METAHEURISTICS FOR THE WASTE COLLECTION VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

A thesis submitted for the degree of

Doctor of Philosophy

by

Aida Mauziah Benjamin

Department of Mathematical Sciences

Brunel University

April 2011

ABSTRACT

In this thesis there is a set of waste disposal facilities, a set of customers at which waste is collected and an unlimited number of homogeneous vehicles based at a single depot. Empty vehicles leave the depot and collect waste from customers, emptying themselves at the waste disposal facilities as and when necessary. Vehicles return to the depot empty. We take into consideration time windows associated with customers, disposal facilities and the depot. We also have a driver rest period. The problem is solved heuristically. A neighbour set is defined for each customer as the set of customers that are close, but with compatible time windows.

This thesis uses six different procedures to obtain initial solutions for the problem. Then, the initial solutions from these procedures are improved in terms of the distance travelled using our phase 1 and phase 2 procedures, whereas we reduce the number of vehicles used using our vehicle reduction (VR) procedure.

In a further attempt to improve the solutions three metaheuristic algorithms are presented, namely tabu search (TS), variable neighbourhood search (VNS) and variable neighbourhood tabu search (VNTS). Moreover, we present a modified disposal facility positioning (DFP), reverse order and change tracking procedures.

Using all these procedures presented in the thesis, four solution procedures are reported for the two benchmark problem sets, namely waste collection vehicle routing problems with time windows (VRPTW) and multi-depot vehicle routing problem with inter-depot routes (MDVRPI).

Our solutions for the waste collection VRPTW problems are compared with the solutions from Kim et al (2006), and our solutions for the MDVRPI problems are compared with Crevier et al (2007). Computational results for the waste collection VRPTW problems indicate that our algorithms produce better quality solutions than Kim et al (2006) in terms of both distance travelled and number of vehicles used. However for the MDVRPI problems, solutions from Crevier et al (2007) outperform our solutions.

ACKNOWLEDGEMENTS

Firstly I would like to express my gratitude to my supervisor, Professor John E Beasley for his guidance and understanding throughout my PhD. His valuable advice and comments during my PhD studies have helped me to finish this thesis and achieve this important goal in my professional career. Also, special thanks to my examiners, Professor Richard Eglese from Lancaster University and Dr. Cormac Lucas from Brunel University for their comments to improve the thesis.

Thanks to everyone in the Mathematical Sciences Department for their knowledge and assistance during my PhD. Working in such a friendly and pleasant environment has contributed to make this an enjoyable experience.

I also wish to acknowledge my indebtedness to my family and all my friends for their fulfilling support and understanding as well as their prayers.

Last but not least, to Ministry of Higher Education, Malaysia for sponsoring my study at Brunel University for three and a half years. Without the financial support from my government this PhD programme would have not been possible. My gratitude also goes to the Mathematical Sciences Department, Brunel University for paying some conferences and workshop expenses that I have attended during my PhD.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The vehicle routing problem (VRP) is one of the most challenging combinatorial optimisation problems. Since it was first proposed by Dantzig and Ramser in 1959 as the truck dispatching problem, the VRP has become a more and more interesting research area due to its wide applicability and economic importance in reducing operational costs in distribution systems.

In the literature, the basic VRP is comprised of a set of vehicles, customers and a depot. It can be defined as the problem of designing least cost routes for identical vehicles of known capacities, which run from a central depot to a set of geographically dispersed customers with non-negative demand. Each customer is to be fully serviced exactly once (typically by one vehicle). The total demand and the length of a route must not exceed the total capacity and the total distance travelled allowed for a vehicle. The vehicles will return to the depot after servicing customers who have been assigned to them.

1.1 VRP variants

In order to satisfy real-life VRP scenarios, more restrictions are usually involved in the problem such as multiple number of depots, different type of vehicles (homogeneous and heterogeneous), different types of customer's demand (deterministic or stochastic), road constraints (one way, prohibited route), types of operations (collection, delivery, and mixed) etc. Including these restrictions implies a significant increase in the complexity of the VRP problem. As a result, variants of the VRP have been introduced in the literature. A paper by Eksioglu, Vural and Reisman (2009) presents a taxonomic framework of VRP for the last three decades. We also address the reader to a VRP web site where the explanation of VRP variants including the VRP instances, solution techniques used to solve VRP as well as the best solutions found are updated on this web site, http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html?/Problem_Instances/instances.html

The VRP variant involved in this study is the Vehicle Routing Problem with Time Windows (VRPTW). This time window constraint restricts the times at which a customer is available to be served by a vehicle. It is usually expressed as a time frame for each customer. Typically if the vehicle arrives early at a customer, then it must wait until start of service is possible.

The VRPTW has been studied extensively in the literature (e.g. Park and Kim, 2010; Russell, Chiang and Zepeda, 2008; Fabri and Recht, 2006; Doerner et el.,2008). This is mainly due to applicability of time window constraints in real-world situations, such as the waste collection problem (WCP) studied in this thesis.

1.2 VRPTW in waste collection

In general a waste collection system involves the collection and transportation of solid waste to disposal facilities. This essential service is receiving increasing attention from many researchers due to its impact on the public concern for the environment and population growth, especially in urban areas. Because this service involves a very high operational cost, researchers are trying to reduce the cost by improving the routing of waste collection vehicles, finding the most suitable location of disposal facilities and the location of collection waste bins as well as minimizing the number of vehicles used. For example, a study by Simonetto and Borenstein (2007) tested a decision support system called SCOLDSS on a real life waste collection problem in Porto Alegre, Brazil. By using SCOLDSS, they stated that it is possible to obtain a mean reduction of 8.82% in the distance to be covered and a reduction of 17.89% in the weekly number of trips by the collection vehicles. This result is very significant to Municipal Department of Urban Cleaning (DMLU) because it can represent savings of around 10% of the DMLU annual budget for solid waste collection per year, considering the operational and maintenance costs.

This thesis considers a vehicle routing problem that arises when a set of customers have waste that must be collected by vehicles. In such situations it is common for the amount of waste to be such that vehicles become full during their working day and have time to visit a waste disposal facility to empty themselves before going on to visit more customers and collect more waste. As such multiple visits to waste disposal facilities may be made during the working day. This problem is a single period node routing problem and is often encountered in terms of waste collection from commercial customers.

In the literature, waste collection can be divided into three categories: residential, commercial and skip waste collection. Residential waste is found in front of the houses in small bins or garbage bags. The collection vehicles will collect all this waste along the streets which have been assigned to them. Therefore, this service is often solved by researchers as an arc routing problem where the exact location of every customer is not needed.

On the other hand the commercial and the skip collection problems are typically solved as node routing problems and the location of every customer is known. This is because the waste collection involved in these strategies is point-to-point collection. Commercial waste can be found at restaurants, retail outlets and apartments in containers. The vehicles will collect waste at collection points before going to the disposal facilities to be emptied.

For the skip problem, waste can be found at construction sites in large containers. The size of containers used in this collection is much larger than the containers used for commercial waste. In general, this problem involves the pickup of full containers, transport to the disposal facility for unloading (emptying) and returning the empty container to the site. Therefore in this problem, the vehicle can often only collect one container at a time. In the literature, this problem is also known as a roll-on roll-off problem. In terms of the total number of customers served residential waste collection problems typically serve more customers than the commercial and skip problems.

Essentially, the commercial waste collection problem focused in this thesis can be described as follows: the problem has an unlimited number of identical (homogeneous) vehicles based at a single depot. Vehicles start/end their routes at the depot empty. This problem involves multiple disposal facilities, so that decisions must be made not only as to when a vehicle should empty itself at a disposal facility, but also which disposal facility it should use. This problem also considers time windows, one associated with each customer that governs when waste can be collected from that customer; another associated with each disposal facility that governs when that facility is open; another associated with the depot that governs when it is open to dispatch/receive vehicles. Each vehicle has a driver rest period (associated with a lunch break during the working day), and a maximum amount of work it can do during the day (both in terms of the total amount of waste collected and the total number of customers dealt with). This is a single period problem, so it is not a periodic routing problem where one has to design routes over multiple periods.

1.3 Thesis structure

This thesis is organised as follows. In Chapter 2 examples of heuristic techniques that have been used to solve the VRP for deliveries are presented. Next, the relevant literature on the waste collection problem, particularly the skip problems and non-skip problems which deal with node routing are reviewed. Finally, a discussion of this chapter is presented.

Chapter 3 is divided into five parts. First, the notation for the problem is presented. Second, the neighbour sets for each customer are defined. The neighbour set for a customer is those customers that are closest to it, but with compatible time windows. Third, six initial solution procedures implemented on two benchmark problem sets used in the thesis are presented. Fourth, initial solutions obtained from the procedures are reported. Finally, a summary of this chapter is presented.

Chapter 4 presents our procedure to evaluate a given route, which involves inserting into the route (if necessary) disposal facility visits. Next, procedures to improve a solution, both in terms of the distance travelled and in terms of the number of vehicles used are presented. Then, computational results for both procedures tested on two benchmark problem sets are reported. Finally, a summary of this chapter is presented.

In Chapter 5 general descriptions of tabu search (TS) and variable neighbourhood search (VNS) are presented. Next, our three metaheuristic algorithms, namely TS, VNS and variable neighbourhood tabu search (VNTS) are presented. VNTS is a metaheuristic algorithm where the variable neighbourhood is searched via tabu search. Then, computational results for our three metaheuristics tested on two benchmark problem sets are reported. Finally, a summary of this chapter is presented.

Chapter 6 presents another route evaluation procedure, namely disposal facility positioning (DFP). Next, two procedures namely, change tracking and reverse order are presented. Then, computational results for these procedures tested on two benchmark problem sets are reported. Finally, a summary of this chapter is presented.

In Chapter 7 a summary of every chapter in the thesis is presented. Then, the contribution of the thesis to knowledge and suggestions for further research are discussed.

CHAPTER 2

LITERATURE REVIEW

In the first part of this chapter, examples of heuristic techniques that have been used to solve the VRP for deliveries are presented. Next, some literature on VRP for collection is reviewed. This includes previous work dealing with waste collection such as arc routing, as well as node routing, particularly skip problems and non-skip problems.

2.1 Heuristics for delivery problems

Basically the VRP for delivery problems can be defined as delivering goods to a number of customers who have placed orders for a certain quantity of these goods from a central depot. Due to some constraints such as load, distance and time, a single vehicle may not be able to serve all the customers. The problem then is to determine the number of vehicles needed to serve the customers as well as the routes that will minimize the total distance travelled by the vehicles. Many heuristics have been introduced in the literature for searching for good solutions to the problem.

For instance the savings algorithm of Clarke and Wright (1964), the sweep algorithm of Gillett and Miller (1974), the cluster-first, route-second heuristic of Fisher and Jaikumar (1981), the path scanning heuristic of Golden, De Armon and Bakers (1983), and the route-first, cluster-second heuristic of Beasley (1983). A detailed survey of major developments in heuristics as well as exact algorithms for solving the VRP can be found in the recent paper by Laporte (2009), but this is a still growing research area.

Many papers in the literature deal with academic test problems. Examples of real-life delivery problems that catch the attention of researchers are newspaper delivery (e.g. Boonkleaw, Suthikarnnarunai and Srinon, 2009; Russell, Chiang and Zepeda, 2008; Song, Lee and Kim, 2002; Ree and Yoon, 1996), food delivery (e.g. Chen, Hsueh and Chang, 2009; Rusdiansyah and Tsao, 2005; Faulin, 2003), as well as postal and parcel delivery (e.g. Bruns, Klose and Stahly, 2000; Novaes and Graciolli, 1999).

2.2 VRP for collection

Essentially, the VRP for collection is dealing with the same type of constraints as in a delivery problem when constructing vehicle routes. Thus, this problem also attempts to determine the number of vehicles needed to serve the customers as well as the routes that will minimize the total distance travelled by the vehicles. However, the vehicle for the collection problem is empty when it starts from the depot, whereas the vehicle for the delivery problem begins its route loaded with customers' goods that need to be delivered. In the collection problem vehicles will collect goods from a set of customers and return to the depot at the end of the working day.

Some applications of collection problems that can be found in the literature are cash collection (e.g. Lambert, Laporte and Louveaux, 1993), collection of raw materials for multi-product dehydration plants (e.g. Tarantilis and Kiranoudis, 2001a; Tarantilis and Kiranoudis, 2001b), and milk collection (e.g. Caramia and Guerriero, 2010).

2.3 Waste collection VRP

Dealing with a waste collection problem is different from the collection problem as discussed in the previous section. There is an additional constraint that needs to be considered in solving this problem. Instead of returning to the depot to unload the collected goods, in a waste collection problem vehicles need to be emptied at a disposal facility before continuing collecting waste from other customers. Thus, multiple trips to the disposal facility occur in this problem before the vehicles return to the depot empty, with zero waste. A complication in the problem arises when more than one disposal facilities are involved. Here one needs to determine the right time to empty the vehicles as well as to choose the best disposal facility they should go to so that the total distance can be minimized. For example it may not be optimal to allow the collection vehicle to become full before visiting a disposal facility.

Increasing quantities of solid waste due to population growth, especially in urban areas, and the high cost of its collection are the main reasons why this problem has become an important research area in the field of vehicle routing. In the next two sections, previous work dealing with waste collection as arc routing problems and as node routing problems are reviewed.

2.4 Arc routing problems

Due to the large number of residential waste locations that have to be collected from this collection problem is often dealt with as an arc routing problem, whereas the collection of commercial waste is dealt with as a node routing problem. In this section some of the previous work dealing with arc routing problems for waste collection is reviewed.

Chang, Lu and Wei (1997) applied a revised multiobjective mixed-integer programming model (MIP) for analyzing the optimal path in a waste collection network within a geographic information system (GIS) environment. They demonstrated the integration of the MIP and the GIS for the management of solid waste in Kaohsiung, Taiwan. Computational results of three cases particularly the current scenario, proposed management scenario (without resource equity consideration) and modified management scenario (with resource equity requirement) are reported. Both the proposed and the modified management scenarios show solutions of similar quality. On average both scenarios show a reduction of around 36.46% in distance travelled and 6.03% in collection time compared to the current scenario.

Mourao and Almeida (2000) solved a capacitated arc routing problem (CARP) with side constraints for a refuse collection VRP using two lower-bounding methods to incorporate the side constraints and a three-phase heuristic to generate a near optimal solution from the solution obtained with the first lower-bounding method. Then, the feasible solution from the heuristic represents an upper bound to the problem. The heuristic they developed is a route-first, cluster-second method.

Bautista and Pereira (2004) presented an ant algorithm for designing collection routes for urban waste. To ascertain the quality of the algorithm, they tested it on three instances from the capacitated arc routing problem literature (i.e. Golden, DeArmon and Baker, 1983; Benavent et al, 1992; and Li and Eglese, 1996) and also on a set of real life instances from the municipality of Sant Boi del Llobregat, Barcelona. The characteristics of each dataset are presented. Computational results for Golden, DeArmon and Baker (1983) and Benavent et al (1992) are within less than 4% of the best known solution, and for Li and Eglese (1996) dataset up to 5.08%.

Mourao and Amado (2005) presented a heuristic method for a mixed CARP, inspired by the refuse collection problem in Lisbon. The proposed heuristic can be used for directed and mixed cases. Mixed cases indicate that waste may be collected on both sides of the road at the same time (i.e. narrow street), whereas waste for the directed cases only can be collected on one side of the road. They reported computational results for the directed case on randomly generated data and for the mixed case on the extended CARP benchmark problems of Lacomme et al. (2002). Computational results for the directed problem, involving up to 400 nodes show the gap values (between their lower bound and upper bound values computed from their heuristic method) varying between 0.8% and 3%. For the mixed problem, comparison results with four other heuristics namely, extended Path-Scanning, extended Ulusoys, extended Augment-Merge and extended Merge are reported. They stated that they were able to get good feasible solutions with gap values (between the lower bound values obtained from Belenguer et al (2003) and their upper bound values) between 0.28% and 5.47%.

Li, Borenstein and Mirchandani (2008) solved a solid waste collection in Porto Alegre, Brazil which involves 150 neighbourhoods, with a population of more than 1.3 million. They design a truck schedule operation plan with the purpose of minimizing the operating and fixed truck costs. In this problem the collected waste is discarded at recycling facilities, instead of disposal facilities. Furthermore, the heuristic approach used in this problem also attempts to balance the number of trips between eight recycling facilities to guarantee the jobs of poor people in the different areas of the city who work at the recycling facilities. Computational results indicate that they reduce the average number of vehicles used and the average distance travelled, resulting in a saving of around 25.24% and 27.21% respectively.

Mourao, Nunes and Prins (2009) proposed two two-phase heuristics and one best insertion method for solving a sectoring arc routing problem (SARC) in a municipal waste collection problem. In SARC, the street network is partitioned into a number of sectors, and then a set of vehicle trips is built in each sector that aims to minimize the total duration of the trips. Moreover, workload balance, route compactness and contiguity are also taken into consideration in the proposed heuristics.

Ogwueleka (2009) proposed a heuristic procedure which consists of a route first, cluster second method for solving a solid waste collection problem in Onitsha, Nigeria. Comparison results with the existing situation show that they use one less collection vehicle, a reduction of 16.31% in route length, a saving of around 25.24% in collection cost and a reduction of 23.51% in collection time.

In some cases, waste collection problems are solved as node and arc routing problems. For example Bautista, Fernandez and Pereira (2008) transformed the arc routing into a node routing problem due to the road constraint such as forbidden turns for solving an urban waste collection problem in the municipality of Sant Boi de Llobregat, Barcelona with 73917 inhabitants using an ant colonies heuristic which is based on nearest neighbour and nearest insertion methods. Computational results show that both methods produce less total distance compared with the current routes. In particular, routes from nearest neighbour and nearest insertion travel 35% and 37% less, respectively.

Furthermore, Santos, Coutinho-Rodriques and Current (2008) presented a spatial decision support system (SDSS) to generate vehicle routes for multi-vehicle routing problems that serve demand located along arcs and nodes of the transportation network. This is mainly due to some streets which are too narrow for standard-sized vehicles to traverse, thus the demand along arcs as well as at network nodes are required for solving waste collection in Coimbra, Portugal.

2.5 Node routing problems

If the location of every collection point is known when solving the waste collection problem then it is a node routing problem. Vehicles will travel from the depot to a customer and then to another customer, etc, to collect waste based on the sequence of visits on the vehicle route. This sequence includes trips to disposal facilities to empty the vehicle and the last visit would be the depot.

In the next section, previous work dealing with node routing problems, particularly the skip problems and non-skip problems are reviewed. Note here that Sbihi and Eglese (2007) have discussed the importance attached to waste management and collection in terms of the "green logistics" agenda.

2.5.1 Container/skip problems

De Meulemeester et al (1997) dealt with the problem of delivering empty skips and collecting full skips from customers. Vehicles can carry only one skip at a time, but skips can be of different types. They stated that the problem was first considered by Cristallo (1994). Their solution approach is based on two simple heuristics and an enumerative approach. They reported computational experience with randomly generated problems involving up to 160 customers and a real-world problem involving 30 customers.

Bodin et al (2000) considered a sanitation routing problem they called the rollon-rolloff vehicle routing problem. In this problem trailers, in which waste is collected, are positioned at customers. A tractor (vehicle) can move only a single trailer at a time. Tractor trips involve, for example, moving an empty trailer from the disposal facility to a customer and collecting the full trailer from the customer. A key aspect of their work is that they assume that the set of trips to be operated is known in advance (so the problem reduces to deciding for these trips how they will be serviced by the tractors).

They presented four heuristic algorithms and gave computational results for problems involving up to 199 trips and a single disposal facility.

Archetti and Speranza (2004) developed a heuristic algorithm called SMART-COLL for a problem motivated by waste collection in Brescia, Italy. In their problem skips are collected from customers and the vehicle can carry only one skip at a time. They call the problem the 1-skip collection problem. They considered skips of different types and time windows are imposed on both the customers and the disposal facilities. Computational experience was reported for real world data involving 51 customers and 13 disposal facilities.

Teixeira, Antunes and de Sousa (2004) developed a three-phase heuristic technique to create collection routes for the collection of urban recyclable waste in the central region of Portugal. Three types of waste in separate containers must be collected individually. The collected containers are emptied at two central depots and vehicles start and terminate a route in one of these depots. Computational results show that the total distance travelled of the proposed solution is 29% less than the historical distance.

Baldacci et al (2006) dealt with an extension of the problem considered by Bodin et al (2000). They considered multiple disposal facilities as well as inventory facilities at which empty trailers are available. They presented an approach based on regarding the problem as a time constrained vehicle routing problem on a directed multi-graph.

Computational results for problems involving up to 75 customers and two disposal facilities were presented.

Le Blanc et al (2006) presented a paper dealing with the collection of containers from end-of-life vehicle dismantlers in the Netherlands. In the problem they considered vehicles can carry two containers at a time. Their heuristic is a two-step procedure, first generating candidate routes, then selecting from these routes using a set partitioning approach. They reported potential cost savings of over 18% compared with the current system.

Even though Aringhieri et al (2004) solved a skip problem in Perugia, Italy, they solve the problem as an arc routing problem due to the different types of containers used to collect different waste. Therefore, the selected arcs for the vehicle to travel depend on the service requests which are characterized by types of waste, container and the collection point.

2.5.2 Non-skip problems

The majority of papers in the literature for non-skip problems are case study papers, focusing on results obtained when algorithms are applied to real world data. Only a few of these papers report computational experience with publicly available waste collection test instances.

Chang and Wei (2002) presented a real-life comparative study between a revised heuristic algorithm and an optimization technique particularly, minimum spanning tree and integer programming model, for investigating the effectiveness of vehicle routing and scheduling in a solid waste collection system. To illustrate the comparison of both techniques, a case study in the city of Kaohsiung, Taiwan which involves 854 collection points was presented. As expected in terms of cost-saving perspective, a set of near-optimal solutions from the heuristic algorithm are not as economic as the optimal solutions from the optimization scheme. Computational results show that the total number of collection vehicles and the total number of crews needed for the optimal solutions are 14 and 56, respectively whereas, the heuristic solutions require 22 vehicles and 84 crews in total. However in terms of total routing distance and collection time, the heuristic solutions show reductions of 12.7% and 0.9%, respectively even though they required more vehicles. Moreover, the authors stated that the heuristic algorithms allow the analysis of a much larger service area of interest within the same computational time as compared to the performance of an optimization model. In their view if an improvement in high performance computing comes into reality in the future it may overcome the present computational limitation of the optimization model.

Tung and Pinnoi (2000) proposed a heuristic procedure to solve a waste collection problem in Hanoi, Vietnam. In their problem there are time windows associated with collection from customers and their heuristic first constructs routes based on an approach due to Solomon (1987) and then improves them. They reported computational experience indicating that they can achieve an operating cost saving of 4.6% when compared with the current situation.

Angelelli and Speranza (2002a) presented an algorithm based on tabu search for the periodic version of the problem where routes must be designed over a planning horizon of more than one time period so as to meet customer service requirements. Their approach is based on the tabu search algorithm for vehicle routing of Cordeau et al (1997). Computational results were presented for problems involving between two and six days in the planning horizon.

Angelelli and Speranza (2002b) proposed a model that fits three different waste collection systems to estimate operational costs. Their solution procedure is based on Angelelli and Speranza (2002a) and results were presented relating to two case studies: Val Trompia, Italy and Antwerp, Belgium.

Sahoo et al (2005) reported how they developed a system called WasteRoute to reduce operating costs for a large company involved in waste collection. They gave one example of an area that went from ten routes to nine, improving route productivity (as measured by the amount collected per hour) by some 11%.

The heuristic used for the WasteRoute system of Sahoo et al (2005) is fully described in Kim et al (2006). Customers have time windows for collection, and there are multiple disposal facilities, as well as a driver rest period. They extended Solomon's (1987) insertion heuristic to cope with both multiple disposal facility visits and the driver rest period and used it to construct routes, which are improved using simulated annealing and a local search exchange procedure called CROSS (Taillard et al, 1997). As their work is motivated by the practical context reported in Sahoo et al (2005) they discussed a number of issues with solutions produced by this heuristic: route compactness, workload balancing and computation time. In order to deal with these issues they also presented a heuristic based on capacitated clustering that generates clusters based on the estimated number of vehicles required, and then routes customers within each cluster. Computational results were presented for ten problem instances, derived from real-world data, involving up to 2100 customers that the authors make publicly available.

Agha (2006) used a mixed integer programming model (MIP) to optimize the routing system for Deir Al-Balah, Gaza Strip. The problem involves 58 pick-up points, one disposal facility and three collection vehicles. Comparison results with the existing routing system are presented in terms of the total distance travelled. The result shows that the solution involves 23.4% less compared to the existing distance. Thus, the monthly cost can be reduced by approximately US$1140.

Ghose, Dikshit and Sharma (2006) combined both skip and non-skip problems to determine the minimum cost/distance efficient collection paths for transporting solid waste to the landfill for the Asansol Municipality Corporation (AMC) of West Bengal State, India. In total, the problem involves 1405 collection bins with three different sizes. Three types of vehicles are used for the collection of these bins. The vehicle type-A and the vehicle type-B serve as skip and non-skip problems, respectively. While the vehicle type-C collects the waste from C-type bins and disposes the waste at its nearest A-type bin. The vehicle will repeat the process until all the waste from C-type bins is collected. Then the vehicle will return to the garage from the location of the last A-type bin served. No comparison with the routing system practised by AMC is made. However, they compare the current annual operating cost AMC is spending with their estimated operating cost with respect to the proposed solution. Comparison result indicates that AMC may save about 66.8% every year if the proposed solution is applied.

Martagan et al (2006) applied classical MIP for a case study in Turkey for transporting metal waste from 17 factories to five potential containers, and from containers to a single disposal centre. The monthly cost of the proposed optimal solution is approximately $48000. Comparison routes with those current practised are not presented. Data is provided by TOSB (TAYSAD Organized Industrial Region) where TAYSAD is an abbreviation of Association of Automotive Parts & Components Manufacturers.

Nuortio et al (2006) considered a problem based on waste collection in two regions of Eastern Finland. Their problem includes time windows and they solved the problem using Guided Variable Neighbourhood Thresholding (Kytöjoki et al 2007).

Apaydin and Gonullu (2007) used Route View Pro^{TM} software for constructing waste collection route in Trabzon, Turkey. The collection involves 777 containers and one disposal site. The solution is compared with the present routes in terms of the total route distances, travelling time as well as the monthly cost. Comparison results indicate that their routes outperform the present. Both distance and travelling time are reduced by up to 59% and 67%, respectively whilst the monthly cost is decreased by up to 24.7%.

Karadimas, Papatzelou and Loumos (2007) presented an ant colony system (ACS) for determining waste collection routes for the Municipality of Athens (MoA). The collection involves 72 loading spots. Comparison results with the empirical method (Kollias, 1993) used by MoA are presented. The route length of the empirical model is 9850, whilst the ACS route is 7328. Thus, the improvement is approximately 25.6%.

Ombuki-Berman et al (2007) presented a multi-objective genetic algorithm that uses a crossover procedure (Best Cost Route Crossover) from Ombuki et al (2006). They reported results from their approach using the test problems of Kim et al (2006), but no computation times were given.

Alagöz and Kocasoy (2008) considered health waste collection in Istanbul. They used a commercial vehicle routing package to consider a number of scenarios relating to the type of facility used for waste disposal. McLeod and Cherrett (2008) considered a problem relating to waste collection in the UK. They used a commercial vehicle routing package and reported that vehicle mileage could be reduced by up to 14%.

Coene, Arnout and Spieksma (2008) proposed several heuristic algorithms for a routing problem of a Belgian company collecting waste at slaughterhouses, butchers and supermarket. The company is responsible for collecting high-risk and low-risk waste categories of animal waste. Both wastes need to be collected separately. The instances can be found at http://www.econ.kuleuven.ac.be/public/N05012/. Comparison results in terms of the total travelling time between the proposed solutions and the current routes are presented. For the low-risk waste, the results indicate that the current total travelling time can be reduced by up to 15.5%, whereas the travelling time for the collection of high-risk waste can be reduced by up to 9%.

Komilis (2008) presented two mixed integer-linear programming models particularly time based optimization model and cost optimization model for the waste collection problem in Athens. The waste is collected from the source nodes and taken to potential intermediate nodes, namely waste production nodes (WPN) and waste transfer stations (WTS), respectively and finally to the landfill as a sink node. The cost modelling approach used in this work has similarities with the cost optimization model used in Badran and El-Haggar (2006) particularly in calculating fuel and maintenance costs as well as labour cost. The problem involves seven WPN, three WTS and one landfill. However, WTS may or may not be included in the optimal path, depending on the solution.

Arribas, Blazquez and Lamas (2010) proposed a methodology for designing an efficient urban waste collection for the west-central zone of the Municipality of Santiago using a combination of mathematical modelling such as linear integer programming and a tabu search algorithm in a GIS environment. The collection involves 1600 bins and the comparison results indicate that the proposed routing system manages to reduce 50% of current monthly cost spent on waste collection system with a reduction of 57% in the number of vehicles as well as a reduction of 57% in the number of workers needed to complete the collection.

Chalkias and Lasaridi (2009) used ArcGIS Network Analyst in their work for the waste collection in the Municipality of Nikea (MoN), Athens, Greece. The problem involves 501 collection bins and one disposal site. Besides constructing collection routes, replacing and reallocating the waste collection bins is also taken into consideration. These two scenarios are compared with the current routes. Computational results demonstrate that both scenarios provide savings in terms of collection time and total travel distance. The first scenario (constructing routes with the current location of the waste bins) saves around 3.0% in collection time and 5.5% in distance travelled, whereas the second scenario (constructing routes after replacing and reallocating the waste bins) saves around 17.0% in collection time and 12.5% in distance travelled.

Hemmelmayr et al (2009) presented a paper motivated by a real world waste collection problem. They consider a periodic problem, where routes must be designed over a multi-day planning horizon so as to meet customer service requirements. They consider a number of constraints motivated by their underlying application and in particular in their application the vehicle need not return to the depot empty. They use dynamic programming to sequence disposal facility visits within a variable neighbourhood search approach. Computational results are presented for instances, involving up to 288 customers, derived from vehicle routing problems given in the literature.

Repoussis et al (2009) considered waste oil collection and recycling in Greece. In their problem vehicles are compartmentalised and they use a list based threshold accepting metaheuristic (Tarantilis et al, 2004) to design vehicle routes. They reported reductions of up to 30% in the cost per unit of waste collected.

Zamorano et al (2009) attempted to reduce the waste collection management costs in Churriana de la Vega, Spain. This objective includes reducing fuel consumption by minimizing the travel time of the collection routes using ArcGIS Network Analyst. Computational results show reductions of 32.3% in travelling time compared to the current routes. Other literature focusing on minimizing fuel consumption are Tavares et al (2008) and Tavares et al (2009).

Here, we also would like to mention some of the previous work dealing with waste collection but where no disposal facilities are involved, such as Firinci et al (2009) and Karadimas et al (2005).

2.6 Discussion

This thesis considers exactly the same waste collection problem as in Kim et al (2006) involving multiple disposal facilities, driver rest period and customer/depot/disposal facility time windows. Because Kim et al (2006) have made their test problems publicly available, a direct computational comparison with their work can be made in this thesis. We compare our solutions with the solutions from Kim et al (2006) in terms of the distance travelled as well as the number of vehicles used. Since the waste collection problem considered in this thesis has an unlimited number of vehicles, our main objective is to produce solutions for the problem with less distance travelled than Kim et al (2006).

Although this thesis focuses directly on waste collection from customers we would briefly mention here that in the context of deliveries to customers an analogous problem is the vehicle routing problem with intermediate replenishment facilities. In problems of this type there are intermediate facilities at which vehicles can replenish/restock with the goods that they need to satisfy demand at customers they have yet to visit before they finally return to the depot at the end of the working day.

An important difference between the collection problem with disposal facilities and the delivery problem with replenishment facilities is that in the collection problem a vehicle visits a disposal facility to empty itself immediately prior to returning to the depot. In the delivery problem there is typically no visit to a replenishment facility for restocking immediately prior to returning to the depot (or conversely immediately after leaving the depot). More as to the delivery problem with intermediate replenishment facilities can be found in Kek et al (2008), Tarantilis et al (2008), Crevier et al (2007) and Angelelli and Speranza (2002a).

Furthermore, this thesis also considers a set of new benchmark instances generated by Crevier et al (2007) from those proposed by Cordeau et al (1997) for the multi-depot VRP (MDVRP). These instances contain up to 288 customers and seven depots. In this case the depots can act as intermediate replenishment facilities along the route of a vehicle. The instances and the best known solutions are available at http://www.hec.ca/chairedistributique/data/. To the best of our knowledge, no previous works are available, thus the comparative results are only made with the authors.

CHAPTER 3

INITIAL SOLUTION (IS) PROCEDURES

This chapter begins with the notation used in the thesis. Next, procedures to define neighbour sets and to find initial solutions are presented. Then, two sets of benchmark problems used in the thesis are presented. Finally, solutions obtained from the procedures are reported.

3.1 Notation

Let C be the set of customers and D be the set of disposal facilities (recall we have just a single depot). To represent the depot, disposal facilities and customers we index them such that 0 is the depot, 1, 2, …, |D| are the disposal facilities and |D|+1, |D|+2, …, |D|+|C| are the customers. The travel time between i and j is denoted by t_{ij} and the distance between them by d_{ij} (where i and j may be the depot, disposal facilities or customers). The notation may be conveniently structured as to that relating to the vehicles or customers/disposal facilities/depot. For the vehicles let:

- Q be the vehicle capacity, so a vehicle filled to this capacity has to be emptied at a disposal facility before any other customer can be visited
- Q^* be the maximum amount a vehicle can deal with per day (over all customers)
- S^* be the maximum number of customers the vehicle can deal with per day

- $[R_1,R_2]$ be the single lunch (rest) time window so that the lunch/rest period must start at some time within this period, the rest duration (how long the vehicle/driver is idle) being R_3

For the customers/disposal facilities/depot let:

- q_i be the quantity to be collected at customer $i \in C$
- V_i be the service time for $i \in C \cup D$ such that the visit to i (for collection/disposal) takes this (fixed) time
- $[E_i,L_i]$ be the time window for $i \in C \cup D \cup \{0\}$ such that the visit to $i \in C \cup D$ (for collection/disposal) must start within this time period and the vehicles must leave/return to the depot within $[E_0,L_0]$

In terms of our heuristic we need to identify the nearest (in terms of travel time) open disposal facility for customer i at time T. We denote this by n(i,T) and it is defined by:

$$n(i,T)=\arg\min[t_{ij} \mid j \in D, T+t_{ij} \in [E_j,L_j]]$$

so the disposal facility associated with customer i at time T is the nearest facility that is open should the vehicle go directly from customer i to the facility. Note here that computationally we do not calculate n(i,T) for all values of i and T, rather we calculate n(i,T) as and when needed in the heuristic below.

3.2 Neighbour sets

A neighbour set of cardinality K for customer i, denoted by N(i,K), is composed of the K customers that are closest to customer i, but with compatible time windows. A customer j is defined in our work to have a compatible time window with customer i (and hence potentially belongs to N(i,K)) if it is possible to visit i at some time in its time window, service i and then go directly onto j to service j without waiting for its time window to open.

