 minutes for MaxWD 20 minutes, and 9.87minutes for MaxWD 30minutes. For average ride time per customers and average excess ride time per customers, there is not much difference.

Table 5.10 The performances of scheduling and routing for three cases

	HCR MaxWD 30	**HCR MaxWD 20**	**HCR MaxWD 10**
Number of routes	242	266	308
Scheduled customers	2363	2363	2363
Scheduled demands	4726	4726	4726
Ave. scheduled customers per vehicle	9.76446	8.88346	7.67208
The number of scheduled ambulatory	2113	2113	2113
The number of scheduled wheelchair	506	506	506
The number of scheduled transferable wheelchair	33	33	33
Ave. travel times per vehicle (min)	169.566	153.718	133.718
Ave. ride times per customer (min)	11.3415	11.6686	11.5032
Ave. waiting times per vehicle (min)	9.86777	4.81203	2.35714
Ave. delay times per customer (min)	0.565806	0.344477	0.886162
Ave. excess ride times per customer (min)	3.77571	4.08464	3.92256

5.4.3 Comparison of results of HCR with MTA's operating

(1) Demands

For comparison of results of HCR with MTA's operation, 4,604 demands of total 4,726 demands are scheduled and routed since operating data of 4,604 demands are available from MTA's database. Table 5.11 shows the distribution of demands by time slots.

Table 5.11 The distribution of demands by time slots

Time slots	Hour	The number of demands
1	0am – 7am	510
2	7am – 10am	1,302
3	10am – 2pm	1,062
4	2pm – 5pm	1,310
5	5pm - 0am	420
Total		4,604

(2) Link Travel Times

Also, for comparison of results of HCR with MTA's operation, in HCR static travel times are used instead of time-dependent travel times. And, it is not possible to exactly compare the result of HCR with MTA's operation since link travel times which are used in MTA's operation were not available in the database. Therefore, two scenarios are introduced. For the first scenario, we assume that actual link speeds are slower by 25% than original link speeds which are set based on speed limits of links. In this case, the value of Speed Factor is 0.75 and modified link speed is calculated as follows:

$$modified\ link\ speed = original\ link\ speed \times Speed\ Factor$$

For the second scenario, we assume that actual link speeds are equal to the original link speeds which are set. In this case, the value of Speed Factor is 1.0.

(3) Results

Actually, in MTA's operation, there is no waiting time, delay time, and excess ride time when all demands are scheduled while total cost includes waiting cost, delay cost, and excess ride cost in HCR. Therefore, it is necessary to analyze total cost according to the variations of waiting cost unit, delay cost unit, and excess ride cost unit in HCR. For this analysis, a total of 24 cases are made as shown in Table 5.12.

Table 5.12 Cases for comparison the results of HCR with MTA's operation

Cases	MaxWD	SpeedFactor	Waiting cost unit($/min)	Delay cost unit($/min)	Excess ride cost unit($/min)
1	30	1.0	0	0	0
2	30	1.0	0.5	0.5	0.5
3	30	1.0	3	3	3
4	30	1.0	5	5	5
5	30	0.75	0	0	0
6	30	0.75	0.5	0.5	0.5
7	30	0.75	3	3	3
8	30	0.75	5	5	5
9	20	1.0	0	0	0
10	20	1.0	0.5	0.5	0.5
11	20	1.0	3	3	3
12	20	1.0	5	5	5
13	20	0.75	0	0	0
14	20	0.75	0.5	0.5	0.5
15	20	0.75	3	3	3
16	20	0.75	5	5	5
17	10	1.0	0	0	0
18	10	1.0	0.5	0.5	0.5
19	10	1.0	3	3	3
20	10	1.0	5	5	5
21	10	0.75	0	0	0
22	10	0.75	0.5	0.5	0.5
23	10	0.75	3	3	3
24	10	0.75	5	5	5

Figure 5.9 and Table 5.13 show the comparison of objective function values of HCR with MTA’s operation. For the comparison, waiting cost, delay cost, and excess ride cost are not included in the objective function values of HCR. The objective function values from HCR are better than those from MTA’s operation for all cases. When SpeedFactor is 1.0, the results of heuristic are better than MTA's operation based on the variation of cost unit (0, 0.5, 3, and 5) and there is a 39.9 to 48.2% savings in total cost. Also, when SpeedFactor is 0.75, the results of heuristic are better than MTA's operation based on the variation of cost unit (0, 0.5, 3, and 5) and there is a 21.4 to 33.3% savings in total cost. Also there is a 40% savings in the routing cost for all cases. This big difference between the results of HCR and MTA’s operation results from the MTA’s route assignment method. As mentioned earlier, MTA assigns a vehicle of specific company to the route which has certain ranges of route number to balance used vehicles from each private company. MTA’s routes assignment method is not reasonable for minimizing total cost. The gap of objective function values between HCR and MTA’s operation for SpeedFactor 1.0 is larger than that for SpeedFactor 0.75. As waiting cost unit, delay cost unit, and excess ride cost unit increase, the savings in total cost slightly decrease. Also, the savings in total cost slightly decrease as MaxWD decreases.

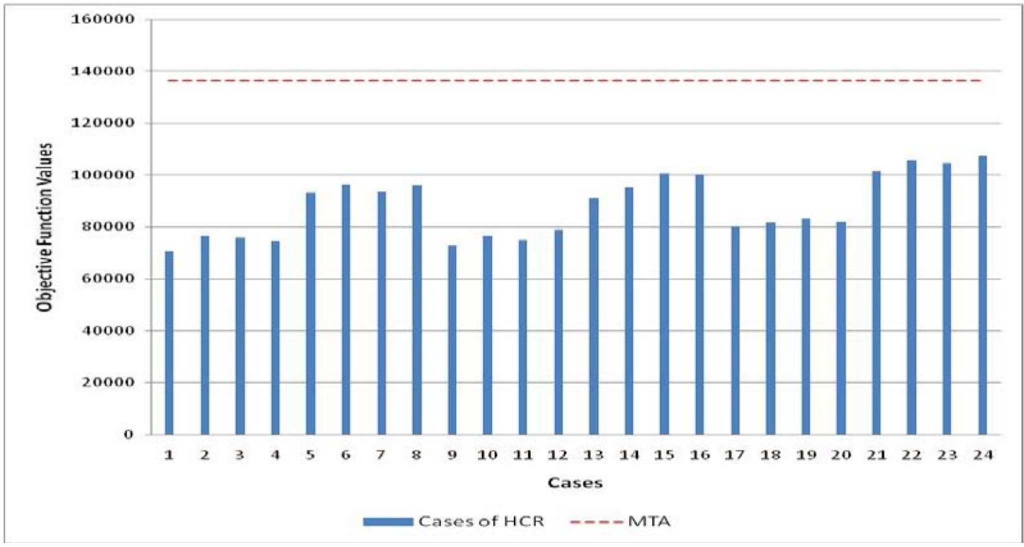


Figure 5.9 The comparison of objective functions of HCR with MTA's operation (I)

Table 5.13 Comparison of results of HCR with MTA's operation (I)

Cases	Total cost	Fixed cost	Routing cost	Total cost saving (%)	Calculation Time (min)
MTA	136473	56400	80073	-	-
1	70642	40000	30642	48.2	341
2	76366	42600	33766	44.0	203
3	75885	41400	34485	44.4	207
4	74784	39800	34984	45.2	217
5	93112	51800	41312	31.8	323
6	96417	51600	44817	29.4	257
7	93547	47200	46347	31.5	257
8	95983	49800	46183	29.7	230
9	72802	41600	31202	46.7	294
10	76541	43200	33341	43.9	196
11	75048	41200	33848	45.0	190
12	78892	44400	34492	42.2	201
13	91012	49000	42012	33.3	287
14	95345	51000	44345	30.1	220
15	100647	54200	46447	26.3	212
16	100297	53400	46897	26.5	214
17	79957	47600	32357	41.4	208
18	81585	47800	33785	40.2	168
19	83349	48600	34749	38.9	170
20	81969	47000	34969	39.9	176
21	101610	57800	43810	25.5	236
22	105799	60200	45599	22.5	201
23	104413	57600	46813	23.5	207
24	107303	60000	47303	21.4	202

Figure 5.10 and Table 5.14 show the comparison of the results of HCR with MTA's operation when waiting cost, delay cost, and excess ride cost are included in the objective function values in HCR. The objective function values from HCR are better than those from MTA's operation except in 4 cases 8, 16, 23, and 24. When SpeedFactor is 1.0, the results of heuristic are better than MTA's operation based on the variation of cost unit (0, 0.5, 3, and 5). But, when SpeedFactor is 0.75, the results of heuristic are better than MTA's operating until cost units increase by 3 and MTA's operation is better than the results of heuristic as cost units increase by so many times. Also there is a 40% savings in the routing cost for all cases.

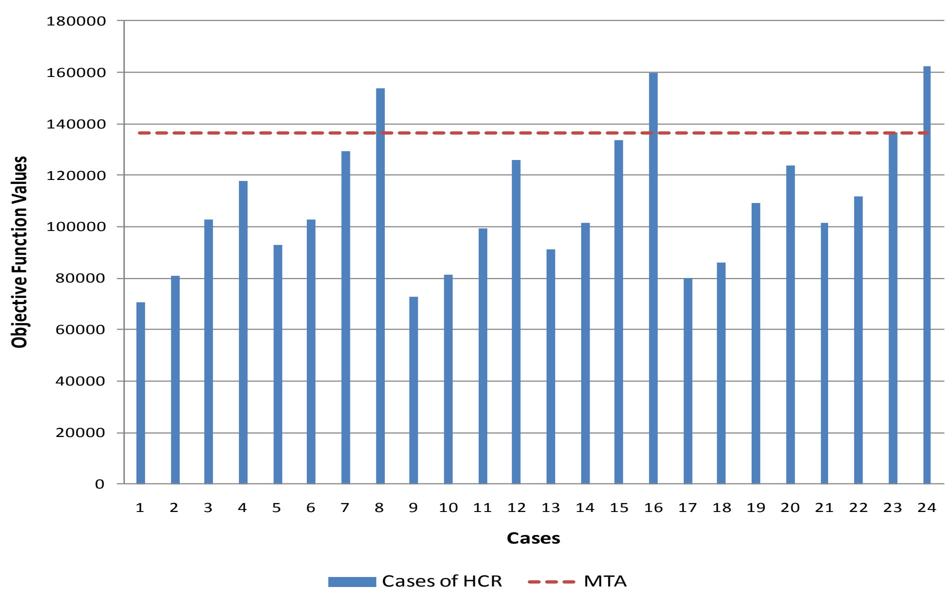


Figure 5.10 The comparison of objective functions of HCR with MTA's operation (II)

Table 5.14 Comparison of results of HCR with MTA's operation (II)

Cases	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost	Total cost saving (%)	Calculation time(min)
MTA	136473	56400	80073	0	0	0	-	-
1	70642	40000	30642	0	0	0	56.7	341
2	80770	42600	33766	698	385.5	3320.5	40.8	203
3	102675	41400	34485	5277	1083	20430	24.8	207
4	117799	39800	34984	7075	2850	33090	13.7	217
5	93112	51800	41312	0	0	0	42.9	323
6	102847	51600	44817	1046	661.5	4722.5	24.6	257
7	129310	47200	46347	8016	2781	24966	5.2	257
8	153838	49800	46183	7260	6135	44470	-12.7	230
9	72802	41600	31202	0	0	0	55.4	294
10	81315	43200	33341	643	479	3652	40.4	196
11	99486	41200	33848	2859	1974	19605	27.1	190
12	125822	44400	34492	3900	8725	34305	7.8	201
13	91012	49000	42012	0	0	0	44.2	287
14	101515	51000	44345	645	730	4795	25.6	220
15	133473	54200	46447	2832	3444	26550	2.2	212
16	159762	53400	46897	6265	10920	42280	-17.1	214
17	79957	47600	32357	0	0	0	51	208
18	86124	47800	33785	264	446.5	3828	36.9	168
19	109251	48600	34749	2001	3615	20286	19.9	170
20	123964	47000	34969	2195	4355	35445	9.2	176
21	101610	57800	43810	0	0	0	37.7	236
22	111741	60200	45599	386	666	4890	18.1	201
23	136714	57600	46813	1449	3276	27576	-0.2	207
24	162308	60000	47303	3055	5535	46415	-18.9	202

Average calculation time for all cases is 225.7 min. These calculation times are reasonable because the computation work can be done before 9pm and there are sufficient times to be ready for the next day's service if the computation work of scheduling and routing for reserved demands starts at 5 pm.

5.5 Sensitivity Analysis

In this section, a sensitivity analysis for the parameters that are used in this model is performed. A sensitivity analysis is the process of varying model input parameter values over a reasonable range and observing the relative change in model response. The purpose of the sensitivity analysis is to demonstrate the sensitivity of the model simulations to uncertainty in values of model input data.