As the time window for i is $[E_i,L_i]$ the time window for arrival at j after servicing i is $[E_i+V_i+t_{ij}, L_i+V_i+t_{ij}]$. Customer j potentially belongs to the neighbour set for i if this time window overlaps with its time window $[E_j,L_j]$. Two time windows overlap if and only if an end point of one time window falls within the other time window. We can therefore define the neighbour sets using:

$\forall i \in C$: set $N(i,K)=\emptyset$

$\forall i \in C, \forall j \in C$ $(j \neq i)$:

if $E_j \leq E_i+V_i+t_{ij} \leq L_j$ or $E_j \leq L_i+V_i+t_{ij} \leq L_j$ or

$E_i+V_i+t_{ij} \leq E_j \leq L_i+V_i+t_{ij}$ or $E_i+V_i+t_{ij} \leq L_j \leq L_i+V_i+t_{ij}$ set $N(i,K)=N(i,K) \cup j$

To ensure that N(i,K) has appropriate cardinality then if after the above calculation we have $|N(i,K)|>K$ we alter N(i,K) to contain only the K nearest customers to i, i.e. sort the customers in N(i,K) in increasing order of their travel time (t_{ij}) from i and set N(i,K) to contain just the first K customers from this ordered list. Note here that we may have $|N(i,K)|<K$ if there are fewer than K customers that have compatible time windows with i.

Figure 3.1 depicts an example of a neighbour set of customer 3 with K = 50. The red dots represent the neighbours of customer 3 which are the closest to customer 3 as well as having compatible time windows with customer 3. The black dots are the other customers in the same data set. Even though there are some black dots close to customer 3 they do not have compatible time windows with customer 3 and so are not neighbours of customer 3. For example, if the time window of customer 3 is [12:00, 15:00] and the time window of customer 5 is [08:00, 10:00] then the vehicle would not be able to visit customer 5 after servicing customer 3, because it is already closed. Thus, customer 5 would not satisfy any of the expressions discussed above.

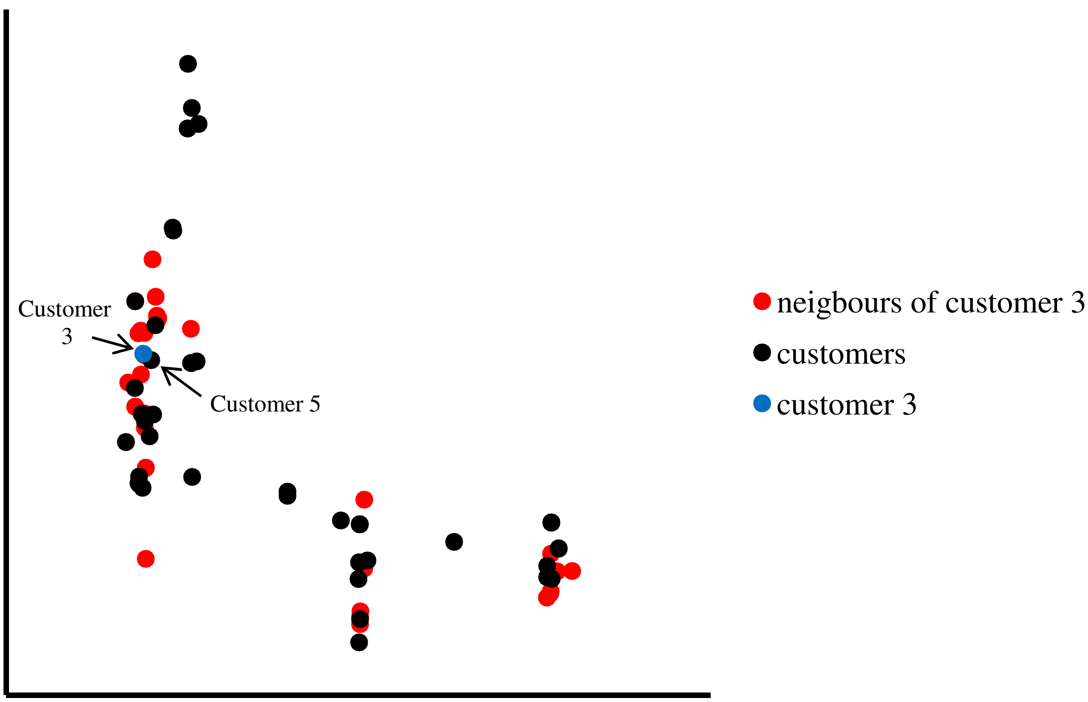


Figure 3.1: An example of a neighbour set with K = 50

Neighbour sets are a key element in our work. This arises for two reasons:

- the nature of our metaheuristics, as will become apparent in later chapters below, is that we use neighbour sets in seeking to improve a route. As such the larger the value of K the larger the neighbourhood we search.
- by varying K we have a variable neighbourhood. This leads in a natural fashion to applying variable neighbourhood search to the problem.

Note here that N(i,K) can be computed before we embark on route construction.

3.3 Current initial solution (IS) procedure

We construct an initial solution by attempting to fully utilise a vehicle over the day (thereby aiming to minimise the total number of vehicles used). Once a vehicle cannot be used any more then we start a new vehicle route with a new vehicle. To deal with the vehicle/driver rest period we attempt to schedule it as early as possible consistent with its time window. Our procedure involves a number of steps, as below.

Initialise

Set B=C (B is a working set of customers that have still to be routed)

Step 1

if $|B|\neq 0$ so there are still customers to be dealt with **then:**

start a new vehicle route at time E_0, when the depot opens

$T=E_0$	T is the current time
$s_{total}=0$	s_{total} is the total number of customers the vehicle has visited
$Q_{total}=0$	Q_{total} is the total load the vehicle has dealt with
$Q_{current}=0$	$Q_{current}$ is the current load on the vehicle

$r=0$ r is the customer at the end of the current emerging vehicle route

$rest=0$ rest is one if the vehicle has had its rest period, else zero

else

all customers have been dealt with so **stop**

end if

Step 2

Check for the rest period – here we start the rest period as soon as practicable

If the vehicle has not had a rest period (rest=0) and $T \in [R_1, R_2]$ **then:**

the vehicle now has its rest period

$rest=1$ update rest

$T=T+R_3$ update the current time

end if

Step 3

The next customer to be visited on the current emerging route is that customer $i \in B$ such that

$$i = \arg\min[t_{rj} \mid j \in B,\ T+t_{rj} \in [E_j, L_j],\ Q_{current}+q_j \leq Q,\ Q_{total}+q_j \leq Q^*,\ s_{total}+1 \leq S^*,$$
$$\theta + t_{j,n(j,\theta)} + V_{n(j,\theta)} + t_{n(j,\theta),0} \leq L_0,\ \theta \leq R_2 \text{ if rest=0, where } \theta = T + t_{rj} + V_j]$$

The customer i is the customer that has the shortest travel time from the customer r at the current end of the route, provided i satisfies the conditions seen above. This expression is relatively complex. Here we consider only those customers j such that when the vehicle arrives at j (at time $T+t_{rj}$) it will be possible to service the customer as the visit will fall in its time window $[E_j, L_j]$ and the load to be collected at j will fit on the vehicle (in terms of the current route, the entire days work and the total number of customers visited). Also j has to be a customer such that if j is visited there is time for the vehicle to visit the nearest (open) disposal facility to j and then return to the depot before the end of the working day. In the above expression θ is the time at which the vehicle finishes servicing j, then the vehicle travels to the nearest open disposal facility $n(j,\theta)$, taking time $t_{j,n(j,\theta)}$, the disposal facility visit takes time $V_{n(j,\theta)}$, and then the vehicle travels to the depot taking time $t_{n(j,\theta),0}$, arriving before the end of the working day (at time L_0). Furthermore j has to be such that if the vehicle has not yet had its rest period (rest=0) there is still time after servicing j for the rest period to be started ($\theta \leq R_2$).

For this step preliminary computational experience indicated that one issue which arises is that we want to avoid excess travel simply because the vehicle has some limited spare capacity. In order to gauge this we compare the travel time from the end of the current route to i (i.e. t_{ri}) to the travel time to the nearest disposal facility n(r,T), i.e. $t_{r,n(r,T)}$. If $t_{ri}>t_{r,n(r,T)}$ and the vehicle is near to capacity (in our work a vehicle is defined to be near to capacity if $\max[Q_{current}/Q, Q_{total}/Q^*, s_{total}/S^*]>0.8$) then we disregard i (i.e. we treat this situation as if we had found no customer satisfying the above expression).

If there is a customer i satisfying the above expression that we can add to the end of the emerging route **then:**

the vehicle travels to i and services the customer

$T=T+t_{ri}+V_i$	update the current time
$r=i$	update the current customer at the end of the route
$Q_{current}=Q_{current}+q_i$	update the current vehicle load
$Q_{total}=Q_{total}+q_i$	update the daily vehicle load

$s_{total}=s_{total}+1$ update the total number of customers visited

$B=B-\{i\}$ update the set of customers B by removing i from it

go to step 2

end if

Step 4

We reach this step when we have not found a customer to add to the end of the emerging route.

If vehicle is not empty ($Q_{current}>0$) **then:**

the vehicle travels to its nearest disposal facility n(r,T) to be emptied

$T=T+t_{r,n(r,T)}+V_{n(r,T)}$ update the current time

$r=n(r,T)$ update the end of the route

$Q_{current}=0$ update the current vehicle load

go to step 2

end if

Step 5

We reach this step when we have not found a customer to add to the end of the emerging route and the vehicle is empty ($Q_{current}=0$). In this case no more work can be done with this vehicle at the current time. It may be possible that the vehicle can do some more collections if it is idle until a time arrives such that it is possible to collect from some customer.

To deal with this situation we look for the customer whose time window will "open" as soon as possible:

$$i=\arg\min[E_j \mid j\in B,\ T+t_{rj}<E_j,\ Q_{current}+q_j\leq Q,\ Q_{total}+q_j\leq Q^*,\ s_{total}+1\leq S^*,$$
$$\theta+t_{j,n(j,\theta)}+V_{n(j,\theta)}+t_{n(j,\theta),0}\leq L_0,\ \theta\leq R_2 \text{ if rest}=0, \text{ where } \theta=E_j+V_j]$$

If there is a customer i satisfying the above expression **then:**

the vehicle travels to i, arrives at $T+t_{ri}$ and waits until time E_i to start the collection

$T=E_i+V_i$	update the current time
r=i	update the current customer at the end of the route
$Q_{current}=Q_{current}+q_i$	update the current vehicle load
$Q_{total}=Q_{total}+q_i$	update the daily vehicle load
$s_{total}=s_{total}+1$	update the total number of customers visited
$B=B-\{i\}$	update the set of customers B by removing i from it

go to step 2

end if

In this step, one further issue that arises with respect to the rest period is when a vehicle finishes servicing the current customer at the end of the emerging route before R_1 ($T<R_1$ and rest=0) and the earliest time windows of the remaining customers will only start at/after R_2 ($E_j \geq R_2$, $\forall j \in B$). In this situation as currently above, either we add customer i with $E_i \geq R_2$ at the end of the emerging route or the vehicle returns to the depot and then we start with a new route to serve these remaining customers. If we add customer i the vehicle will not have a rest break because after servicing customer i, T is already exceeding the rest time window.

To deal with this situation we set the vehicle to have the rest break before servicing customer i:

If we have not found a customer that satisfies the above expression because rest=0 and $E_j \geq R_2$, $\forall j \in B$ **then:**

$$i = \arg\min[E_j \mid j \in B,\ T+t_{rj}<E_j,\ Q_{current}+q_j \leq Q,\ Q_{total}+q_j \leq Q^*,\ s_{total}+1 \leq S^*,$$
$$\theta+t_{j,n(j,\theta)}+V_{n(j,\theta)}+t_{n(j,\theta),0} \leq L_0, \text{ where } \theta=E_j+V_j]$$

the vehicle travels to i, waits until E_i, and then services the customer

$T=\max[T+ t_{ri}, R_1]+R_3$ update the current time before servicing i

$T=\max[T, E_i]+ V_i$ update the current time after servicing i

rest=1 **update rest**

r=i update the current customer at the end of the route

$Q_{current}=Q_{current}+q_i$ update the current vehicle load

$Q_{total}=Q_{total}+q_i$ update the daily vehicle load

$s_{total}=s_{total}+1$ update the total number of customers visited

B=B-{i} update the set of customers B by removing i from it

go to step 3

else

the vehicle travels back to the depot (as it is empty it does not need to visit a disposal facility first) and a new vehicle must now be used

go to step 1

end if

In step 1 above we start a new vehicle route and initialise the various counters we need to keep track of the use made of the vehicle. In step 2 we attempt to schedule the rest period. In step 3 we add a customer to the end of the emerging route such that it is feasible to add the customer both in terms of the vehicle load and in terms of “look-ahead” for the vehicle to return to the depot empty. In step 4 the vehicle is emptied whilst in step 5 the vehicle waits for a customer time window to open. The above procedure terminates once all of the customers have been dealt with.

We have set out our initial solution procedure in detail in order that the reader can clearly see the steps involved and the counters that are updated as a route is constructed.

For the remainder of this thesis we, for brevity, use higher level pseudocode in terms of presenting our algorithms.

Figure 3.2 illustrates an example of how our IS works in constructing initial vehicle routes for a problem which consists of one single depot, ten customers and two disposal facilities. All time windows are [9:00, 19:00] except customer 3 is [17:00, 18:00]. The rest time window is [11:00, 12:00] and the rest time duration is one hour. The amount of waste that needs to be collected at every customer is one cubic meter except for customers 4 and 7 which are two cubic meters each. The vehicle capacity is five cubic meters and two vehicle routes are required to serve the customers. In this example we, for simplicity, have not given explicit travel times between customers. Rather we have assumed underlying travel times are such as to illustrate the procedure we outlined above. The processes for constructing both routes are explained as follow:

- Initialise set B=C
- Do step 1: We construct first vehicle route and initialise all counters.
- Do step 2: Rest period is checked. Currently, T=9:00 (E_0) so no rest break is required yet.
- Do step 3: The first vehicle will start its route by travelling to a customer closest to the depot. In this case, suppose customer 3 is closest to the depot but its time window starts at 17:00 and still not ready to be served if the vehicle travels to it. Thus, the vehicle travels to the next customer which is closest to the depot (customer 4 which can be served). After collecting waste at customers 4, 5, and 7, total waste

on the vehicle is 2+1+2=5 cubic meters. Thus, no other customers can be served because it would exceed the capacity of the vehicle.

- Do step 4: The vehicle travels to the nearest available disposal facility (here disposal facility 1) to be emptied.
- Do step 3 to find the next customer to be served. So from disposal facility 1, it goes to customer 9 and finishes servicing customer 9 at 11:15.
- Do step 2: The driver takes a rest break for an hour before serving other customers.

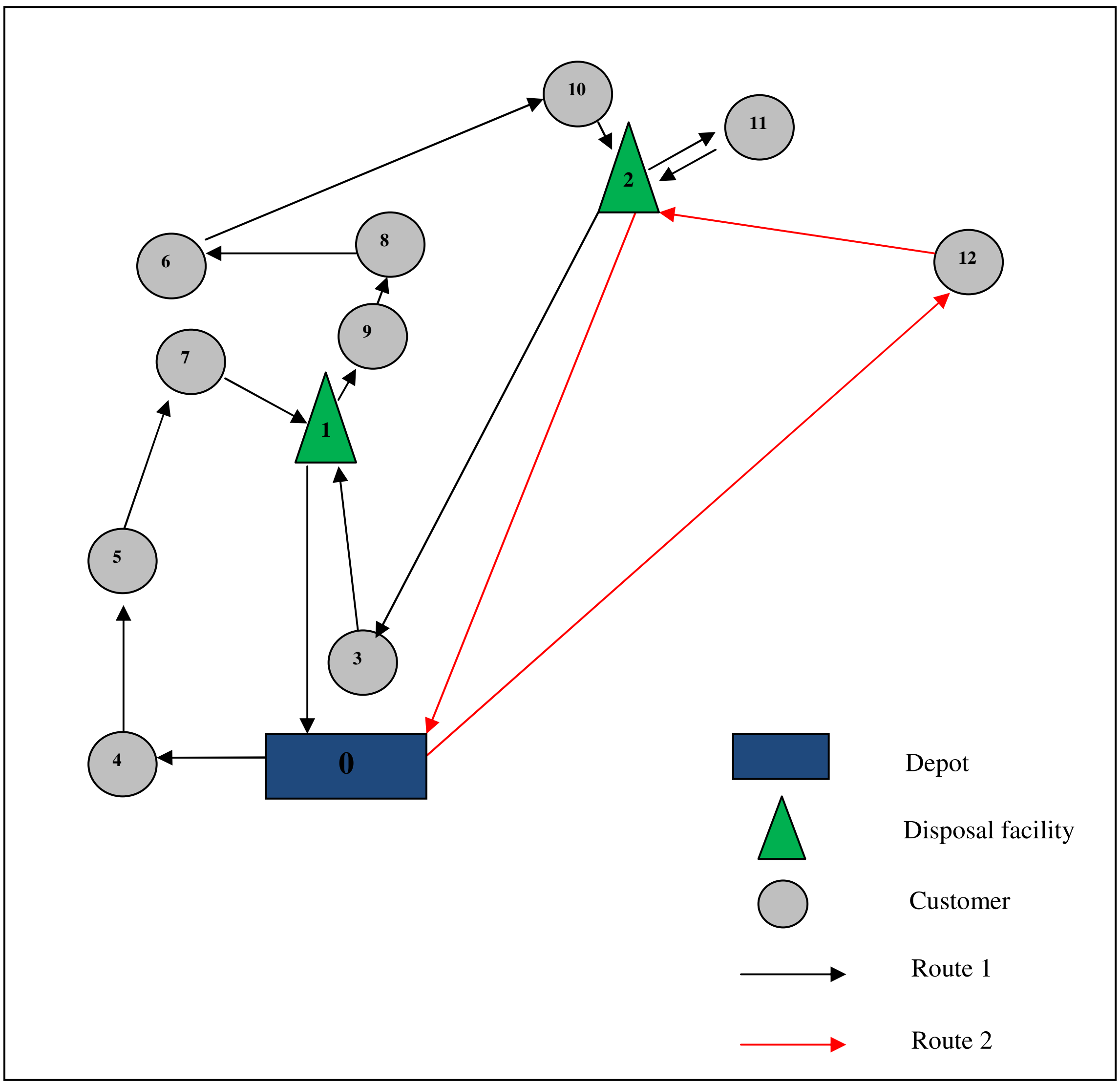


Figure 3.2: Examples of two vehicle routes for a waste collection problem

- Do step 3: After the break, the vehicle goes to customer 8, followed by customers 6 and 10. Now the total waste on the vehicle is four cubic meters which is approximately 80% (4/5) full. Our procedure treats this situation as we had found no customer satisfying the expressions in step 3. Here we will check whether the vehicle should continue with another customer (customer 11) or should unload the waste at a disposal facility. The figure shows that disposal facility 2 is closer to customer 10 than disposal facility 1. This facility is also closer to customer 10 than customer 11 (its next nearest customer that can be served). Hence, the vehicle goes to disposal facility 2 to be emptied and then travels to the nearest customer (i.e. customer 11). After serving customer 11, the remaining customers (customers 3 and 12) do not satisfy the expression in step 3. For example, if the vehicle travels to customer 3 it will arrive before 17:00 and customer 3 is still not ready to be served. Whereas, if the vehicle travels to customer 12, the vehicle will not be able to return to the depot before 19:00 (i.e. $T>L_0$).
- Do step 4: The vehicle is not empty so it goes to the nearest disposal facility to be emptied.
- Do step 5: We will check whether the vehicle should return to the depot or continue servicing other customers, whose time window will open as soon as possible. Here, customer 3 satisfies the expression. Thus, the vehicle travels to customer 3 and waits until the customer is ready to be served.
- Do step 3: After servicing customer 3, only one unrouted customer is left (customer 12). However if the vehicle continued to service customer 12, it will arrive at the depot after 19:00. Thus, no more customers can be served.

- Do step 4: The vehicle goes to the nearest disposal facility (here disposal facility 1) to be emptied.
- Do step 3: After unloading waste, the vehicle continues to serve another customer. Since only customer 12 is left and the vehicle cannot serve this customer because the depot is already closed if it arrived after servicing customer 12. Thus, no more customers can be served.
- Do step 5: The vehicle is empty and returns to the depot.
- Do step 1: We construct a second vehicle route and initialise all counters.
- Do step 2: We check rest period. Currently, T=9:00 (E_0) so no rest break is required yet.
- Do step 3: From the depot, the second vehicle starts its route by travelling to customer 12. Now, $|B|=0$ (where all customers have been served).
- Do step 4: The vehicle is not empty and goes to the nearest disposal facility to unload the waste.
- Do step 5: The vehicle returns to the depot. Since the vehicle completes its route before 11:00 ($T<R_1$), the driver would have the rest period at the depot.
- Note that in our procedure every time the vehicle finishes servicing a customer or unloading waste at a disposal facility, we always do step 2 because we want to start the rest period as soon as practicable.

The IS procedure discussed in this section has been designed for solving the waste collection VRPTW problems by Kim et al (2006). In addition, the effectiveness of our IS procedure is also tested on a multi-depot VRP with inter-depot routes (MDVRPI) benchmark problem generated by Crevier et al (2007). There are some differences between MDVRPI and waste collection VRPTW problems such as:

- no time windows (either customers or depots)
- no rest period
- limited number of homogenous vehicles
- much smaller problems, the largest problem of the waste collection VRPTW involves 2092 customers, whereas the MDVRPI involves 288 customers.

The need to cope with the very large waste collection VRPTW problems has affected the design of the heuristic procedures we present in this thesis. For example our use of neighbour sets.

When applying our IS procedure on the MDVRPI problems, in step 1 we set rest=1 so in the solution no rest time will be included. However, computational results indicate that solutions by Crevier et al (2007) are much better than ours. Even though our initial solutions have been improved after applying our local search algorithm, yet the solutions by Crevier et al (2007) still outperform our solutions. Thus in trying to improve the solution, in this thesis we apply other IS procedures to this problem. Some of the procedures are adapted from the literature.

Moreover, these IS procedures also have been tested on the waste collection VRPTW problems for investigating the effectiveness of the procedures. Both sets of benchmark problems are presented in section 3.6.

3.4 Other IS procedures

In this section we present two other IS procedures which we modified from the IS procedure presented in the last section, namely

- Farthest from depot procedure
- Different initial customer procedure.

3.4.1 Farthest from depot

Currently, in our IS procedure every vehicle route starts with a customer closest to the depot. Then the vehicle continues travelling to another customer closest to the last customer/disposal facility added on the route. Thus, the customers which are far from the depot are dealt with late in the process of route construction. As a result, the last vehicle that serves these customers may travel more distance than the other earlier vehicle routes even if there is only a small number of customers left on this route.

Hence, to overcome this problem we modify our current IS procedure particularly in Step 3. In this new procedure, when we start a new route the first customer on the route would be the customer farthest from the depot. Then new customers on the route are added using the same expressions as in Step 3. Other parts of the current IS procedures remain the same. Below is the pseudocode of Step 3 with the changes highlighted:

Step 3

The next customer to be visited on the current emerging route is that customer $i \in B$ such that

if r = 0 then

this is the first customer to be placed on the route, take the customer farthest from the depot

$$i=\arg\ \mathbf{max}[t_{rj} \mid j\in B,\ T+t_{rj}\in [E_j,L_j],\ Q_{current}+q_j\leq Q,\ Q_{total}+q_j\leq Q^*,\ s_{total}+1\leq S^*,$$
$$\theta+t_{j,n(j,\theta)}+V_{n(j,\theta)}+t_{n(j,\theta),0}\leq L_0,\ \theta\leq R_2 \text{ if rest=0, where } \theta=T+t_{rj}+V_j]$$

else

$$i=\arg\ \mathbf{min}[t_{rj} \mid j\in B,\ T+t_{rj}\in [E_j,L_j],\ Q_{current}+q_j\leq Q,\ Q_{total}+q_j\leq Q^*,\ s_{total}+1\leq S^*,$$
$$\theta+t_{j,n(j,\theta)}+V_{n(j,\theta)}+t_{n(j,\theta),0}\leq L_0,\ \theta\leq R_2 \text{ if rest=0, where } \theta=T+t_{rj}+V_j]$$

endif

Note here that r is the node at the end of the current emerging vehicle route. Thus if r=0 (current node is the depot), the next node on the route would be the farthest customer from the depot. Then, after updating r, the process on selecting the next customer to be visited on the current emerging route is the same as in our IS procedure (closest to r).

3.4.2 Different initial customer based on current IS procedure

In this section we construct |C| sets of initial solutions using the current IS procedure. For this new procedure we do |C| runs, each time with a different initial customer as the first customer on route 1. The processes for constructing vehicle routes remain the same such that we start a new route with a customer closest to the depot. Here we add Step 0 and also make a change in Step 3 in the current IS procedure. The pseudocode of both steps is therefore

Step 0

For all customers $i\in C$:

- Set flag = 0, this is a flag to indicate that we are going to construct route 1

- startNode = i, this would be the first customer on route 1
- Perform IS(startNode, flag)

end for

Step 3

The next customer to be visited on the current emerging route is that customer $i \in B$ such that

if flag = 0 then

i = startNode

flag = 1

else

$i=\arg\,\mathbf{min}[t_{rj} \mid j \in B,\ T+t_{rj} \in [E_j,L_j],\ Q_{current}+q_j \leq Q,\ Q_{total}+q_j \leq Q^*,\ s_{total}+1 \leq S^*,$

$\theta+t_{j,n(j,\theta)}+V_{n(j,\theta)}+t_{n(j,\theta),0} \leq L_0,\ \theta \leq R_2$ if rest=0, where $\theta=T+t_{rj}+V_j]$

endif

In step 0 above we set a flag to indicate that we are going to construct a new set of initial solutions with a different initial customer as the first customer on route 1. In step 3 we add a customer to the end of the emerging route.

3.5 Other IS procedures using disposal facility positioning (DFP)

In further attempts to construct other initial solutions for the benchmark problems, in this section three IS procedures adapted from the literature are presented, namely

- The savings approach of Clarke and Wright (1964)
- The sweep algorithm of Gillett and Miller (1974)

- The different initial customer procedure based on the sweep algorithm of Gillett and Miller (1974)

Since the waste collection problem involves disposal facilities trips on the routes, a procedure by Hemmelmayr et al (2009), namely disposal facility positioning (DFP) is added in the above IS procedures for choosing the best disposal facilities to go on the route. In this thesis the DFP procedure is presented in detail in Chapter 6.

3.5.1 The savings approach

In this section, an initial solution based on the savings approach of Clarke and Wright (1964) is constructed. For all pairs of customers (i,j) let $s_{ij} = (d_{0i} + d_{i0}) + (d_{0j} + d_{j0}) – (d_{0i} + d_{ij} + d_{j0}) = d_{i0} + d_{0j} - d_{ij}$ be the distance saving resulting from having i and j together on the same route, rather than serviced individually on two separate routes. The larger the distance saving the more attractive it is to have i and j on the same route. Note that these savings values can be computed once for the problem, they do not depend on the routes that will be constructed later.

The new savings algorithm is therefore:

Step 1

Choose the customer pair (i,j), where $i \in N(j,K)$ and $j \in N(i,K)$ with the largest saving. Then, evaluate the route depot–i-j-depot using the DFP and reverse procedures. If the route is not FEASIBLE then consider the customer pair with the next largest saving. Repeat the process until a customer pair such that the route is feasible is found.

Reloop:

Step 2

Let α and β be the two customers at the end of this emerging feasible route. Consider all unrouted customers $i \in N(\alpha,K)$ and $j \in N(\beta,K)$ and from the savings [$s_{\alpha i}$ $\forall i \in N(\alpha,K)$ i unrouted; $s_{\beta j}$ $\forall j \in N(\beta,K)$ j unrouted]:

- consider them in descending order until a feasible customer to add to the end of the merging route is found, i.e.:
    - o for each savings value add the corresponding customer to the end of the emerging route and evaluate the route using the DFP and reverse procedures
    - o if the route is FEASIBLE then keep the added customer, reset α and β to the customers at the end of the emerging route and go to **Reloop** to find a new customer to add; else consider the next largest saving value

Step 3

At this step we have added as many customers as we can to the emerging route in which case this route is done. If there are still unrouted customers then go to **Step 1** to construct a new route with a new vehicle.

In step 1 above we start a new vehicle route by adding a pair of customers so that they would be the two customers at the end of the emerging feasible route. In step 2 a neighbour (with the largest saving value) of the two customers who are currently at the end of the emerging feasible route is added to the end of the emerging route. In step 3 we check whether there are still unrouted customers left. The above procedure terminates once all of the customers have been dealt with.

3.5.2 The sweep algorithm

In this section, an initial solution based on the sweep algorithm of Gillett and Miller (1974) is constructed. Here we consider the customers' coordinates (x_i,y_i) are in ascending order of the angle between the line that connects the customer with the depot (x_0,y_0) and the horizontal. The coordinates relative to the depot of customer i are (x_i-x_0, y_i-y_0) and the polar angle that the line connecting the depot to the customer makes with the horizontal is $\theta = \tan^{-1}[(y_i-y_0)/(x_i-x_0)]$ or $\theta = \tan^{-1}[y/x]$ where $y = y_i-y_0$ and $x = x_i-x_0$.

To obtain θ in the interval [0, 2π) radians (moving anti-clockwise), an adjustment has been made using the following rules:

$$\theta = \begin{cases} \tan^{-1}[y/x] & \text{if } x > 0 \text{ and } y \geq 0 \\ \tan^{-1}[y/x] + 2\pi & \text{if } x > 0 \text{ and } y < 0 \\ \tan^{-1}[y/x] + \pi & \text{if } x < 0 \\ \pi/2 & \text{if } x = 0 \text{ and } y > 0 \\ 3\pi/2 & \text{if } x = 0 \text{ and } y < 0 \\ 0 & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

Again in this section, the DFP procedure is added in the sweep algorithm for choosing suitable disposal facilities on the route. The new initial solution procedure using a sweep approach is as below.

Step 1

Choose the unrouted customer i, $i \in B$ that has the minimum polar angle to be the first customer on the route.

Reloop:

Step 2

- Let α be the customer at the end of the emerging feasible route. Consider all unrouted customers i (with polar angles ≥ polar angle of α) in **ascending** order of their polar angles until a feasible customer to add to the end of the merging route is found, i.e.:
  - for each unrouted customer i (whose polar angle ≥ polar angle of α) add it to the end of the emerging route and evaluate the route using the DFP procedure
  - if the route is FEASIBLE then keep the added customer, reset α to the customer at the end of the emerging route and go to **Reloop** to find a new customer to add; else consider the next customer with a larger polar angle.

Step 3

At this step we have added as many customers as we can to the emerging route in which case this route is done. If there are still unrouted customers then go to **Step 1** to construct a new route with a new vehicle.

In step 1 above we start a new vehicle route by adding a customer who has the minimum polar angle. In step 2 we add a customer who has a larger or equal polar angle than the polar angle of a customer who is currently at the end of the emerging feasible route. In step 3 we check whether there are still unrouted customers left. The above procedure terminates once all of the customers have been dealt with.

3.5.3 Different initial customer based on the sweep algorithm

A procedure presented in this section is quite similar to the IS procedure discussed in section 3.4.2 where |C| sets of initial solutions are obtained and each solution starts with a different customer on route 1 using the sweep algorithm as presented in the previous section. The processes for constructing vehicle routes remain the same such that we start a new route with a customer who has the minimum polar angle. In this new procedure step 0 is added in the sweep algorithm, whilst step 1 of the sweep algorithm is modified. The pseudocode of both steps is therefore

Step 0

For all customers $i \in C$:

- Set flag = 0, this is a flag to indicate that we are going to construct route 1
- startNode = i, this would be the first customer on route 1
- Perform IS(startNode, flag)

end for

Step 1

The next customer to be visited on the current emerging route is

If flag = 0 **then**

Choose **startNode** to be the first customer on route 1 and set flag = 1

else

Choose the unrouted customer i, $i \in B$ that has the minimum polar angle to be the first customer on the route.

end if

In step 0 above we set a flag to indicate that we are going to construct a new set of initial solutions with a different initial customer as the first customer on route 1. In step 1 we add a customer to be visited on the current emerging route.

Note here that after running this procedure, we have |C| sets of initial solutions from the sweep algorithm. Even though the total computational time for constructing these initial solutions is high, the approach may be worthwhile for trying to find the best initial solution that we can have using the sweep algorithm.

3.6 Benchmark problems

Two sets of benchmark problems are used to test the algorithms in the thesis. The details of both problem sets are explained in the next sub-section.

3.6.1 Waste collection VRPTW benchmark problems

The main objective of this thesis is to develop good metaheuristics for solving waste collection VRPTW, particularly a real life waste collection benchmark problem obtained from Kim et al (2006). It consists of ten test problems, involving up to 2092 customers and 19 waste disposal facilities as publicly available at: http://www.postech.ac.kr/lab/ie/logistics/WCVRPTW_Problem/benchmark.html. The main characteristic of the test problems is shown in Table 3.5.1.

The driver rest time window in this problem is [11:00, 12:00] and an unlimited number of homogeneous vehicles are considered in this problem. The maximum number of customers on each vehicle route is limited to 500 customers. Moreover, the coordinates of the nodes (depot/customers/disposal facilities) provided in the data sets are in feet and then we calculate the distances between the nodes (in miles) using Manhattan distance as stated in Kim et al (2006).

Table 3.1: Characteristics of the waste collection VRPTW benchmark problems

Problem	Number of customers	Number of disposal facilities	Capacity of a vehicle	Capacity allowed for a vehicle per day
102	99	2	280	400
277	275	1	200	2200
335	330	4	243	400
444	442	1	200	400
804	784	19	280	10000
1051	1048	2	200	800
1351	1347	3	255	800
1599	1596	2	280	800
1932	1927	4	462	2000
2100	2092	7	462	2000

3.6.2 Multi-depot VRP with inter-depot routes (MDVRPI) benchmark problems

The second set of benchmark problems that we use to test our algorithms is a MDVRPI benchmark set generated by Crevier et al (2007). It consists of ten test problems, involving up to 288 customers and seven depots. Note here that in this problem, the depots can act as intermediate replenishment facilities along a vehicle route. The data sets and the best known solutions are available at http://www.hec.ca/chairedistributique/data/. The main characteristic of the test problems is shown in Table 3.2.

No time windows and driver rest period are considered in this problem. However, the number of collection vehicles used in this problem is limited for each test problem. Furthermore, the duration of a rotation (i.e. duration of a vehicle when it starts from the depot until it returns to the depot after servicing customers) must not exceed the total duration allowed for a vehicle. For these problems the distances between the nodes are calculated using Euclidean distance.

Table 3.2: Characteristics of the MDVRPI benchmark problems

Problem	Number of customers	Number of depots	Number of vehicles	Maximum duration of a rotation	Capacity of a vehicle
a2	48	5	4	600	150
b2	96	5	4	1150	200
c2	144	5	4	1700	250
d2	192	5	3	2250	300
e2	240	5	3	2800	350
f2	288	5	3	3350	400
g2	72	7	4	950	175
h2	144	7	4	1800	250
i2	216	7	3	2650	325
j2	288	7	3	3500	400

3.7 Results

The algorithms presented in this thesis are coded in C++ and run on a 3.16GHz pc (Intel Core2 Duo) with 3.23Gb memory. Computational results of all IS procedures used to construct initial routes for both sets of benchmark problems are reported in the next sections.

3.7.1 Computational results for the waste collection VRPTW problems

Table 3.3(a) shows the solutions from Kim et al (2006) for the waste collection VRPTW problems using their clustering heuristic (using simulated annealing), in terms of the number of vehicles used, total distance travelled and computation time. Note here that results are presented in Kim et al (2006) for an insertion heuristic (also using simulated annealing). However some of the insertion heuristic results reported are incorrect (involving for example fewer vehicles than can possibly be used) and based on Kim (2009) we disregard these results. In any event for seven of the ten test problems the results for the clustering heuristic are (in terms of distance travelled) better than the results for the insertion heuristic.

Table 3.3 a) Solutions from Kim et al (2006) for the waste collection VRPTW problems

Problem	Total number of vehicles used	Total distance (mile)	Total computation time (s)
102	3	205.1	3
277	3	527.3	10
335	6	205.0	11
444	11	87.0	16
804	5	769.5	92
1051	18	2370.4	329
1351	7	1039.7	95
1599	13	1459.2	212
1932	17	1395.3	424
2100	16	1833.8	408
Average			160

Their clustering heuristic (using simulated annealing) was implemented in the C language and run on a Microsoft Server 2003 machine with a 3.05 GHz Xeon processor with 1.0 GB of RAM. Computation times for Kim et al (2006) in the last column are taken from their paper, and relate to a different computer than we used. Utilising Dongarra (2009) it is possible to make an ***approximate*** estimate of the relative speed of the hardware involved. On this basis we estimate that the Kim et al (2006) heuristic would (on average) require 31 seconds on our 3.16GHz pc. Hereafter, we assume Kim et al (2006) average computation time is 31 seconds for comparison with our work. In terms of the number of vehicles used, in total they use 99 collection vehicles to solve the test problems.

Table 3.3(b) shows computational results of the test problems using our IS procedure (refer to Section 3.3). A number of these test problems contain customers for which the amount to be collected is zero, but in our results we explicitly visit these customers (based on Kim, 2009) to be comparable with the results of Kim et al (2006). Note too here that for some of these test problems the daily vehicle capacity is such that the vehicle finishes its work and returns to the depot before the driver rest period (associated with a lunch break). For these problems we regard the driver rest period as being taken at the depot.

Table 3.3 b) Solutions from our IS procedure for the waste collection VRPTW problems

Problem	Total number of vehicles used	Total distance (mile)	Total computation time (s)	% Improvement in distance over Kim et al (2006)
102	3	206.8	1	-0.83
277	3	473.8	1	10.15
335	6	213.3	2	-4.05
444	11	92.9	3	-6.78
804	6	863.3	8	-12.19
1051	17	2645.1	13	-11.59
1351	8	984.3	20	5.33
1599	14	1578.1	29	-8.15
1932	16	1346.1	41	3.53
2100	16	1823.6	49	0.56
Average			16.7	-2.40

The last column in Table 3.3(b) gives the percentage improvement in distance when compared to the result of Kim et al (2006), namely 100(Kim et al (2006) solution distance – our solution distance)/(Kim et al (2006) solution distance). Using this measure a positive number indicates we have a better result, a negative number indicates we have a worse result. The results indicate that four of the ten test problems using our IS procedure are better. However, on average our initial solutions travel approximately 2.4% more than Kim et al (2006) with 16.7 seconds of computational time. With respect to the number of vehicles used our initial solutions involve (in total) 100 vehicles, those of Kim et al (2006) 99 vehicles, so slightly worse.

Note that the solutions reported in Kim et al (2006) and given in Table 3.3(a) are the final solutions after applying their metaheuristics. They did not report their initial solutions for the test problems. Thus, the comparison results reported in this section are between their final solutions and our initial solutions. In later chapters we will present our metaheuristics to improve our initial solutions.

Figure 3.3 shows an example of initial routes from our IS procedure for the waste collection VRPTW problems, in particular the test problem 102. This problem consists of 99 customers and two disposal facilities. Solution for this problem as shown in Table 3.3(b) indicates that our solution travels approximately 0.83% more distance than Kim et al (2006). With respect to the number of vehicles used our solution involves three vehicles as in Kim et al (2006).

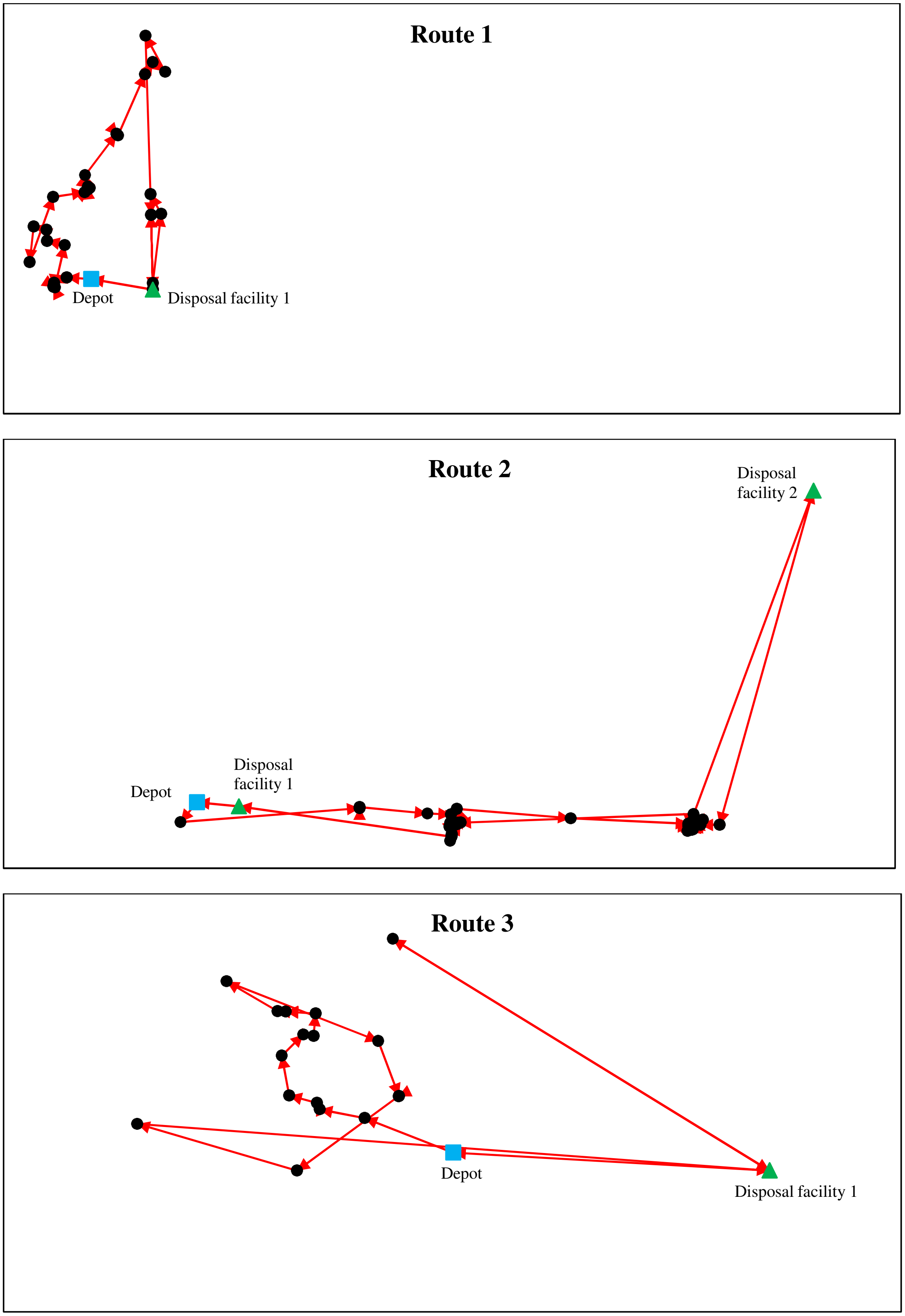


Figure 3.3: Initial routes for the 102 problem using our IS procedure

Table 3.3(c) shows the summary of the computational results of the same test problems using other IS procedures. Here for reasons of space we have chosen a summary comparison rather than present full detailed results. The table shows the algorithm used to solve the test problems, the section number where the algorithm is presented in this chapter, the total number of vehicles used, the average of the computation time in seconds and as well as the average of the percentage improvement in distance when compared to the result of Kim et al (2006). These averages are computed over the ten test problems given by Kim et al (2006).

Table 3.3 c) Solutions from other IS procedures for the waste collection VRPTW problems

Algorithm	Section	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure as Table 3.3(b)	3.3	100	16.7	-2.40
Farthest from depot	3.4.1	103	18.0	-6.41
Savings approach	3.5.1	103	39.8	-2.80
Sweep algorithm	3.5.2	155	19.0	-183.90
Different initial customer: a) current IS procedure	3.4.2	**98**	413.7	**4.75**
b) sweep algorithm	3.5.3	150	4331.2	-161.67

Of all the algorithms used in this thesis to construct initial solutions for the waste collection VRPTW problems, the solution from the different initial customer procedure based on our IS procedure shows the best result in terms of the number of vehicles used as well as the average of the percentage improvement in distance. Here, we use one less vehicle than Kim et al (2006) and a reduction of around 4.75% in average of the total distance travelled. Even though we use more computation time than Kim et al (2006), but note that in this procedure, we run |C| times for each test problem. The total number of customers involved in the ten test problems is 9940. Thus, in total we have 9940 initial solutions. The total computation time to construct the 9940 solutions is 4137 seconds. That means we only take less than 0.5 seconds for each initial solution.

Considering Table 3.3(c) it appears that the sweep algorithm is not suitable for constructing initial solutions for the test problems. Besides travelling more distance, the solutions also involve more vehicles than Kim et al (2006) even though we have run |C| times to get the best initial solution that we can have for every test problem. On average we can only reduce the distance from 183.9% (sweep algorithm) to 161.67% (different initial customer based on sweep algorithm) and reduce five vehicles (from 155 to 150 vehicles used). Yet, these solutions are still far from Kim et al (2006). Hence, we hope that these initial solutions can be significantly improved after applying our metaheuristics.

Initial solutions from the farthest from depot procedure and savings approach also on average travel more distance than those of Kim et al (2006), particularly 6.41% and 2.80% respectively. However, the solutions are not as bad as solutions from the sweep algorithm. Both procedures use four more vehicles than Kim et al (2006).

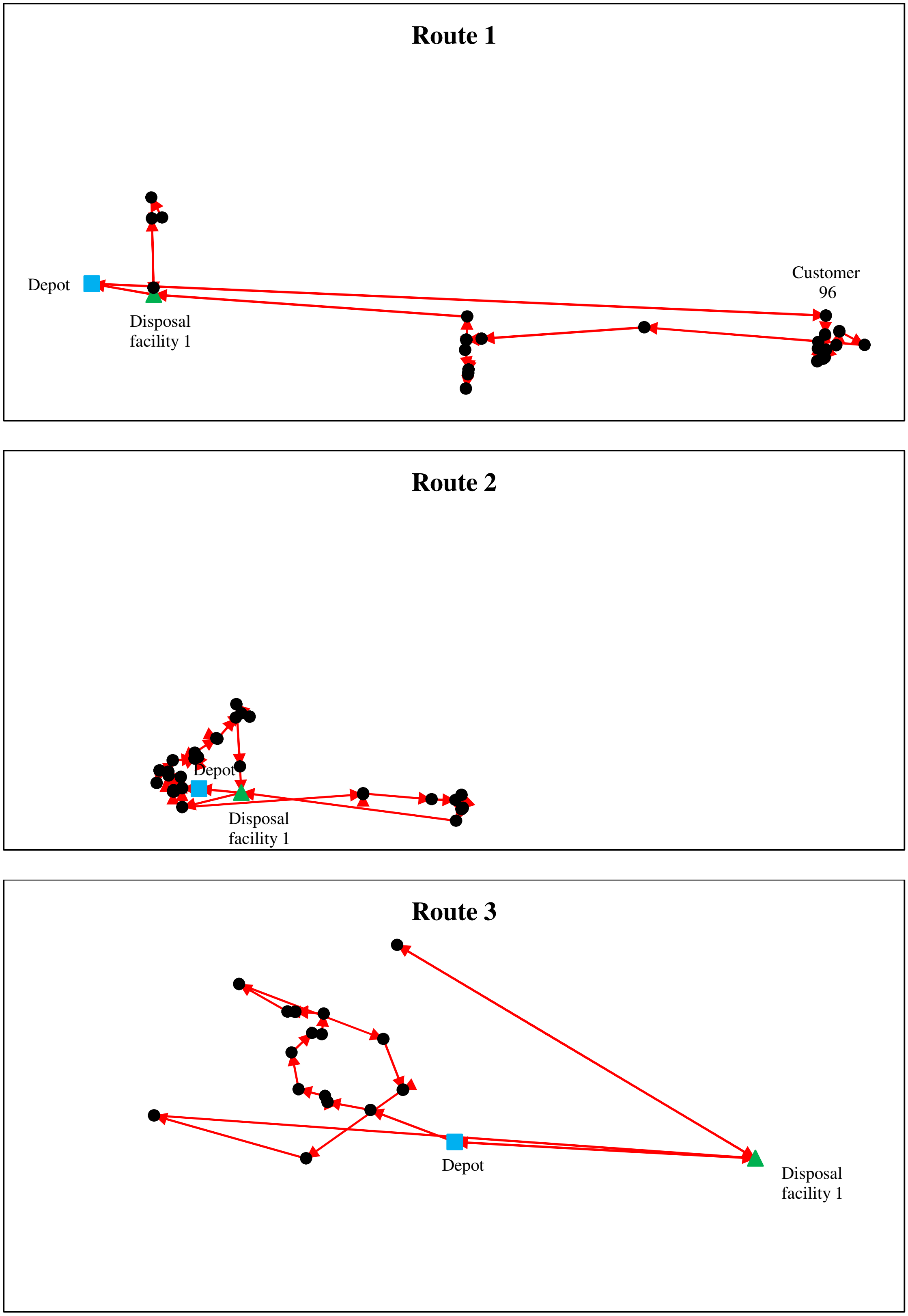


Figure 3.4: Initial routes for the 102 problem using the different initial customer (here customer 96) based on our IS procedure

Since the different initial customer based on our IS procedure produces the best solution (as shown in Table 3.3(c)) for the waste collection VRPTW problems, Figure 3.4 shows an example of initial routes for the 102 problem constructed by this procedure. Note here that this procedure produces |C|=99 number of solutions for the 102 problem. Each solution starts with a different customer as the first customer on the first route constructed. Computational results indicate that the first route (as shown in the figure) which starts with customer 96 has the lowest distance travelled compared to the other solutions. Comparison result with the solution from Kim et al (2006) shows that this solution reduces the total distance travelled of around 10.14% and uses three vehicles as in Kim et al (2006).

3.7.2 Computational results for the MDVRPI problems

Table 3.4(a) shows computational results of the MDVRPI problems from Crevier et al (2007) for their tabu search algorithm as well as integer programming model, in terms of the number of vehicles used and total distance travelled. Computation time is not reported in their paper.

Their tabu search algorithm was coded in the C language and run on a Prosys, 2 Ghz computer. In total they use 31 vehicles to solve the test problems. Computational results using our IS procedure (refer to Section 3.3) is presented in Table 3.4(b).

Table 3.4 a) Solutions from Crevier et al (2007) for the MDVRPI problems

Problem	Total number of vehicles used	Total distance
a2	4	997.94
b2	3	1307.28
c2	3	1747.61
d2	3	1871.42
e2	3	1942.85
f2	3	2284.35
g2	3	1162.58
h2	3	1587.37
i2	3	1972.00
j2	3	2294.06

Table 3.4 b) Solutions from our IS procedure for the MDVRPI problems

Problem	Total number of vehicles used	Total distance	Total computation time (s)	% Improvement in distance over Crevier et al (2007)
a2	5	1604.36	1	-60.77
b2	3	1636.97	1	-25.22
c2	3	2344.79	1	-34.17
d2	3	2556.26	1	-36.59
e2	3	2802.56	1	-44.25
f2	3	3326.04	2	-45.60
g2	4	1572.16	1	-35.23
h2	3	2113.21	1	-33.13
i2	3	2678.79	1	-35.84
j2	3	3439.59	2	-49.93
Average			1.2	-40.07
Number of infeasible solutions	1			

The last row in Table 3.4(b) shows the number of infeasible solutions obtained from the IS procedure in solving the ten test problems. And the last column in the table gives the percentage improvement in distance when compared to the result of Crevier et al (2007), namely 100(Crevier et al (2007) solution distance – our solution distance)/(Crevier et al (2007) solution distance). Using this measure a positive number indicates we have a better result, a negative number indicates we have a worse result. The results clearly show that solutions from Crevier et al (2007) outperform our initial solutions using the IS procedure presented in section 3.3. Besides involving two extra collection vehicles (for test problems a2 and g2), we are also travelling more distance than those of Crevier et al (2007), on average over these ten problems approximately 40.07% more with 1.2 seconds of computational time.

Note here that the solutions reported in Crevier et al (2007) are the final solutions after applying their metaheuristics. They did not report their initial solutions for the test problems. Thus in this section, we are also comparing our initial solutions with their final solutions. In later chapters we will present our metaheuristics to improve the initial solutions.

As stated before, our IS procedure is developed for the waste collection VRPTW problems which have an unlimited number of vehicles. Thus when it is applied to this problem, unsurprisingly there are some of the initial solutions are not feasible. For example initial solution for a2 problem in Table 3.4(b) actually is not feasible. The total number of vehicles available for a2 is four (refer to Table 3.2) but our initial solution needs five vehicles. However, this is an initial solution so we hope that this solution can be improved after the improvement phase.

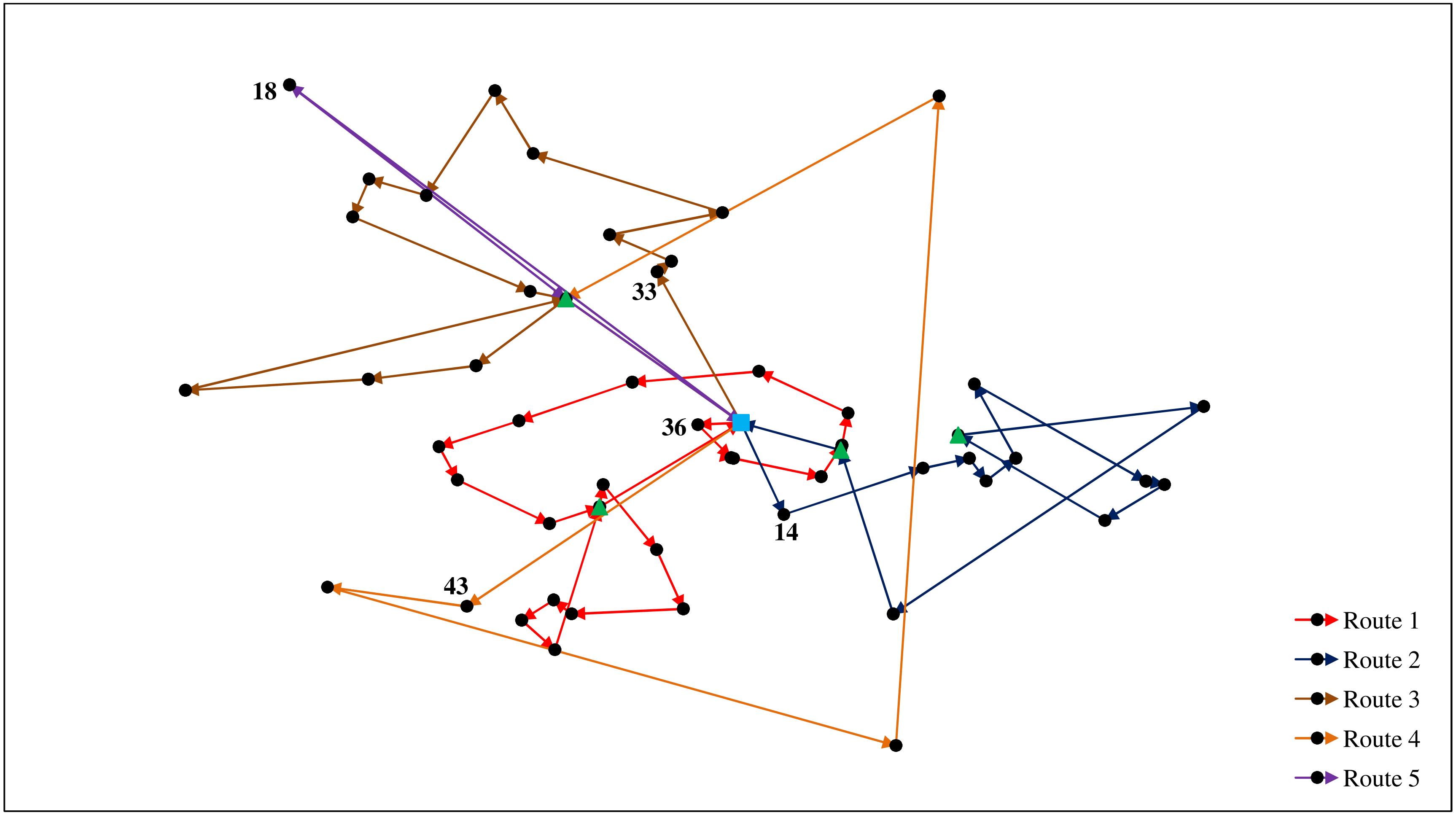


Figure 3.5: Initial routes for the a2 problem using our IS procedure

Figure 3.5 shows five initial routes constructed using our IS procedure for the a2 problem. The first customer on each route is shown in the figure. These customers are the closest customers to the depot when the route is constructed. For example customer 36 is the closest customer to the depot when route 1 is constructed and customer 14 is the closest to the depot when route 2 is constructed. When route 5 (last route) is constructed there is only one customer left (customer 18) with waste not having being collected. As mentioned earlier the initial solution for this problem is not feasible because it involves one extra vehicle. Because of route 5 has only one customer, this route can possibly be eliminated when we apply our route improvement procedures in later chapters.

Other IS procedures as applied to the waste collection VRPTW problems are also applied to these test problems to investigate the effectiveness of other IS procedures for this problem. The summary of the computational results using these procedures is shown in Table 3.4(c) which has the same format as Table 3.3(c) except that the last column in this table shows the number of infeasible solutions obtained from the procedures in solving the ten test problems.

Results in Table 3.4(c) also indicate that solutions from Crevier et al (2007) are better than other IS procedures applied on the test problems. Even though initial solutions from the savings approach and the different initial customer procedure based on our IS procedure involve the same number of vehicles used as in Crevier et al (2007), but on average both procedures travel approximately 20.73% more distance for the savings approach and 25.28% more distance for the different initial customer based on our IS procedure.

Table 3.4 c) Solutions from other IS procedures for the MDVRPI problems

Algorithm	Section	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
Current IS procedure as Table 3.4(b)	3.3	33	1.2	-40.07	1
Farthest from depot	3.4.1	34	1.2	-42.94	1
Savings approach	3.5.1	31	1.2	-20.73	0
Sweep algorithm	3.5.2	43	1.4	-164.05	8
Different initial customer: a) current IS procedure	3.4.2	31	1.7	-25.28	0
b) sweep algorithm	3.5.3	40	10.5	-152.37	5

Once again, it appears that the sweep algorithm is not suitable for constructing initial solutions for the test problems. Besides travelling more distance, the solutions also used more vehicles than what they have. Thus, most of the initial solutions obtained from this procedure are not feasible.

3.8 Conclusion

In this chapter we have presented a number of IS procedures used for constructing initial solutions for two benchmark problem sets, waste collection VRPTW and MDVRPI. Some of the IS procedures are adapted from the literature.

Because both problem sets have some characteristic differences such as

- the size of the test problems (i.e. waste collection VRPTW problems involved up to 2092 customers, whereas the MDVRPI problems only involved up to 288 customers)
- time windows of the nodes (depot/customer/disposal facility)
- limited/unlimited number of vehicles
- rest time period

the best initial solution for both problems are obtained from the different IS procedures. For example, computational results show that the best initial solution for the waste collection VRPTW problems is obtained from the different initial customer based on our IS procedure. The logic here is that our IS procedure is designed for the waste collection VRPTW problems. Thus, when we run this procedure |C| times, each time with a different initial customer as the first customer on route 1, we found the best initial solution for each test problem. Computational results show that our initial solution without applying metaheuristics is already better than solutions by Kim et al (2006) in terms of the total number of vehicles used as well as the total distance travelled. Hence, if we spend more computation time to run |C| times for the ten test problems, this solutions show that our IS procedure is able to construct good feasible initial routes for the waste collection VRPTW problems.

On the other hand, initial solutions from the savings approach and the different initial customer procedure based on our IS procedure are the best procedures for the MDVRPI problems because both procedures manage to produce feasible initial routes for the ten test problem. However in terms of the total distance, the solution from the savings approach travelled less distance than the different initial customer solution procedure.

CHAPTER 4

ROUTE IMPROVEMENT PROCEDURES

This chapter is divided into five sections. The first section presents our procedure to evaluate a given route, which involves inserting into the route (if necessary) disposal facility visits. In the second and the third sections, we present procedures to improve a solution, both in terms of the distance travelled and in terms of the number of vehicles used. In the fourth section, computational results for both procedures tested on two benchmark problem sets are reported. Finally, a summary of this chapter is presented.

4.1 Route evaluation

In this section we indicate how we evaluate a given route. One complication here is that in the local search procedure we present later in this chapter, we move customers between routes. If we move a customer onto a route then it is possible that the route after addition of the customer will be infeasible when we evaluate it (e.g. because the vehicle exceeds its collection capacity). However if we were to schedule into the route an extra disposal facility visit the route may become feasible. Preliminary computational experience indicated that incorporating extra disposal facility visits was of benefit, and so in evaluating a given route we allow such extra visits to be incorporated.

In a similar fashion there may be benefit in allowing the time at which the rest period occurs to vary from the time that was initially scheduled as we evaluate a route and so we also allow this to change (although the rest period must still occur within its time window). **In evaluating a given route we regard it as comprising a fixed sequence of places - starting at the depot, then a mix of customers and disposal facilities, finally a disposal facility (to empty the vehicle), followed by the depot.**

Note here that in our work in evaluating the quality of any given set of feasible vehicle routes we evaluate them using total distance travelled. We do not constrain the distance travelled by vehicles, rather we constrain vehicle operating times via the depot time window. Amending the algorithms presented below to deal with vehicle distance constraints is however a simple task. In pseudocode our procedure for route evaluation is:

Start the route at time E_0 (set $T=E_0$)

Repeat until all places on the route have been dealt with:

- Perform Step 2 of our IS procedure (refer to Section 3.3 in Chapter 3) to schedule the rest period if possible
- If travelling to the next place in the fixed sequence would exceed the vehicle capacity then schedule in an extra visit to the nearest disposal facility; formally suppose the current place at the end of the route is customer r and the current time is T, then insert a visit to disposal facility n(r,T) at this point in the sequence
- Travel to the next place (customer/disposal facility/depot) in the fixed sequence

- If the vehicle arrives before the time window for the place opens then wait until the time window opens (we wait as we are trying to operate the sequence)
- Deal with this place (collection or disposal or arrival back at the depot)

As we run through the fixed sequence we update the current time T, also keeping track of the loads – where the procedure returns INFEASIBLE if at any point we violate the constraints of the problem (e.g. vehicle load exceeded or the vehicle arrives after the time window for a place has closed), otherwise the procedure returns FEASIBLE (also returning the sequence used since we may have added extra disposal facility visits, and the total distance associated with the route).

4.2 Route improvement – local search

In order to improve vehicle routes we adopt a local search procedure which is based on our route improvement procedure presented in Benjamin and Beasley (2010) (refer to Appendix 1). The procedure published in that paper is tested only on the waste collection VRPTW problems. The computational results show that our routes outperform previous work presented in the literature. However when it was tested on the MDVRPI problems, the solutions from Crevier et al (2007) are better than our solutions. Thus in a further attempt to improve the solution, we consider more movements between the customers on the routes to be evaluated.

In Benjamin and Beasley (2010), two different phases were associated with this procedure:

- moving customers/disposal facilities elsewhere on the same route; also changing disposal facilities on the same route
- interchanging the positions of two customers in the different routes.

However, in this thesis we consider moving customers and interchanging the positions of two customers on the same route as well as on the different routes. We deal with each of these in turn. In both phases, we use the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. The procedures for both phases are presented in the next two sub-sections.

4.2.1 Phase 1

In this phase we evaluate repositioning customers and the disposal facilities. In pseudocode we:

For all customers $i \in C$:

For all customers $j \in N(i,K)$:

- Move j immediately before/after i (here we check positioning j before/after i)
    - Evaluate these new routes with j moved to this new position and if they are better than the original route (both are FEASIBLE and of lower total distance) then keep them, else do not. Note here that this includes two cases, one where i and j are on the same route, one where i and j are on different routes

end for

end for

For all routes:

For all disposal facilities $i \in D$ on the route:

- Remove i from the route
- Add i to every possible position on the route in turn
- Evaluate this new route with i added to this new position and if it is better than the original route (FEASIBLE and of lower total distance) then keep it, else do not

end for

end for

Preliminary computational experience indicated that we could also improve routes by changing disposal facilities. This can happen (for example) if we have a disposal facility visited as the last place on the vehicle route before travel back to the depot, but in fact there is a better disposal facility to use in terms of travel back to the depot. In addition it is worthwhile to check for whether disposal facilities on the route can be removed (since in the route evaluation procedure we only ever add disposal facilities, never remove them). Therefore as part of this phase we also do:

For all routes:

For all disposal facilities $i \in D$ on the route:

- Remove disposal facility i from the route
- Evaluate this new route with the disposal facility removed and if it is better than the original route (FEASIBLE and of lower total distance) then keep it, else do not

end for

end for

For all routes:

For all disposal facilities $i \in D$ on the route:

For all disposal facilities $j \in D$, $j \neq i$:

- Replace disposal facility i on the route by disposal facility j
- Evaluate this new route with the disposal facilities changed and if it is better than the original route (FEASIBLE and of lower total distance) then keep it, else do not

end for

end for

end for

4.2.2 Phase 2

In a further attempt to improve the solution, in this phase we interchange the positions of two customers on the same route as well as on the different routes. In pseudocode we:

For all customers $i \in C$:

For all customers $j \in N(i,K)$:

- we interchange customers i and j, i.e. customer i moves to the position that customer j occupied on its route and customer j moves to the position that customer i occupied on its route
- Evaluate the two routes that are involved in this interchange. If both are FEASIBLE and their total distance is lower than the total distance for the two routes before the interchange then keep the interchange, else do not. Note here that this includes two cases, one where i and j are on the same route, one where i and j are on different routes

end for

end for

Computationally we repeat phases 1 and 2 in turn until no further improvement can be achieved. We will then have a locally optimal solution. Hereafter, finding the initial solution followed by the phase 1 and phase 2 procedures are called ISP1P2 in this thesis.

4.3 Vehicle reduction (VR) procedure

Our solution procedure has no direct control over the number of vehicles used, although in this thesis we also attempt to minimise the number of vehicles used by utilising a vehicle as much as possible. Examination of preliminary computational results indicated that, for some test problems, the number of customers serviced on the last vehicle route constructed from our IS procedure (refer to Section 3.3) was so small that (given judicious rearrangement of customers on earlier routes) it might well be possible to reduce the number of vehicles used. Thus in reducing the number of vehicles used, the VR procedure as presented in Benjamin and Beasley (2010) moved customers from the last vehicle route constructed to earlier routes (provided that this is feasible, and irrespective of the effect on distance travelled).

However, solution obtained from other IS procedures (i.e. savings approach, sweep algorithm etc) shows that the small number of customers serviced is not only on the last route. It may also happen on earlier routes. In this thesis, we change the VR procedure presented in Benjamin and Beasley (2010) to move customers from the route that has the smallest number of customers (which may or may not be the last route constructed in the initial solution) to other routes of the solution. In pseudocode we:

Repeat until no more customers can be moved from the route that has the smallest number of customers (say route α):

For all customers $i \in C$ that are on route α:

- Add i to every possible position on every other route in turn
- Evaluate this new route with i added to this new position and if it is FEASIBLE then keep it, else do not

end for

Perform phases 1 and 2 above, but excluding from consideration in those phases route α (since we are seeking to eliminate all customers from that route)

end repeat

If all customers have been moved from route α then keep the routes else do not

The logic here is that we, provided it is feasible, move customers off the route that has the smallest number of customers to other routes, where we use ISP1P2 to reorder customers on these other routes (thereby potentially enabling further customers to be moved off the route). Note here that if we are already using a minimal number of vehicles, as is the case if $\max[|C|/S^*, \sum_{i \in C} q_i/Q^*]$ (when rounded up to the nearest integer) is equal to the number of vehicles used, there is no point in applying this procedure.

4.4 Results

This section reports improvements results of the initial solutions (reported in Chapter 3) using the ISP1P2 and VR procedures. For both sets of benchmark problems waste collection VRPTW and MDVRPI, two sets of solutions are reported. First, a set of solutions without VR procedure (i.e. initial solutions are improved using the ISP1P2) is reported. Then, we report the effect of the VR procedure on the solution.

4.4.1 Computational results for the waste collection VRPTW problems

In this section, computational results of the waste collection VRPTW problems using our improvement procedures are reported. Note here that we use the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. In improving the solutions, preliminary computational experience indicated that the suitable size of the neighbour set for the waste collection VRPTW problems is K=50. Computational results using our ISP1P2 when K = 50 are reported in Table 4.1(a) and (b). Recall here that ISP1P2 starts from the solution given after our IS procedure and the computation times given in both tables Table 4.1(a) and (b) include the time taken to generate this solution. In Table 4.1(a) we also include final solution from Kim et al (2006) and our IS solution as presented in Chapter 3 (refer to Table 3.3(a) and (b), respectively).

Examining Table 4.1(a) it is clear that our ISP1P2 solutions use less distance for all test problems than those of Kim et al (2006), on average approximately 9.31% less. With respect to the number of vehicles used, ISP1P2 solutions also involve (in total) 100 vehicles as our IS solution, those of Kim et al (2006) 99 vehicles, so slightly worse. In terms of computation time, ISP1P2 solutions take a longer time than the Kim et al (2006) heuristic, but produce solutions involving significantly less distance.