The parameters for sensitivity analysis in this section are taken from the formulation in chapter 3. The objective function of our proposed mathematical model has many parameters such as *fc* (fixed cost of one vehicle, $/vehicle), *rc* (routing cost for the unit travel time, $/min), *pe* (excess ride time penalty, $/min), *pw* (waiting penalty, $/min), and *pd* (delay penalty, $/min). We set a base value and a range for each model parameter. Table 5.15 shows that the base value and the range of the parameters. We test three cases, MaxWD 10, 20, and 30 minutes, respectively.

Table 5.15 Sensitivity test ranges of parameters

Cost	Base Value	Range
Fixed cost	200	0, 100, 200, 500
Routing cost	1	0, 1, 5, 50
Waiting cost	0.5	0, 0.5, 5, 50
Delay cost	0.5	0, 0.5, 5, 50
Excess ride cost	0.5	0, 0,5, 5, 50

5.5.1 Fixed cost

The fixed cost is the capital cost for a vehicle. Table 5.16 shows the result of the fixed cost sensitivity analysis. We change the fixed cost unit range from 0 to 500 while other parameters have the fixed value with their base values. In all three cases, there is no change in the number of vehicles except for fixed cost unit, 0. The number of used vehicles remains the same when the fixed cost unit increases from 100 to 500. In the case of MaxWD 30 minutes, when we set the fixed cost unit as 0, 247 vehicles are used. And, 242 vehicles are used for fixed cost unit, 100, 200, and 500. In the case of MaxWD 20 minutes, when we set the fixed cost unit as 0, 291 vehicles are used. And, 266 vehicles are used for fixed cost unit, 100, 200, and 500. In the case of MaxWD 10 minutes, 310 vehicles are used for fixed cost unit, 0. And, 308 vehicles are used for the fixed cost unit, 100, 200, and 500.

Table 5.16 Sensitivity Analysis of Fixed Cost

Problems	Fixed cost unit	Number of vehicles	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost
HCR MaxWD 30	0	247	46948	0	40442	1025	912	4569
	100	242	71559	24200	41035	1194	669	4461
	200	242	95759	48400	41035	1194	669	4461
	500	242	168359	121000	41035	1194	669	4461
HCR MaxWD 20	0	291	45911	0	40519	711	443	4238
	100	266	73362	26600	40889	640	407	4826
	200	266	99962	53200	40889	640	407	4826
	500	266	179762	133000	40889	640	407	4826
HCR MaxWD 10	0	310	47863	0	41331	347	1490	4696
	100	308	78030	30800	41185	363	1047	4635
	200	308	108830	61600	41185	363	1047	4635
	500	308	201230	154000	41185	363	1047	4635

Figure 5.11 shows the cost comparison for three cases of problems. The cost is the total cost minus the fixed cost. As shown in Figure 5.11, there are not big

changes in the total cost minus the fixed cost for the three problems when the fixed cost unit is between 100 and 500.

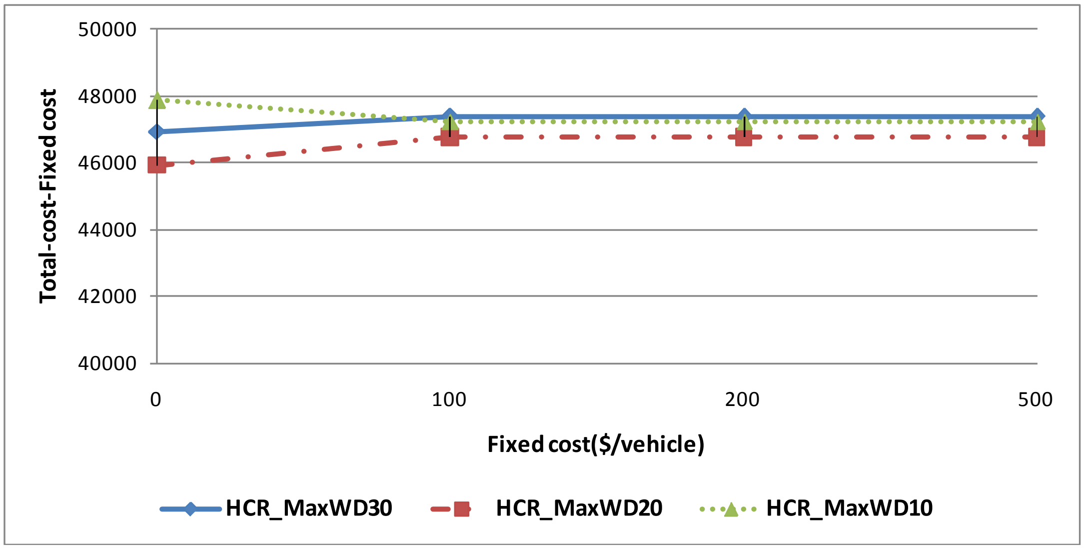


Figure 5.11 Sensitivity Analysis of Fixed Cost

The sensitivity analysis result of the fixed cost shows that the model keeps the minimum number of vehicles after the fixed cost unit increases more than 100. We cannot reduce operating cost beyond the required minimum level, even when we can use unlimited number of vehicles with 0 fixed cost.

5.5.2 Routing cost

The routing cost is related with the link travel times. Table 5.17 shows the results of the sensitivity analysis for the three cases, MaxWD 10, 20, and 30 minutes, respectively. We can see that as the routing cost unit increase, routing times decreases In case of MaxWD 30 minutes, the model spends 43,474 minutes for routing when routing cost unit is 0 and the routing time is 38,307 minutes when routing cost unit is 50. In case of MaxWD 20 minutes, the model spends 42,095 minutes for routing when routing cost unit is 0 and the routing time is 38,687 minutes when routing cost unit is 50. Also, in case of MaxWD 10 minutes, the model spends 43,578 minutes for routing when routing cost unit is 0 and the routing time is 40,130 minutes when unit routing cost is 50. This shows that it is possible to control the routing time with the routing cost.

Table 5.17 Sensitivity Analysis of Routing Cost

Problems	Routing cost unit	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost	Routing time (min)
HCR MaxWD 30	0	59211	53600	0	981	574	4056	43474
	1	95759	48400	41035	1194	669	4461	41035
	5	254659	51000	196220	1622	1018	4799	39244
	50	1974670	50800	1915350	1718	1710	5092	38307
HCR MaxWD 20	0	59058	51800	0	783	1941	4534	42905
	1	99962	53200	40889	640	407	4826	40889
	5	252193	50800	194780	992	932	4689	38956
	50	1997889	53800	1934350	1222	3202	5316	38687
HCR MaxWD 10	0	64381	59200	0	347	520	4315	43578
	1	108830	61600	41185	363	1047	4635	41185
	5	268308	60200	201750	385	961	5013	40350
	50	2072759	59400	2006500	446	1112	5301	40130

As the routing cost unit increases, the fixed cost does not change much except for the case of MaxWD 30 minutes. In case of MaxWD 30 minutes, fixed cost for 0 routing cost unit is higher than those for other routing cost units. We can see that when routing cost is 0, more vehicles are needed to reduce other costs of waiting, delay, and excess ride which are larger than routing cost. In case of waiting, delay, and excess ride cost, they slightly increase as the routing cost unit increases.

Figure 5.12 shows the relation between the routing cost unit and the routing time generated in the model.

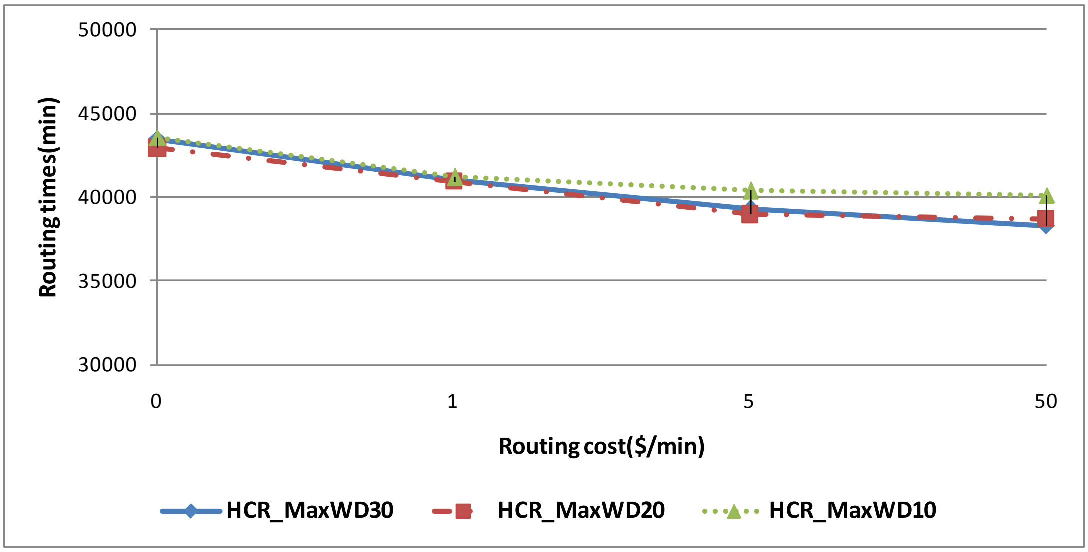


Figure 5.12 Sensitivity Analysis of Routing Cost and Routing Time

5.5.3 Waiting cost

Waiting cost represents penalty cost of vehicles when vehicles arrive before earliest allowable time windows. We change the waiting cost unit from 0 to 50 with base value for other parameters. Table 5.18 shows the results of the sensitivity analysis for three cases for the waiting cost variation.

In Table 5.18, we can see that the waiting cost variation does not much affect the fixed cost for range from 0.5 to 5. For example, in case of MaxWD 30minutes, when the waiting cost unit is 0.5, 242 routes are made and when the waiting cost unit is 5, 243 routes are made. After that, the fixed cost increases. The routing cost, delay cost, and excess ride cost for each case increase when the waiting cost unit increases.

Table 5.18 Sensitivity Analysis of Waiting Cost

Problems	Waiting cost unit	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost	Waiting time (min)
HCR MaxWD 30	0	97434	52200	40179	0	526	4529	3197
	0.5	95759	48400	41035	1194	669	4461	2388
	5	99631	48600	41723	2720	1539	5049	544
	50	115893	50600	42371	15000	2825	5097	300
HCR MaxWD 20	0	97434	52200	40179	0	526	4529	3197
	0.5	99962	53200	40889	640	407	4826	1280
	5	101391	49800	42151	3235	1300	4905	647
	50	113085	51200	42044	12600	2226	5015	252
HCR MaxWD 10	0	105048	58200	41323	0	908	4617	1209
	0.5	108830	61600	41185	363	1047	4635	726
	5	110209	60200	42366	1320	1409	4914	264
	50	124732	55400	42064	20850	1417	5002	417

Figure 5.13 shows that the relations between the waiting time unit and the waiting time generated in the model. We can recognize that the waiting time is reduced when the waiting cost unit increases. In case of MaxWD 10 minutes, the degree of variation of delay time is smaller than those of MaxWD 30 and 20 minutes. With the base waiting cost value, 0.5, 2388 minutes are spent for the case of MaxWD 30 minutes, 1280 minutes for MaxWD 20 minutes, and 726 minutes for MaxWD 10 minutes. For waiting time unit, 50, the waiting time is reduced to 300, 252, and 417 minutes. This result shows that we can control the early arrival at the demand node with the waiting time cost and the proposed model works well for the role as expected.

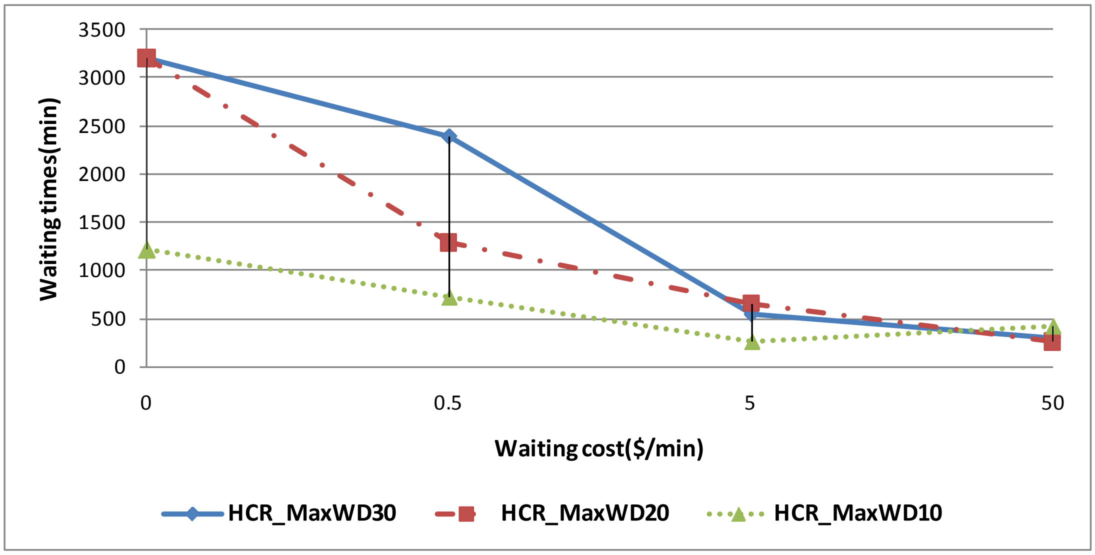


Figure 5.13 Sensitivity Analysis of Waiting Cost and Waiting Time

Figure 5.14 shows the trend of the total cost minus waiting cost as the waiting cost unit increases. For three cases, total cost except waiting cost does not much change in the range from 0.5 to 5 of waiting cost unit.