Table 4.1 a) ISP1P2 solutions for the waste collection VRPTW problems for our IS procedure

Problem	Algorithm	Total number of vehicles used	Total distance (mile)	Total computation time (s)	% Improvement in distance over Kim et al (2006)
102	Kim et al (2006)	3	205.1	3	
	IS	3	206.8	1	-0.83
	ISP1P2	3	181.9	3	11.31
277	Kim et al (2006)	3	527.3	10	
	IS	3	473.8	1	10.15
	ISP1P2	3	462.9	15	12.21
335	Kim et al (2006)	6	205.0	11	
	IS	6	213.3	2	-4.05
	ISP1P2	6	196.8	28	4.00
444	Kim et al (2006)	11	87.0	16	
	IS	11	92.9	3	-6.78
	ISP1P2	11	79.9	57	8.16
804	Kim et al (2006)	5	769.5	92	
	IS	6	863.3	8	-12.19
	ISP1P2	6	716.7	199	6.86
1051	Kim et al (2006)	18	2370.4	329	
	IS	17	2645.1	13	-11.59
	ISP1P2	17	2131.8	257	10.07
1351	Kim et al (2006)	7	1039.7	95	
	IS	8	984.3	20	5.33
	ISP1P2	8	907.5	305	12.72
1599	Kim et al (2006)	13	1459.2	212	
	IS	14	1578.1	29	-8.15
	ISP1P2	14	1395.2	362	4.39
1932	Kim et al (2006)	17	1395.3	424	
	IS	16	1346.1	41	3.53
	ISP1P2	16	1187.8	695	14.87
2100	Kim et al (2006)	16	1833.8	408	
	IS	16	1823.6	49	0.56
	ISP1P2	16	1676.8	829	8.56
Average	Kim et al (2006)	99		160	
	IS	100		16.7	-2.40
	ISP1P2	100		**275**	**9.31**

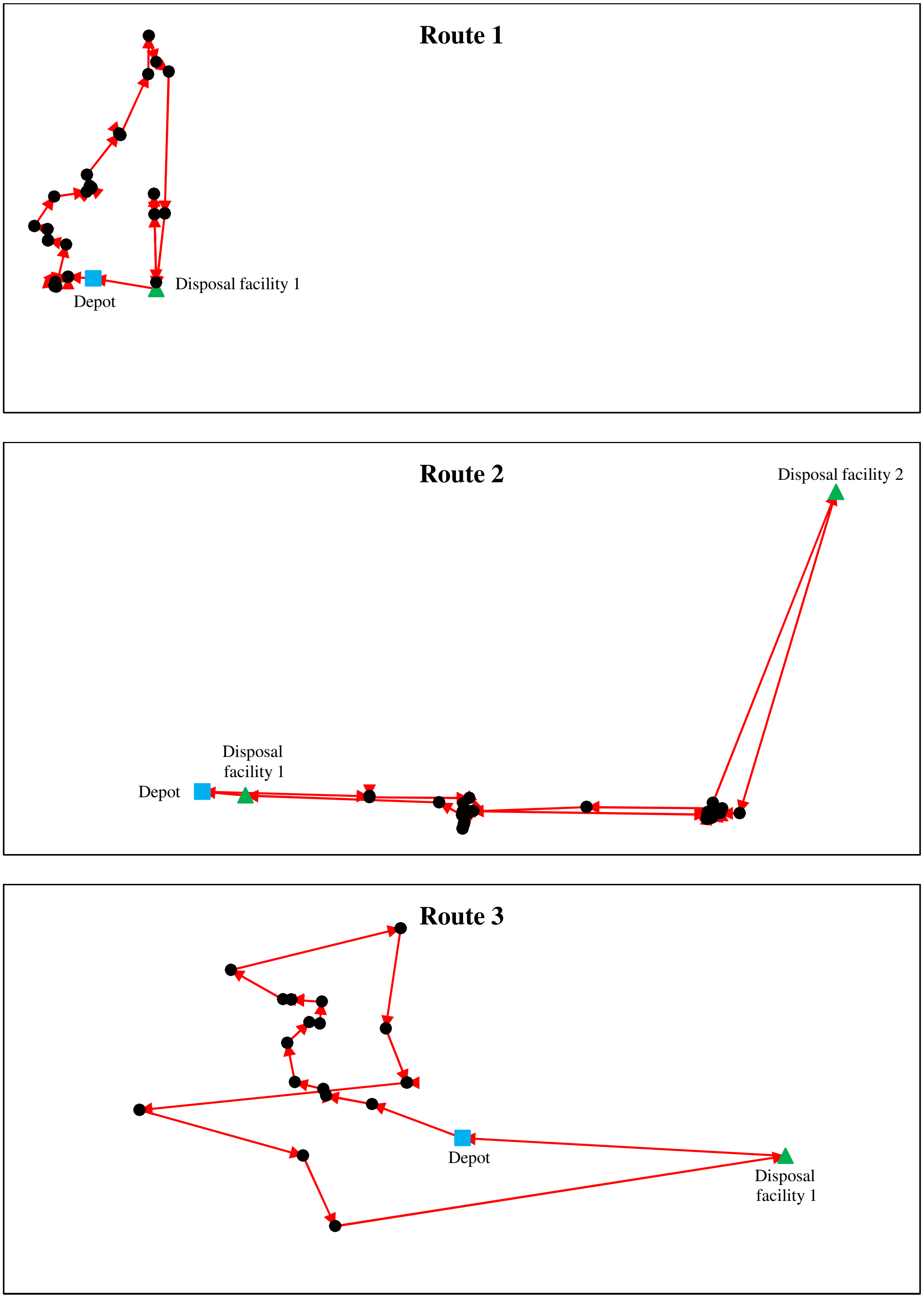


Figure 4.1: ISP1P2 routes for the 102 problem starting from the initial solution using our IS procedure

Figure 4.1 shows our ISP1P2 routes for the 102 problem. Comparison of the result with the initial solution shows that our phase 1 and phase 2 procedures reduce the total distance travelled of around 100(206.8-181.9)/206.8=12.04%. Thus, our solution for the 102 problem now travels 11.31% less distance than Kim et al (2006).

Table 4.1(b) shows ISP1P2 solutions starting from the solution given after other IS procedures. For easy comparison between solutions from the IS and ISP1P2, in Table 4.1(b) we include IS solutions reported in the last chapter (refer to Table Table 3.3(c)).

Table 4.1 b) ISP1P2 solutions for the waste collection VRPTW problems for other IS procedures

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
IS – current IS procedure as Table 3.3(b)	100	16.7	-2.40
ISP1P2 as Table 4.1(a)	100	275.0	9.31
IS - Farthest from depot	103	18.0	-6.41
ISP1P2	103	388.3	6.23
IS - Savings approach	103	39.8	-2.80
ISP1P2	101	411.2	6.94
IS - Sweep algorithm	155	19.0	-183.90
ISP1P2	154	290.7	-18.76
IS - Different initial customer: a) current IS procedure	**98**	413.7	**4.75**
ISP1P2	**98**	657.0	**13.00**
b) sweep algorithm	150	4331.2	-161.67
ISP1P2	150	4611.4	-16.16

Results in Table 4.1(b) indicate that ISP1P2 starting from the solution given after the different initial customer based on our IS procedure produces the best solutions (on average). The distance travelled is reduced by up to 13% with one less vehicle used than Kim et al (2006). In addition, ISP1P2 starting from the solutions given after the farthest from depot procedure and the savings approach also produce better results than Kim et al (2006) with a reduction of 6.23% and 6.94%, respectively in the distance travelled. However in terms of the number of vehicles used, both solutions involve more vehicles than Kim et al (2006). On the other hand, ISP1P2 starting from the solutions given after the sweep algorithm and the different initial customer based on sweep algorithm are still worse compared to Kim et al (2006), in terms of distance travelled and also the number of vehicles used.

Figure 4.2 shows our best solution for the 102 problem starting from the solution given after the different initial customer based on our IS procedure. This solution travels 158.28 miles (approximately 22.8% less distance than Kim et al (2006)) with three vehicles.

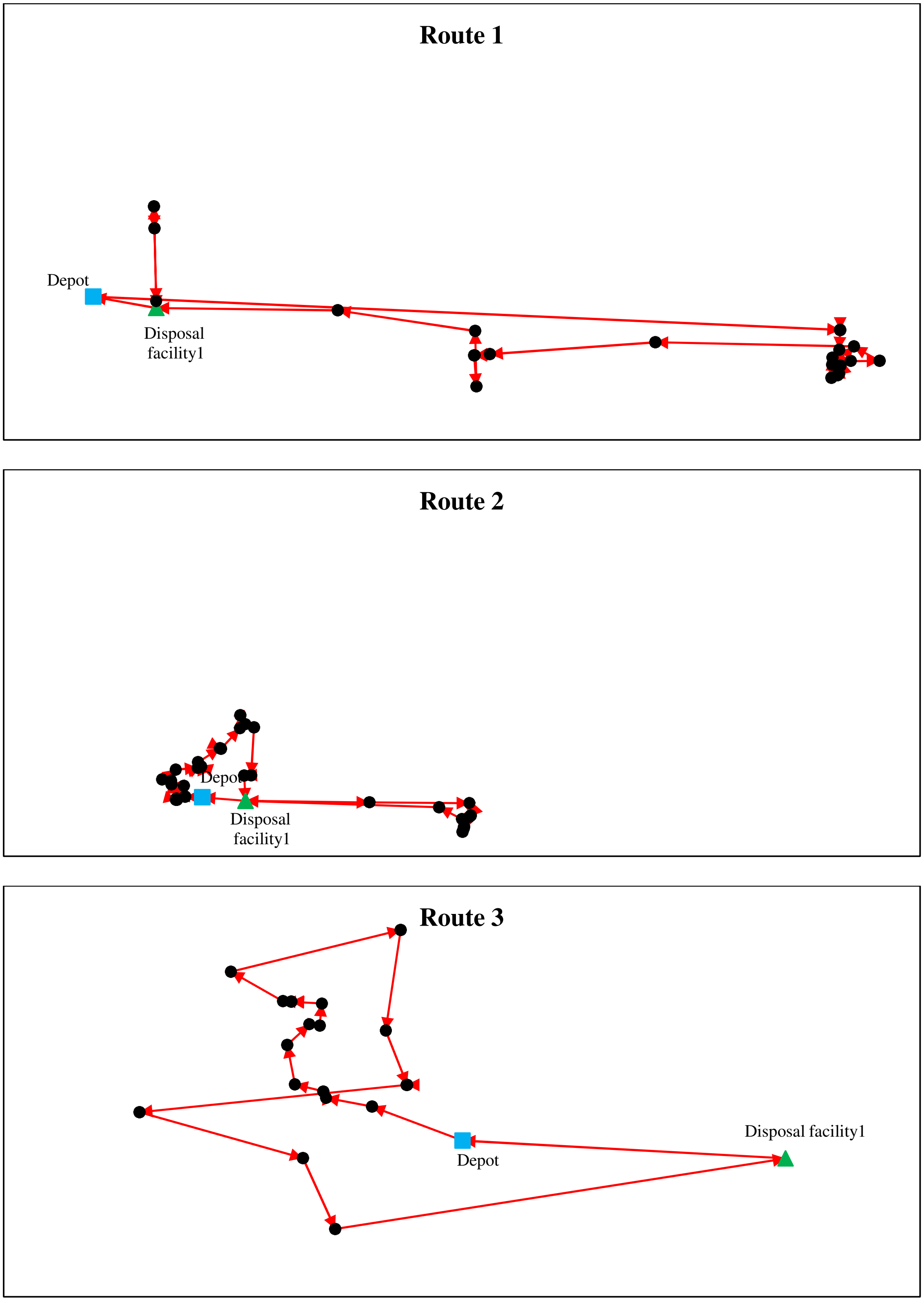


Figure 4.2: ISP1P2 routes for the 102 problem starting from the initial solution using the different initial customer (here customer 96) based on our IS procedure

Results in Table 4.1(a) and (b) were produced without using our VR procedure. To illustrate the effect of the VR procedure, the results obtained when it is applied (denoted by VRISP1P2) are shown in Table 4.1(c) and (d), respectively. For reasons of space we only show in Table 4.1(c) those problems where a reduction in the number of vehicles was achieved. Note here that for two of the problems shown in Table 4.1(a) (problems 102 and 335) our solutions already use the minimal number of vehicles (as can be deduced from consideration of total customer demand and vehicle capacity).

Table 4.1 c) VRISP1P2 solutions for the waste collection VRPTW problems for our IS procedure

Problem	Algorithm	Total number of vehicles used	Total distance (mile)	Total computation time (s)	% Improvement in distance over Kim et al (2006)
1351	Kim et al (2006)	7	1039.7	95	
	IS	8	984.3	20	5.33
	ISP1P2	8	907.5	305	12.72
	VRISP1P2	7	1010.9	545	2.77
Average	Kim et al (2006)	99		160.0	
	IS	100		16.7	-2.40
	ISP1P2	100		275	9.31
	VRISP1P2	99		**296.6**	**8.32**

Table 4.1(c) has the same format as Table 4.1(a), except that now we apply our vehicle reduction procedure to the routes that result from ISP1P2. For ease of comparison the averages shown at the foot of Table 4.1(c) are the averages over all ten problems, computed by combining the results for the problem explicitly shown in Table 4.1(c) with the results shown in Table 4.1(a) for the other nine problems. Note here that the average time given at the foot of Table 4.1(c) includes the time for applying our vehicle reduction procedure to all problems (whether successful or not). Considering Tables

4.1(a) and (c) then with respect to the number of vehicles used our solutions now involve (in total) 99 vehicles, as in Kim et al (2006).

Table 4.1 d) VRISP1P2 solutions of the waste collection VRPTW problems for other IS procedures

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
IS - Current IS procedure	100	16.7	-2.40
ISP1P2	100	275.0	9.31
VRISP1P2	99	296.6	8.32
IS - Farthest from depot	103	18.0	-6.41
ISP1P2	103	388.3	6.23
VRISP1P2	101	430.0	4.30
IS - Savings approach	103	39.8	-2.80
ISP1P2	101	411.2	6.94
VRISP1P2	99	479.3	7.05
IS - Sweep algorithm	155	19.0	-183.90
ISP1P2	154	290.7	-18.76
VRISP1P2	107	2427.9	-22.86
IS - Different initial customer: a) current IS procedure	98	413.6	4.75
ISP1P2	98	657.0	13.00
VRISP1P2	97	678.2	11.26
b) Sweep algorithm	150	4331.2	-161.67
ISP1P2	150	4611.4	-16.16
VRISP1P2	109	6545.3	-21.58

Table 4.1(d) shows the improvement solution of ISP1P2 for other IS solutions after applying our VR procedure. The results indicate that all VRISP1P2 solutions use less vehicles compared to ISP1P2 solutions. The decreasing numbers in the last column indicate that VRISP1P2 solutions involve more distance than ISP1P2 solutions even though they use fewer vehicles except for VRISP1P2 solution for savings approach. This solution has improved in terms of number of vehicles used (from 101 to 99

vehicles) as well as the total distance travelled (from 6.94% to 7.05%). VRISP1P2 solution for the sweep algorithm and the different initial customer based on sweep algorithm are still worse than Kim et al (2006).

4.4.2 Computational results for the MDVRPI problems

In this section, computational results of the MDVRPI problems using our improvement procedures are reported. Note here that we use the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. Since the size of the test problems of MDVRPI is much smaller than waste collection VRPTW, we choose to evaluate all possible moves/interchanges of customers on the routes. Thus, for this problem we use K=|C| for the size of the neighbour set. Computational results using our ISP1P2 when K=|C| are reported in Table 4.2(a) and (b). Recall here that ISP1P2 starting from the solutions given after other IS procedures presented in the previous chapter and the computation times given in both tables include the time taken to generate these solutions. In Table 4.2(a) we also include final solution from Crevier et al (2007) and the IS solutions as reported in Chapter 3 (refer to Table 3.4(a) and (b), respectively).

Table 4.2 a) ISP1P2 solutions for the MDVRPI problems for our IS procedure

Problem	Algorithm	Total number of vehicles used	Total distance	Total computation time (s)	% Improvement in distance over Crevier et al (2007)
a2	Crevier et al (2007)	4	997.94		
	IS	5	1604.36	1	-60.77
	ISP1P2	5	1134.93	2	-13.73
b2	Crevier et al (2007)	3	1307.28		
	IS	3	1636.97	1	-25.22
	ISP1P2	3	1503.85	7	-15.04
c2	Crevier et al (2007)	3	1747.61		
	IS	3	2344.79	1	-34.17
	ISP1P2	3	1978.75	23	-13.23
d2	Crevier et al (2007)	3	1871.42		
	IS	3	2556.26	1	-36.59
	ISP1P2	3	2130.00	46	-13.82
e2	Crevier et al (2007)	3	1942.85		
	IS	3	2802.56	1	-44.25
	ISP1P2	3	2415.59	101	-24.33
f2	Crevier et al (2007)	3	2284.35		
	IS	3	3326.04	2	-45.60
	ISP1P2	3	2639.74	138	-15.56
g2	Crevier et al (2007)	3	1162.58		
	IS	4	1572.16	1	-35.23
	ISP1P2	4	1352.61	3	-16.35
h2	Crevier et al (2007)	3	1587.37		
	IS	3	2113.21	1	-33.13
	ISP1P2	3	1767.10	15	-11.32
i2	Crevier et al (2007)	3	1972.00		
	IS	3	2678.79	1	-35.84
	ISP1P2	3	2347.10	66	-19.02
j2	Crevier et al (2007)	3	2294.06		
	IS	3	3439.59	2	-49.93
	ISP1P2	3	2835.93	139	-23.62
Average	IS	33		1.2	-40.07
	ISP1P2	33		**54**	**-16.60**
Number of infeasible solutions	1				

Note here the negative numbers on the last column indicate that our solutions are worse than Crevier et al (2007). Thus, results in Table 4.2(a) clearly show that for the ten test problems our ISP1P2 solutions are worse than Crevier et al (2007). Besides involving two extra collection vehicles (for test problems a2 and g2), we also travel more distance than those of Crevier et al (2007), on average over these ten problems approximately 16.60% more with 54 seconds of computational time and one infeasible solution, namely the solution of the a2 problem.

Figure 4.3 shows our ISP1P2 solution for the a2 problem. Because of our solution still involves five vehicles, the solution for this problem is not feasible. Besides involving one extra vehicle, our solution also travels approximately 13.7% more distance than Crevier et al (2007). However compared with the initial solution, our phase 1 and phase 2 procedures produce less total distance travelled. In particular, our ISP1P2 routes travels approximately 100(1604.36-1134.93)/1604.36=29.3% less.

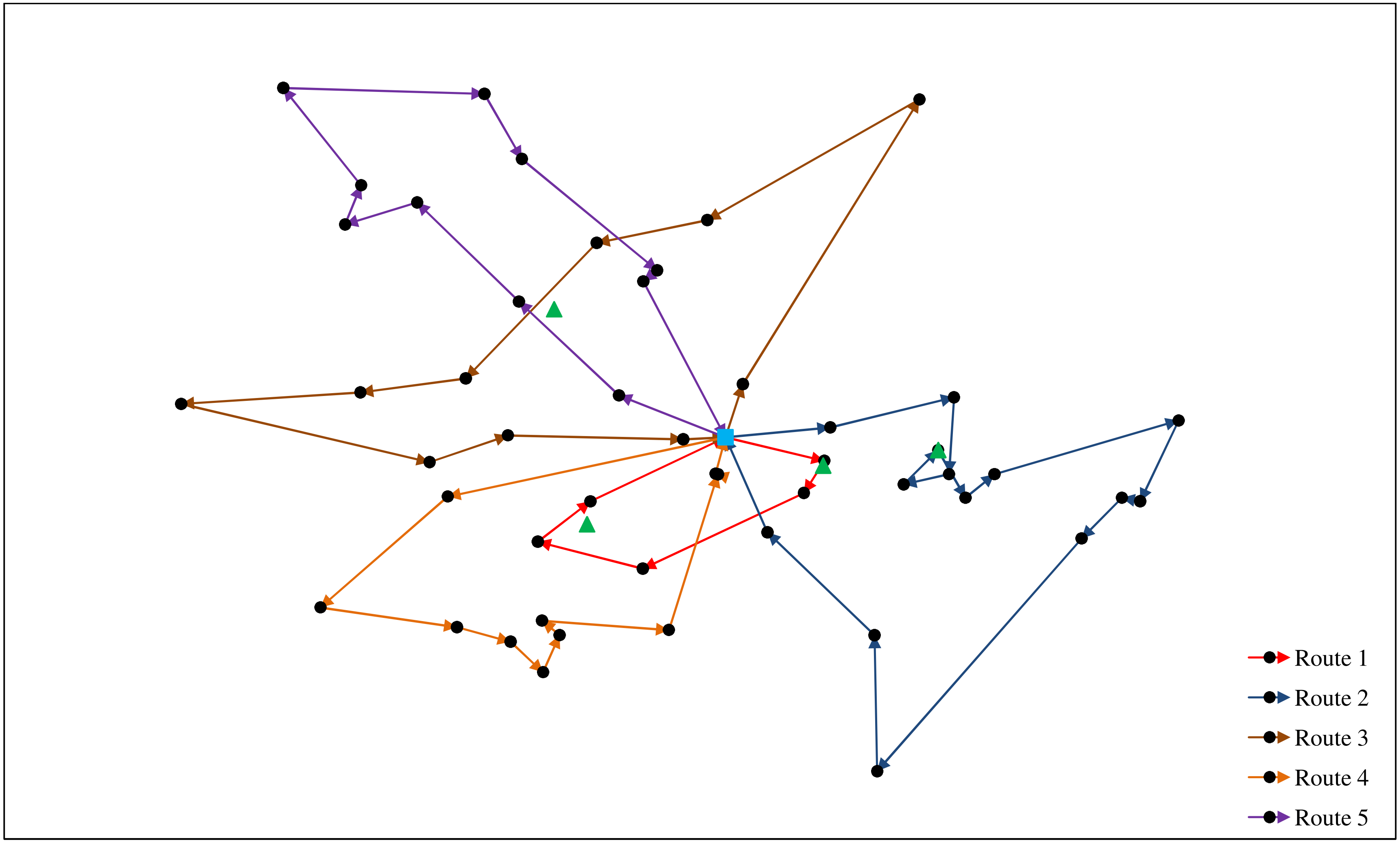


Figure 4.3: ISP1P2 routes for the a2 problem starting from the initial solution using our IS procedure

Table 4.2(b) shows ISP1P2 starting from the solution given after other IS procedures. In this table we include IS solutions shown in the last chapter (refer to Table 3.4(c)). The computation times given in Table 4.2(b) include the time taken to generate these solutions.

Table 4.2 b) ISP1P2 solutions for the MDVRPI problems for other IS procedures

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
IS - Current IS procedure as Table 3.4(b)	33	1.2	-40.07	1
ISP1P2	33	54.0	-16.60	1
IS - Farthest from depot	34	1.2	-42.94	1
ISP1P2	**33**	56.7	-18.29	**0**
IS - Savings approach	31	1.2	-20.73	0
ISP1P2	31	36.3	-14.14	0
IS - Sweep algorithm	43	1.4	-164.05	8
ISP1P2	**41**	58.1	-14.36	**6**
IS - Different initial customer: a) current IS procedure	31	1.7	-25.28	0
ISP1P2	31	52.8	-13.57	0
b) sweep algorithm	40	10.5	-152.37	5
ISP1P2	40	110.8	-15.20	5

Results in Table 4.2(b) show that the distance for all IS solutions have been reduced after applying our ISP1P2. However the solution from Crevier et al (2007) still outperforms all ISP1P2 solutions. From the table, we can say that the ISP1P2 solutions are slightly different for all IS procedures, on average they travel approximately 13.57% to 16.60% more distance than Crevier et al (2007). In terms of number of vehicles used, two ISP1P2 solutions use less vehicles compared to their IS solutions, particularly

ISP1P2 for farthest from depot (reduce from 34 to 33 vehicles) and ISP1P2 for sweep algorithm (reduce from 43 to 41 vehicles). Thus, ISP1P2 for farthest from depot now produces feasible solutions for all the test problems, whereas ISP1P2 for sweep algorithm reduces two infeasible solutions.

Results in Table 4.2(a) and (b) were produced without using our VR procedure. To illustrate the effect of our VR procedure, the results obtained when it is applied (denoted by VRISP1P2) are shown in Table 4.2(c) and (d), respectively. For reasons of space we only show in Table 4.2(c) those problems where a reduction in the number of vehicles was achieved.

Table 4.2 c) VRISP1P2 solutions for the MDVRPI problems for our IS procedure

Problem	Algorithm	Total number of vehicles used	Total distance	Total computation time (s)	% Improvement in distance over Crevier et al (2007)
a2	Crevier et al (2007)	4	997.94		
	IS	5	1604.36	1	-60.77
	ISP1P2	5	1134.93	2	-13.73
	VRISP1P2	4	1113.98	4	-11.63
g2	Crevier et al (2007)	3	1162.58		
	IS	4	1572.16	1	-35.23
	ISP1P2	4	1352.61	3	-16.35
	VRISP1P2	3	1330.60	6	-14.45
Average	IS	33		1.2	-40.07
	ISP1P2	33		54	-16.60
	VRISP1P2	**31**		**55.3**	**-16.20**
Number of infeasible solutions	0				

Table 4.2(c) has the same format as Table 4.2(a), except that now we apply our vehicle reduction procedure to the routes that result from ISP1P2. For ease of comparison the averages shown in Table 4.2(c) are the averages over all ten problems, computed by combining the results for the two problems explicitly shown in Table 4.2(c) with the results shown in Table 4.2(a) for the other eight problems. Note here that the average time given at the foot of Table 4.2(c) includes the time for applying our vehicle reduction procedure to all problems (whether successful or not). Considering Tables 4.2(a) and (c) then with respect to the number of vehicles used our solutions now involve (in total) 31 vehicles, as in Crevier et al (2007). In addition, VRISP1P2 produces feasible solutions for all test problems.

Table 4.2(d) shows the improvement solution of ISP1P2 for other IS solutions after applying our VR procedure. The results indicate that the VR procedure does not give any effect to VRISP1P2 for farthest from depot procedure, savings approach and different initial customer based on our IS procedure. However VRISP1P2 for another three IS procedures are improved in terms of the number of vehicles used, total distance travelled as well as the number of infeasible solutions. Improved solutions are highlighted in Table 4.2(b).

Table 4.2 d) VRISP1P2 solutions for the MDVRPI problems for other IS procedures

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
IS - Current IS procedure	33	1.2	-40.07	1
ISP1P2	33	54.0	-16.60	1
VRISP1P2	**31**	55.3	**-16.20**	**0**
IS - Farthest from depot	34	1.2	-42.94	1
ISP1P2	33	56.7	-18.29	0
VRISP1P2	33	57.7	-18.29	0
IS - Savings approach	31	1.2	-20.73	0
ISP1P2	31	36.3	-14.14	0
VRISP1P2	31	37.3	-14.14	0
IS - Sweep algorithm	43	1.4	-164.05	8
ISP1P2	41	58.1	-14.36	6
VRISP1P2	**38**	79.1	**-14.41**	**3**
IS - Different initial customer: a) current IS procedure	31	1.7	-25.28	0
ISP1P2	31	52.8	-13.57	0
VRISP1P2	31	53.8	-13.57	0
b) Sweep algorithm	40	10.5	-152.37	5
ISP1P2	40	110.8	-15.20	5
VRISP1P2	**37**	134.3	**-14.82**	**3**

4.5 Conclusion

In this chapter, we have outlined how we can improve the initial solution using the two phases discussed in Section 4.2. We also discussed how we can reduce the number of vehicles used (refer to Section 4.3). Computational results of these two procedures are reported separately in Section 4.4 so that we can clearly see the effectiveness of both procedures on the test problems. The results indicate that even though our VR procedure reduces the number of vehicles used for some test problems the total distance travelled may/may not be reduced. Because the MDVRPI problems have a limited number of vehicles available to use, the VR procedure is important to solve these test problems to generate feasible solutions by reducing the number of vehicles used. In the next chapter we discuss our metaheuristic algorithms for the problems.

CHAPTER 5

METAHEURISTICS

This chapter begins with general descriptions of tabu search (TS) and variable neighbourhood search (VNS). Next, our three metaheuristic algorithms used in this thesis, namely TS, VNS and variable neighbourhood tabu search (VNTS) are presented. VNTS is a metaheuristic where the variable neighbourhood is searched via tabu search. Then, computational results of these metaheuristics tested on two benchmark problem sets are reported. Finally, a summary of this chapter is presented.

5.1 General description of tabu search (TS)

In this section, we describe the main ideas behind the metaheuristic algorithm called tabu search (TS) which was developed by Fred Glover in 1986. A basic TS algorithm is based on an adaptive memory that allows a search heuristic such as a hill climbing method to escape local optima and seek better solutions by allowing non-improving moves. In the hill climbing method a neighbour solution of the current solution is generated. The solution is accepted only if it is better than the current one. This process is repeated until a solution is found such that in its neighbourhood it cannot be improved and then the algorithm will terminate. The problem with the hill climbing method is that it can easily get stuck in a local optimum. However, this problem can be solved by using TS which is an extension of the hill climbing method. In the TS algorithm, we allow a search heuristic to go beyond points of local optimality by permitting non-improving moves through an adaptive memory called a tabu list.

The tabu list is used to prevent cycling when moving away from local optima through non-improving moves because it keeps the information as to the past moves of the search. Once the information as to past move enters into the tabu list, this move (or its reverse) cannot be made and will stay in the tabu list for a certain period of time. Here, tabu tenure is set to indicate how long moves are going to stay in the tabu list.

However if a tabu move produces a new solution with an objective value better than the current best-known solution, the TS algorithm will accept this solution because it has definitely not been explored yet. The function in TS that allows revoking the tabu move while exploiting new solutions is called the aspiration criteria.

Moreover, in the process of finding new solutions using TS we may also get trapped in a space of local optimum. Thus in order to search other parts of the solution space to look for the global optimum, search diversification is performed in TS where it is applied to diversify the solution space to the areas that are largely unexplored so far.

Here we have highlighted the main functions in the TS algorithm. Further information with regard to TS can be found in Glover and Laguna (1993, 1997) and Gendreau (2003). Our TS algorithm used in the thesis is presented in Section 5.3.

5.2 General description of variable neighbourhood search (VNS)

In this section the metaheuristic algorithm called variable neighbourhood search (VNS) is briefly described. VNS algorithm was developed by Hansen and Mlandenovic in 1997 (Hansen and Mlandenovic, 1997). As with other metaheuristics in the literature, VNS also attempts to avoid being trapped in local optimum while improving the current solution. This algorithm escapes a local optimum by a systematic change of neighbourhood within a local search algorithm. In other words, a VNS algorithm involves iterative exploration of a different structure of neighbourhoods (e.g. starts with a small neighbourhood to a larger and larger neighbourhood) around a given local optimum until an improvement is found after which time the search across expanding neighbourhoods is repeated.

In the process of finding new solutions, a local optimum for one neighbourhood structure may not be a local optimum for another neighbourhood structure. Thus by switching from one neighbourhood to the other, different parts of the search space can be explored and better solutions could be found. In this case it is clear that a global optimum is a local optimum for all possible neighbourhood structures. As with all metaheuristics neither TS nor VNS can guarantee to find the global optimum.

For further information relating to VNS see Mladenović and Hansen (1997), Hansen and Mladenović (2001). Our VNS algorithm used in the thesis is presented in Section 5.4.

5.3. Tabu search (TS)

In this section we discuss our metaheuristic algorithm for the problem using tabu search (TS). The move that we consider in our TS heuristic is an interchange of two customers, who may or may not be on the same route. This move is similar to that in the phase 2 procedure discussed in Chapter 4 (refer to Section 4.2.2). Here we again use the neighbour set for a customer to prevent the number of customer interchanges we have to examine being excessive. We do these interchanges however in a tabu search framework (so we allow interchanges that worsen the solution).

In our approach we apply tabu status to customers. So if a customer is tabu it cannot be considered for any possible move. Note that we do not consider disposal facilities with regard to using tabu. This is because of the fact that a disposal facility may appear on more than one route, and so considering disposal facilities for tabu would entail keeping track of which route they are on (since we may wish to move a disposal facility on one route but leave it in its current position, i.e. tabu, on another route). Customers, by contrast, can only be on one route. For our TS heuristic let:

- Δ be the tabu tenure, how long a customer stays tabu for
- M be an iteration counter
- $\delta(i)$ be the last iteration at which customer i was moved, we use $\delta(i)$ to judge whether moving i is tabu or not
- $Z_{current}$ be the current solution value (this being the solution from which we are examining potential moves)
- Z_{move} be the solution value associated with the move we are currently examining

- Z_{best} be the value of the best solution we have encountered during our algorithm (before we start TS both $Z_{current}$ and Z_{best} will be the value of the locally optimal solution as derived in the previous chapter)
- m be a counter of the number of times we examine all pairs of customers without improving Z_{best}
- Z_{non} be the value of the solution associated with the "best" non-improving move, and α and β be the customers associated with this best non-improving move
- ϕ be a diversification factor such that any non-improving solution from $Z_{current}$ that we consider has to have value $\geq Z_{current}+\phi$

The role of ϕ is to diversify the solution by forcing the new solution after a non-improving move to be "far" from the current solution. Computational experiences for choosing a suitable value for ϕ are shown in Appendix 2. Five different values of ϕ (0.01, 0.001, $0.01(Z_{best})$, $Z_{best}/20|C|$ and $Z_{best}/30|C|$) have been tested on the test problems. Here for reasons of space we only show results of the test problem 102. Computational results indicated that our solutions are improved when $\phi = Z_{best}/20|C|$ and $Z_{best}/30|C|$. These two cases produce appropriate gap between Zbest and Zcurrent. Because $Z_{best}/20|C|$ involved more iterations than $Z_{best}/30|C|$, in this thesis we choose $Z_{best}/20|C|$ for ϕ.

The majority of papers in the literature used $\Delta=7$. In this thesis, preliminary computational experience also indicated that $\Delta=7$ produced good solutions for the waste collection VRPTW problems within reasonable computational times. Thus, our metaheuristic solutions reported in this thesis are obtained when $\Delta=7$.

In pseudocode our TS heuristic is:

Initialise

Set M=m=0 (counters set to zero); $\delta(i)=-(\Delta+1)\ \forall i\in C$ (this ensures that all customers are not tabu)

Step 1

Set $Z_{non}=\infty$; set m=m+1; set flag=0, this is a flag to signify whether we have changed Z_{best} or $Z_{current}$ during this step

For all customers $i\in C$:

For all customers $j\in N(i,K)$:

- Interchange customers i and j, i.e. customer i moves to the position that customer j occupied on its route and customer j moves to the position that customer i occupied on its route (these customers may be on the same route, may be on two different routes) and evaluate the routes that result
- **If** the route(s) involved in this interchange are not FEASIBLE **then** disregard the interchange and go to consider a new pair, i.e. go to DONEPAIR.

  Here the route(s) are FEASIBLE, check for:
  - improving Z_{best}, irrespective of tabu status (so aspiration)
  - improving $Z_{current}$ (if not tabu)
  - a better non-improving move Z_{non} (if not tabu)

  Improving the best solution (aspiration)

If the total distance, Z_{move}, associated with the entire (feasible) solution after interchange (so considering not just any routes involved in the interchange but also any routes not involved) is strictly less than Z_{best} (i.e. $Z_{move} < Z_{best}$) **then:**

> we keep the interchange (i.e. keep the moved customers where they are); update Z_{best} to the value of this new improved solution, so $Z_{best}=Z_{move}$; update the current solution, so $Z_{current}=Z_{best}$; set the tabu status for customers i and j using $\delta(i)= \delta(j)=M$; set M=M+1; reset the value of the solution for the best non-improving move $Z_{non}=\infty$; set m=0 as Z_{best} has been improved; set flag=1 to indicate that the solution has changed; and go to DONEPAIR to consider a new pair.

end if

Tabu status (check for whether the customers are tabu)

If $|M - \delta(i)| \leq \Delta$ or $|M - \delta(j)| \leq \Delta$ (so the move is tabu) **then** go to DONEPAIR.

Improving the current solution

If $Z_{move}<Z_{current}$ (so the move improves on the current solution $Z_{current}$) **then**:

> update the current solution, so $Z_{current}=Z_{move}$; set the tabu status for customers i and j using $\delta(i)= \delta(j)=M$; set M=M+1; reset the value of the solution for the best non-improving move $Z_{non}=\infty$; set flag=1 to indicate that the solution has changed; and go to DONEPAIR to consider a new pair.

end if

Better non-improving move

If $Z_{move}<Z_{non}$ (so the move improves on the current non-improving move Z_{non}) and $Z_{move} \geq Z_{current}+\phi$ (so the move is sufficiently far from $Z_{current}$) **then**:

> set the best non-improving move solution $Z_{non}=Z_{move}$ and record the customers associated with this best non-improving move using $\alpha=i$, $\beta=j$

end if

DONEPAIR:

end for

end for

Step 2

Terminate if sufficient iterations have been performed without improving the best solution Z_{best}. In our work we stop if m=5.