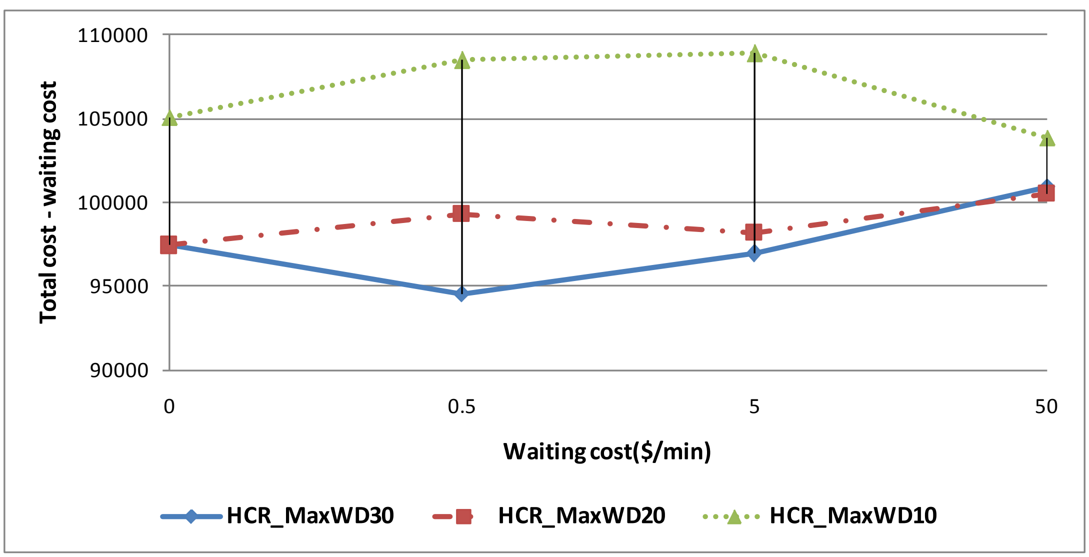


Figure 5.14 Sensitivity Analysis of Waiting Cost

5.5.4 Delay cost

Delay cost and excess ride cost represent the customer inconvenience cost by breaking time windows. We change the delay cost unit from 0 to 50 with base value for other parameters.

Table 5.19 shows the results of the sensitivity analysis for three cases for the delay cost variation. We can see that the fixed cost and routing cost increases slowly according to increasing of the delay cost unit, but not so sensitively.

The waiting cost increases as the delay cost unit increases. For example, in case of MaxWD 30 minutes, when the delay cost unit is 0.5, the waiting cost is 1194 and when the delay cost unit is 50, the waiting cost is 1694. In case of MaxWD 20 minutes and 10 minutes, when the delay cost unit is 0.5, the waiting cost is 640 and 363. However, when the delay cost unit is 50, the waiting cost increases to 1176 and 509.

Table 5.19 Sensitivity Analysis of Delay Cost

Problem	Delay cost unit	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost	Delay time (min)
HCR MaxWD 30	0	96001	51000	39912	769	0	4321	5583
	0.5	95759	48400	41035	1194	669	4461	1337
	5	112317	52800	41278	1325	1235	4564	2470
	50	141028	52200	41619	1694	40850	4665	817
HCR MaxWD 20	0	94432	49800	39524	614	0	4495	5102
	0.5	99962	53200	40889	640	407	4826	814
	5	102094	52400	41394	922	2710	4669	542
	50	126933	53600	41555	1176	26000	4603	520
HCR MaxWD 10	0	101286	56000	40504	300	0	4483	2582
	0.5	108830	61600	41185	363	1047	4635	2904
	5	118362	62000	42921	514	8105	4822	1621
	50	207068	60400	42361	509	98950	4849	1979

As the delay cost unit increases, the delay time decreases for all three cases. In case of MaxWD 10 minutes, the degree of variation of delay time is smaller than those of cases of MaxWD 30 and 20 minutes. With the base delay cost value, 0.5, 1337 minutes are spent for the case of MaxWD 30 minutes, 814 minutes for MaxWD 20 minutes, and 2904 minutes for MaxWD 10 minutes. For the delay time unit, 50, the delay time reduced to 817, 520, and 1979 minutes. Therefore, the delay time also can be controlled by the delay cost unit like the waiting time. Figure 5.15 shows that the relation between the delay cost unit and the delay time generated in the model.

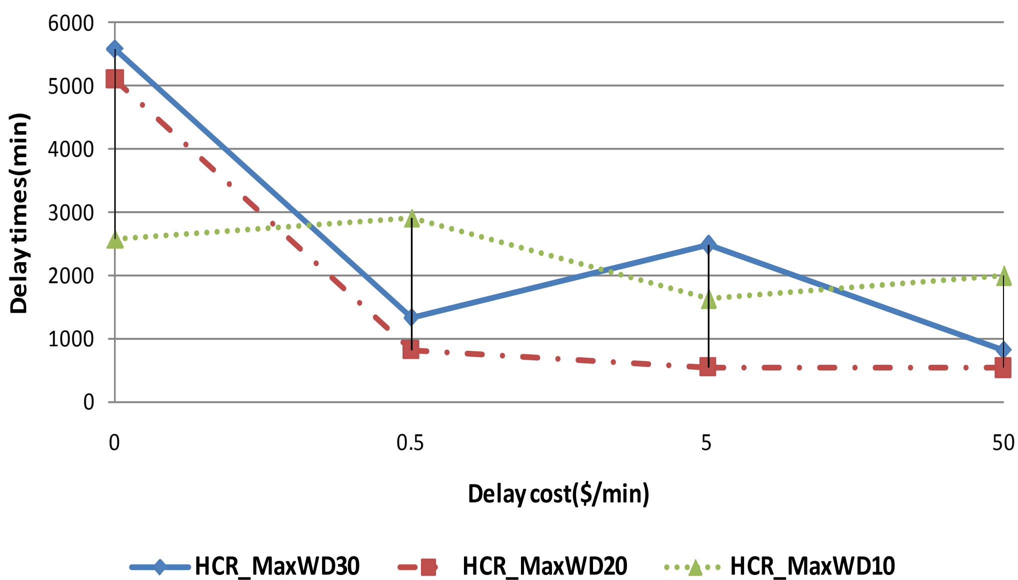


Figure 5.15 Sensitivity Analysis of Delay Cost and Delay Time

Figure 5.16 shows the trend of the total cost minus the delay cost as the unit delay cost increases. There is not much change from 0.5 to 50 of delay cost unit.

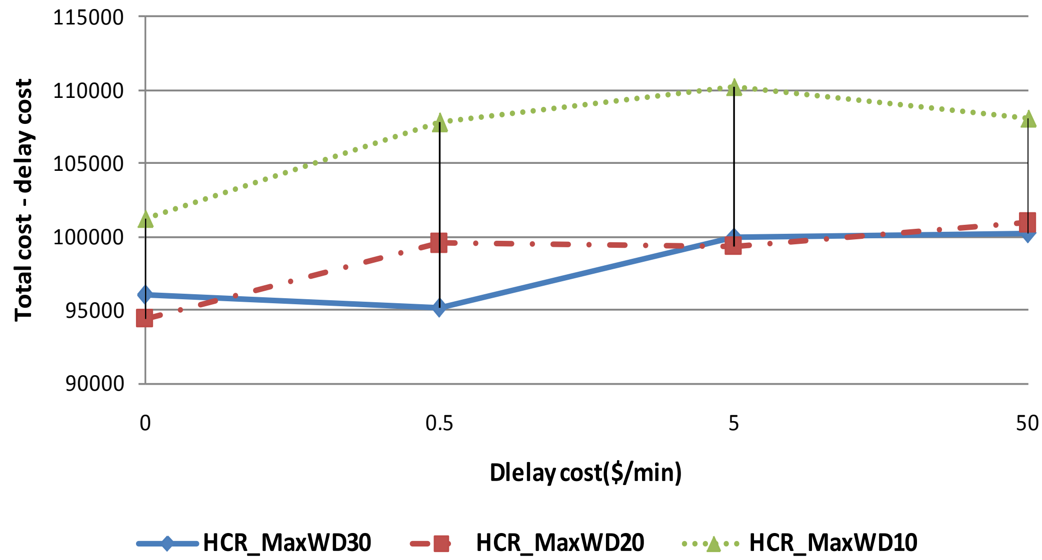


Figure 5.16 Sensitivity Analysis of Delay Cost

5.5.5 Excess ride cost

In this section, we describe the sensitivity analysis with respect to the excess ride cost. Customers' excess ride time is used as a proxy for bad customer service. We change the excess ride cost unit from 0 to 50 with base value for other parameters.

Table 5.20 shows the results of the sensitivity analysis for three cases for the excess ride cost variation. Fixed cost does not change much as excess ride cost unit increase, but routing cost increase slowly as the excess ride cost increases for all three cases. For example, in case of MaxWD 30 minutes, when the excess ride cost unit is base value, 0.5, the routing cost is 41035 and when the excess ride cost unit is 50, the routing cost is 42629. Also waiting cost and delay cost increase as the excess ride cost increases except when excess ride cost unit is 50.

Table 5.20 Sensitivity Analysis of Excess Ride Cost

Problem	Excess ride cost unit	Total cost	Fixed cost	Routing cost	Waiting cost	Delay cost	Excess ride cost	Excess ride time (min)
HCR MaxWD 30	0	93759	51800	40336	1063	560	0	9864
	0.5	95759	48400	41035	1194	669	4461	8922
	5	132599	52200	41949	1403	837	36210	7242
	50	473144	49200	42629	1408	4807	375100	7502
HCR MaxWD 20	0	95890	54200	40235	746	709	0	9981
	0.5	99962	53200	40889	640	407	4826	9652
	5	134600	52000	42114	1417	999	38070	7614
	50	460647	51600	42454	1007	2337	363250	7265
HCR MaxWD 10	0	102962	61200	40904	261	598	0	9014
	0.5	108830	61600	41185	363	1047	4635	9269
	5	144694	58000	42659	727	1263	42045	8409
	50	510797	59200	43382	444	771	407000	8140

In Figure 5.17, we can see that the relation between the excess ride cost unit and the excess ride time generated in the model. As excess ride cost increases, excess ride time decreases. In case of MaxWD 10 minutes, the degree of variation of delay time is smaller than those of cases of MaxWD 30 and 20 minutes.

Figure 5.18 shows the trend of the total cost minus the excess ride cost as the excess ride cost unit increases. Total cost except the excess ride cost is stable for the variation of the excess ride cost unit.