If flag=1 **then:**

we have made a change (either to Z_{best} or to $Z_{current}$), so go to Step 1

else

we have not made a change so we make the best non-improving move. In other words we interchange the customers α and β associated with the best non-improving move; update the current solution $Z_{current}$ to the solution after the move; set the tabu status for customers α and β using $\delta(\alpha)=\delta(\beta)=M$; set M=M+1; and go to Step 1.

end if

5.4. Variable neighbourhood search (VNS)

In our VNS heuristic we consider the same move as in our TS heuristic above. However whilst that heuristic operates with a fixed value of K, the number of neighbours a customer has, in our VNS heuristic we vary K. In our VNS heuristic define K^*={set of values of K we will consider}. As for TS we start from the locally optimal solution as derived in the previous chapter. In terms of neighbourhood search (for a specified value of K) we use the same neighbourhood as in our TS heuristic. However, unlike our TS heuristic we only accept moves that improve the best solution Z_{best}. Our VNS heuristic is:

Set $\Gamma = K^*$ (initialise the set of K values we will consider)

while $|\Gamma| \neq 0$ so there are still values of K to consider:

Set $K = \min[k \mid k \in \Gamma]$ to choose the smallest value from Γ and set $\Gamma = \Gamma - \{K\}$

For all customers $i \in C$:

For all customers $j \in N(i,K)$:

- Interchange customers i and j and evaluate the routes that result
- **If** the solution after interchange is FEASIBLE and better than Z_{best}

  then: accept the move, update Z_{best}; set $\Gamma = K^*$ (as we have improved the solution we are willing to reconsider all possible values of K) **else** disregard the interchange

end for

end for

end while

Our VNS heuristic terminates when we have a solution that cannot be improved by any move associated with any of the K values in K^*.

5.5. Variable neighbourhood tabu search (VNTS)

In our VNTS heuristic we adopt the same variable neighbourhood as in our VNS heuristic above. However, whilst our VNS heuristic searches each neighbourhood for improved solutions, in VNTS we allow non-improving moves, i.e. we search each neighbourhood in a TS fashion. Our VNTS heuristic is:

Set $\Gamma = K^*$ (initialise the set of K values we will consider)

while $|\Gamma| \neq 0$ so there are still values of K to consider:

Set $K = \min[k \mid k \in \Gamma]$ to choose the smallest value from Γ and set $\Gamma = \Gamma - \{K\}$

Apply our TS heuristic with this value of K starting from Z_{best}

If the best solution after applying TS has improved **then** set $\Gamma = K^*$

end while

Our VNTS heuristic terminates when we have a solution that cannot be improved by TS associated with any of the K values in K^*.

5.6. Results

This section reports our three metaheuristics (TS, VNS and VNTS) solution for both sets of benchmark problems waste collection VRPTW and MDVRPI.

5.6.1. Computational results for the waste collection VRPTW problems

Computational results for the waste collection VRPTW problems using our metaheuristics, namely TS, VNS and VNTS are reported in this section. Note here that we use the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. For this set of benchmark problems, TS solutions are obtained when K=50 whilst for VNS and VNTS, we use K*={5,10,25,50}. Solutions from these metaheuristics which start from the ISP1P2 solutions discussed in Chapter 4 (refer to Table 4.1(a) and (b)) are reported in Table 5.1(a) and (b). Thus, the computation times given in both tables include the time taken to generate these solutions.

Table 5.1 a) Metaheuristic solutions for the waste collection VRPTW problems for our IS procedure without our VR

Problem	Algorithm	Total number of vehicles used	Total distance (mile)	Total computation time (s)	% Improvement in distance over Kim et al (2006)
102	ISP1P2	3	181.9	3	11.31
	TS	3	181.8	5	**11.36**
	VNS	3	181.8	4	**11.36**
	VNTS	3	181.8	4	**11.36**
277	ISP1P2	3	462.9	15	12.21
	TS	3	462.8	30	**12.23**
	VNS	3	462.9	20	12.21
	VNTS	3	462.8	21	**12.23**
335	ISP1P2	6	196.8	28	4.00
	TS	6	196.8	51	4.00
	VNS	6	196.8	34	4.00
	VNTS	6	196.8	35	4.00
444	ISP1P2	11	79.9	57	8.16
	TS	11	79.9	74	8.16
	VNS	11	79.9	65	8.16
	VNTS	11	79.9	66	8.16
804	ISP1P2	6	716.7	199	6.86
	TS	6	716.7	250	6.86
	VNS	6	716.7	225	6.86
	VNTS	6	716.7	227	6.86
1051	ISP1P2	17	2131.8	257	10.07
	TS	17	2127.6	377	**10.24**
	VNS	17	2127.6	294	**10.24**
	VNTS	17	2127.6	308	**10.24**
1351	ISP1P2	8	907.5	305	12.72
	TS	8	907.5	426	12.72
	VNS	8	907.5	366	12.72
	VNTS	8	907.5	372	12.72
1599	ISP1P2	14	1395.2	362	4.39
	TS	14	1392.3	572	**4.58**
	VNS	14	1395.2	432	4.39
	VNTS	14	1395.2	439	4.39
1932	ISP1P2	16	1187.8	695	14.87
	TS	16	1187.8	991	14.87
	VNS	16	1187.8	786	14.87
	VNTS	16	1187.8	811	14.87
2100	ISP1P2	16	1676.8	829	8.56

	TS	16	1676.5	1079	**8.58**
	VNS	16	1676.8	943	8.56
	VNTS	16	1676.8	954	8.56
Average	ISP1P2	100		275.0	9.31
	TS	100		385.5	**9.36**
	VNS	100		316.9	9.34
	VNTS	100		323.7	9.34

From the averages at the foot of Table 5.1(a), it is clear that our metaheuristic solutions produce better results than Kim et al (2006) in terms of the distance travelled. In particular, TS produces the highest reduction in distance approximately 9.36% less than Kim et al (2006), whilst solutions from VNS and VNTS produce routes of similar quality (approximately 9.34% less distance than Kim et al (2006)). Improved solutions compared to the ISP1P2 solutions are highlighted in the table. However in some test problems, our metaheuristics do not improve on ISP1P2 solutions (i.e. 335, 444, 804, 1351 and 1932 problems). With respect to the number of vehicles used, no reduction occurs in any of the metaheuristic solutions. Thus, our metaheuristic solutions also involve (in total) 100 vehicles as our ISP1P2 solutions, those of Kim et al (2006) 99 vehicles, so slightly worse.

Results in Table 5.1(a) indicate that our three metaheuristics (TS, VNS and VNTS) produce solutions of similar quality for the 102 problem. Thus Figure 5.1 shows only our VNS routes for the problem. Compared with our ISP1P2 solution, our metaheuristics just slightly improved the ISP1P2 solutions by some 100(181.9-181.8)/181.9=0.05%. Thus, our solution for the 102 problem travels approximately 11.36% less distance than Kim et al (2006) with three vehicles.

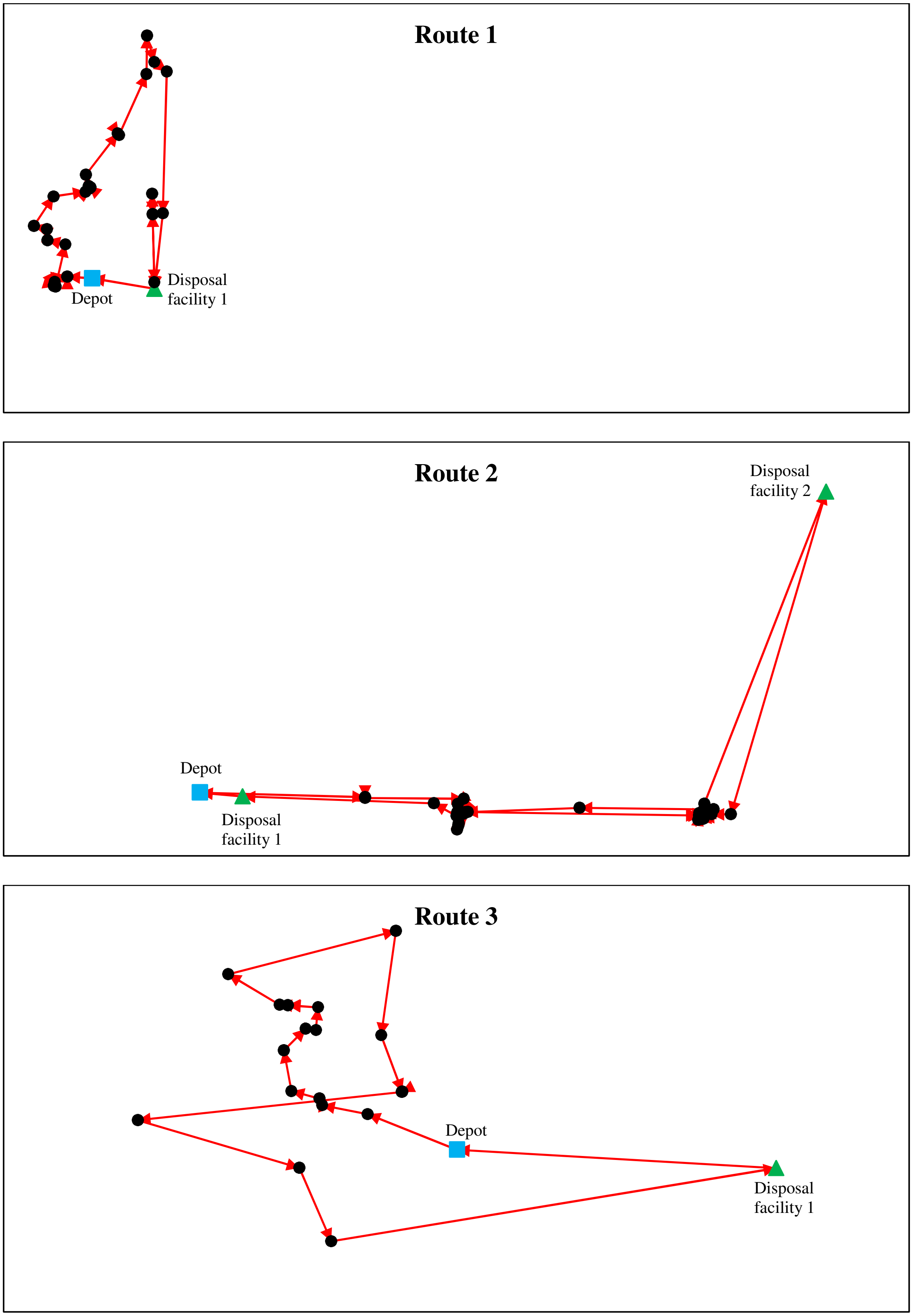


Figure 5.1: VNS routes for the 102 problem

Table 5.1 b) Metaheuristic solutions for the waste collection VRPTW problems for other IS procedures without our VR

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
ISP1P2 - current IS procedure as Table 4.1(a)	100	275	9.31
TS	100	385.5	**9.36**
VNS	100	316.9	**9.34**
VNTS	100	323.7	**9.34**
ISP1P2 - Farthest from depot	103	388.3	6.23
TS	103	473.7	6.23
VNS	103	432.1	6.23
VNTS	103	458.4	6.23
ISP1P2 - Savings approach	101	411.2	6.94
TS	101	501.3	**6.95**
VNS	101	455.6	**6.95**
VNTS	101	464.3	**6.95**
ISP1P2 - Sweep algorithm	154	290.7	-18.76
TS	154	385.7	**-18.73**
VNS	154	339.4	**-18.72**
VNTS	154	362.3	**-18.64**
Different initial customer:			
a) ISP1P2 - current IS procedure	98	657	13.00
TS	98	741.3	**13.02**
VNS	98	697.6	13.00
VNTS	98	702.2	**13.02**
b) ISP1P2 - sweep algorithm	150	4611.4	-16.16
TS	150	4715.3	**-16.14**
VNS	150	4679.8	**-16.14**
VNTS	150	4715.1	**-16.14**

Table 5.1(b) shows our metaheuristic solutions starting from the ISP1P2 solution for other IS procedures. Examining Table 5.1(b) all ISP1P2 solutions are slightly improved after applying our metaheuristics (as highlighted) except for the ISP1P2 solution for the farthest from depot. Our TS and VNTS solutions for the different initial solution based on our IS procedure produces the best solution for the waste collection VRPTW

problems. On average those metaheuristics (TS and VNTS) produce a reduction of around 13.02% in distance travelled and involve one less vehicle than Kim et al (2006). Even though both metaheuristics show solution of similar quality, but VNTS requires less computation time than TS. On the other hand, the negative numbers for the metaheuristic solutions for the sweep algorithm and the different initial solution based on sweep algorithm indicate that they are still worse than Kim et al (2006).

Metaheuristic solutions presented in Table 5.1(a) and (b) were produced starting from the ISP1P2 solutions without using our VR procedure. Next in Table 5.1(c) and (d), metaheuristic solutions starting from the VRISP1P2 solutions are reported.

Table 5.1 c) Metaheuristic solutions for the waste collection VRPTW problems for our IS procedure with our VR

Problem	Algorithm	Total number of vehicles used	Total distance (mile)	Total computation time (s)	% Improvement in distance over Kim et al (2006)
1351	VRISP1P2 as Table 4.1(c)	7	1010.9	545	2.77
	TS	7	1010.9	665	2.77
	VNS	7	1010.9	605	2.77
	VNTS	7	1010.9	613	2.77
Average	VRISP1P2	99		296.6	8.32
	TS	99		410.1	**8.37**
	VNS	99		341.5	8.34
	VNTS	99		348.5	8.34

As we can see in Table 5.1(c) our metaheuristics do not improve on the VRISP1P2 solution for the 1351 problem. The total distance travelled for the 1351 problem is 1010.9 miles which is approximately 2.77% less distance than Kim et al (2006). However, on average our TS produces the best routes, with a reduction of 8.37% in the distance travelled and uses 99 vehicles as in Kim et al (2006).

Table 5.1 d) Metaheuristic solutions for the waste collection VRPTW problems for other IS procedures with our VR

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
VRISP1P2 - Current IS procedure	99	296.6	8.32
TS	99	410.1	**8.37**
VNS	99	341.5	**8.34**
VNTS	99	348.5	**8.34**
VRISP1P2 - Farthest from depot	101	430	4.30
TS	101	519.5	4.30
VNS	101	478.9	4.30
VNTS	101	507.7	4.30
VRISP1P2 - Savings approach	99	479.3	7.05
TS	99	568.2	7.05
VNS	99	525.9	7.05
VNTS	99	534.5	7.05
VRISP1P2 - Sweep algorithm	107	2427.9	-22.86
TS	107	2551.1	**-22.78**
VNS	107	2486.3	**-22.77**
VNTS	107	2509.9	**-22.84**
Different initial customer:	107		
VRISP1P2 - current IS procedure	97	678.2	11.26
TS	97	765.8	**11.28**
VNS	97	722.1	**11.27**
VNTS	97	726.7	**11.28**
VRISP1P2 - sweep algorithm	109	6545.3	-21.58
TS	109	6651.5	-21.58
VNS	109	6615.8	-21.58
VNTS	109	6646.9	-21.58

Table 5.1(d) shows the metaheuristic solutions starting from the VRISP1P2 solutions for other IS procedures. As highlighted in the table, three VRISP1P2 solutions are slightly improved after applying our metaheuristics (i.e. VRISP1P2 solutions for current IS procedure, sweep algorithm and the different initial customer based on our IS procedure). The rest remain the same as VRISP1P2 solutions. Of all our three

metaheuristics, the results show that VNS has lower average computation times than either TS or VNTS.

5.6.2. Computational results for the MDVRPI problems

In this section, computational results of the MDVRPI problems using our metaheuristics (TS, VNS and VNTS) are reported. Here we also use the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. For this set of benchmark problems, TS solutions are obtained when K=|C| whilst for VNS and VNTS, we use K*={5,10,25,|C|}. Solutions from these metaheuristics which starting from the ISP1P2 solutions discussed in Chapter 4 (refer to Table 4.2(a) and (b)) are reported in Table 5.2(a) and (b). Thus, the computation times given in both tables include the time taken to generate these solutions.

Table 5.2 a) Metaheuristic solutions for the MDVRPI problems for our IS procedure without our VR

Problem	Algorithm	Total number of vehicles used	Total distance	Total computation time (s)	% Improvement in distance over Crevier et al (2007)
a2	ISP1P2	5	1134.93	2	-13.73
	TS	5	1134.93	3	-13.73
	VNS	5	1134.93	3	-13.73
	VNTS	5	1134.93	3	-13.73
b2	ISP1P2	3	1503.85	7	-15.04
	TS	3	1503.85	10	-15.04
	VNS	3	1503.85	9	-15.04
	VNTS	3	1503.85	9	-15.04
c2	ISP1P2	3	1978.75	23	-13.23
	TS	3	1978.75	31	-13.23
	VNS	3	1978.75	27	-13.23
	VNTS	3	1978.75	27	-13.23

d2	ISP1P2	3	2130.00	46	-13.82
	TS	3	**2129.29**	72	**-13.78**
	VNS	3	2130.00	54	-13.82
	VNTS	3	2130.00	54	-13.82
e2	ISP1P2	3	2415.59	101	-24.33
	TS	3	2415.59	137	-24.33
	VNS	3	2415.59	116	-24.33
	VNTS	3	2415.59	116	-24.33
f2	ISP1P2	3	2639.74	138	-15.56
	TS	3	2639.74	191	-15.56
	VNS	3	2639.74	160	-15.56
	VNTS	3	2639.74	161	-15.56
g2	ISP1P2	4	1352.61	3	-16.35
	TS	4	1352.61	5	-16.35
	VNS	4	1352.61	4	-16.35
	VNTS	4	1352.61	4	-16.35
h2	ISP1P2	3	1767.10	15	-11.32
	TS	3	1767.10	24	-11.32
	VNS	3	1767.10	19	-11.32
	VNTS	3	1767.10	19	-11.32
i2	ISP1P2	3	2347.10	66	-19.02
	TS	3	2347.10	92	-19.02
	VNS	3	2347.10	77	-19.02
	VNTS	3	2347.10	77	-19.02
j2	ISP1P2	3	2835.93	139	-23.62
	TS	3	2835.93	192	-23.62
	VNS	3	2835.93	161	-23.62
	VNTS	3	2835.93	162	-23.62
Average	ISP1P2	33		54.0	-16.60
	TS	33		75.7	-16.60
	VNS	33		63.0	-16.60
	VNTS	33		63.2	-16.60
Number of infeasible solutions	1				

Results in Table 5.2(a) show that only the ISP1P2 solution for the d2 problem is improved using TS. The total distance travelled for this problem is slightly reduced from 2130 to 2129.29. However, this small reduction does not improve on the averages at the foot of the table. We travel approximately 16.6% more distance and use two extra vehicles than Crevier et al (2007). Here TS requires more computational time than either VNS or VNTS even though they produce routes of similar quality.

Table 5.2 b) Metaheuristic solutions for the MDVRPI problems for other IS procedures without our VR

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
ISP1P2 -Current IS procedure as	33	54	-16.60	1
TS	33	75.7	-16.60	1
VNS	33	63.0	-16.60	1
VNTS	33	63.2	-16.60	1
ISP1P2 - Farthest from depot	33	56.7	-18.29	0
TS	33	**80.0**	**-18.16**	0
VNS	33	65.9	-18.29	0
VNTS	33	**66.3**	**-18.16**	0
ISP1P2 - Savings approach	31	36.3	-14.14	0
TS	31	**56.6**	**-14.10**	0
VNS	31	44.8	-14.11	0
VNTS	31	**45.0**	**-14.10**	0
ISP1P2 - Sweep algorithm	41	58.1	-14.36	6
TS	41	85.3	-14.33	6
VNS	41	66.8	-14.35	6
VNTS	41	**67.1**	**-14.25**	6
Different initial customer:				6
a) ISP1P2 - current IS procedure	31	52.8	-13.57	0
TS	31	78.4	-13.52	0
VNS	31	61.6	-13.57	0
VNTS	31	**62.1**	**-13.45**	0
b) ISP1P2 - sweep algorithm	40	110.8	-15.20	5
TS	40	133.3	-15.20	5
VNS	40	119.5	-15.20	5
VNTS	40	119.7	-15.20	5

Table 5.2(b) shows our metaheuristic solutions starting from the ISP1P2 solutions for other IS procedures. The results clearly show that solutions from Crevier et al (2007) are still better than us. However, our metaheuristics do improve on some of the ISP1P2 solutions. For example, on average our TS and VNTS solutions improve the ISP1P2 solutions for the farthest from depot (from -18.29% to -18.16%) and the savings approach (from -14.14% to -14.10%). In both improvements, VNTS has lower computation time than TS even though they produce solutions of similar quality. In addition, on average VNTS also produce the biggest reduction in distance travelled than either TS or VNS for the ISP1P2 solutions for the sweep algorithm and the different initial customer based on our IS procedure (from -14.36% to -14.25% and from -13.57% to -13.45%, respectively). On the other hand, our metaheuristics do not improve on ISP1P2 solutions for the current IS procedure and the different initial customer based on sweep algorithm. Since there is no vehicle reduction in the metaheuristic solutions, the numbers of infeasible solutions in the last column of the table remain the same as ISP1P2 solutions.

Table 5.2(c) has the same format as Table 4.2(c). Examining results in Table 5.2(c), our three metaheuristics (TS, VNS and VNTS) do not improve on VRISP1P2 solutions for both test problems a2 and g2. After applying our metaheuristics, on average we travel approximately 16.20% more distance than Crevier et al (2007) with 31 vehicles. Again, from the foot of the table on average VNS has a lower computation time than either TS or VNTS.

Table 5.2 c) Metaheuristic solutions for the MDVRPI problems for our IS procedure with our VR

Problem	Algorithm	Total number of vehicles used	Total distance	Total computation time (s)	% Improvement in distance over Crevier et al (2007)
a2	VRISP1P2	4	1113.98	4	-11.63
	TS	4	1113.98	5	-11.63
	VNS	4	1113.98	5	-11.63
	VNTS	4	1113.98	5	-11.63
g2	VRISP1P2	3	1330.60	6	-14.45
	TS	3	1330.60	8	-14.45
	VNS	3	1330.60	7	-14.45
	VNTS	3	1330.60	7	-14.45
Average	VRISP1P2	31		55.3	-16.20
	TS	31		76.9	-16.20
	VNS	31		**64.3**	-16.20
	VNTS	31		64.5	-16.20
Number of infeasible solutions	0				

Table 5.2 d) Metaheuristic solutions for the MDVRPI problems for other IS procedures with our VR

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
VRISP1P2 - Current IS procedure	31	55.3	-16.20	0
TS	31	76.9	-16.20	0
VNS	31	**64.3**	-16.20	0
VNTS	31	64.5	-16.20	0
VRISP1P2 - Sweep algorithm	38	79.1	-14.41	3
TS	38	105.4	**-14.39**	3
VNS	38	**88.0**	**-14.39**	3
VNTS	38	88.4	**-14.39**	3
Different initial customer:				
VRISP1P2 - sweep algorithm	37	134.3	-14.82	3
TS	37	156.6	-14.82	3

VNS	37	**142.2**	-14.82	3
VNTS	37	142.3	-14.82	3

Table 5.2(d) shows improved solutions compared to the VRISP1P2 solutions for other IS solutions after applying our three metaheuristics (TS, VNS and VNTS). However, only the VRISP1P2 solution for the sweep algorithm has been improved (as highlighted in the table). It is clearly shows that all metaheuristics produce routes of similar quality but on average VNS has the lowest computation times.

5.7. Conclusion

In a further attempt to improve the solutions for both sets of benchmark problems waste collection VRPTW and MDVRPI, in this chapter we have presented our three metaheuristics namely TS, VNS and VNTS. These metaheuristics have been applied on the solutions of the ISP1P2 and VRISP1P2 which have been discussed in Chapter 4. Based on the computational results reported above, our metaheuristics give only small improvement on those solutions.

Of all solutions obtained from our metaheuristics, the solution for the different initial customer based on our IS procedure gives the best solution for the waste collection VRPTW problems. This solution outperforms Kim et al (2006) in terms of the distance travelled as well as the number of vehicles used. This is followed by the solution from our IS procedure and the solution for the savings approach. Both solutions are better than Kim et al (2006) in terms of the distance travelled and use 99 vehicles as in Kim et al (2006). On the other hand, solution for the farthest from depot is only better than

Kim et al (2006) in terms of the distance travelled but use extra two vehicles than Kim et al (2006). Moreover, solutions from the sweep algorithm and the different initial customer based on sweep algorithm are worse than Kim et al (2006) in terms of both distance travelled and number of vehicles used.

For the MDVRPI problems, computational results reported above indicate that all our metaheuristic solutions are worse than Crevier et al (2007) in terms of the distance travelled. However, three solutions (current IS procedures, savings approach and the different initial customer based on current IS procedure) involve (in total) 31 vehicles as in Crevier et al (2007).

Examining the results in the tables presented above we could say that all our metaheuristics produce routes of almost similar quality for the test problems. On this basis we would be justified in choosing the metaheuristic involving the lowest computation time. From the average of all table presented above, clearly VNS is to be preferred, having a lower average time than either TS or VNTS.

In the next chapter other procedures for route evaluation are presented. Since some of the solutions from both sets of benchmark problems are still worse even after applying our metaheuristics, in the next chapter the route evaluation procedures are only tested on some of the solutions. For the waste collection VRPTW problems, we choose the solution procedures which travel less distance than Kim et al (2006) and used at least the same number of vehicles as in Kim et al (2006). Hence, three IS procedures are selected for further examination in the next chapter:

- Current IS procedure
- Savings approach
- Different initial customer based on current IS procedure

However, for the MDVRPI problems, all of our solutions are worse than Crevier et al (2007) in terms of the distance travelled. On this basis, we choose the solution procedures that produce feasible solutions for all the test problems. Hence, four IS procedures are selected for further examination in the next chapter:

- Current IS procedure
- Farthest from depot
- Savings approach
- Different initial customer based on current IS procedure

CHAPTER 6

OTHER LOCAL SEARCH PROCEDURES

In the first section of this chapter a route evaluation procedure, namely disposal facility positioning (DFP) is presented. In the second and third sections, two procedures, namely change tracking and reverse order are presented. The change tracking procedure aims to reduce the computation time by evaluating only necessary changes on the routes whereas the reverse order procedure aims to reduce the distance travelled. In the fourth section, computational results for all these procedures tested on two benchmark problem sets are reported. Finally, a summary of this chapter is presented.

6.1 Disposal facility positioning procedure (DFP)

In this section a route evaluation procedure, disposal facility positioning (DFP) originally proposed by Hemmelmayr et al (2009) is presented. Since the waste collection problem involves disposal facilities trips on the routes, this procedure is used for choosing the best disposal facilities to go on the route. In Chapter 3 (refer to Section 3.5), we have used this procedure for evaluating initial routes constructed from the savings approach and the sweep algorithm. In this chapter, this procedure is used to evaluate routes for our phase 1 and phase 2 procedures as well as the three metaheuristic algorithms (TS, VNS and VNTS) in solving both sets of benchmark problems waste collection VRPTW and MDVRPI.

Note here that in evaluating a given route we regard it as comprising a fixed sequence of places - starting at the depot, then a mix of customers and disposal facilities, finally a disposal facility (to empty the vehicle), followed by the depot. Although in our route evaluation procedure presented in Chapter 4 (refer to Section 4.1) allows the insertion of extra disposal facilities to try (if necessary) to ensure that the route remains feasible with respect to capacity it is clear that as we evaluate a large number of routes, and as we move customers between routes as we attempt to improve the solution, the positioning of disposal facilities is crucial.

In the DFP procedure presented below, dynamic programming is used to generate a candidate route by “optimally” positioning disposal facilities. Here we use “optimal” in inverted commas, as we cannot actually optimally position disposal facilities, as we have to take into account the rest period and time windows for the waste collection VRPTW problems. This contrasts with the problem considered by Hemmelmayr et al (2009) where such complications did not arise. However, this approach can be used to suggest positions for disposal facilities in a route that we can then evaluate taking the rest period and time windows into account.

As in the route evaluation procedure we start with a fixed sequence of places for which we wish to investigate disposal facility positioning. Suppose the route (but now excluding any disposal facilities in this fixed sequence) is given by $\Gamma(1)\rightarrow\Gamma(2)\rightarrow\ldots\rightarrow\Gamma(m)$ so there are m places on the route with $\Gamma(1)=\Gamma(m)$ being the depot, $\Gamma(2)$ being the first customer, $\Gamma(3)$ the second customer, etc and $\Gamma(m\text{-}1)$ the last customer.

Clearly we will need to position a disposal facility been the last customer $\Gamma(m-1)$ and the depot $\Gamma(m)$, but we may also need to position other disposal facilities as well (to try and feasibly operate the route).

In order to try and avoid excessive computation time we, for all pairs of customers/depot, pre-determine the best disposal facility to insert. More formally let:

$$b(i,j) = \arg\min [d_{ik}+d_{kj} \mid k\in D] \qquad i\in \{0\}\cup C;\ j\in \{0\}\cup C;\ i\neq j$$

Here $b(i,j)$ is the best disposal facility to insert between i and j based on the total distance travelled. This (obviously) ignores the issue of customer and disposal facility time windows. Note here that $b(i,j)$ can be computed once before any routes are calculated.

We can now construct a graph to assist in disposal facility positioning. This graph is a directed, acylic graph where there are m nodes, node i ($i\neq 1,m$) representing customer $\Gamma(i)$ and nodes 1 and m being the depot. Here we distinguish the depot twice, once at the start of the route (node 1) and once at the end of the route (node m).

In this graph there is an arc between nodes i and j ($j>i$) if and only if it is feasible (in terms of vehicle capacity) to visit all of the customers between $\Gamma(i)$ and $\Gamma(j-1)$ inclusive i.e. if and only if $\sum_{k=i}^{j-1} q_{\Gamma(k)} \leq Q$ (where we define $q_0=0$ to ease the notation). If this arc between nodes i and j exists then it indicates that we will visit the customers between

Γ(i) and Γ(j-1) in their fixed sequence, then go to disposal facility b(Γ(j-1),Γ(j)) before going to customer Γ(j). The cost π(i,j) of this arc is given by the total travel distance involved, i.e. $\pi(i,j) = \sum_{k=i}^{j-2} d_{\Gamma(k),\, \Gamma(k+1)} + d_{\Gamma(j-1), b(\Gamma(j-1),\, \Gamma(j))} + d_{b(\Gamma(j-1),\, \Gamma(j)),\, \Gamma(j)}$

If we now find the shortest (least cost) path in this graph between node 1 and node m, where the arc costs are given by π(i,j) then we will have "optimally" positioned disposal facilities. In other words the arcs that make up the least cost path from node 1 to node m each (by definition) have a disposal facility associated with them. Thus, by knowing the arcs used in the least cost path we know which disposal facilities to position where in the route.

For example, suppose the shortest (least cost) path in this graph between node 1 and node m involves the arc from 1 to k and the arc from k to m. Then we will be positioning two disposal facilities:

- one just before Γ(k), this will be the disposal facility b(Γ(k-1),Γ(k))
- one just before Γ(m), this will be the disposal facility b(Γ(m-1), Γ(m))

We then evaluate the fixed sequence of places depot=Γ(1)→Γ(2) →…→Γ(k-1)→ b(Γ(k-1), Γ(k))→Γ(k) →Γ(k+1)→…→ Γ(m-1) → b(Γ(m-1),Γ(m))→ Γ(m)=depot

This evaluation is done using our route evaluation procedure so we take into account the rest period and the time windows. There is no guarantee that this route with disposal facilities "optimally" positioned will (once we take account of the rest period and time windows) be either feasible or better than the route (involving disposal facilities) we had originally. However this procedure does offer one way to position disposal facilities across the whole route that may be worthwhile.

Finding the least cost path between node 1 and node m in the directed graph with arc costs $\pi(i,j)$ can be accomplished using dynamic programming, e.g. using an algorithm due to Dijkstra (1959). However since here we have an acyclic graph we can use a more specialised algorithm than the general algorithm for finding shortest paths.

6.1.1 Least cost path finding

In this section a procedure for finding least cost paths in a computationally efficient manner is presented. Here, we use the above procedure for disposal facility positioning each time we evaluate a route (fixed sequence of places). Although finding a least cost path in a graph can be accomplished using algorithms such as those due to Dijkstra (1959) the graph in which we are finding a least cost path has a special structure, namely that all arcs go from some node i to some node j, **where j>i**. As such we can find the least cost path from 1 to m in the manner presented below. In the procedure presented we in fact combine establishing whether arcs in the graph exist (and if so their cost $\pi(i,j)$) simultaneous with calculating the least cost path. This is because we can move forward in the fixed sequence, from node 1 to node 2 to node 3, etc in an obvious manner.

Let $\Omega(i)$ be the cost of the least cost path from node 1 to node i and let $\sigma(i)$ be the node before i in that path (i.e. the arc $\sigma(i)\rightarrow i$ is used in the path).

Initialise $\Omega(i)=\infty$ and $\sigma(i)=0$ i=1,…m

For all i=1,….,m-1 in turn do:

Here we try an establish arcs from node i to later nodes in the fixed sequence

Set $D_{temp}=Q_{temp}=0$; these are running totals of the distance/capacity involved in the arcs from i

For all j=(i+1),…,m in turn do:

If $Q_{temp}+q_{\Gamma(j-1)} > Q$ then go to DONEJ, as we cannot add an arc from i to j since the vehicle capacity would be exceeded, and this by implication means we cannot add an arc from i to any other node k>j

Here we can add an arc from i to j

Set $\pi(i,j)=D_{temp}+d_{\Gamma(j-1),b(\Gamma(j-1),\ \Gamma(j))}+d_{b(\Gamma(j-1),\ \Gamma(j)),\ \Gamma(j)}$ set the arc length

Update $\Omega(j)$ and $\sigma(j)$ using:

If i=1 and $\pi(i,j)< \Omega(j)$ **then:**

$\Omega(j)= \pi(i,j)$ update the length of the shortest path from 1 to j

$\sigma(j)=i$ update the node before j in the shortest path

end if

If $\Omega(i)+\pi(i,j)< \Omega(j)$ **then:**

$\Omega(j)= \Omega(i)+\pi(i,j)$ update the length of the shortest path from 1 to j

$\sigma(j)=i$ update the node before j in the shortest path

end if

Set $Q_{temp}=Q_{temp}+q_{\Gamma(j-1)}$ update the capacity used

Set $D_{temp}=D_{temp}+d_{\Gamma(j-1), \Gamma(j)}$ update the distance travelled

end for

DONEJ:
end for

Computationally this procedure requires $O(m^2)$ operations.

In order to trace out the sequence of disposal facilities (and customers) at the end of the procedure (for use in evaluating whether this sequence, when we consider rest periods and time windows, is feasible) we start from node m (=depot) in the procedure below.

Set i=m
While $i \neq 1$ do:
Insert disposal facility $b(\Gamma(i-1), \Gamma(i))$ before customer $\Gamma(i)$
Set $i=\sigma(i)$
end while

Figure 6.1 illustrates an example of a vehicle route when the DFP procedure is applied for positioning disposal facilities on the route. The route consists of 20 nodes (starts from node 1 to node 20) where node 1=node 20=depot. In this example the shortest (least cost) path between node 1 and node 20 involves four arcs:

- Arc from node 1 to node 5
- Arc from node 5 to node 12
- Arc from node 12 to node 18

- Arc from node 18 to node 20

After the shortest (least cost) path is defined, then the disposal facilities will be inserted on the route. Using the procedure presented above, a disposal facility will be inserted just before node i (b(Γ(i-1), Γ(i)). Here we start with i=m=20. Thus, the first disposal facility is inserted before node 20. The second disposal facility is inserted before node 18 (i.e. i=σ(i) =σ(20)=18). This process is repeated if i≠1. So in this example the last disposal facility inserted on the route is before node 5. Then the procedure is terminated (i.e. i=σ(5)=1). The logic here is that node 1=depot. So when i=1 no disposal facility can be inserted before the depot.

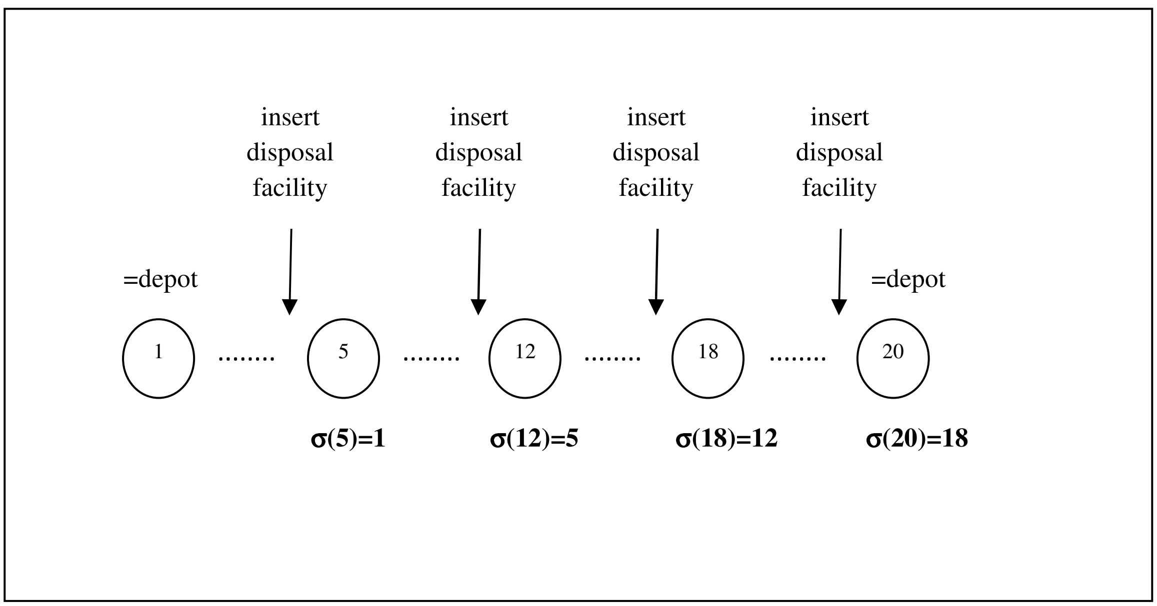


Figure 6.1: An example of a vehicle route when the DFP procedure is applied

Preliminary computational experiments indicated that this procedure is effective by reducing distance travelled for the test problems which have multiple disposal facilities. But for a problem with only one disposal facility it does not have much effect and will produce almost the same quality routes as our route evaluation procedure. This is because our procedures (refer to Section 4.1 and Section 4.2) do allow us to move disposal facilities. If we have just one disposal facility, the DFP procedure adds little to our existing approaches.

Preliminary computational experiments indicated that this DFP procedure is very time consuming. For example, on the largest problem with 2092 customers including the DFP procedure requires approximately 1.5 hours to solve the problem. Hence in the next section we present a procedure, namely change tracking, to overcome this problem by evaluating only necessary changes on the routes.

6.2 Change tracking procedure

In our local search algorithms (e.g. phase 1 and phase 2 procedures) the majority of the changes that we examine involve two customers (who may or may not be on the same route), so for example we:

- add customer j before/after a customer i
- interchange two customers i and j

With the DFP procedure the evaluation of any change is now computationally quite expensive. So one thing we can do is cut down on “unnecessary” evaluations. Suppose we examine an interchange of two customers i and j and find it is not successful (in terms of leading to an improvement in the current solution) – we go on to examine many more changes but at some point we may come back to examining the same pair of customers. Now if since we last examined this pair:

- i is still on the same route and the route that i is on has not changed at all; AND
- j is still on the same route and the route that j is on has not changed at all

then it is clear that re-examining the interchange of i and j cannot be successful in terms of leading to an improvement in the current solution.

Hence if we can identify when we come to examine a pair that any change cannot be successful (i.e. it satisfies the above condition) we can skip evaluating it – saving on computation time.

To do this:

- set up a counter COUNT that is increased by one each time we do a route evaluation (using our new DFP approach)
- let ROUTE(v) v=1,2,….,number of vehicles be the COUNT value at which route v was last changed
- Initially ROUTE(v)=0 ∀v and when we make a change to a route at any point in the algorithm set ROUTE(v) for that route equal to the current COUNT value

Define

- last_time(i,j,1) to be the COUNT value when we last examined placing j before customer i (both i and j on the same route);
- last_time(i,j,2) to be the COUNT value when we last examined placing j after customer i (both i and j on the same route);
- last_time(i,j,3) to be the COUNT value when we last examined placing j before customer i (both i and j on the different routes);

- last_time(i,j,4) to be the COUNT value when we last examined placing j after customer i (both i and j on the different routes);
- last_time(i,j,5) to be the COUNT value when we last examined interchanging customers i and j, both are on the same route (since when we swap the pair (i,j) this is the same as the pair (j,i) we need to always have last_time(i,j,5)= last_time(j,i,5))
- last_time(i,j,6) to be the COUNT value when we last examined interchanging customers i and j, both are on the different routes (since when we swap the pair (i,j) this is the same as the pair (j,i) we need to always have last_time(i,j,6)= last_time(j,i,6))

Now when we come to examine a change we need only bother with examining this change if:

- placing j before customer i (on the same route):
    - last_time(i,j,1) < ROUTE(route i on), since if this is true then the route that i is on has changed since the last time we examined placing j before i
- placing j after customer i (on the same route):
    - last_time(i,j,2) < ROUTE(route i on), since if this is true then the route that i is on has changed since the last time we examined placing j after i
- placing j before customer i (on the different routes):
    - last_time(i,j,3) < maximum (ROUTE(route i on), ROUTE(route j on)), since if this is true then the routes that i and j are on have changed since the last time we examined placing j before i
- placing j after customer i (on the different routes):
    - last_time(i,j,4) < maximum (ROUTE(route i on), ROUTE(route j on)), since if this is true then the routes that i and j are on have changed since the last time we examined placing j after i
- interchanging i and j (on the same route):
    - last_time(i,j,5) < ROUTE(route i on), since if this is true then the route that i and j is on has changed since the last time we examined interchanging i and j

- interchanging i and j (on the different routes):
    - last_time(i,j,6) < maximum (ROUTE(route i on), ROUTE(route j on)) since if this is true then the routes that i and j are on have changed since the last time we examined interchanging i and j

The change tracking procedure discussed above is added only at certain parts of the phase 1 and phase 2 procedures where repositioning/interchanging of the customers are involved. Thus, our phase 1 and phase 2 procedures become as below

Before phase 1 and phase 2, set:

COUNT=0

last_time(i,j,1)=last_time(i,j,2)=last_time(i,j,3)=last_time(i,j,4)=last_time(i,j,5)=last_time(i,j,6)=-1 ∀i,j (setting it to –1 ensures we always examine a change at least once)

ROUTE(v)=0 ∀v

Phase 1

In this phase we evaluate repositioning customers using our new DFP procedure. In pseudocode we:

For all customers i∈C:

For all customers j∈N(i,K):

If i and j are on the same route:

then:

If last_time(i,j,1) < ROUTE(route i on) **then**:

- set COUNT=COUNT+1; set last_time(i,j,1)=COUNT
- Add j immediately before i (here we check positioning j before i on the same route as it was originally on)
- Evaluate this new route with j added to this new position **using our new DFP procedure** and if it is better than the original route (FEASIBLE and of lower total distance) then keep it (and set ROUTE(route i on)=COUNT), else do not

endif

If last_time(i,j,2) < ROUTE(route i on) then:

- set COUNT=COUNT+1; set last_time(i,j,2)=COUNT
- Add j immediately after i (here we check positioning j after i on the same route as it was originally on)
- Evaluate this new route with j added to this new position **using our new DFP procedure** and if it is better than the original route (FEASIBLE and of lower total distance) then keep it (and set ROUTE(route i on)=COUNT), else do not

end if

endif

If i and j are on the different routes:

then:

If last_time(i,j,3) <maximum (ROUTE(route i on), ROUTE(route j on)) **then**:

- set COUNT=COUNT+1; set last_time(i,j,3)=COUNT
- Add j immediately before i (here we check positioning j before i on the routes as they were originally on)
- Evaluate the two routes that are involved in this change **using our new DFP procedure** and if both are FEASIBLE and their total distance is<the total distance for the two routes before the change then keep them (and set ROUTE(route i on)=ROUTE(route j on)=COUNT), else do not

endif

If last_time(i,j,4) < maximum (ROUTE(route i on), ROUTE(route j on)) **then**:

- set COUNT=COUNT+1; set last_time(i,j,4)=COUNT
- Add j immediately after i (here we check positioning j after i on the routes as they were originally on)
- Evaluate the two routes that are involved in this change **using our new DFP procedure** and if both are FEASIBLE and their total distance is<the total distance for the two routes before the change then keep them (and set ROUTE(route i on)=ROUTE(route j on)=COUNT), else do not

end if

endif

end for

end for

Phase 2

In this phase we evaluate interchanging customers using our new DFP procedure. In pseudocode we:

For all customers i∈C:

For all customers j∈N(i,K):

If i and j are on the same route AND last_time(i,j,5) < ROUTE(route i on) **then:**

- set COUNT=COUNT+1; set last_time(i,j,5)= last_time(j,i,5)=COUNT;
- we interchange customers i and j, i.e. customer i moves to the position of customer j and customer j moves to the position of customer i on the route
- Evaluate this new route **using our new DFP procedure** and if it is better than the original route (FEASIBLE and of lower total distance) then keep it (and set ROUTE(route i on)=COUNT), else do not

end if

If i and j are on the different routes (serviced by a different vehicle) AND last_time(i,j,6) < maximum (ROUTE(route i on), ROUTE(route j on))
then:

- set COUNT=COUNT+1; set last_time(i,j,6)= last_time(j,i,6)=COUNT;
- we interchange customers i and j, i.e. customer i moves to the position that customer j occupied on its route and customer j moves to the position that customer i occupied on its route

- Evaluate the two routes that are involved in this interchanges **using our new DFP procedure**. If both are FEASIBLE and their total distance is < the total distance for the two routes before the interchange then keep the interchange (and set ROUTE(route i on)=ROUTE(route j on)=COUNT), else do not

end if

end for

end for

Computationally we repeat phase 1 and phase 2 in turn until no further improvement can be achieved. We will then have a locally optimal solution.

Note here this procedure aims to reduce the computational time when the DFP procedure is used to evaluate vehicle routes. Preliminary computational experiments indicated that including the DFP procedure in our phase 1 and phase 2 procedures for the test problems with more than 500 customers could take approximately 25 minutes. For example, Table 6.1 shows total computational time for the test problems of the waste collection VRPTW which involved more than 500 customers with/without applying the change tracking procedure. The last column in the table gives the percentage reduction of the total time when the change tracking procedure is used, namely 100(total computation time without the change tracking procedure - total computation time with the change tracking procedure)/(total computation time without the change tracking procedure). The last row in the table shows the average of the computation time of the phase 1 and phase 2 procedures with/without the change tracking procedure.

Table 6.1: Total computational time for the phase 1 and phase 2 procedures with/without the change tracking procedure

Problem	Total computation time (s)		% reduction
	Without change tracking procedure	With change tracking procedure	
804	1529	1151	24.72
1051	840	390	**53.57**
1351	2491	1905	23.52
1599	2127	1581	25.67
1932	3486	2966	14.92
2100	5153	3337	35.24
Average	1562.6	1133.0	**29.61**

Examining Table 6.1 it shows that including the change tracking procedure in our phase 1 and phase 2 procedures, the total computational time of the test problems could be reduced by up to 53.57%. On average the computation time is decreased by some 29.6%. Furthermore, both solution sets with/without change tracking procedure have the same quality routes. This is because by the nature of the change tracking procedure we must end up with the same routes as we end up with when we do not apply the change tracking procedure. Thus, it is proved that our change tracking procedure is effective when it is included in our phase 1 and phase 2 procedures with the DFP procedure.

Since the MDVRPI problems only involved up to 288 customers, this procedure is not included in our phase 1 and phase 2 procedures when generating the solutions for the problems.

6.3 Reverse order procedure

In a further attempt to improve the solution, this section presents our reverse order procedure where the objective of this procedure is to reduce crossing routes when we interchange two customers on the same route. In this procedure after interchanging the two customers on the same route, the positions of other customers between this pair are changed in the reverse order.

Figure 6.2 illustrates an example of a new route constructed with/without the reverse order procedure after two customers on the route have interchanged. This example consists of one single depot, one disposal facility and five customers. The original route (i.e. route a) starts from the depot→1→2→3→4→5→disposal facility→depot. After interchanging between customer 2 and customer 5, the new route is depot→1→5→3→4→2→disposal facility→depot (i.e. route b).

Here, we can see that many crossing arcs have occurred on this route. Preliminary computational experiments indicated that by reducing these crossing arcs, the distance travelled can be improved. In order to reduce the crossing arcs, the positions of customers 4 and 3 (customers between 5 and 2) are changed in a reverse order. Thus, after applying our reverse order procedure the new route becomes depot→1→5→4→3→2→disposal facility→depot (i.e. route c) and definitely with less crossing arcs.

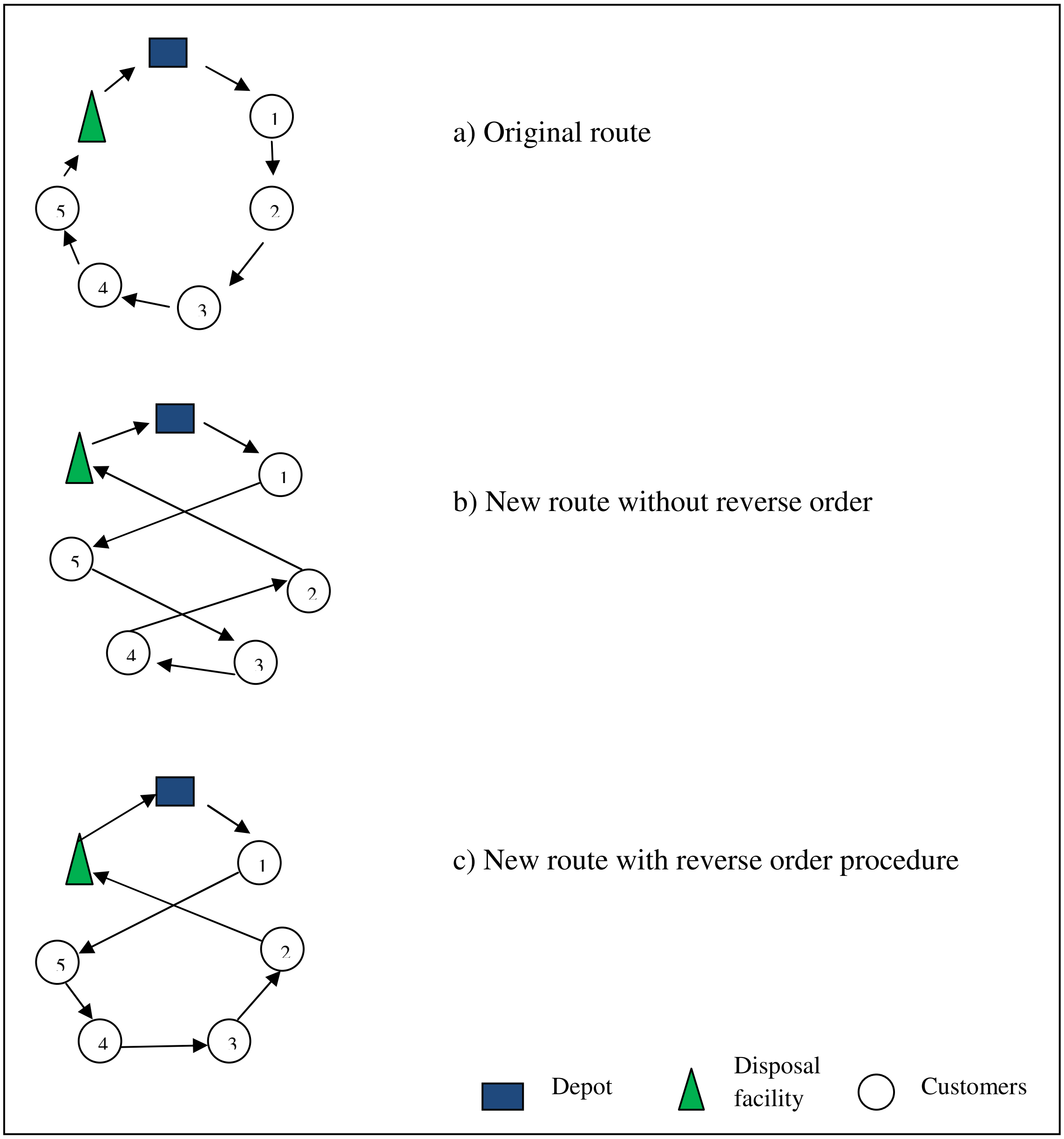


Figure 6.2: An example of two routes with/without the reverse order procedure

This procedure can be applied only in our phase 2 procedure but not in our phase 1 procedure. The logic here is that in our phase 1 procedure we move a customer immediately before/after its neighbour who may or may not be on the same route (refer to Section 4.2.1 in Chapter 4). Thus, there is no customer between this pair that the reverse order can be applied to. On the other hand, in our phase 2 procedure we interchange the positions of two customers that may or may not be on the same route. But we can reduce crossing arcs that occurred on the same route. Thus, this procedure is only included in our phase 2 procedure where we interchange two customers on the same route.

Phase 2

In pseudocode we:

For all customers $i \in C$:

For all customers $j \in N(i,K)$:

If i and j are on the same route **then:**

- we interchange customers i and j, i.e. customer i moves to the position of customer j and customer j moves to the position of customer i on the route
- **reverse the positions** of the customers between i and j on the route
- Evaluate this new route. If it is better than the original route (FEASIBLE and of lower total distance) then keep it, else do not

end if

end for

end for

6.4 Results

This section reports computational results for the phase 1 and phase 2 procedures as well as our three metaheuristics (TS, VNS and VNTS) with the DFP, reverse order and change tracking procedures discussed above. For both sets of benchmark problems waste collection VRPTW and MDVRPI, three sets of solutions are reported:

- First, computational results with reverse order only (DFP=No and Reverse=Yes). For this solution, every move (i.e. moving/swapping customers) that we have made on the vehicle routes with the reverse order procedure is evaluated using our route evaluation procedure discussed in Chapter 4 (refer to Section 4.1).
- Second, computational results with DFP procedure only (DFP=Yes and Reverse=No). For this solution, the reverse order procedure is not applied in improving the vehicle routes. Every move (i.e. moving/swapping customers) that we have made on the vehicle routes is evaluated using the DFP procedure.
- Third, computational results with DFP and reverse order procedures (DFP=Yes and Reverse=Yes). For this solution, every move (i.e. moving/swapping customers) that we have made on the vehicle routes with the reverse order procedure is evaluated using the DFP procedure.

For each set of solutions, ISP1P2, TS, VNS and VNTS solutions are reported before/after applying our VR procedure discussed in Chapter 4 (refer to Section 4.3).

6.4.1 Computational results for the waste collection VRPTW problems

Computational results for the DFP and reverse order procedures tested on waste collection VRPTW problems are reported in this section. Table 6.2(a) reports solutions of ISP1P2 and our three metaheuristics (TS, VNS and VNTS) with DFP=No, Reverse=Yes before applying the VR procedure. Note here that we use the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. In the same manner as results reported in the previous chapter, here ISP1P2 and TS solutions are also obtained when K=50 and for the VNS and VNTS, we use K*={5,10,25,50).

Table 6.2 a) DFP=No, Reverse=Yes without our VR procedure for the waste collection VRPTW problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure:			
ISP1P2	100	406.6	12.52
TS	100	596.2	12.53
VNS	100	527.3	12.53
VNTS	100	553.2	12.53
Savings approach:			
ISP1P2	102	432.6	10.17
TS	102	712.4	10.21
VNS	102	617.8	10.18
VNTS	102	744.4	10.19
Different initial customer based on current IS procedure:			
ISP1P2	98	850.7	14.48
TS	98	1080.7	14.49
VNS	98	1018.0	14.48
VNTS	**98**	**1127.6**	**14.50**

Results in Table 6.2(a) shows that our solutions from three different IS procedures (i.e. current IS, savings approach and different initial customer based on current IS procedure) travel less distance than Kim et al (2006). However, in terms of the total number of vehicles used only solutions from the different initial customer based on current IS procedure use one less vehicle than Kim et al (2006) whilst solutions from the IS procedure use one extra vehicle and the solutions from the savings approach use three extra vehicles. Of all solutions reported in the table, VNTS from the different initial customer based on current IS procedure produces the best solution, travels approximately 14.5% less distance and uses one less vehicle than Kim et al (2006) with 1127.6 seconds of computational time. Note here that for space reasons we, in this table as in other tables in this chapter, do not give individual results for the test problems.

Table 6.2 b) DFP=No, Reverse=Yes with our VR procedure for the waste collection VRPTW problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure:			
ISP1P2	97	545.8	11.67
TS	97	737.0	11.68
VNS	97	669.1	11.68
VNTS	97	698.8	11.68
Savings approach:			
ISP1P2	97	573.2	10.20
TS	97	1059.7	10.23
VNS	97	891.5	10.21
VNTS	97	1124.0	10.23
Different initial customer based on current IS procedure:			
ISP1P2	97	920.8	13.54
TS	97	1202.7	13.56
VNS	97	1113.0	13.54
VNTS	**97**	**1224.8**	**13.57**

Table 6.2(b) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=No, Reverse=Yes after applying our VR procedure. Now, all of our solutions are better than Kim et al (2006) in terms of the total number of vehicles used as well as the distance travelled. Here, all solutions use two less vehicles than Kim et al (2006). However, the reduction of the vehicles used does not guarantee that the distance travelled also will be reduced. For example, before applying VR procedure the best solution from VNTS for the different initial customer based on current IS procedure travel approximately 14.5%

less distance than Kim et al (2006) with 98 vehicles (from Table 6.2(a)). Now with 97 vehicles, the solution travel approximately 13.57% less distance than Kim et al (2006).

Table 6.3 a) DFP=Yes, Reverse=No without our VR procedure for the waste collection VRPTW problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure:			
ISP1P2	100	1032.4	11.29
TS	100	1567.3	11.30
VNS	100	1301.0	11.30
VNTS	100	1368.6	11.30
Savings approach:			
ISP1P2	101	937.0	7.83
TS	101	1438.0	7.83
VNS	101	1165.7	7.83
VNTS	101	1206.6	7.83
Different initial customer based on current IS procedure:			
ISP1P2	98	1470.2	13.40
TS	**98**	**1958.8**	**13.41**
VNS	98	1755.7	13.40
VNTS	**98**	**1844.1**	**13.41**

Table 6.3(a) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=No before applying the VR procedure. Results in the table show that the best solutions are from TS and VNTS for the different initial customer based on current IS procedure. Both solutions are better than Kim et al (2006) in terms of the number of vehicles used (i.e. one less vehicle than Kim et al (2006)) as well as distance traveled (i.e. approximately 13.41% less distance). However VNTS is to be preferred, having a lower average time than TS. On the other hand, other solutions (i.e. for the current IS

procedure and savings approach) involve more vehicles than Kim et al (2006), yet in terms of the distance traveled they are still outperform solutions from Kim et al (2006).

Table 6.3 b) DFP=Yes, Reverse=No with our VR procedure for the waste collection VRPTW problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure:			
ISP1P2	97	1345.8	9.88
TS	97	1908.1	9.88
VNS	97	1622.6	9.88
VNTS	97	1694.4	9.88
Savings approach:			
ISP1P2	97	1191.1	7.60
TS	97	1671.2	7.60
VNS	97	1434.7	7.60
VNTS	97	1498.6	7.60
Different initial customer based on current IS procedure:			
ISP1P2	97	1555.0	12.08
TS	**97**	**2046.3**	**12.09**
VNS	97	1791.3	12.08
VNTS	**97**	**1812.7**	**12.09**

Table 6.3(b) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=No after applying the VR procedure. Results in the table show that all solutions have reduced the number of vehicles used (e.g. solutions for the current IS procedure reduce from 100 vehicles to 97 vehicles). The decreasing positive numbers on the last column in the table compared with the last column in Table 6.3(a) indicate that with fewer vehicles all solutions in this table travel more distance than before. However, all solutions shows in the table are better than Kim et al (2006) in terms of the total number of vehicles used as well as distance travelled. Again, VNTS and TS for the

different initial customer based on current IS procedure produce the best solutions. They travel approximately 12.09% less distance and use two less vehicles than Kim et al (2006). Yet, VNTS has lower computation times than TS.

Table 6.4 a) DFP=Yes, Reverse=Yes without our VR procedure for the waste collection VRPTW problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure:			
ISP1P2	100	1189.9	12.85
TS	100	1819.7	12.85
VNS	100	1513.6	12.85
VNTS	100	1606.7	12.87
Savings approach:			
ISP1P2	101	1176.7	10.23
TS	101	1815.7	10.24
VNS	101	1515.7	10.24
VNTS	101	1606.2	10.24
Different initial customer based on current IS procedure:			
ISP1P2	98	1731.1	14.87
TS	98	2532.1	14.89
VNS	98	2062.3	14.88
VNTS	**98**	**2143.3**	**14.90**

Table 6.4(a) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=Yes before applying the VR procedure. As results presented in Table 6.2(a) and 6.3(a) above, here in this table all solutions also travel less distance than Kim et al (2006) but only solutions for the different initial customer based on current IS procedure use fewer vehicles than Kim et al (2006). The best solution is from VNTS involves 98 vehicles and travels approximately 14.9% less distance than Kim et al (2006) with 2143.3 seconds of computation time.

Table 6.4 b) DFP=Yes, Reverse=Yes with our VR procedure for the waste collection VRPTW problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure:			
ISP1P2	97	1607.7	11.85
TS	97	2289.4	11.89
VNS	97	1959.6	11.85
VNTS	97	2102.4	11.86
Savings approach:			
ISP1P2	97	1559.1	10.03
TS	97	2260.1	10.10
VNS	97	1904.6	10.05
VNTS	97	1993.7	10.05
Different initial customer based on current IS procedure:			
ISP1P2	97	1937.2	14.09
TS	97	2740.7	14.11
VNS	97	2268.7	14.10
VNTS	**97**	**2349.3**	**14.12**

Table 6.4(b) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=Yes after applying the VR procedure. Our three metaheuristics (TS, VNS and VNTS) produce solutions of almost similar quality with their ISP1P2 solutions. They clearly show that all solutions are better than Kim et al (2006) in terms of the total number of vehicles used and the distance travelled. The best solution travels approximately 14.12% less distance than Kim et al (2006) with 97 vehicles and 2349.3 seconds of computation time (as highlighted in the table).

6.4.2 Conclusions for the waste collection VRPTW problems

In order to choose the best initial solution procedure for the waste collection VRPTW problems, Table 6.5 reports the best metaheuristic solution for every initial solution procedure taken from the tables presented above.

Table 6.5: The best metaheuristic solution for every initial solution procedure

Algorithm	Table	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
Current IS procedure	6.3(b)	97	2289.4	11.89
Savings approach	6.1(b)	97	1059.7	10.23
Different initial customer based on current IS procedure	**6.3(b)**	**97**	**2349.3**	**14.12**

Examining Table 6.5 it shows that all solutions use two less vehicles than Kim et al (2006). But in terms of the distance travelled, on average the solution from the different initial customer based on current IS procedure shows the highest percentage improvement over Kim et al (2006), travelling approximately 14.12% less distance than Kim et al (2006). Moreover our solution from this procedure even without our VR procedure is still better than Kim et al (2006) (refer to Table 6.4(a)) in terms of the number of vehicles used (i.e. one less vehicle) as well as the distance travelled (i.e. approximately 14.90% less distance travelled). Thus, in this thesis we may conclude that the different initial customer based on current IS procedure is the best initial solution procedure for the waste collection VRPTW problems.

This thesis reports four sets of computational results for the waste collection VRPTW problems (as shown in Table 6.6).

Table 6.6: Four solution procedures with the DFP and reverse order

		Reverse	
		Yes	**No**
DFP	**Yes**	(a)	(c)
	No	(b)	(d)

Note here that computational results for the DFP=No, Reverse=No (d) case are reported in Chapter 5 whereas other results (a, b, and c) are reported in this section. In order to choose the best solution procedure for the waste collection VRPTW problems, Table 6.7 reports the best solution from every case for the different initial customer based on current IS procedure (since this is the best procedure for the waste collection VRPTW problems). Our metaheuristic solutions discussed in the previous section (Section 6.4.1) indicate that our VNS has a lower average computation time than either TS or VNTS. However in terms of the distance travelled, solutions from our VNTS and TS are better than the solution from our VNS. Both procedures VNTS and TS produce routes of similar quality yet our VNTS has a lower computation time than TS. Thus, VNTS is to be preferred for solving the waste collection VRPTW problems.

Table 6.7: Four solution procedures for the different initial customer based on current IS procedure

Procedures	Table	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Kim et al (2006)
DFP=No, Reverse=No	5.1(d)	97	726.7	11.28
DFP=No, Reverse=Yes	6.1(b)	97	1224.8	13.57
DFP=Yes, Reverse=No	6.2(b)	97	1812.7	12.09
DFP=Yes, Reverse=Yes	**6.3(b)**	**97**	**2349.3**	**14.12**

Four solution procedures from our chosen metaheuristic (VNTS) in Table 6.7 show that they use two less vehicles than Kim et al (2006). On average DFP=Yes, Reverse=Yes solution produces the largest improvement in distance over Kim et al (2006) compared to other solution sets (as highlighted in the table). Even though this solution has the highest average computation time (2349.3 seconds which is approximately 40 minutes) than other solutions, the time taken to obtain the best solution that we can achieve with the DFP and reverse order procedures is worthwhile to solve the waste collection VRPTW problems. Moreover, solutions in this table show that the reverse order procedure has a better effect on the test problems (approximately 13.57% less distance travelled) compared to the DFP procedure (approximately 12.09% less distance travelled). Combining these two procedures with our phase 1 and phase 2 procedures as well as our VNTS produce the best solution for the waste collection VRPTW problems.

6.4.3 Computational results for the MDVRPI problems

This section reports computational results for the DFP and reverse order procedures tested on MDVRPI problems. Table 6.8(a) reports solutions of ISP1P2 and our three metaheuristics (TS, VNS and VNTS) with DFP=No, Reverse=Yes before applying the VR procedure. Note here that we use the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. In the same manner as results reported in the previous chapter, here ISP1P2 and TS solutions are also obtained when K=|C| and for the VNS and VNTS, we use K*={5,10,25,|C|).

Table 6.8 a) DFP=No, Reverse=Yes without our VR procedure for the MDVRPI problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
Current IS procedure:				
ISP1P2	33	110.3	-10.69	1
TS	33	170.4	-10.50	1
VNS	33	133.0	-10.69	1
VNTS	33	133.2	-10.56	1
Farthest from depot:				
ISP1P2	33	90.2	-12.63	0
TS	33	155.3	-12.53	0
VNS	33	117.7	-12.59	0
VNTS	33	112.6	-12.58	0
Savings approach:				
ISP1P2	31	97.3	-11.85	0
TS	31	166.1	-11.73	0
VNS	31	119.4	-11.85	0
VNTS	31	120.3	-11.78	0
Different initial customer based on current IS procedure:				
ISP1P2	31	97.1	-10.21	0
TS	**31**	**153.5**	**-10.16**	**0**
VNS	31	120.0	-10.21	0
VNTS	31	120.7	-10.21	0

Note here the negative numbers on the fourth column indicate that our solutions for four different IS procedures (i.e. current IS, farthest from depot, savings approach and different initial customer based on current IS procedure) reported in this chapter are worse than Crevier et al (2007). The smallest negative number on that column is the best solution that we have for the test problems (as highlighted in the table). Solutions for the savings approach and the different initial customer based on current IS procedure involve the same total number of vehicles used as in Crevier et al (2007) whilst other solutions used two extra vehicles. Since our phase 1 and phase 2 procedures have no direct control over the number of vehicles used, solutions for our IS procedure produce one infeasible solution for the ten test problems. On the other hand, other solutions are feasible.

Table 6.8 b) DFP=No, Reverse=Yes with our VR procedure for the MDVRPI problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
Current IS procedure:				
ISP1P2	31	111.4	-9.72	0
TS	**31**	**171.7**	**-9.50**	**0**
VNS	31	134.1	-9.72	0
VNTS	31	134.4	-9.57	0
Farthest from depot:				
ISP1P2	31	92.4	-12.09	0
TS	31	157.7	-11.94	0
VNS	31	119.8	-12.04	0
VNTS	31	114.8	-11.99	0

Table 6.8(b) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=No, Reverse=Yes after applying the VR procedure. Here, we only show solutions that have vehicles reduction after using the VR procedure (i.e. solutions for the current IS and farthest from depot procedures). Both solutions reduce two vehicles. Solutions for our IS procedure now are all feasible for the test problems. In terms of the distance travelled, our solutions are still worse than Crevier et al (2007) but TS solutions for our IS procedure is the best of all solutions reported in the table. The solution involves 31 vehicles as in Crevier et al (2007) and travels approximately 9.50% more distance than Crevier et al (2007) with 171.7 seconds of the computation time.

Table 6.9 a) DFP=Yes, Reverse=No without our VR procedure for the MDVRPI problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
Current IS procedure:				
ISP1P2	33	300.4	-12.51	1
TS	33	464.4	-12.49	1
VNS	33	357.2	-12.51	1
VNTS	33	359.0	-12.49	1
Farthest from depot:				
ISP1P2	33	274.2	-13.45	0
TS	33	443.1	-13.44	0
VNS	33	330.0	-13.45	0
VNTS	33	331.3	-13.44	0
Savings approach:				
ISP1P2	31	205.9	-12.29	0
TS	31	355.0	-12.26	0
VNS	31	260.4	-12.29	0
VNTS	31	262.2	-12.27	0
Different initial customer based on current IS procedure:				

ISP1P2	31	279.7	-10.49	0
TS	31	397.6	-10.48	0
VNS	31	322.4	-10.49	0
VNTS	**31**	**325.3**	**-10.46**	**0**

Table 6.9(a) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=No before applying the VR procedure. Again, solutions from Crevier et al (2007) are better than us in terms of the distance travelled. Solutions for the savings approach and the different initial customer based on current IS procedure use 31 vehicles as in Crevier et al (2007) whilst other solutions use two extra vehicles. All solutions are feasible except for one solution for our IS procedure. The best solution shown in the table is the VNTS solution for the different initial customer based on current IS procedure. It travels approximately 10.46% more distance than Crevier et al (2007) with 31 vehicles and 325.3 seconds of the computation time.