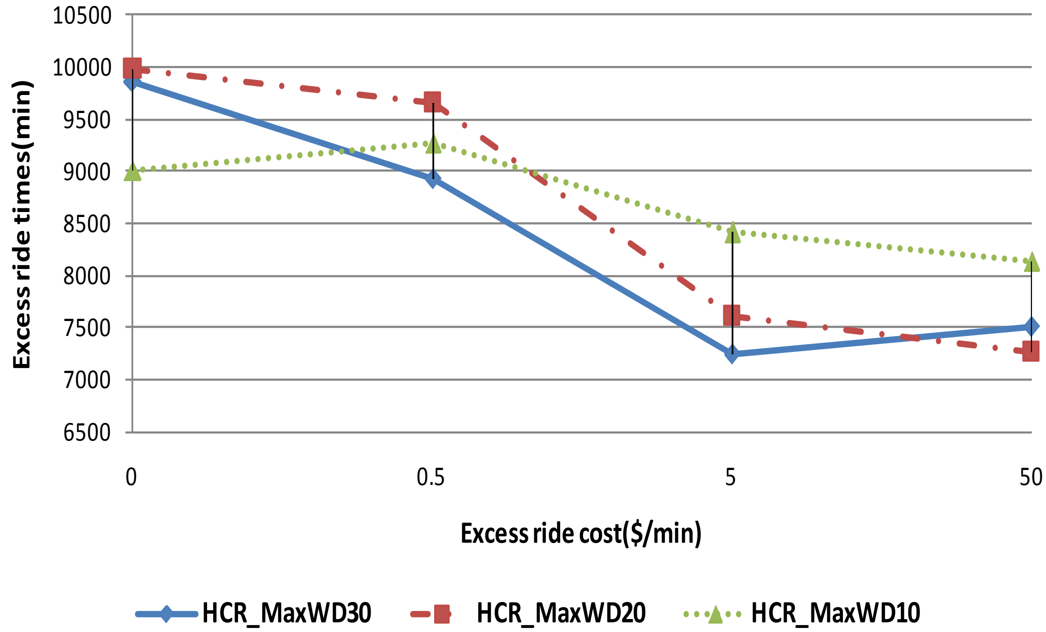


Figure 5.17 Sensitivity Analysis of Excess Ride Cost and Excess Ride Times

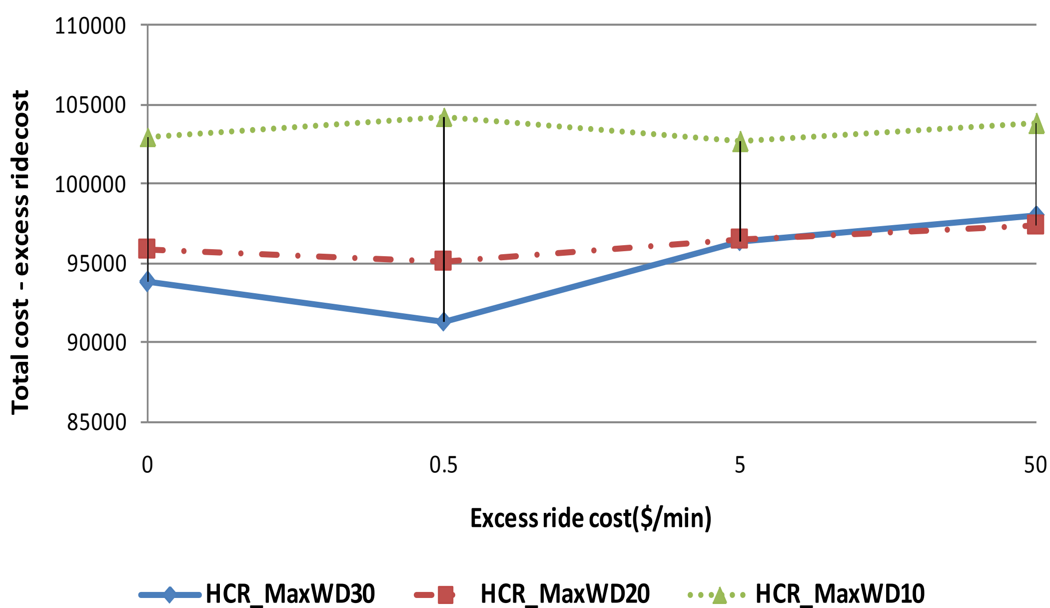


Figure 5.18 Sensitivity Analysis of Excess Ride Cost

5.6 Summary

In this chapter, a case study for real world large-scale static DARP was presented. For the case study, Maryland Transit Administration (MTA)'s real operation of Dial-a-ride service was introduced and compared with the results of developed heuristic.

The whole problem was decomposed into 5 time slots problem to solve it in reasonable time. The 5 time slots were made based on the distributions of request times of demands. After comparing the three heuristic in a preliminary test, HCR was selected for solving the large-scale DARP. For comparison of results of HCR with MTA's operation, 4604 demands of total 4726 demands were scheduled and routed. Since link travel times which had been used in MTA's operation were not available to us, two scenarios were introduced. For the first scenario, we assumed that actual link speeds are slower by 25% compared to the original link speeds which are set based on speed limits of links. In this case, the value of Speed Factor is 0.75. For the second scenario, we assumed that actual link speeds are equal to the original link speeds which are set. In this case, the value of Speed Factor is 1.

We analyzed the total cost by varying waiting cost unit, delay cost unit, and excess ride cost unit in HCR. The objective function values from HCR were better than those from MTA's operation for all cases when waiting cost, delay cost, and excess ride cost were not included in the objective function values of HCR. When waiting cost, delay cost, and excess ride cost were included in the objective function values of HCR, the objective function values from HCR are better than those from MTA's operation except in 4 cases 8, 16, 23, and 24.

A sensitivity analysis with respect to the parameters that are used in this model was performed. These parameters included the fixed costs, the routing costs, the waiting cost, the delay cost, and the excess ride cost. The results indicated that the proposed model performed as expected with respect to changes in these parameters.

Chapter 6: Heuristic Algorithm for the Real-Time DARP

There are two modes of Dial-a-Ride service. In the static mode, all requests are known in advance by reservations on one day before the service day. Also there are cancellations in advance. In the dynamic mode, part of the requests is dynamically generated. And these dynamic demands need to be inserted into the routes that are made in static problem and scheduled in real time. Also, other dynamic events such as customer's no show, cancellation in a day, and accidents in a network which may happen should be considered together for scheduling and routing both static and dynamic demands. At each time interval, routes and demands are updated using real time travel times. It is necessary to develop a heuristic method for responding to dynamic events in a short time and updating scheduling and routing of real-time DARP.

A vehicle is not to be diverted from its immediate destination for a new customer. However, diversion is allowed after the first stop, because the location of that stop is known to the dispatcher.

In this chapter, online heuristic algorithm for the real-time DARP is presented and its performance is tested on several cases and the results of cases are compared with each other. For this work, a simulation framework is made based on MTA's operation and scheduling and routing plan from static problem.

6.1 Dynamic Events

In reality, there may be many dynamic events in Dial-a-ride service. Dynamic events that are considered in this research are similar to those by Xiang et al. (2008).

Breakdown event is not considered in this research since breakdown is very rare in real situations. And, accident event is newly introduced. The dynamic events which are considered in this research are summarized according to their priority as follows:

(1) No-shows: some customers may not show up.

(2) Accidents: there may be accidents in a network.

(3) Cancellations: some customers may cancel their requests on the service day.

(4) New requests: new requests may arrive

(5) Travel times: Link speed varies according to the time of a day.

In principle, the earlier the customers make the trip requests, the more flexibility a service planner can have to schedule the requests and the more efficiency a service can have. Therefore, it is assumed that a new request should arrive at the dispatch center one hour before their desired service time. Also, as mentioned earlier, a minimum of two hours advanced notice is required for cancellation.

Any event may arrive within each time interval in a queue. In the event queue, events are sorted in ascending order with respect to their priorities. In the same kind of events, first-come first-serve is applied.

6.2 Online Heuristic Algorithm

As an on-line heuristic algorithm, insertion-based heuristic is applied to real-time DARP since an insertion-based heuristic is computationally efficient and it could be easily adapted for real-time DARP. The on-line insertion-based heuristic can be described as follows:

Step 0: time t=0 and time interval i =0. Set time interval length. Input scheduling and routing plan from static problem based on reserved demands and expected link travel times.

Step 1: If the current time is time interval, go to step 2. Otherwise, t= t+1.

Step 2: Update the status of routes and customer.

Step 3: If there is any dynamic events, go to Step 4. Otherwise, go to Step 5.

Step 4: Update scheduling and routing considering dynamic events. Also, routes are improved through the operator of combining vehicles.

Step 5: Update real-time travel times. i =i+1.

Step 6: If the current time is the end of service time, stop. Otherwise, t= t+1 and go to Step 1.

Figure 6.1 show the framework of on-line heuristic developed for real-time DARP.

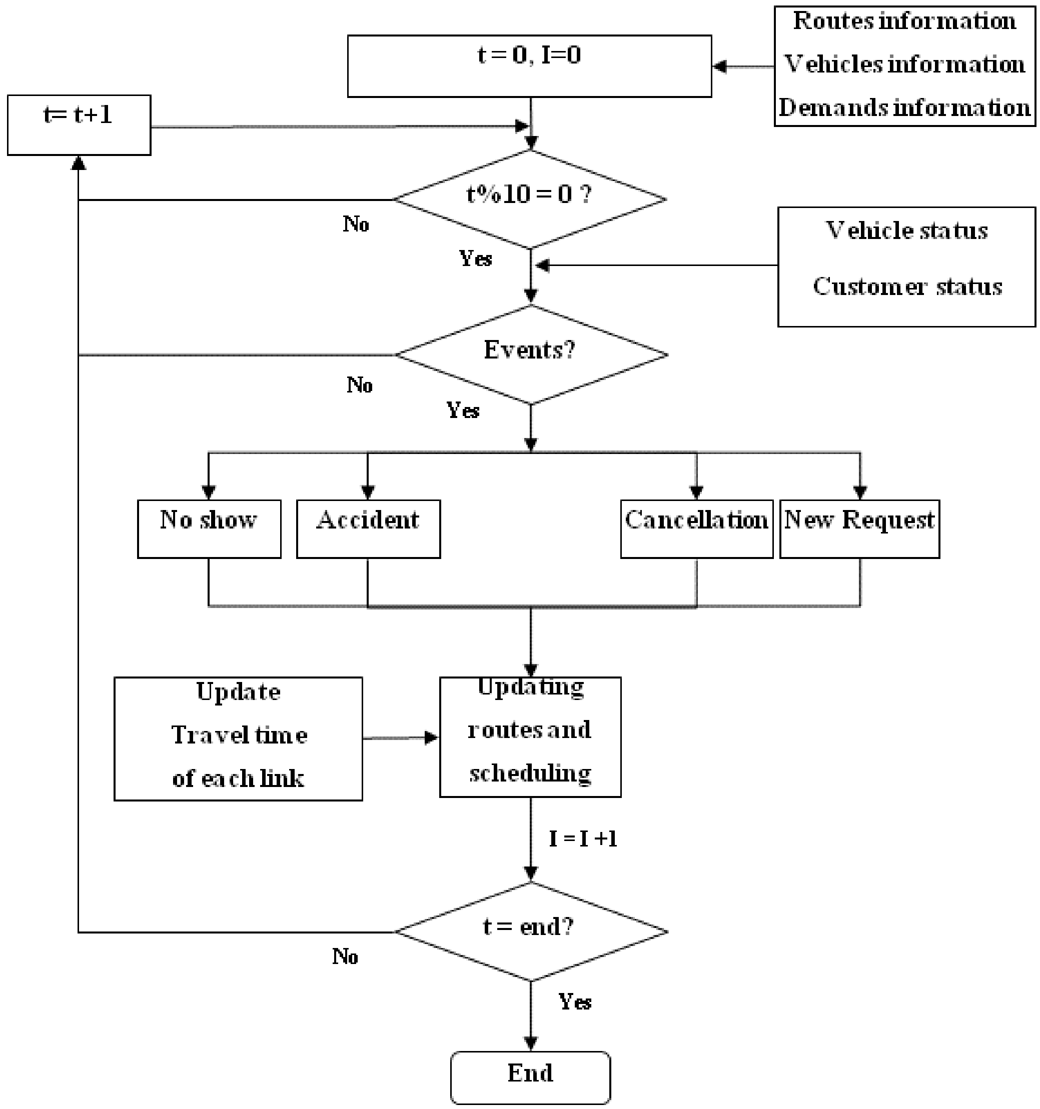


Figure 6.1 The framework of on-line heuristic for real-time DARP

6.2.1 Updating route and customer status

It is necessary to check the status of routes and customer at each time interval before considering dynamic events. The status of routes and customers are divided by 5 kinds of status, respectively as follows:

(1) **Route(vehicle) status**

1: At depot (not used still)

2: IDLE (waiting customer at stop)

3: At stop (on service)

4: On the way to next stop

5: Finished

(2) **Customer status**

1: Before service time

2: On board

3: Serviced

4: No show

5: Cancellation

6.2.2 No-show events

In practice, some customers may not show up without any notice and cancellation when a vehicle arrived at those customers' location. It is necessary to consider no show event and update the route that already include those no show customers. The steps of no-show event are described as follows:

1. Check whether there is any customer no show or not within last time interval.
2. If there is any no show, check the routing sequence of drop off demand of the absent customer in the route. If the next sequence demand of the absent customer's pickup demand in the route is not drop off demand of the absent customer, move drop off demand to the next sequence of the absent customer's pickup demand in the route. Also change node id of drop off demand to that of pickup demand.
3. Update service time, departure time, and load at pickup demand and drop off demand of the absent customer (service time =0, load =0, arrival time = departure time, waiting time at pickup demand (5minutes) for updating routes and scheduling)
4. Update arrival times of remnant demands in the route considering real time link speed. If the waiting time at the next demand of the absent customer in the route is greater than acceptable waiting and delay, it is allowed.
5. If there are some demands in the route that violate constraints, after removing them from the route, try to insert them to other routes that can accept them. Finally insert them to a route that has minimum additional cost. If there is no route that can accept a violating demand, then a new route is added.