Table 6.9 b) DFP=Yes, Reverse=No with our VR procedure for the MDVRPI problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
Current IS procedure:				
ISP1P2	31	302.0	-11.72	0
TS	31	466.0	-11.70	0
VNS	31	358.8	-11.72	0
VNTS	31	360.6	-11.70	0
Farthest from depot:				
ISP1P2	31	276.6	-13.65	0
TS	31	445.8	-13.64	0
VNS	31	332.4	-13.65	0
VNTS	31	333.7	-13.64	0

Table 6.9(b) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=No after applying the VR procedure. Solutions for the savings approach and the different initial customer based on current IS procedure remain the same as reported in Table 6.9(a). On the other hand, solutions for the current IS procedure and the farthest from depot reduce two vehicles. After reducing the vehicles, the distance travelled for the current IS solutions have reduced as well (e.g. on average the % improvement in distance over Crevier et al (2007) for ISP1P2 solution reduces from -12.51% (refer to Table 6.9(a)) to -11.72%) whilst the distance travelled for the farthest from depot solutions have increased (e.g. on average the % improvement in distance over Crevier et al (2007) for ISP1P2 solution increases from -13.45% (refer to Table 6.9(a)) to -13.65%). However, all solutions are still worse than Crevier et al (2007) but they involve the same total number of vehicles used as in Crevier et al (2007).

Table 6.10(a) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=Yes before applying the VR procedure. Here, all solutions are feasible for the ten problems but solutions from Crevier et al (2007) are better than us in terms of the distance travelled. In terms of the total number of vehicles used, only solutions for the farthest from depot involve two extra vehicles than Crevier et al (2007). The rest use the same number of vehicles as in Crevier et al (2007). TS solution for the current IS procedure is the best solution that we have which travels approximately 7.87% more distance than Crevier et al (2007) with 31 vehicles and 544 seconds of computational time.

Table 6.10 a) DFP=Yes, Reverse=Yes without our VR procedure for the MDVRPI problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
Current IS procedure:				
ISP1P2	31	359.2	-7.88	0
TS	**31**	**544.0**	**-7.87**	**0**
VNS	31	428.2	-7.88	0
VNTS	31	430.1	-7.88	0
Farthest from depot:				
ISP1P2	33	358.0	-10.92	0
TS	33	528.6	-10.92	0
VNS	33	425.6	-10.92	0
VNTS	33	427.4	-10.90	0
Savings approach:				
ISP1P2	31	280.9	-10.62	0
TS	31	475.9	-10.62	0
VNS	31	350.4	-10.62	0
VNTS	31	352.0	-10.62	0
Different initial customer based on current IS procedure:				
ISP1P2	31	305.3	-9.49	0
TS	31	531.8	-9.42	0
VNS	31	401.1	-9.49	0
VNTS	31	382.7	-9.45	0

Table 6.10 b) DFP=Yes, Reverse=Yes with our VR procedure for the MDVRPI problems

Algorithm	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)	Number of infeasible solutions
Farthest from depot:				
ISP1P2	32	359.5	-10.58	0
TS	32	530.2	-10.58	0
VNS	32	427.1	-10.58	0
VNTS	32	428.9	-10.56	0

Table 6.10(b) reports solutions of ISP1P2, TS, VNS and VNTS with DFP=Yes, Reverse=Yes after applying the VR procedure. Only solutions for the farthest from depot have an effect after applying the VR procedure. They use one less vehicle and travel less distance than before (e.g. the average % improvement in distance over Crevier et al (2007) decreases from -10.92% (refer to Table 6.10(a)) to -10.58% for the ISP1P2 solution). However even after applying the VR procedure, this solution still uses one extra vehicle than Crevier et al (2007) but other solutions use the same number of vehicles as in Crevier et al (2007). In terms of the distance travelled all solutions are worse than Crevier et al (2007).

6.4.4 Conclusions for the MDVRPI problems

In order to choose the best initial solution procedure for the MDVRPI problems, Table 6.11 reports the best metaheuristic solution for every initial solution procedure taken from the tables presented above.

Examining Table 6.11 it is clear that all solutions involve the same number of vehicles as in Crevier et al (2007). However the negative numbers in the last column indicate that in terms of the distance travelled, our solutions are worse than Crevier et al (2007). The best solution that we can achieve for the test problems is from our current IS procedure, travelling approximately 7.87% more distance than Crevier et al (2007) with 544 seconds of the computation time. Thus, in this thesis we may conclude that our current IS procedure is the best initial solution procedure for the MDVRPI problems.

Table 6.11: The best metaheuristic solution for every initial solution procedure

Algorithm	Table	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)
Current IS procedure	**6.10(a)**	**31**	**544.0**	**-7.87**
Farthest from depot	6.8(b)	31	157.7	-11.94
Savings approach	6.10(a)	31	350.4	-10.62
Different initial customer based on current IS procedure	6.10(a)	31	531.8	-9.42

This thesis also reports four set of computational results for the MDVRPI problems (refer to Table 6.6). Computational results for the DFP=No, Reverse=No (d) case for this problems are reported in Chapter 5 whereas other results (a, b, and c) are reported in this section. In order to choose the best solution procedure for the MDVRPI problems, Table 6.12 reports the best solution from every case for the current IS procedure (since this is the best procedure for the MDVRPI problems).

Table 6.12: Four solution procedures for our IS procedure

Procedures	Metaheuristic	Table	Total number of vehicles used	Average computation time (s)	Average % improvement in distance over Crevier et al (2007)
DFP=No, Reverse=No	VNS	5.2(d)	31	64.3	-16.20
DFP=No, Reverse=Yes	TS	6.4(b)	31	171.7	-9.50
DFP=Yes, Reverse=No	VNTS	6.5(b)	31	360.6	-11.70
DFP=Yes, Reverse=Yes	**TS**	**6.6(a)**	**31**	**544.0**	**-7.87**

Four solution procedures in Table 6.12 are obtained from our different metaheuristics. All solutions use the same total number of vehicles as in Crevier et al (2007). The VR procedure is applied for all solution procedures in order to reduce the number of vehicles used except for the DFP=Yes, Reverse=Yes. However in terms of the distance travelled, all solutions are worse than Crevier et al (2007). Our TS solution with the DFP and reverse order procedures is the best that we can achieve for the MDVRPI problems (as highlighted in the table). Moreover, solutions in this table show that the reverse order procedure have a better effect on the test problems (approximately 9.5% more distance travelled) compared to the DFP procedure (approximately 11.7% more distance travelled). Since the average computation time shown in the table is reasonable for every solution procedure, our metaheuristic that produces the best solution (TS) is to be preferred. Thus, in this thesis we can conclude that combining DFP and the reverse order procedures with our phase 1 and phase 2 procedures as well as our chosen metaheuristic (TS) are the best algorithms for solving the MDVRPI problems.

6.5 Conclusion

In this chapter, we have presented another route evaluation procedure namely DFP procedure based on Hemmelmayr et al (2009). This procedure improves the solution for both benchmark problems waste collection VRPTW and MDVRPI by choosing the best disposal facilities and the best position to place the disposal facilities on the routes. However, this DFP procedure is very time consuming. Thus to overcome this problem we have presented one procedure, namely change tracking, to evaluate only necessary changes on the routes. The effectiveness of this procedure when it is included in our phase 1 and phase procedures is shown in Table 6.1.

In a further attempt to improve the solution we also present one procedure, namely reverse order, to improve distance travelled by reducing crossing arcs which occurred after two customers on the same route are interchanged.

Four solution procedures are reported in this thesis:

- DFP=No, Reverse=No (both procedures DFP and reverse order are not applied)
- DFP=No, Reverse=Yes (only reverse order procedure is applied)
- DFP=Yes, Reverse=No (only DFP procedure is applied)
- DFP=Yes, Reverse=Yes (both procedures DFP and reverse order are applied)

As mentioned earlier, all solution procedures have been discussed in this chapter except for the DFP=No, Reverse=No. This solution procedure is discussed in Chapter 5.

Computational results show that when our phase 1 and phase 2 procedures as well as our chosen metaheuristics (i.e. VNTS for the waste collection VRPTW problems and TS for the MDVRP problems) are combined with the DFP and reverse order procedures (DFP=Yes, Reverse=Yes), they produce the best solutions for both benchmark problems waste collection VRPTW and MDVRPI. However in terms of the time taken to generate the solutions, this solution procedure has the largest average time than either DFP=No, Reverse=Yes or DFP=Yes, Reverse=No. Yet with our change tracking procedure, the time taken to generate the solutions is still reasonable. Thus, including both procedures DFP and reverse order in our phase 1 and phase 2 procedures and our chosen metaheuristics (VNTS and TS) are worthwhile.

In this chapter, we also conclude that the best initial solution procedure to construct initial routes for the MDVRPI problems is our current IS procedure. Our metaheuristics for this procedure produce the best solutions compared to metaheuristics solutions for other IS procedures used in this thesis. However, the best solution that we can achieve for the MDVRPI problems is still worse than Crevier et al (2007). On average our solution travels approximately 7.87% more distance than Crevier et al (2007) with 31 vehicles (in total) as in Crevier et al (2007).

On the other hand, the best initial solution procedure to construct initial routes for the waste collection VRPTW problems is the different initial customer based on our IS procedure. Our metaheuristics solutions for this procedure outperform the solution from Kim et al (2006). The best solution for the waste collection VRPTW problems involves 97 vehicles (in total) which is two less vehicles than Kim et al (2006) and on average it travels approximately 14.12% less distance than Kim et al (2006).

Note here that our solutions for the waste collection VRPTW problems have been published in Benjamin and Beasley (2010). However, there are some differences (as shown in Table 6.13) regarding the procedures used in that paper with the procedures used in the thesis that produces the best solution for the waste collection VRPTW problems.

Table 6.13: Procedures differences used in Benjamin and Beasley (2010) and the thesis

Procedure	Benjamin and Beasley (2010)	Thesis
Initial solution	Current IS procedure	Different initial customer based on current IS procedure
Phase 1 and Phase 2	• Moving customers/disposal facilities on the **same** route • Interchanging two customers on **different** routes • DFP=No, Reverse=No	• Moving customers/disposal facilities on the **same/different** route(s) • Interchanging two customers on the **same/different** route(s) • DFP=Yes, Reverse=Yes
VR	Use our route evaluation procedure	Use the DFP procedure
TS, VNS and VNTS	DFP=No, Reverse=No	DFP=Yes, Reverse=Yes

Solutions in Benjamin and Beasley (2010) indicate that our metaheuristics produce routes of similar quality. On this basis we have justified in choosing the metaheuristic involving the lowest computation time. In that paper our chosen metaheuristic was VNS, having a lower average time than either TS or VNTS. The last column in Table 6.13 shows the procedures that are results indicate our best. In terms of which metaheuristic to use we would choose VNTS.

Note here that in Benjamin and Beasley (2010) two solutions with/without our VR procedure for the waste collection VRPTW problems have been reported. The solution in the paper and our new solution from this thesis without the VR procedure are compared in Table 6.14 whereas both solutions with the VR procedure are compared in Table 6.15.

Table 6.14: Comparison results without our VR procedure

Problem	Solution	Number of vehicles used	Total distance (mile)	Total computation time (s)	% improvement in distance over Kim et al (2006)
102	Paper (2010)	3	183.5	3	10.53
	Thesis	3	156.9	16	23.50
277	Paper (2010)	3	464.5	8	11.91
	Thesis	3	454.7	272	13.77
335	Paper (2010)	6	204.5	16	0.24
	Thesis	6	186.7	219	8.93
444	Paper (2010)	11	89.1	19	-2.41
	Thesis	11	79.7	273	8.39
804	Paper (2010)	6	756.3	62	1.72
	Thesis	5	641.8	2152	16.60
1051	Paper (2010)	17	2251.6	124	5.01
	Thesis	17	2123.8	637	10.40
1351	Paper (2010)	8	915.1	119	11.98
	Thesis	8	874.7	2698	15.87
1599	Paper (2010)	14	1410.4	172	3.34
	Thesis	13	1206.1	2556	17.35
1932	Paper (2010)	16	1262.8	285	9.50
	Thesis	16	1127.7	6541	19.18
2100	Paper (2010)	16	1749.0	332	4.62
	Thesis	16	1558.1	6069	15.03
Average	Paper (2010)	**100**		**107**	**5.64**
	Thesis	**98**		**2143.3**	**14.90**

Solutions for the ten test problems in Table 6.14 show that our new solutions in the thesis have improved in terms of the distance travelled as well as the number of vehicles used compared to our solution in Benjamin and Beasley (2010). On average the percentage of the improvement in distance over Kim et al (2006) has increased from 5.64% to 14.90% and now we use one less vehicle than Kim et al (2006). Since our new solution in the thesis saves two more vehicles than before (100-98) and provides a reduction in distance compared to Kim et al (2006) of some 100(14.90-5.64)/5.64 = 164% over and above the reduction provided by Benjamin and Beasley (2010), the average time shown at the foot of the table is worthwhile.

Figure 6.3 and 6.4 show our final solution for the 102 problem from Benjamin and Beasley (2010) and from the thesis, respectively. Our solution in the thesis has improved by some 100(183.5-156.9)/183.5=14.5% compared with our solution from the paper. However, both of our solutions are better than Kim et al (2006). In particular, our solutions from the paper and the thesis travel approximately 10.53% and 23.50% less distance respectively. With respect to the number of vehicles used, both solutions involve the same number as in Kim et al (2006).

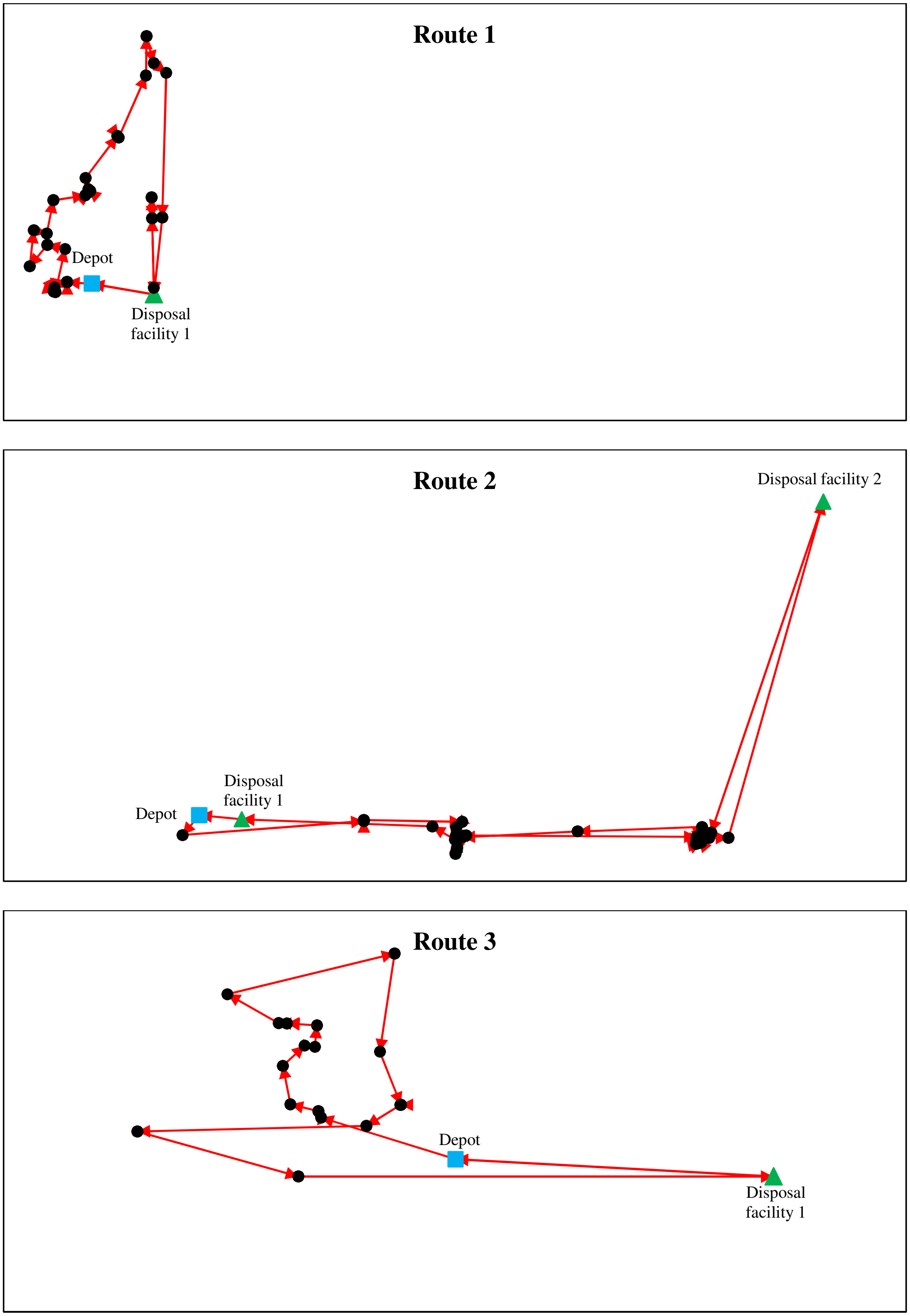


Figure 6.3: Final routes for the 102 problem from Benjamin and Beasley (2010)

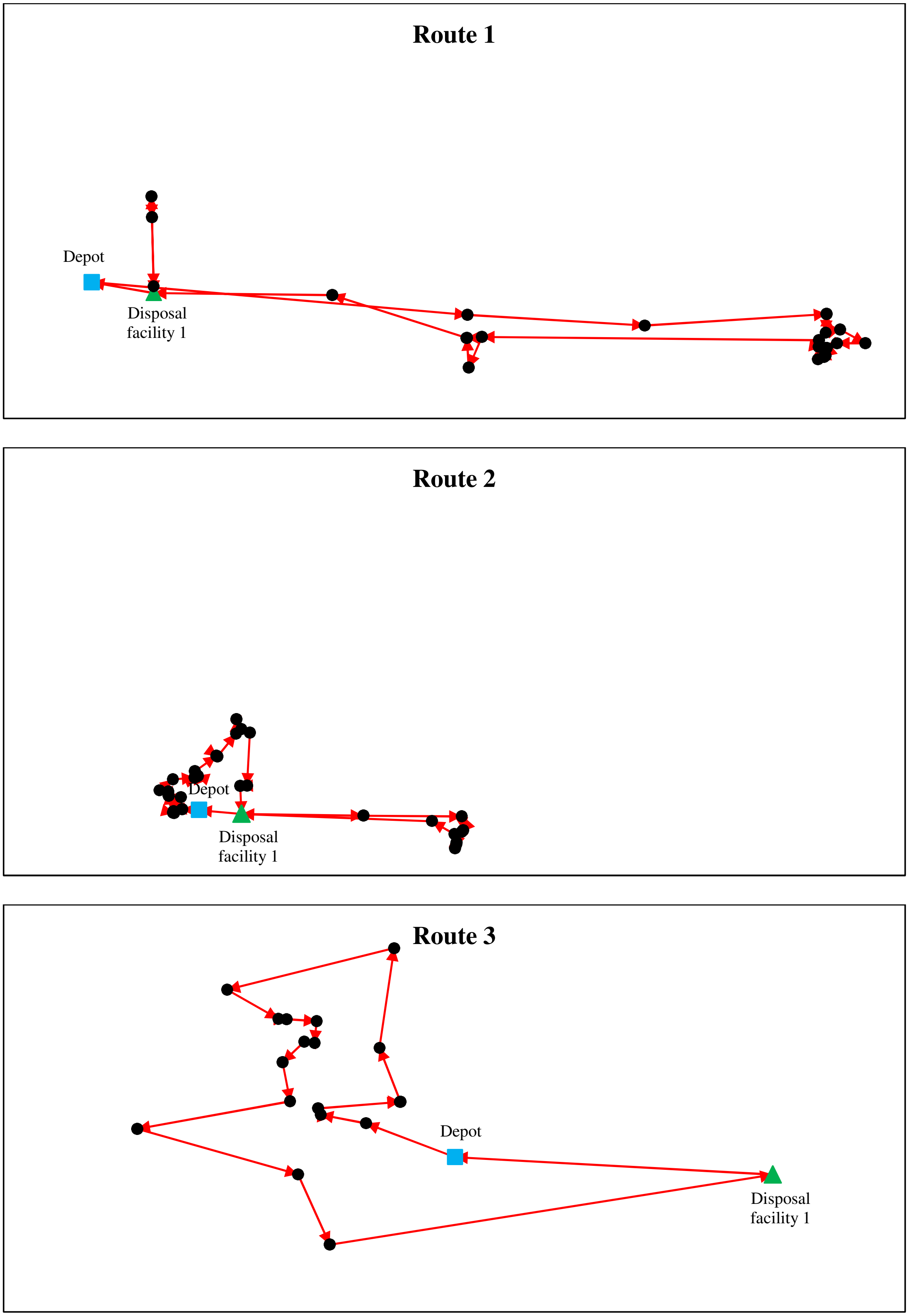


Figure 6.4: Final routes for the 102 problem from the thesis

Table 6.15: Comparison results with our VR procedure

Problem	Solution	Number of vehicles used	Total distance (mile)	Total computation time (s)	% improvement in distance over Kim et al (2006)
804	Paper (2010)	5	725.6	72	5.71
1351	Paper (2010)	7	1011.9	193	2.67
	Thesis	7	955.7	4736	8.08
1599	Paper (2010)	13	1364.7	252	6.48
Average	Paper (2010)	**97**		**123.0**	**5.43**
	Thesis	**97**		**2349.3**	**14.12**

Solutions in Table 6.15 show test problems where a reduction in the number of vehicles was achieved when the VR procedure is applied. Both solutions involve two less vehicles than Kim et al (2006). On average our new solution provides a reduction in distance compared to Kim et al (2006) of some 100(14.12-5.43)/5.43 = 160% over and above the reduction provided by Benjamin and Beasley (2010). Again we could say that the average computation time, 2349.3 seconds to generate this new solution is worthwhile since it shows huge improvement from our solutions in Benjamin and Beasley (2010).

In the next chapter conclusions from every chapter in this thesis are presented.

CHAPTER 7

CONCLUSIONS

This thesis considers a vehicle routing problem (VRP) that arises when a set of customers have waste that must be collected by vehicles. In such situations it is common for the amount of waste to be such that vehicles become full during their working day and have time to visit a waste disposal facility to empty themselves before going on to visit more customers and collect more waste. As such multiple visits to waste disposal facilities may be made during the working day.

Moreover, there were a significant set of constraints relating to real-world considerations. Specifically we took into consideration time windows associated with customers, disposal facilities and the depot. We also took into consideration a driver rest period. This problem is a single period node routing problem and is often encountered in terms of waste collection from commercial customers.

This thesis considers exactly the same real life waste collection benchmark problems as in Kim et al (2006) involving multiple disposal facilities, driver rest period and customer/depot/disposal facility time windows. It consists of ten test problems, involving up to 2092 customers and 19 waste disposal facilities as publicly available at: http://www.postech.ac.kr/lab/ie/logistics/WCVRPTW_Problem/benchmark.html.

Because Kim et al (2006) have made their test problems publicly available, a direct computational comparison with their work is made in this thesis.

Furthermore, this thesis also considers a set of new benchmark instances generated by Crevier et al (2007) from those proposed by Cordeau et al (1997) for the multi-depot VRP (MDVRP). These instances contain up to 288 customers and seven depots. In this case the depots can act as intermediate replenishment facilities along the route of a vehicle. The instances and the best known solutions are available at http://www.hec.ca/chairedistributique/data/. To the best of our knowledge, no previous works are available, thus the comparative results are only made with the authors.

7.1 Chapter overview

In Chapter 2 examples of heuristic techniques that have been used to solve the VRP for deliveries were presented. We also reviewed some literature on VRP for collection. This included previous work dealing with waste collection as arc routing, as well as node routing; particularly skip problems and non-skip problems.

In Chapter 3 the notation and neighbour sets used in the thesis were presented. Furthermore we also presented a number of initial solution (IS) procedures used for constructing initial routes for two set of benchmark problems. Some of the IS procedures are adapted from the literature. The IS procedures used in this thesis:

- Our current IS procedure
- Different initial customer based on our current IS procedure

- Farthest from depot procedure
- The savings approach of Clarke and Wright (1964)
- The sweep algorithm of Gillett and Miller (1974)
- Different initial customer procedure based on the sweep algorithm of Gillett and Miller (1974)

The main characteristics of the waste collection VRPTW problems (i.e. from Kim et al (2006)) and the MDVRPI problems (i.e. from Crevier et al (2007)) were shown in Chapter 3. However, both benchmark problems have some characteristic differences such as:

- the size of the test problems (i.e. waste collection VRPTW problems involved up to 2092 customers, whereas MDVRPI problems only involved up to 288 customers)
- time windows of the nodes (depot/customer/disposal facility)
- limited/unlimited number of vehicles
- rest time period

Thus, the best initial solution for both problems were obtained from the different IS procedures. For example, computational results show that the best initial solution for the waste collection VRPTW problems is obtained from the different initial customer based on our IS procedure. This initial solution already outperforms the solution from Kim et al (2006). In total it uses 98 vehicles (one less vehicle than Kim et al (2006)) and a reduction of around 4.75% on average of the total distance travelled.

On the other hand, initial solution from the savings approach is the best for the MDVRPI problems. This procedure produces feasible solution for the ten test problems, involving 31 vehicles (in total) as in Crevier et al (2007). However in terms of the distance travelled, on average it travels approximately 20.73% more distance than Crevier et al (2007).

In Chapter 4 we presented our procedure to evaluate a given route, which involves inserting into the route (if necessary) disposal facility visits. Moreover, we have outlined how we can improve the initial solution using our phase 1 and phase 2 procedures. We also discussed how we can reduce the number of vehicles used using our vehicle reduction (VR) procedure. In order to prevent the number of customer moves/interchanges we have to examine being excessive when using both procedures, we use the neighbour set for a customer (i.e. K=50 for the waste collection VRPTW problems whereas K=|C| for the MDVRPI problems) to improve the solutions. Computational results for these two procedures were reported separately so that we can clearly see the effectiveness of both procedures on the two benchmark problem sets, waste collection VRPTW and MDVRPI.

The best ISP1P2 solution for the waste collection VRPTW problems provides a reduction in distance compared to Kim et al (2006) of some 173.7% over and above the reduction provided by initial solution from the different initial customer based on current IS procedure. The ISP1P2 solution involves 98 vehicles (in total). On the other hand, the best ISP1P2 solution for the MDVRPI problems is worse than Crevier et al

(2007). On average it travels approximately 14.14% more distance than Crevier et al (2007). With respect to the number of vehicles used, the ISP1P2 solution involves 31 vehicles (in total) as in Crevier et al (2007).

After applying our VR procedure, the results indicated that even though the procedure reduces the number of vehicles used for some test problems, the total distance travelled may/may not be reduced. For example the VRISP1P2 solution for the different initial customer based on current IS procedure for the waste collection VRPTW problems now uses one less vehicle than before. But this solution travels more distance than ISP1P2 solution. On average the percentage improvement in distance over Kim et al (2006) has decreased from 13.0% to 11.26%.

For the MDVRPI problems, the VRISP1P2 solution for the different initial customer based on current IS procedure shows the best solution for the problems. The solution involves the same number of vehicles used as in Crevier et al (2007). However in terms of the distance travelled, on average it travels approximately 13.57% more distance than Crevier et al (2007).

In Chapter 5 we have presented our three metaheuristics namely TS, VNS and VNTS. Here we also used the neighbour set for a customer to prevent the number of customer moves/interchanges we have to examine being excessive. For the waste collection VRPTW problems, the TS solutions were obtained when K=50 and for the VNS and VNTS, we used K*={5,10,25,50). For the MDVRPI problems the TS solutions were obtained when K=|C| and for the VNS and VNTS, we used K*={5,10,25,|C|}. Solutions from these metaheuristics started from the ISP1P2 solutions and the VRISP1P2 solution. Computational results showed that our metaheuristics give only small improvement on those solutions.

Of all solutions obtained from our metaheuristics, the solution for the different initial customer based on our IS procedure gives the best solution for the waste collection VRPTW problems. This solution outperforms the solution from Kim et al (2006) in terms of the distance travelled as well as the number of vehicles used. On average the best distance travelled that we achieved is approximately 13.02% less distance than Kim et al (2006) with 98 vehicles (in total). However if we want to use the minimum number of vehicles to serve the customers, our solution shows that we may use 97 vehicles and travel approximately 11.28% more distance than Kim et al (2006). In this chapter, we could say that all our metaheuristics produce almost the same quality routes for the test problems. On this basis we are justified in choosing the metaheuristic involving the lowest computation time. The results show that VNS is to be preferred, having a lower average time than either TS or VNTS.

For the MDVRPI problems, our metaheuristic solutions were worse than Crevier et al (2007) in terms of the distance travelled. However, three solutions (current IS procedures, savings approach and the different initial customer based on current IS procedure) involve (in total) 31 vehicles as in Crevier et al (2007). Solution for the different initial customer based on current IS procedure shows the best solution that we can achieve, travelling approximately 13.45% more distance than Crevier et al (2007).

In Chapter 6 we have presented another route evaluation procedure namely DFP procedure based on Hemmelmayr et al (2009). However, this DFP procedure is very time consuming. Thus to overcome this problem we have presented one procedure, namely change tracking, to evaluate only necessary changes to the routes. In a further attempt to improve the solution in this chapter we also present one procedure, namely reverse order, to improve distance travelled by reducing crossing arcs which occurred after two customers on the same route are interchanged.

Since some of the solutions of both sets of benchmark problems presented in Chapter 5 are still worse even after applying our metaheuristics, in this chapter the DFP and reverse order procedures are only tested on some of the solutions. For the waste collection VRPTW problems, three IS procedures are selected (i.e. current IS procedure, savings approach and the different initial customer based on current IS procedure) whereas for the MDVRPI problems, four IS procedures are selected (i.e. current IS procedure, farthest from depot, savings approach and the different initial customer based on current IS procedure).

This thesis reports four solution procedures:

- DFP=No, Reverse=No (both procedures DFP and reverse order are not applied)
- DFP=No, Reverse=Yes (only reverse order procedure is applied)

- DFP=Yes, Reverse=No (only DFP procedure is applied)
- DFP=Yes, Reverse=Yes (both procedures DFP and reverse order are applied)

All solution procedures are discussed in Chapter 6 except for the DFP=No, Reverse=No. This solution procedure was discussed in Chapter 5.

Computational results presented in this chapter indicate that our best solution for the waste collection VRPTW problems is obtained from our VNTS with the DFP and reverse order procedures (DFP=Yes, Reverse=Yes) for the different initial customer based on current IS procedure. On the other hand, the best solution for the MDVRPI problems is obtained from our VNS with the DFP and reverse order procedures (DFP=Yes, Reverse=Yes) for current IS procedure. Even though the best solution for these two benchmark problem sets has an higher average time than other solutions presented in this thesis, the time taken is still reasonable and worthwhile to achieve the best solution for the problems.

For the MDVRPI problems, the best solution from the algorithms mentioned earlier is still worse than Crevier et al (2007). On average our solution travels approximately 7.87% more distance than Crevier et al (2007) with 31 vehicles (in total) as in Crevier et al (2007). On the other hand, the best solution for the waste collection VRPTW problems outperforms the solution from Kim et al (2006). It involves 97 vehicles (in total) which is two less vehicles than Kim et al (2006) and on average it travels approximately 14.12% less distance than Kim et al (2006). However if we want to travel less distance than this solution, our solution shows that in total we could use 98 vehicles (still better than Kim et al (2006)) and travel approximately 14.9% less distance than Kim et al (2006).

Our solutions for the waste collection VRPTW problems have been published in Benjamin and Beasley (2010) in Computers and Operations Research. However the solutions in that paper do not include the DFP and reverse order procedures. Since the solutions indicate that our metaheuristics produce routes of similar quality, VNS becomes our preferred metaheuristic due to the lower average computation time than TS and VNTS.

Comparison between our solutions in Benjamin and Beasley (2010) and the new solutions in this thesis for the waste collection VRPTW problems are discussed. Before applying our VR procedure, Benjamin and Beasley (2010) use one extra vehicle than Kim et al (2006) but now our new solution is better than Kim et al (2006). We use one less vehicle than Kim et al (2006). In terms of the distance travelled, both of our solutions travel less distance than the solution from Kim et al (2006). On average the percentage of the improvement in distance over Kim et al (2006) has increased from 5.64% to 14.90%. After applying our VR procedure, both of our solutions involve two less vehicles than Kim et al (2006). Here, on average our new solution provides a reduction in distance compared to Kim et al (2006) of some 160% over and above the reduction provided by Benjamin and Beasley (2010).

In addition on average the time taken to generate our new solutions in this thesis with/without VR procedure are 2143.3 seconds and 2349.3 seconds, respectively. Both computation times are reasonable and worthwhile since our new solutions show a large improvement from our solutions in Benjamin and Beasley (2010).

7.2 Contributions to knowledge

In this section the contributions of this thesis to knowledge are summarised as follows:

- This thesis presents three metaheuristic algorithms (TS, VNS and VNTS) for waste collection VRPTW problems that produce better quality solutions than previous work presented in the literature.
- Our algorithms for the waste collection VRPTW problems have been published in Benjamin and Beasley (2010) in Computers and Operations Research.
- Our initial solution procedure (indeed all our algorithms) are able to deal effectively with real world constraints such as time windows and rest periods.
- Our initial solution procedure was especially developed for the waste collection VRPTW problem considered in this thesis.
- This thesis presents a modified DFP procedure (originally proposed by Hemmelmayr et al (2009)) to select/position disposal facilities on a vehicle route for a problem with time windows and rest period. This procedure is very time consuming but very effective for a problem with multiple disposal facilities.
- Our metaheuristics are suitable to solve small/large test problems due to the change tracking and neighbour sets procedures presented.

7.3 Suggestions for further research

This section presents a few suggestions to extend this thesis. First, some experiments could be done to improve the solutions for the MDVRPI problems. For example, we could vary the tabu tenure or number of iterations used in our metaheurstics for the MDVRPI problems.

Moreover, we could also extend our metaheuristics by including the phase 1 procedure in a further attempt to improve the solutions. New and emerging metaheuristic ideas (such as ant colony optimisation) could be applied to the problem.

In a further attempt to test the effectiveness of our metaheuristics, these metaheuristics can be tested on other VRP for collection problems. For a collection problem with single depot and multiple trips per day, our metaheuristics can directly solve this problem without any changes. However, for a collection problem with single depot and one trip per day, our metaheuristics need slight changes.

REFERENCES

Agha, S.R. (2006). Optimizing routing of municipal solid waste collection vehicles in Deir El-Balah-Gaza Strip. *The Islamic University Journal (Series of Natural Studies and Engineering)*, 14(2), 75-89.

Alagöz, A. Z., & Kocasoy, G. (2008). Improvement and modification of the routing system for the health-care waste collection and transportation in Istanbul. *Waste Management, 28*(8), 1461-1471. doi:10.1016/j.wasman.2007.08.024

Angelelli, E., & Speranza, M. G. (2002). The application of a vehicle routing model to a waste-collection problem: Two case studies. *Journal of the Operational Research Society, 53*(9), 944-952. doi:10.1057/palgrave.jors.2601402

Angelelli, E., & Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research, 137*(2), 233-247.