6.2.3 Accident events

Accidents may often happen in a network such as an urban area. Accidents affect link speeds and travel times of those links where accidents happen and arrival times of demands in the route that travels using those links. Also, drivers' distraction may affect link speeds and travel times in the network. However, drivers' distraction is not considered in this research. Accident events are considered as follows:

1. Check whether there is any accident or not within last time interval.
2. If there is any accident, find the link on which it happened and the severity of the accident.
3. For those links on which accident happened, change link speed value as 0 during accident clear time. Therefore, travel time of any link that have accident is a large number. Accident clearing times vary according to the severity of accident as shown below:

   Light: not using the link for next 10minutes,

   Medium: not using the link for next 20 minutes,

   Heavy: not using the link for next 30 minutes.
4. Consider the new link speeds for finding shortest travel time and updating arrival times of demands at the time interval
5. After the accident is cleared, reset the link speed to the real travel speed at the new time interval.

6.2.4 Cancellation events

In practice, some customers may cancel their requests. This cancellation affects scheduling and routing of the route that includes customers who cancel. Cancellation events are considered as follows:

1. Check whether there is any cancellations or not within the last time interval
2. If there is any cancellation, remove customer's pickup and dropoff demand in the route.
3. Check whether the route originally scheduled for this customer is running or not at current time. If yes, go to 4, otherwise go to 5.
4. Update arrival times of demands for that route considering real time link speed. And for violating demands, remove them from originally scheduled route, try to insert them into a route that can accept them, and finally insert them into the route which has a minimum additional cost. If there is no route that can accept a violating demand, add a new route.
5. Calculate marginal time of the route and change the start time of route. And update arrival times of demands for the route considering real time link speed. For violating demands, remove them from originally scheduled route, try to insert them into a route that can accept them, and finally insert them into the route which has a minimum additional cost. If there is no route that can accept a violating demand, add a new route.

6.2.5 New request events

New requests may arrive between 6am and 6pm. These new requests cannot be rejected. New request events are considered as follows:

1. Check whether there is any new request or not within last time interval.
2. If there is any new request, try to insert the new request into the current routes (starting time of the route ≤ request time≤ ending time of the route)
   2.1. Find the route that has a minimum additional cost, and insert the new request into that route.
   2.2. Update arrival times considering real-time link speed and loads for each demand in this route.
3. If there is no route that can accept a new request, add a new route.

6.2.6. Improving by combining vehicles

After the routes have been updated considering dynamic events, the solution is improved by the operator of combining vehicles. There may be some routes which have a few demands. These routes can be combined into other routes that can accept customers from them route without violating constraints. Combining vehicles is necessary to reduce the objective function values since the fixed cost accounts for a significant part of the total cost. It is allowed for a vehicle to be combined into another vehicle that starts from a different depot.

The procedure of combining vehicles is described as follows:

1. Choose a route that has fewer demands than $min\_demands$ guaranteeing that each route has at least some demands.

2. Find other routes into which the chosen route can be combined and calculate saving cost for each case.
3. Combine the chosen route into the route which has maximum saving cost.

6.2.7 Updating arrival times

At this step, real-time travel times are considered for updating arrival times of demands in routes. After improving by combining vehicles, arrival times of demands in routes that is running at current time are updated considering real time link speeds. The procedure of updating arrival times is as follows.

1. Check real time link speeds.
2. For those routes that are running at current time, update arrival times of demands in routes.
3. For violating demands, remove them from originally scheduled routes, and try to insert them into a route that can accept them, and finally insert them into the route which has a minimum additional cost.
4. If there is no route that can accept a violating demand, add a new route.

6.3 Simulation Settings

6.3.1 Case study I

In real time situation, link flow speeds on the network within each time interval is available through various surveillance mechanisms in real time. If there is no real-time data available, average travel speeds based on historical data can be used. Given link flow speeds we can calculate the expected travel time between origin and destination at starting time using a time dependent shortest path algorithm. After some time passed, the network conditions may be different from the expected condition at time T. Therefore, there are gaps between the expected and actual link speed. The new condition of link speed is generated as indicated in Table 6.1. The heuristic is tested in simulation scenarios according to the fluctuation of link speeds.

Table 6.1 Gaps between the expected and the actual link speed

	Changing Gap at each link		
	0%	**25%**	**50%**
Cases	Case 1	Case 2	Case 3

Also we need to test the results according to the variation of time interval length. At each time interval, after all dynamic events that have arrived within last time interval are processed, a dispatcher communicates with drivers and guides routes. Therefore, the time interval must be long enough length for this entire processing time. For time interval length, 10 minutes and 20 minutes are tested for simulation as shown in Table 6.2.

Table 6.2 Time interval length

	1	2
Time interval length	10 minutes	20 minutes

6.3.2 Case study II

Usually a dynamic algorithm should be evaluated through the competitive analysis. As competitive analyses, the cost of dynamic case is compared with the cost of static case, where all requests are known in advance. The scenario for competitive analysis is shown in Table 6.3.

Table 6.3 Competitive analysis scenarios

	Considering new demands	
	Dynamic case	**Static case**
Time interval 10minutes	Scenario 1	Scenario 2

In this analysis, other dynamic events such as no-show, cancellation, and accidents in the dynamic case are not considered for comparing with static case.

To test the difference between dynamic case and static case according to the variation of percentage of new requests, four cases are used as shown in Table 6.4.

Table 6.4 The variations of new requests

Cases	Variations of new requests
Case 1	Total new requests = total reserved requests * 0.05 (5%)
Case 2	Total new requests = total reserved requests * 0.1 (10%)
Case 3	Total new requests = total reserved requests * 0.2 (20%)
Case 4	Total new requests = total reserved requests * 0.3 (30%)

6.3.3 Case study III

Similar to case study II, there may be extreme cases as shown in Table 6.5.

Table 6.5 The extreme cases for new requests

Cases	Variations of new requests
Case 1	Total new requests = total reserved requests * 0.0 (0%)
Case 2	Total new requests = total reserved requests * 1.0 (100%)

For the case 1, the dynamic case is the same as static case because there is no new request and other dynamic events such as no-shows, accidents, cancellations, and travel time fluctuations are not considered. Therefore, only case 2 is tested for comparing dynamic case with static case. In case 2, there are a total of 1,000 demands as reserved requests for dynamic case. These demands are randomly chosen from 4,726 demands.

6.3.4 Case study IV

It is assumed that new requests arrive between 6 AM to 6 PM. Also, it is possible for all new requests to arrive in first and second time intervals. Therefore, 2 scenarios are introduced as shown in Table 6.6. These two scenarios are tested for comparison of objective function values.

Table 6.6 The scenarios for arrivals of new requests

Scenarios	Arrivals of new requests
Scenario 1	New requests arrive between 6 am to 6 pm
Scenario 2	New requests arrive in first and second time intervals

Also, 3 cases are used for each scenario as shown in Table 6.7.

Table 6.7 The variations of new requests

Cases	Variations of new requests
Case 1	Total new requests = total reserved requests * 0.01 (1%)
Case 2	Total new requests = total reserved requests * 0.03 (3%)
Case 3	Total new requests = total reserved requests * 0.05 (5%)

6.3.5 Generating dynamic events

In this section, we explain how we generate random samples to emulate dynamic events such as no-show, accidents, cancellations, and new requests in order to develop a simulation model.

(1) Generating no-shows

Among total customers, 5% of total customers are no-shows. The parameters for generating no-shows are as follows:

No show rate = 0.05 (5% of total customer)

No show customer ID = uniform [1, total customer]

(2) Generating Accidents

Accident may happen at any link and accident locations are uniformly distributed over the network area. The probability of an accident happening time is based on the statistic about crashes by time of day in 2006 Maryland Traffic Safety Facts. The parameters for generating accidents are as follows:

Total accident = 20

Link ID of accident = uniform [1, total links]

Probability of an accident happening between 6am and 9am: 0.22

Probability of an accident happening between 9am and 12pm: 0.2

Probability of an accident happening between 12pm and 3pm: 0.25

Probability of an accident happening between 3pm and 6pm: 0.33

Probability of a light accident: 0.5,

Probability of a medium accident: 0.3

Probability of a heavy accident: 0.2

(3) Generating cancellations

Among total customers, 5% of total customers have cancellations of requests. The received times of cancellations are uniformly distributed between 0 and scheduled pickup time of customer – 60minute. The parameters for generating cancellations are as follows:

Cancellation rate: 0.05 (5% of total customers)

Customer Id of a cancellation = uniform [1, total customers]

Received time of a cancellation = uniform [0, scheduled time-60min]

(4) Generating new requests

There are new requests equivalent to 5% of reserved requests and new requests can arrive between 6am and 6pm. It is assumed that 30% of new requests occur between 6am and 9am, 20% between 9am and 12pm, 20% between 12pm and 3pm, and 30% between 3pm and 6pm, respectively. For the type of new requests, 50% of new request are ambulatory, 30% wheelchair, and 20% transferable wheelchair, respectively. The pickup and drop off locations are uniformly distributed in the network area. The parameters for generating cancellations are as follows:

New request rate: 0.05 (5% of reserved requests)

Probability of a request between 6am and 9am: 0.3

Probability of a request between 9am and 12am: 0.2

Probability of a request between 12am and 3pm: 0.2

Probability of a request between 3pm and 6pm: 0.3

Probability of type 1 request (ambulatory): 0.5

Probability of type 2 request (wheelchair): 0.3

Probability of type 3 request (transferable wheelchair): 0.2

Node of a pickup demand = uniform [1, total nodes]

Node of a drop off demand = uniform [1, total nodes]

The desired pickup time = uniform [received time+30min, 7pm]

The desired drop off time = uniform [received time+60min, 8pm]

6.3.6 Parameters

The width of time window is 30 minutes, Maximum route duration is 540 minutes, Maximum acceptable waiting and delay is 30 minutes, the fixed cost for used vehicle is \$200/vehicle, the travel cost is \$1/minute, the penalty cost for waiting time is \$0.5/minute, the penalty cost for delay time is \$0.5/minute, and the penalty cost for customers' excess ride time is \$0.5/minute. The service times at demand node are 2 minutes for a regular passenger, 4 minutes for a passenger using transferable wheelchair, and 6 minutes for a passenger using wheelchair, respectively.

The heuristic algorithm was coded in C++. All computations were carried out on a machine with 2.93GHZ Intel Core i7 CPU and 8GB memory in Windows 7 environment.

6.4 Results of Simulation

6.4.1 Results of case study I

Figure 6.2 shows the total costs of three cases for time interval 10 and 20 minutes. Total cost increases as gap between the expected and actual link speed increases and the total costs for time interval 20minutes are larger than those for time interval 10 minutes. The differences of total cost for 10 minutes time interval and 20 minutes time interval increases as gap between the expected and actual link speed increases. The differences of total costs according to the time interval length are not much when the gaps between the expected and actual link speed are between 0% and 25% while the differences of total costs according to the time interval length become larger as the gaps between the expected and actual link speed exceed 25%. Figure 6.3 shows that there is no difference of the number of used vehicles between case 1 and case 2 for time interval 10minutes and 20 minutes.