Apaydin, O., & Gonullu, M. T. (2007). Route optimization for solid waste collection: Trabzon (turkey) case study. *Global Nest Journal, 9*(1), 6-11.

Archetti, C., & Speranza, M. G. (2004). Vehicle routing in the 1-skip collection problem. *Journal of the Operational Research Society, 55*(7), 717-727. doi:10.1057/palgrave.jors.2601743

Aringhieri, R., Bruglieri, M., Malucelli, F., & Nonato, M. (2004). An asymmetric vehicle routing problem arising in the collection and disposal of special waste. *Electronic Notes in Discrete Mathematics*, 17(20), 41-47.

Arribas, C.A., Blazquez, C.A., & Lamas, A. (2010). Urban solid waste collection system using mathematical modelling and tools of geographic information systems. *Waste Management & Research,* 28(4), 355-363.

Badran, M.F., & El-Haggar, S.M. (2006). Optimization of municipal solid waste management in Port-Said-Egypt, *Waste Management*, 26, 534–545.

Baldacci, R., Bodin, L., & Mingozzi, A. (2006). The multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. *Computers & Operations Research, 33*(9), 2667-2702. doi:10.1016/j.cor.2005.02.023

Bautista, J., & Pereira, J. (2004). Ant algorithms for urban waste collection routing. *Ant Colony Optimization and Swarm Intelligence, Proceedings, 3172*, 302-309.

Bautista, J., Fernandez, E., & Pereira, J. (2008). Solving an urban waste collection problem using ants heuristics. *Computers & Operations Research, 35*(9), 3020-3033. doi:10.1016/j.cor.2007.01.029

Beasley, J.E. (1983). Route first—Cluster second methods for vehicle routing. *Omega,* 11(4), 403-408.

Belenguer, J.M., Benavent, E., Lacomme, P., & Prins, C. (2003). Heuristics and lower bounds for the mixed capacitated arc routing problem. Working Paper, Univ. Tecnologie de Troyes, France.

Benavent, E., Campos, V., Corberan, A., & Mota, E. (1992). The capacitated arc routing problem: lower bounds. *Networks*, 22: 669–690. doi: 10.1002/net.3230220706

Benjamin, A.M., & Beasley, J.E. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operations Research,* 37(12), 2270-2280.

Bodin, L., Mingozzi, A., Baldacci, R., & Ball, M. (2000). The rollon-rolloff vehicle routing problem. *Transportation Science, 34*(3), 271-288.

Boonkleaw, A., Nanthi, S., & Srinon, R. (2009). Strategic planning and vehicle routing algorithm for newspaper delivery problem: Case study of morning newspaper, Bangkok, Thailand. *Wcecs 2009: World Congress on Engineering and Computer Science, Vols i and Ii,* , 1067-1071.

Bruns, A., Klose, A., & Stahly, P. (2000). Restructuring of Swiss parcel delivery services. *OR Spektrum, 22*(2), 285-302.

Caramia, M., & Guerriero, F. (2010). A milk collection problem with incompatibility constraints. *Interfaces, 40*(2), 130-143. doi:10.1287/inte.1090.0475

Chalkias, C., & Lasaridi, K. (2009). Optimizing municipal solid waste collection using GIS, presented at 5th International Conference on Energy, Environment, Ecosystems and Sustainable Development/2nd International Conference on Landscape Architecture, Greece, September 2009. In Proceedings of Energy, Environment, Ecosystems, Development and Landscape architecture, 45-50.

Chang, N. B., Lu, H. Y., & Wei, Y. L. (1997). GIS technology for vehicle routing and scheduling in solid waste collection systems. *Journal of Environmental Engineering-Asce, 123*(9), 901-910.

Chang, N. B., & Wei, Y. L. (2002). Comparative study between the heuristic algorithm and the optimization technique for vehicle routing and scheduling in a solid waste collection system. *Civil Engineering and Environmental Systems, 19*(1), 41-65. doi:10.1080/10286600290012788

Chen, H., Hsueh, C., & Chang, M. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research, 36*(7), 2311-2319. doi:10.1016/j.cor.2008.09.010

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568-581.

Coene, S., Arnout, A., & Spieksma, F. C. R. (2008). The periodic vehicle routing problem: A case study. Working paper (2008) available from http://www.econ.kuleuven.be/public/n05012/

Cordeau, J. F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks, 30*(2), 105-119.

Crevier, B., Cordeau, J., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research, 176*(2), 756-773. doi:10.1016/j.ejor.2005.08.015

Cristallo, G. (1994). Optimisation de tournees de vehicules de transport container. Memoire de licence en Sciences Economiques et Sociales, Facultes Universitaires Notre-Dame de la Paix, Namur, Belgium.

DeMeulemeester, L., Laporte, G., Louveaux, F. V., & Semet, F. (1997). Optimal sequencing of skip collections and deliveries. *Journal of the Operational Research Society, 48*(1), 57-64.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271

Doerner, K. F., Gronalt, M., Hartl, R. F., Kiechle, G., & Reimann, M. (2008). Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows. *Computers & Operations Research, 35*(9), 3034-3048. doi:10.1016/j.cor.2007.02.012

Dongarra, J.J. (2009). Performance of various computers using standard linear equations software. Report CS-89-85, University of Tennessee, USA, September 2009. Available from http://www.netlib.org/benchmark/performance.ps last accessed October 1st 2009.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering, 57*(4), 1472-1483. doi:10.1016/j.cie.2009.05.009 ER

Fabri, A., & Recht, P. (2006). On dynamic pickup and delivery vehicle routing with several time windows and waiting times. *Transportation Research Part B: Methodological, 40*(4), 335-350. doi:10.1016/j.trb.2005.04.002

Faulin, J. (2003). Applying MIXALG procedure in a routing problem to optimize food product delivery. *Omega-International Journal of Management Science, 31*(5), 387-395. doi:10.1016/S0305-0483(03)00079-3

Fırıncı, N., Çelik, A., Akün, E., & Md. Atif Khan (2009). A Pilot Study for the Optimization of Routes for Waste Collection Vehicles for the Göçmenköy District of Lefkosa. *World Academy of Science, Engineering and Technology,* 49, 1055-1058.

Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle-routing. *Networks, 11*(2), 109-124.

Gendreau, M. (2003). An introduction to tabu search, in *Handbook of Metaheuristics*, F. Glover and G.A. Kochenberger (eds), pp 37-54, Kluwer Academic Publishers: Dordrecht, The Netherlands.

Ghose, M. K., Dikshit, A. K., & Sharma, S. K. (2006). A GIS based transportation model for solid waste disposal - A case study on asansol municipality. *Waste Management, 26*(11), 1287-1293. doi:10.1016/j.wasman.2005.09.022

Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research, 22*(2), 340.

Glover, F., & Laguna, M. (1997). Tabu Search, Kluwer Academic Publishers: Dordrecht, The Netherlands.

Golden, B. L., Dearmon, J. S., & Baker, E. K. (1983). Computational experiments with algorithms for a class of routing-problems. *Computers & Operations Research, 10*(1), 47-59.

Hansen, P., & Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research, 130*(3), 449-467. doi:10.1016/S0377-2217(00)00100-4

Hemmelmayr, V., Doerner, K.F., Hartl, R.F., & Rath, S. (2009). Metaheuristics for a real world solid waste collection problem. Working paper available from the third author at Department of Business Administration, University of Vienna, Bruenner Strasse 72, 1210 Vienna, Austria.

Karadimas, N.V., Kouzas, G., Anagnostopoulos, I., & Loumos, V. (2005). Urban solid waste collection and routing: The ant colony strategic approach. *I.J. of Simulation,* 6(12-13). 45-53.

Karadimas, N.V., Papatzelou, K., & Loumos, V.G. (2007). Optimal solid waste collection routes identified by the ant colony system algorithm. *Waste Management & Research,* 25(2), 139-147.

Kek, A. G. H., Cheu, R. L., & Meng, Q. (2008). Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling, 47*(1-2), 140-152. doi:10.1016/j.mcm.2007.02.007

Kim, B.I. (2009). Private communication.

Kim, B. I., Kim, S., & Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research, 33*(12), 3624-3642. doi:10.1016/j.cor.2005.02.045

Kollias, P. (1993). Waste: Urban – Industrial. Kollias Panagiotis, Athens (in Greek).

Komilis, D. P. (2008). Conceptual modeling to optimize the haul and transfer of municipal solid waste. *Waste Management, 28*(11), 2355-2365. doi:10.1016/j.wasman.2007.11.004

Kytöjoki, J., Nuortio,T., Bräysy, O., & Gendreau, M. (2007). An efficient variable neighbourhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research*, 34, 2743-2757.

Lacomme,P., Prins, C., & Ramdane-Cherif, W. (2002). Fast algorithm for general arc routing problems. Presented in: IFORS 2002 Conference, Edinburgh, UK.

Lambert, V., Laporte, G., & Louveaux, F. (1993). Designing collection routes through bank branches. *Computers & Operations Research, 20*(7), 783-791.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science, 43*(4), 408-416. doi:10.1287/trsc.1090.0301

le Blanc, I., van Krieken, M., Krikke, H., & Fleuren, H. (2006). Vehicle routing concepts in the closed-loop container network of ARN - a case study. *OR Spectrum, 28*(1), 53-71. doi:10.1007/s00291-005-0003-6

Li, J., Borenstein, D., & Mirchandania, P. B. (2008). Truck scheduling for solid waste collection in the city of Porto Alegre, Brazil. *Omega-International Journal of Management Science, 36*(6), 1133-1149. doi:10.1016/j.omega.2006.04.007

Li, L.Y.O., & Eglese, R.W. (1996). An interactive algorithm for vehicle routing for winter-gritting. *Journal of the Operational Research Society*, 47, 217–228.

Martagan, T.G., Ertek, G., Birbil, I., Yasar, M., Cakir, A., Okur, N., Gullu, G., Hacioglu, A., & Serim, O. (2006). Optimizing waste collection in an organized industrial region-A case study. In Proceedings of 4th International Logistics and Supply Chain Congress (2006), Izmir University of Economics Publication, 178-186.

McLeod, F., & Cherrett, T. (2008). Quantifying the transport impacts of domestic waste collection strategies. *Waste Management, 28*(11), 2271-2278. doi:10.1016/j.wasman.2007.09.041

Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*(11), 1097-1100. doi:10.1016/S0305-0548(97)00031-2

Mourão, M. C., & Almeida, M. T. (2000). Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of Operational Research, 121*(2), 420-434. doi:10.1016/S0377-2217(99)00045-4

Mourão, M. C., & Amado, L. (2005). Heuristic method for a mixed capacitated arc routing problem: A refuse collection application. *European Journal of Operational Research, 160*(1), 139-153. doi:10.1016/j.ejor.2004.01.023

Mourao, M. C., Nunes, A. C., & Prins, C. (2009). Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research, 196*(3), 856-868. doi:10.1016/j.ejor.2008.04.025

Novaes, A. G. N., & Graciolli, O. D. (1999). Designing multi-vehicle delivery tours in a grid-cell format. *European Journal of Operational Research, 119*(3), 613-634.

Nuortio, T., Kytojoki, J., Niska, H., & Braysy, O. (2006). Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications, 30*(2), 223-232. doi:10.1016/j.eswa.2005.07.009

Ogwueleka, T.Ch. (2009). Municipal solid waste characteristics and management in Nigeria. *Iranian Journal of Environmental Health Science & Engineering*, 6(3), 173-180.

Ombuki, B., Ross, B. J., & Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence, 24*(1), 17-30. doi:10.1007/s10489-006-6926-z

Ombuki-Berman, B.M., Runka, A., & Hanshar, F.T. (2007). Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms, presented at Computational Intelligence, Banff, July 2007. In Proceedings of Computational Intelligence (CI 2007), R. Andonie (ed), 91-97, Acta Press, Calgary, Canada.

Park, J., & Kim, B. (2010). The school bus routing problem: A review. *European Journal of Operational Research, 202*(2), 311-319. doi:DOI: 10.1016/j.ejor.2009.05.017

Ree, S. B., & Yoon, B. S. (1996). A two-stage heuristic approach for the newspaper delivery problem. *Computers & Industrial Engineering, 30*(3), 501-509.

Repoussis, P. P., Paraskevopoulos, D. C., Zobolas, G., Tarantilis, C. D., & Ioannou, G. (2009). A web-based decision support system for waste lube oils collection and recycling. *European Journal of Operational Research, 195*(3), 676-700. doi:10.1016/j.ejor.2007.11.004

Rusdiansyah, A., & Tsao, D. B. (2005). An integrated model of the periodic delivery problems for vending-machine supply chains. *Journal of Food Engineering, 70*(3), 421-434. doi:10.1016/j.jfoodeng.2004.05.073

Russell, R., Chiang, W., & Zepeda, D. (2008). Integrating multi-product production and distribution in newspaper logistics. *Computers & Operations Research, 35*(5), 1576-1588. doi:DOI: 10.1016/j.cor.2006.09.002

Sahoo, S., Kim, S., Kim, B. I., Kraas, B., & Popov, A. (2005). Routing optimization for waste management. *Interfaces, 35*(1), 24-36. doi:10.1287/inte.1040.0109

Santos, L., Coutinho-Rodrigues, J., & Current, J. R. (2008). Implementing a multi-vehicle multi-route spatial decision support system for efficient trash collection in portugal. *Transportation Research Part A-Policy and Practice, 42*(6), 922-934. doi:10.1016/j.tra.2007.08.009

Sbihi, A., & Eglese, R. W. (2007). Combinatorial optimization and green logistics. *4or-a Quarterly Journal of Operations Research, 5*(2), 99-116. doi:10.1007/s10288-007-0047-3

Simonetto, E. D. O., & Borenstein, D. (2007). A decision support system for the operational planning of solid waste collection. *Waste Management, 27*(10), 1286-1297. doi:10.1016/j.wasman.2006.06.012

Solomon, M. M. (1987). Algorithms for the vehicle-routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254-265.

Song, S. H., Lee, K. S., & Kim, G. S. (2002). A practical approach to solving a newspaper logistics problem using a digital map. *Computers & Industrial Engineering, 43*(1-2), 315-330.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science, 31*(2), 170-186.

Tarantilis, C. D., & Kiranoudis, C. T. (2001). An efficient meta-heuristic algorithm for routing product collecting vehicles of dehydration plants. I. Algorithm development. *Drying Technology, 19*(6), 965-985.

Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2008). A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. *Informs Journal on Computing, 20*(1), 154-168. doi:10.1287/ijoc.1070.0230

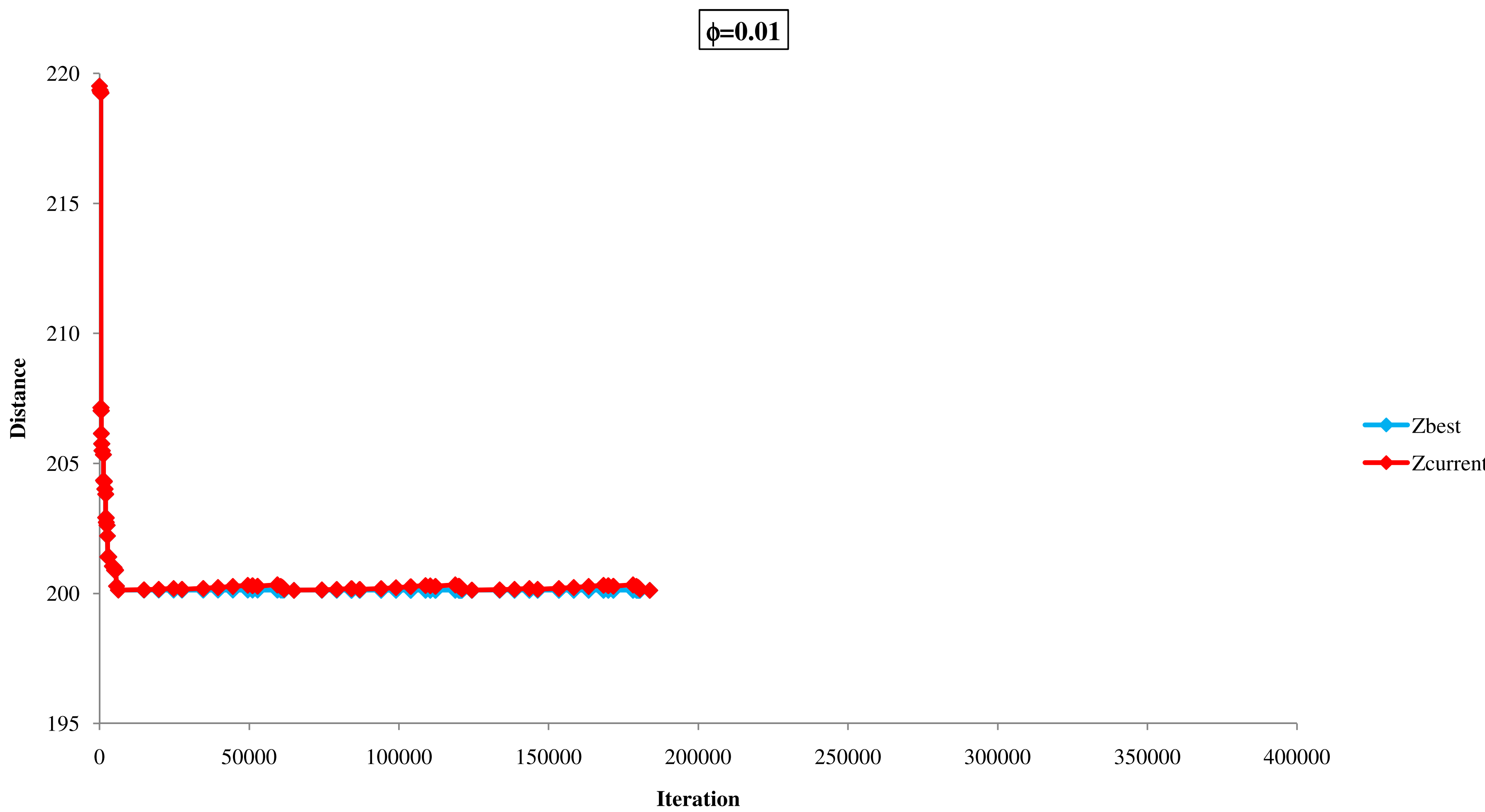
Tavares, G., Zsigraiova, Z., Semiao, V., & Carvalho, M. G. (2009). Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. *Waste Management, 29*(3), 1176-1185. doi:10.1016/j.wasman.2008.07.013

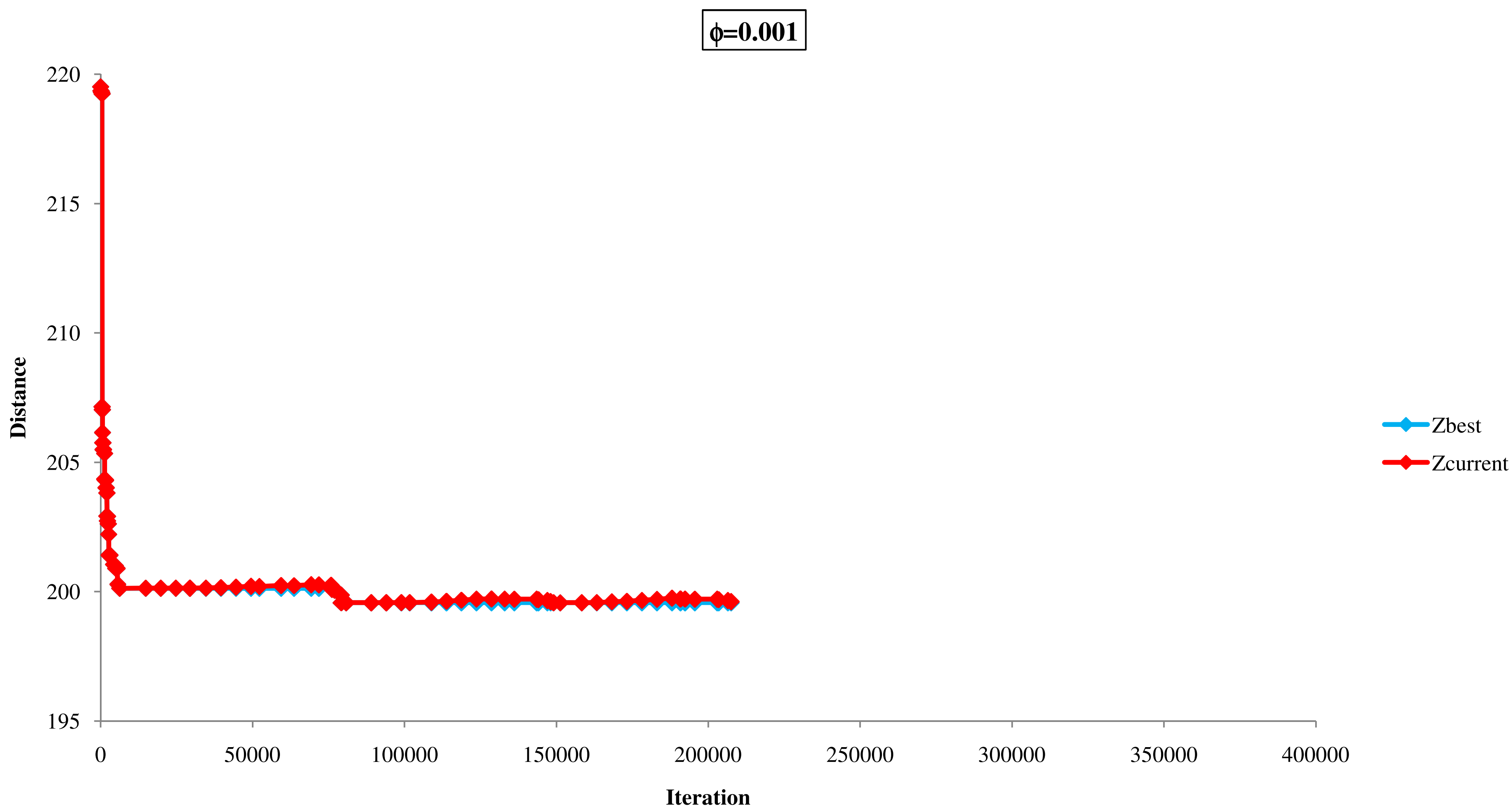
Tavares, G., Zsigraiova, Z., Semiao, V., & da Graça Carvaiho, M. (2008). A case study of fuel savings through optimisation of MSW transportation routes. *Management of Environmental Quality: An International Journal, 19*(4), 444-454. doi:10.1108/14777830810878632

Teixeira, J., Antunes, A. P., & de Sousa, J. P. (2004). Recyclable waste collection planning - a case study. *European Journal of Operational Research, 158*(3), 543-554. doi:10.1016/S0377-2217(03)00379-5

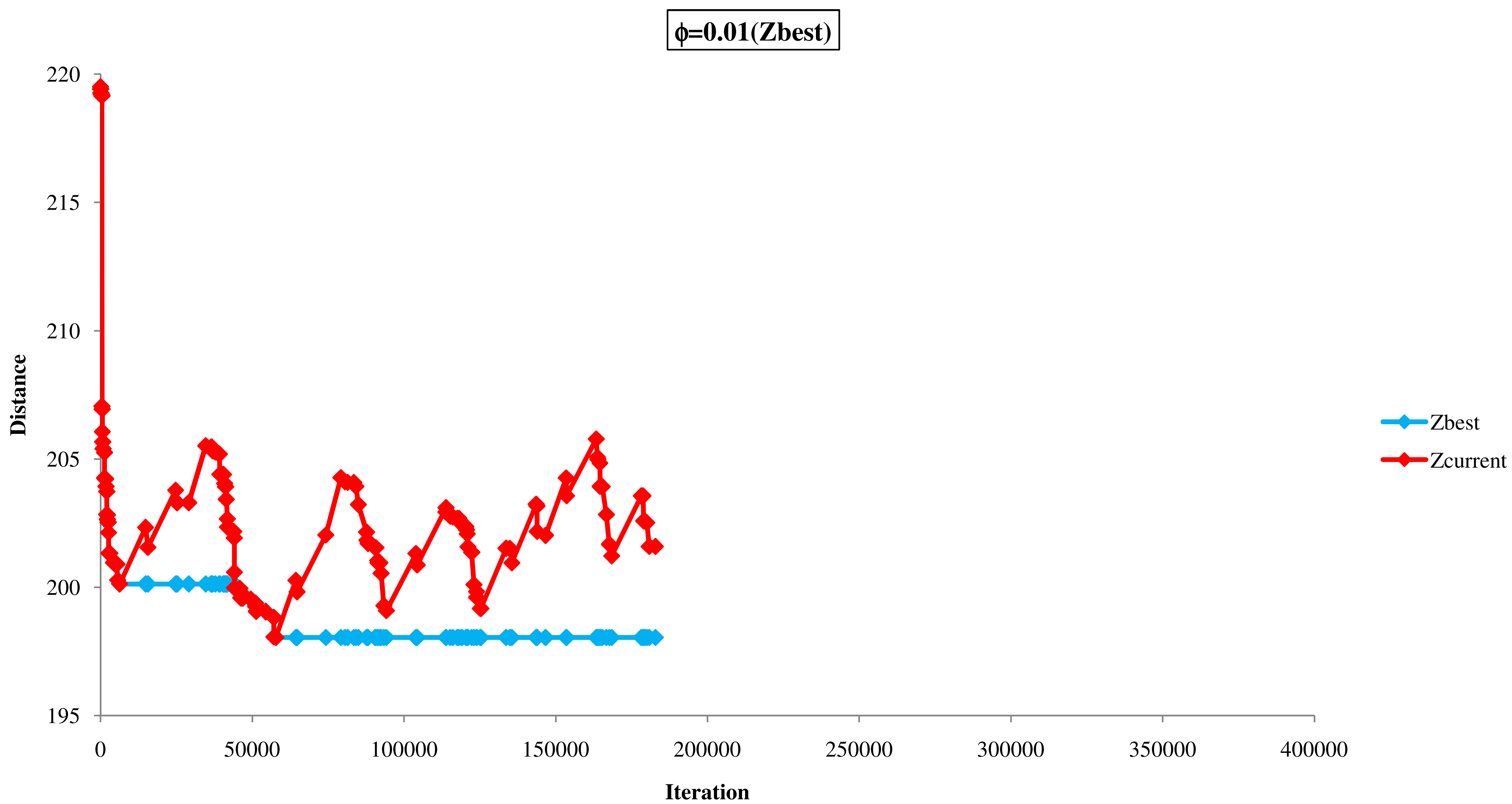
Tung, D. V., & Pinnoi, A. (2000). Vehicle routing-scheduling for waste collection in Hanoi. *European Journal of Operational Research, 125*(3), 449-468.

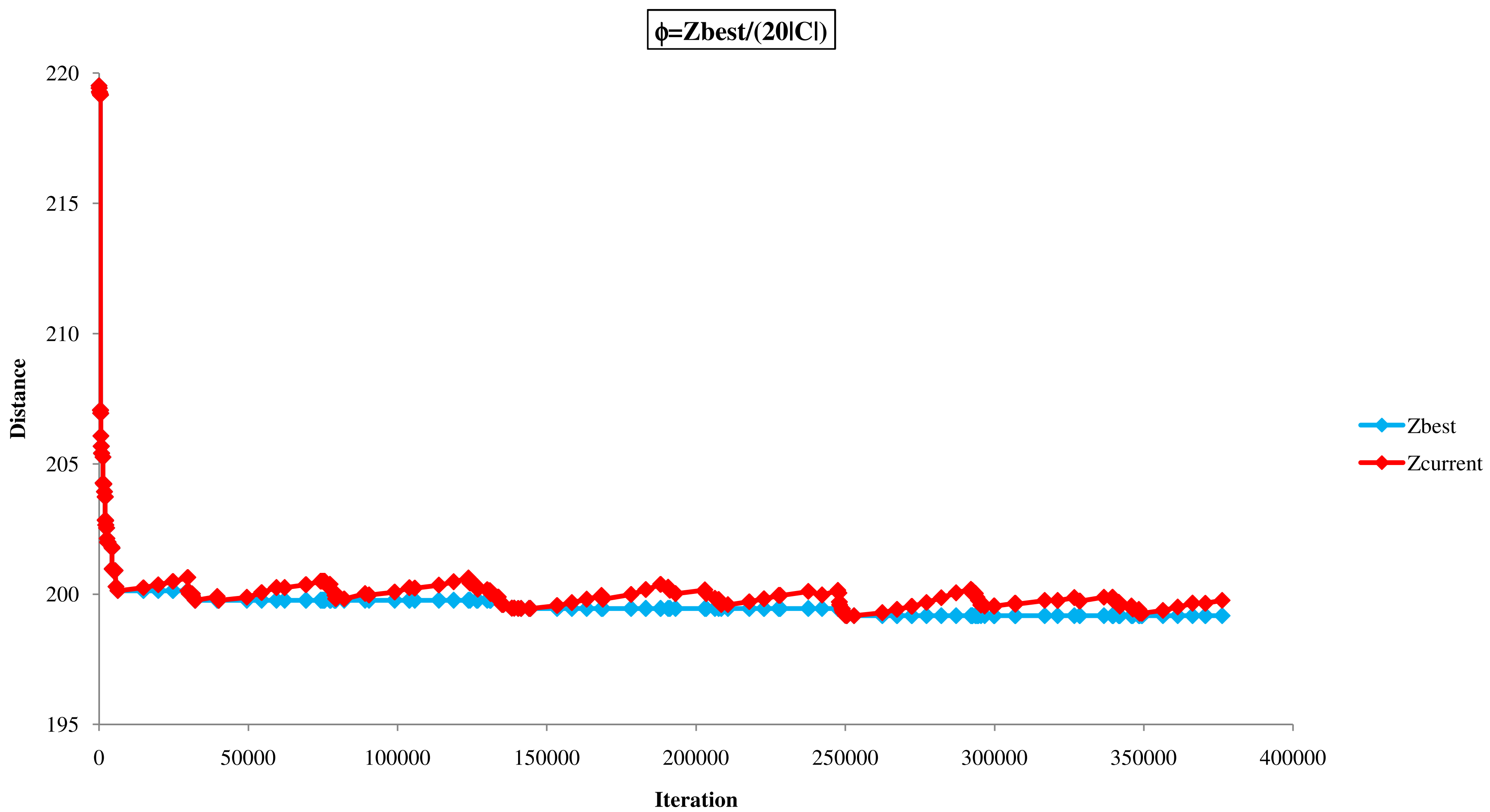
Zamorano, M., Molero, E., Grindlay, A., Rodriguez, M. L., Hurtado, A., & Calvo, F. J. (2009). A planning scenario for the application of geographical information systems in municipal waste collection: A case of Churriana de la Vega (Granada, Spain). *Resources Conservation and Recycling, 54*(2), 123-133. doi:10.1016/j.resconrec.2009.07.001


ϕ=0.01
Distance
220
215
210
205
200
195
0
50000
100000
150000
200000
250000
300000
350000
400000
Iteration
Zbest
Zcurrent

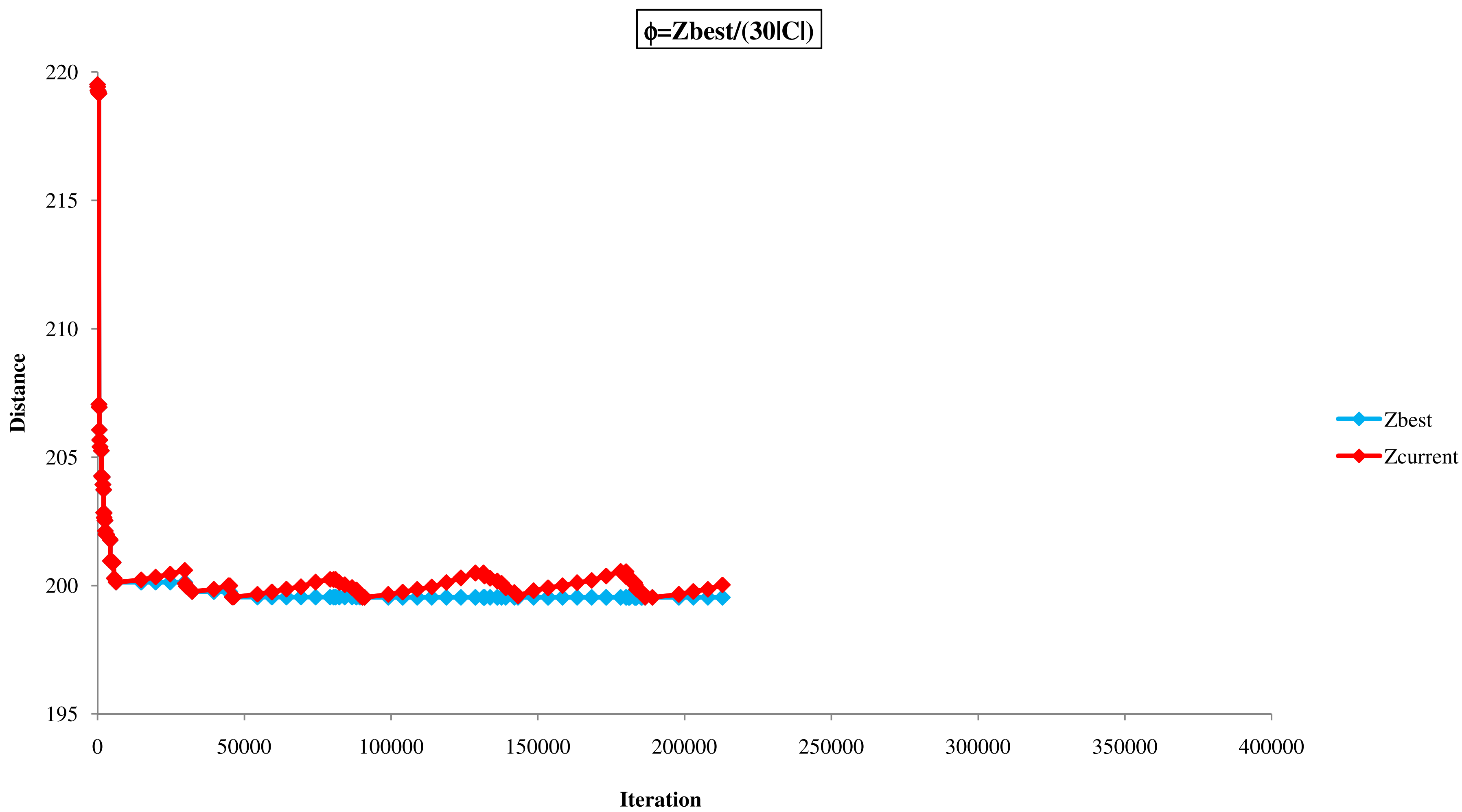


ϕ=0.001
220
215
210
205
200
195
Distance
0
50000
100000
150000
200000
250000
300000
350000
400000
Iteration
Zbest
Zcurrent


ϕ=0.01(Zbest)
Distance
220
215
210
205
200
195
0
50000
100000
150000
200000
250000
300000
350000
400000
Iteration
Zbest
Zcurrent



ϕ=Zbest/(20|C|)
220
215
210
205
200
195
Distance
0
50000
100000
150000
200000
250000
300000
350000
400000
Iteration
Zbest
Zcurrent



ϕ=Zbest/(30|C|)
220
215
210
205
200
195
Distance
0
50000
100000
150000
200000
250000
300000
350000
400000
Iteration
Zbest
Zcurrent