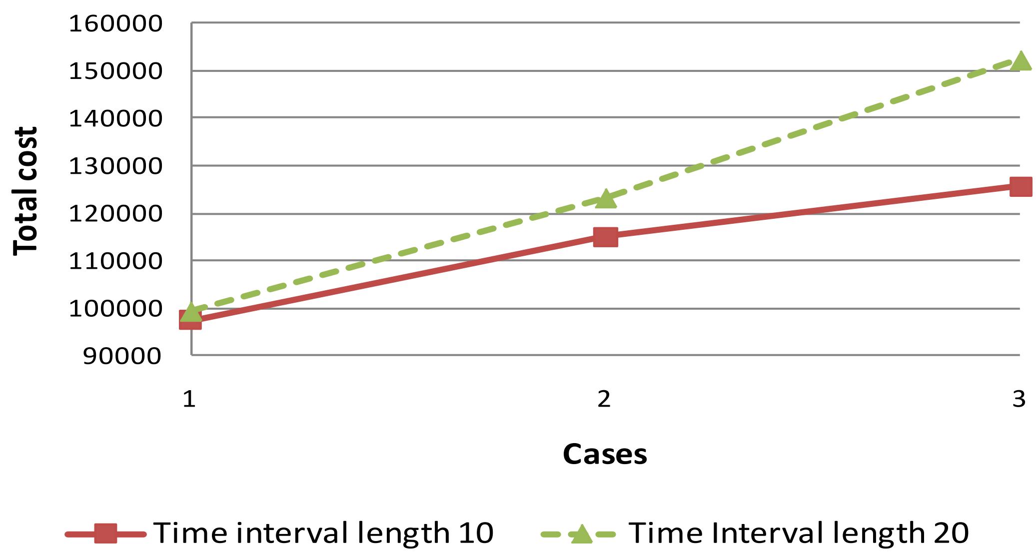


Figure 6.2 The total costs of three cases for time interval 10 and 20 minutes

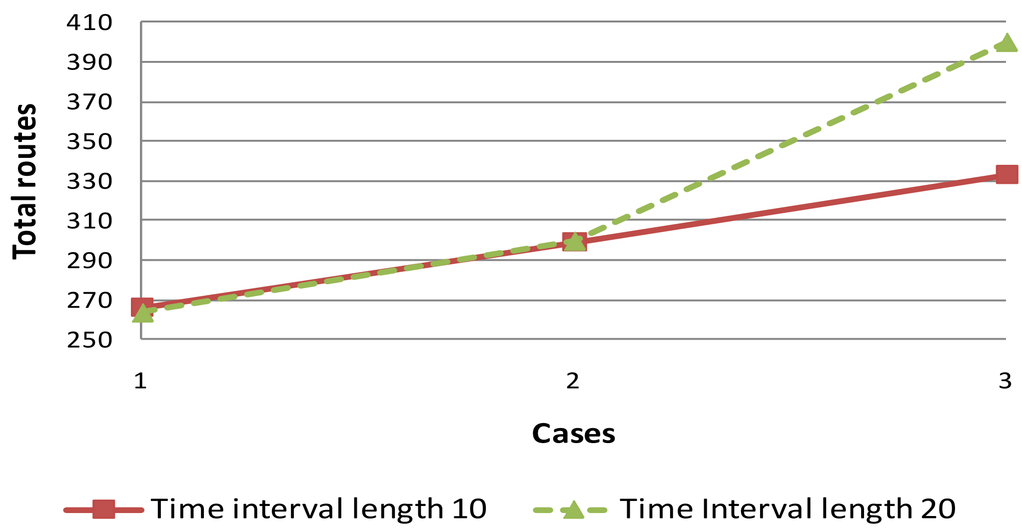


Figure 6.3 Total used routes of three cases for time interval 10 and 20 minutes

As time interval length is longer, there may be more dynamic events and uncertainty of link speeds, and more vehicles and routing times should be needed to satisfy constraints. We can see that the traffic conditions are very unstable and fluctuate seriously, shorter time intervals can save more cost.

Table 6.8 shows average processing time at time interval according to time interval length and gaps between the expected and the actual link speed. We can see that average processing time at time interval increases with time interval length and fluctuation of traffic condition. This occurs because the longer time interval, the more new insertion or removing-insertion operation may be involved in the computation.

Table 6.8 Average processing time at time interval

Gaps between the expected and the actual link speed	Time Interval Length	Average processing time at time interval	Δ (%)
0%	10 minutes	103.4 sec	-
	20 minutes	115.6 sec	11.8%
25%	10 minutes	130.4 sec	-
	20 minutes	156.3 sec	19.9%
50%	10 minutes	140.0 sec	-
	20 minutes	163.6 sec	16.9%

6.4.2 Results of case study II

For competitive analysis, the results of dynamic case are compared with the results of static case, where all requests are known in advance. Competitive ratio (CR) is calculated as follows:

$$\mathit{Competitive\ ratio} = \frac{\mathit{The\ cost\ of\ dynamic\ case} - \mathit{The\ cost\ of\ static\ case}}{\mathit{The\ cost\ of\ static\ case}} \times 100\%$$

Five replications are generated for each case to deal with the randomness of new requests, and the statistics reported are the average over five replications. Figure 6.4 shows the change of competitive ratio according to the variation of new requests. The percentages of new requests are 5% for case 1, 10% for case 2, 20% for case 3, and 30% for case 4, respectively. As percentage of new requests increases, the competitive ratio decreases. For example, competitive ratio is 1.6% for case 1, 1.3% for case 2, 0.4% for case 3, and -0.3% for case 4, respectively. Smaller CRs are obtained with more number of new requests. This result comes from that more new requests provide more chances to adjust scheduling and routing. Also this result gives the confidence that the on-line heuristic is very flexible to cope with new request.

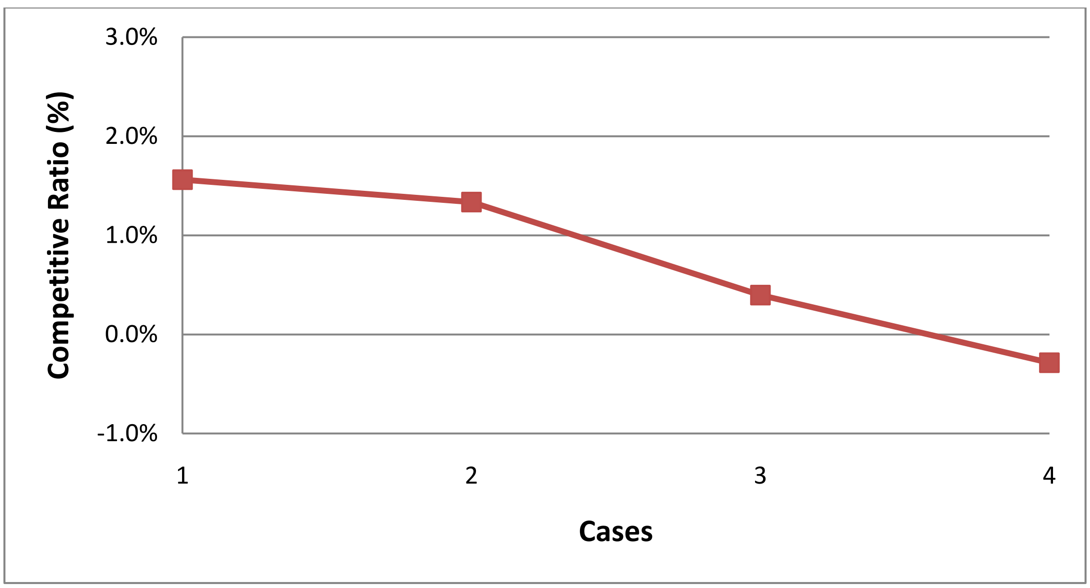


Figure 6.4 The competitive ratio according to the variation of new requests.

Figure 6.5 shows the changes of total cost according to the variation of new requests. For both static case and dynamic case, total cost increases as percentage of new requests increases. As percentages of new requests increase, the differences of total costs between the static case and the dynamic case decrease. For example, for case1, the difference of total cost is 1615.2, for case 2, 1478.6, for case 3, 492.6 and for case 4, -393. This result shows that the on-line heuristic well works with uncertainty of new requests as it is expected.

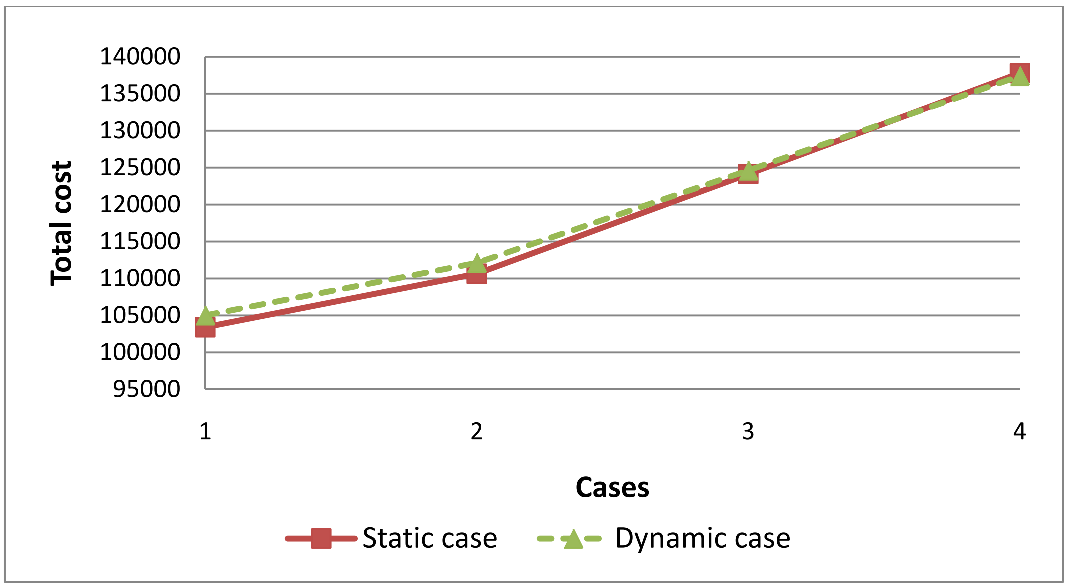


Figure 6.5 The total costs according to the variation of new requests

Figure 6.6 shows the changes of total used routes according to the variation of new requests. For both static case and dynamic case, total used vehicles increases as percentage of new requests increases. We can see that for dynamic case, the slope of change of total used routes is less steep than the slope of change of total used routes for static case. This result shows that the on-line heuristic well works with uncertainty of new requests as it is expected.

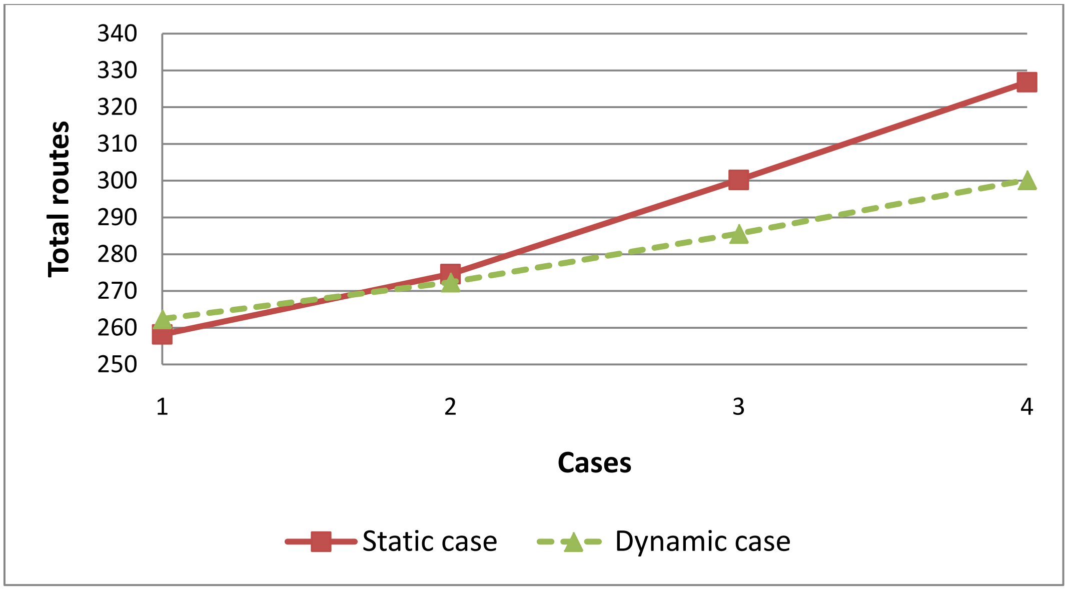


Figure 6.6 Total used routes according to the variation of new requests

6.4.3 Results of case study III

For the extreme case, competitive analysis of dynamic case to static case is performed. In this case, total new requests are equivalent to 100% of total reserved requests.

Five replications are generated for the extreme case to deal with the randomness of new requests. Figure 6.7 shows the competitive ratio for each replication. Objective function values of dynamic case are better than those of static case except for replication 4. Average of competitive ratio is -1.7%. Also this result gives the confidence that the on-line heuristic is very flexible to cope with new request.

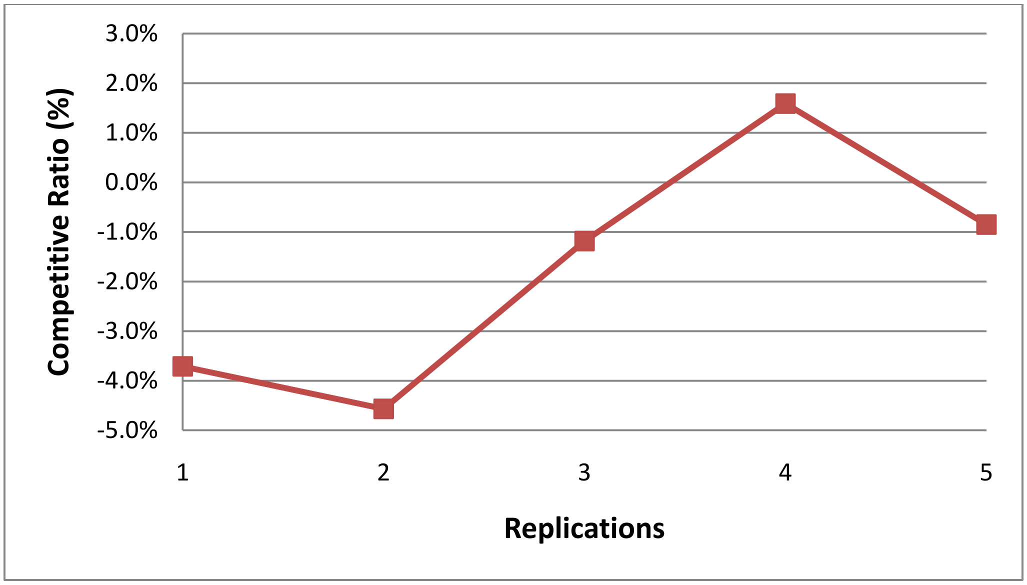


Figure 6.7 The competitive ratio for the extreme case

6.4.4 Results of case study IV

Two scenarios are tested for comparison of objective function values. In scenario 1, it is assumed that new requests arrive between 6 AM to 6 PM. In scenario 2, all new requests arrive in first and second time intervals.

Figure 6.8 shows the change of objective function values to the variation of new requests. The percentages of new requests are 1% for case 1, 3% for case 2, and 5% for case 3, respectively. As percentage of new requests increases, the difference of total costs between the scenario 1 and scenario 2 slightly increases. For example, the difference of total cost is 267.5 for case 1, -447 for case 2, and -657 for case 3, respectively. Smaller CRs are obtained with more number of new requests. Also, we can see that objective function values of scenario 2 are better than those of scenario 1 as percentage of new requests increases.

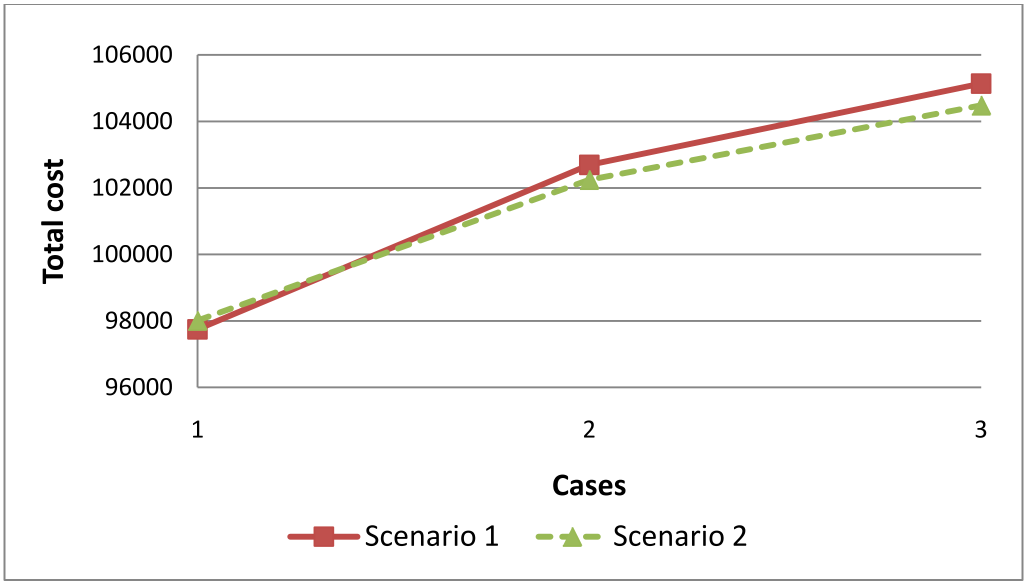


Figure 6.8 The total costs for two scenarios

6.5 Summary

In this chapter, online heuristic algorithm for the real-time DARP was presented and its performance was tested on several cases and the results of cases were compared with each other. For this work, simulation framework was made based on MTA's operation and scheduling and routing plan from static problem.

As an on-line heuristic algorithm, insertion-based heuristic was applied to real-time DARP since an insertion-based heuristic is computationally efficient and it could be easily adapted for real-time DARP. Also, for dynamic events, customer no-shows, accidents, cancellation, and new requests are considered in real-time DARP.

The simulation results were compared with each other according to the gaps between the expected and actual link speed and time interval length. In all cases, total cost increases as gap between the expected and actual link speed increases and the total costs for time interval 20minutes are larger than those for time interval 10 minutes. The differences of total costs according to the time interval length are not much when the gaps between the expected and actual link speed are between 0% and 25%. Also, the average processing time at time interval increases with time interval length.

For competitive analysis, the results of dynamic case are compared with the results of static case, where all requests are known in advance. As percentage of new requests increases, the competitive ratio decreases. For example, competitive ratio is 1.6% for case 1, 1.3% for case 2, 0.4% for case 3, and -0.3% for case 4, respectively.

For the extreme case, competitive analysis of dynamic case to static case is performed. In this case, total new requests are equivalent to 100% of total reserved requests. Objective function values of dynamic case are better than those of static case except for replication 4.

Two scenarios are tested for comparison of objective function values. In scenario 1, it is assumed that new requests arrive between 6AM to 6 PM. In scenario 2, all new requests arrive in first and second time intervals. Objective function values of scenario 2 are better than those of scenario 1 as percentage of new requests increases.

We conclude that the more unpredictable the demands are, the more cost can be saved by heuristic. Also, when the traffic conditions are very unstable and fluctuate seriously, shorter time interval can save more cost.

Chapter 7: Conclusions and Future Research

7.1 Summary and Conclusions

This research studies a static and real-time dial-a-ride problem with time varying travel times, soft time windows, and multiple depots. There is a potential for our model for application in real world as follows: Our model has a comprehensive objective function combining service provider's cost and customers' cost, and complex constraints in order to explain and reflect real world operation more reasonably than other models. Also, the mathematical formulation of this model is proposed. It is assumed that dial-a-ride service is provided from multiple depots in wide geographical areas like metropolitan cities. We can get a larger feasible solution set by loosening time constraints using soft time windows. In static DARP, the routing and scheduling are done considering time varying travel times in each link to reflect real situation and increase efficiency of service. Of course, travel times in each link are updated at every time interval in dynamic DARP. Our model is implemented to large real world problem and the results of model are compared with those of operation in real world.

In this research, a static DARP model considering time varying travel times, soft time windows, and multiple depots is formulated as a mixed integer programming. The objective of the formulation is to minimize the total cost that consists of the service provider's cost and the customers' inconvenience cost. The service provider's cost includes fixed costs of used vehicles, the routing costs, and vehicle waiting cost, while the customers' inconvenience cost includes customers' excess ride time cost and delayed service cost. In order to validate the model, several random small network problems are solved using commercial optimization package, CPLEX. The three heuristic algorithms based on sequential insertion, parallel insertion, and clustering first-routing second are proposed to solve static DARP within a reasonable time for implementation in a real-world situation. Also, the results of the three heuristic methods are compared with the results obtained from the exact solution by CPLEX to validate and evaluate the three heuristic algorithms. Computational results show that the three heuristic algorithms are superior compared to the exact algorithm in terms of the calculation time as the problem size (in terms of the number of demands) increases. As the number of customers exceeds 3 with service period of 10 time intervals, the calculation time of exact method increases exponentially and becomes unreasonable. The largest DARP problem size that could be solved in a reasonable time by exact method was 5 customers with service period of 10 time intervals. The gaps of the objective function values between the exact method and the three heuristic algorithms are less than 0.006%. For most of the cases, the three heuristic algorithms solved the problems within less than 0.2 second while the exact method could not solve the problem which has 5 customers and 15 time intervals. Among the three heuristic algorithms, the heuristic algorithm based on parallel insertion has a little better performance based on calculation times and objective function values.

Next, the three heuristic algorithms are tested on lager problems and compared with each other. Among the three heuristic algorithms, the heuristic algorithm based on sequential insertion is more efficient than other heuristic algorithms that are based on parallel insertion and clustering first-routing second. HSI performs better than HCR and HPI based on calculation times for most cases. For example, in case of 100 customers, HSI solved the problem 42.4% faster than HCR for MaxWD of 30 minutes, 27.5% for MaxWD of 20 minutes, and 34.8% for MaxWD of 10 minutes, respectively. Even in worst cases, the difference between the calculation times of HSI and HCR is less than -2.0%. Also considering the objective function values, HSI is better than HPI and HCR for most cases. For example, in case of 100 customers, the solution of HSI is 26.7% better than that of HCR for MaxWD of 30 minutes, 9.8% for MaxWD of 20 minutes, and 9.0% for MaxWD of 10 minutes, respectively. Even in worst cases, the difference between the objective function values of HSI and HCR is less than -5%.

For the case study, Maryland Transit Administration (MTA)'s real operation of Dial-a-ride service is introduced and compared with the results of developed heuristic. The objective function values from HCR are better than those from MTA's operation except in 4 cases 8, 16, 23, and 24. As SpeedFactor is 1.0, the results of heuristic are better than MTA's operating according to the variation of cost unit (0, 0.5, 3, and 5). But, as SpeedFactor is 0.75, the results of heuristic are better than MTA's operation until cost units increase by 3 and MTA's operation is better than the results of heuristic as cost units exceed 3.

A sensitivity analysis for the parameters that are used in this model was performed with respect to the fixed costs, the routing costs, the waiting cost, the delay cost, and the excess ride cost. The results indicated that the proposed model performed as expected with respect to changes in these parameters.

Also, the algorithm for real-time DARP considering dynamic events such as customer no shows, accidents, cancellations, and new requests is developed based on the static DARP. The algorithm is tested in simulation framework. In the simulation test, we compared the results of cases according to degree of gap between expected link speeds and real link speeds. For competitive analysis, the results of dynamic case are compared with the results of static case, where all requests are known in advance. As percentage of new requests increases, the competitive ratio decreases. The simulation test shows that the on-line heuristic method could save cost as the uncertainty in new requests is high.

The overall conclusions of this research can be outlined as followed.

1. The mathematical model for static DARP with multi depot, heterogeneous vehicles, soft time window considering time varying travel times was proposed

2. A heuristic methodology based on sequential insertion, parallel insertion and cluster first-routing second is proposed to solve this problem within a reasonable time for implementation in a real-world situation.

3. The heuristic algorithm based on clustering first-routing second was applied to real world DARP for case study. The results of heuristic is better than MTA's operating when the waiting cost, delay cost, and excess ride cost unit are between 0 to 3.

4. The on-line insertion-based heuristic was developed to solve real-time DARP considering dynamic events such as no shows, accidents, cancellations, and new requests.

5. Real-time DARP is tested on simulation framework based on real problem. The simulation test shows that the more unpredictable the demands are, the more cost can be saved by heuristic. Also, when the traffic conditions are very unstable and fluctuate seriously, shorter time interval can save more cost.

7.2 Future Research

Although many achievements have been made in this research, there are still many problems that are unsolved and are left for future research. Some of these are as follows:

1. In this model, there is maximum route duration constraint. Still, driver constraints such as break time, maximum working time, and maximum driving time are not considered. Since the operation of a system is supported by both fleet and crew, it is important to consider the corresponding crew scheduling problem. This model can be extended for comprehensive routing and scheduling system including crew scheduling.

2. For small problems, we can obtain optimal solutions from CPLEX, though the computational time might be long. For large size problems, CPLEX cannot provide optimal solutions. Therefore, we need to develop lower bound method for the DARP. The simple way of the Lower Bound solution procedure is to minimize the number of integer variables by LP relaxation. In this research, the lower bound was calculated from the reformulation of the problem that reduced the integer variables. We solved 5 customers and 15 time intervals by the lower bound and showed that heuristic algorithm performs well compared to the lower bounds for this

problem size. However, problems with 5 customers and 15 time intervals are still not large enough compared to the real world problems. Therefore, a more efficient method to produce lower bound for larger size problem is required.

3. In this research, at the construction phase, violation of maximum acceptable waiting and delay time and maximum excess ride time are allowed to get the initial solution within a short computational time. And, the quality of the initial solution may be low and many computational times are needed for improving the initial solution at the improvement phase. Therefore, it is necessary to develop a method to get a good initial solution within a short computational time.

4. In this research, the proposed heuristic algorithm was applied to real-world large DARP. And, the whole problem was decomposed into 5 time slots problem to solve it in reasonable time. Still there is a possibility to develop an efficient decomposition method for large problems.

5. In this research, it is assumed that vehicles have the same fixed unit costs, regardless of the vehicle's type. But, we have to consider that in reality fixed unit costs should be different according to the vehicle's type.

6. The proposed heuristic algorithm for DARP in this research can be implemented to pickup and delivery problems.

7. This model can be implemented with GIS technology. Recently, GIS technologies are widely used in logistics and transportation since ArcGIS tool can provide a user-friendly graphic interface and decision support system. We can develop a routing and scheduling system based on ArcGIS using heuristic algorithm developed in this research.

REFERENCES

Attanasio, A., Cordeau, J.F., Ghiani, G., and Laporte G. (2004), Parallel Tabu search heuristic for the dynamic multi-vehicle dial-a-ride problem, *Parallel Computing*, Vol. 30, pp. 377-387.

Baugh, J., Kakivaya, G., and Stone, J.R. (1998), Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. *Engineering Optimization*,. Vol. 30, pp. 91-123.

Beaudry, A., Laporte, G., Melo, T., and Nickel, S. (2010), Dynamic transportation of patients in hospitals, *OR Spectrum*, Vol. 32, No. 1, pp. 77-107.

Bodin, L., Golden, B., Assad, A., and Ball, M. (1983), Routing and Scheduling of vehicles and crews: the state of the art. *Computers and Operations Research*, Vol. 10, No. 2, pp. 63-211

Bodin, L.D. and Sexton, T. (1986), The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in Management Science*, 26, pp. 73-86.

Borndörfer, R., Grötschel, M., Klosteimer, F., and Küttner, C. (1999), Telebus berlin: vehicle scheduling in a dial-a-ride system. *Lecture notes in economics and mathematical systems*, vol. 471, pp.391-422

Chan S.Y.E. (2004), *Metaheuristics for solving the dial-a-ride problem*. Ph.D. Dissertation, Department of Operations Research, North Carolina State University, Raleigh, NC.

Cherkassky, B.V., Goldberg, A.V., and Radzik, T. (1993), *Shortest path algorithms: theory and experimental evaluation*. Technical Report 93-1480, Computer Science Department, Stanford Universtiy.

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

Cordeau, J.F. and Laporte, G. (2003a), The dial-a-ride problem: Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies, forthcoming*. Vol. 1, No. 2, pp. 89-101.

Cordeau, J.F. and Laporte, G. (2003b), 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. (2007), The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, Vol. 153, pp. 29-46.

Daganzo, C. (1978), Approximate analytic model of many-to-many demand responsive transportation systems. *Transportation Research*. Vol. 12, pp. 325-333.

De Paepe, W. E., Lenstra J. K., Sgall, J., and Leen Stougie, R.A.S. (2004), *INFORMS Journal on Computing*, Vol 16, No. 2, pp. 120-132.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M.M., and Soumis F. (2002), VRP with pickup and delivery, in Toth, P. and Vigo, D. (eds.) *The Vehicle Routing Problem.* SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp. 225-242.

Desrosiers, J., Dumas, Y., and Soumis, F. (1986), A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences* 6, pp. 301-325.

Desrosiers, J., Dumas, Y., and Soumis, F. (1988), The multiple vehicle dial-a-ride problem, in J.R. Daduna, A. Wren (eds.) Computer Aided Transit Scheduling, Proceedings of the Fourth International Workshop on Computer-Aided Scheduling of Public Transport, *Lecture Notes in Economics and Mathematical Systems* 308, Springer Verlag, pp. 15-27.

Desrosiers, J., Dumas, Y., Solomon, M.M., and Soumis, F. (1995), Time constrained routing and scheduling, in Ball, M.O., Magnant, T.L., Monma, C.L., and Nemhauser, G.L., eds. *Network Routing, Handbooks in Operation Research and Management Science,* Volume 8, INFORMS, Elsevier Science, pp. 35-13 9.

Dial R.B. (1995), Autonomous dial-a-ride transit introductory overview. *Transportation Research Part C*. Vol. 3, No. 5, pp. 261-275.

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.

Feuerstein, E. and Stougie, L. (2001), On-line server dial-a-ride problems. *Theoretical Computer Science*. Vol. 268, pp. 91-105.

Fu, L. (1996), *Real-time vehicle routing and scheduling in dynamic and stochastic traffic networks*, Ph. D. dissertation, University of Alberta, Edmonton, Alberta, Canada.

Fu, L., Rilett, L.R. (1998), Shortest path problems in traffic networks with dynamic and stochastic link travel times. *Transportation Research, Part B*: Methodological B 32 (7), pp. 499-515.

Fu, L. (1999a), On-line and off-line routing and scheduling of dial-a-ride Paratransit Vehicles. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 14, pp. 309-319

Fu, L. (1999b), Improving paratransit scheduling by accounting for dynamic and stochastic variations in travel time. *Transportation Research Record* 1666, Transportation Research Board, National Research Council, Washington, DC. pp. 74-81.

Fu, L. (2001), Simulation Model for evaluating intelligent paratransit systems. *Transportation Research Record* 1760, Transportation Research Board, National Research Council, Washington, DC. pp. 93-99.

Fu, L. (2002a), A simulation model for evaluating advanced dial-a-ride paratransit systems. . *Transportation Research Part A*, Vol. 36, pp. 291-307.

Fu, L. (2002b), Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. *Transportation Research Part B*. Vol. 36, pp. 485-506.

Fu, L. and Ishkhanov, G. (2004), Fleet size and mix optimization for paratransit services. *Transportation Research Record* 1884, Transportation Research Board, National Research Council, Washington, DC. pp. 39-46.

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

Goldberg, D.E. (1989), *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley.

Healy, P. and Moll, R. (1995), A new extension of local search applied to the dial-a-ride problem, *European Journal of Operations Research*, Vol. 83, pp. 83-104.

Ioachim, I., Desrosiers, J., Dumas, Y., and Solomon, M.M. (1995), A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science* 29, pp. 63-78.

Jaw, J.J. (1984), *Heuristic algorithms for multi-vehicle, advance-request dial-a-ride problems.* Ph.D. dissertation, Department of Aeronautics and Astronautics, MIT, Cambridge, MA.

Jaw, J., Odoni, A.R., Psaraftis, 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 B,* 20 (3), pp. 243-257.

Jorgensen, R.M., Larsen J. and Bergvinsdottir, K.B. (2007), Solving the dial-a-ride problem using genetic algorithms. *Journal of the operational research society*, Vol. 58, pp. 1321-1331.

Jung, S. (2000), *A genetic algorithm for the vehicle routing problem with time-dependent travel times*, Ph.D. dissertation, University of Maryland.

Jung, S. and Haghani, A. (2000), Genetic algorithm for a pickup and delivery problem with time windows. *Transportation Research Record* 1733, pp. 1-7.

Jung, S. and Haghani, A. (2001), Genetic algorithm for the time-dependent vehicle routing problem. *Transportation Research Record* 1771, pp. 164-171.

Kikuchi, S. (1984), Scheduling of demand-responsive transit vehicles. *Journal of Transportation Engineering* 110, pp. 511-520.

Kikuchi, S. and Rhee, J. (1989), Scheduling method for demand-responsive transportation system. *Journal of Transportation Engineering* 115, pp. 630-645.

Kikuchi, S. and Donnelly, R.A. (1992), Scheduling demand-responsive transportation vehicles using fuzzy-set theory. *Journal of Transportation Engineering* 118, pp.391-409.

Luo, Y. and Schonfeld, P. (2005), *Tour length of many-to-one dial-a-ride service*. 2005 TRB 84th Annual Meeting at Washington D.C.

Luo, Y. (2006), *Heuristics and performance metamodels for the dynamic dial-a-ride problem*. Ph. D. Dissertation, University of Maryland.

Luo, Y. and Schonfeld. P. (2007), A rejected-reinsertion heuristic for static Dial-A-Ride Problem. *Transportation Research Part B*, Vol. 41, No. 7, 2007, pp. 736-755.

Luo, Y. and Schonfeld. P. (2011), Online rejected-reinsertion heuristic for the dynamic multi-vehicle dial-a-ride problem. 2011 TRB 90th Annual Meeting at Washington D.C.

Luo, Y. and Schonfeld. P. (2011), Performance metamodels for dial-a-ride services with time constraints. 2011 TRB 90th Annual Meeting at Washington D.C.

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

Melachrinoudis, E., Ilhan, A. B., and Min, H. (2007), A dial-a-ride problem for client transportation in a health-care organization, *Computers and Operations Research*, Vol. 34, No. 3, pp. 742-759.

Mitrovic-Minic, S. (1998), *Pickup and delivery problem with time window: A Survey*.

SFU CMPT TR 1998-12, Simon Fraser University, Canada.

Mitrovic-Minic, S. (2001), *The dynamic pickup and delivery problem with time window.* Ph. D. Dissertation, Department of Computing Science, Simon Fraser University, Canada.

Parragh, S.N. (2009), *Ambulance Routing Problems with Rich Constraints and Multiple Objective*, Ph. D. Thesis, University of Vienna, Faculty of Business, Economics and Statistics.

Parragh, S.N., Doerner, K.F., Hartl, F.R., and Gandibleux, X. (2009), A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks,* Vol. 54 No. 4, pp. 227-242.

Parragh, S.N., Doerner, K.F., and Hartl, F.R. (2010), Variable neighborhood search for the dial-a-ride problem. *Computers and Operations Research*, Vol. 37, No. 6, pp. 1129-1138.

Potvin, J.Y. and Rousseau, J.M. (1992), Constraint-directed search for the advanced request dial-a-ride problem with service quality constraints, in O. Balci, R. Shrada, S.A. Zenios (eds.) *Computer Science and Operation Research: New Developments in Their Interfaces*, Pergamon Press, Oxford, England, pp. 457-474.

Psaraftis, H.N. (1980), A dynamic programming approach to the single-vehicle, many-to-many immediate request diala-ride problem. *Transportation Science* 14, pp. 130-154.

Psaraftis, H.N. (1983), An exact algorithm for the single-vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17, pp. 351-357.

Rekiek, B., Delchambre, A., and Saleh, H. A. (2006), Handicapped person transportation: an application of the grouping genetic algorithm. *Engineering Applications of Artificial Intelligence*, Vol. 19, No. 5, pp. 511-520.

Savelsbergh, M.W.P. and Solomon, M. (1995), The general pickup and delivery problem. *Transportation Science*, Vol. 29, No. 1, pp. 17-29.

Sexton, T. and Bodin, L. (1985a). Optimizing single vehicle many-to-many operation with desired delivery times: I. Scheduling. *Transportation Science* 19, pp. 378-410.

Sexton, T. and Bodin, L. (1985b). Optimizing single vehicle many-to-many operation with desired delivery times: II. Routing. *Transportation Science* 19, pp. 411-435.

Stein, D.M. (1978), Scheduling dial-a-ride transportation systems. *Transportation Science* 12, pp. 232-249.

Sung K., Bell, M.G.H, Seong M., and Park S. (2000), Shortest paths in a network with time dependent-flow speeds, *European Journal of Operational Research*, Vol. 121, pp. 32-39.

Swihart, Michael R. (1994), *A stochastic and dynamic model for the single vehicle dial-a-ride problem*, Master thesis, Purdue University.

Teodorovic, D. and Vukadinovic, K. (1998), *Traffic control and transport planning: A fuzzy sets and neural networks approach*. Kluwer, Boston.

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

Toth, P. and Vigo, D. (1996), Fast local search algorithms for the handicapped persons transportation problem. In: Osman, I.H., Kelly, J.P. (Eds.), *Meta-Heuristics: Theory and Applications*. Kluwer, Boston, pp. 677-690.

Toth, P. and Vigo, D. (1997), Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* 31, pp. 60-71.

Uchimura, K., Takahnashi, H., and Saitoh T. (2002), *IEEE Transactions on vehicular technology*. Vol. 51, No. 4, pp. 760-766;

Van der Bruggen, L.J.J., Lenstra, J.K., and Schuur P.C. (1993), Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transportation Science* Vol. 27, pp. 298-311.

Wilson, N.H.M., Sussman, J., Wong, H., and Higonnet, B. (1971), *Scheduling Algorihms for dial-a-ride systems*. Technical Report USL TR-70-13, Urban Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Wilson, N.H.M., Weissberg, R. W., Higonnet, B.T., and Hauser, J. (1975), *Advanced dial-a-ride algorithms research project: Interim report*. Technical Report R75-27, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Wilson, N.H.M. and Weissberg, H. (1976), *Advanced dial-a-ride algorithms research project: Final report.* Technical Report R76-20, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Wilson, N.H.M. and Colvin, N.H. (1977), *Computer Control of Rochester dial-a-ride system.* Technical Report R77-31, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Wong, K.I. and Bell, M. G. H. (2006), Solution of the dial-a-ride problem with multi-dimensional capacity constraints. *International Transactions in Operational Research* 13(3), pp. 195-208.

Xiang, Z., Chu, C., and Chen, H. (2006), A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints, *European Journal of Operational Research*, Vol. 174, 2006, pp. 1117-1139.

Xiang, Z., Chu, C., and Chen, H. (2008), The Study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research* 185(2), pp. 534-551.

Zhan F.B. (1997), Three fastest shortest path algorithm on real road networks: data structures and procedures, *Journal of Geographic Information and Decision Analysis*, Vol. 1, No. 1, pp. 69-82.

Americans with Disabilities Act of 1990, http://www.usdoj.gov/crt/ada/adahom1.htm

American Public Transportation Association, http://www.apta.com

Maryland Transit Administration, http://www. mtamaryland.com

Maryland Traffic Safety Facts 2006, Maryland State Highway Administration, Nov. 2007.

Public Transportation Fact Book 2011, American Public Transportation Association, April 2011.