INTEGRATED FORWARD AND REVERSE LOGISTICS NETWORK DESIGN

Sally S. Kassem

A Thesis
In the Department
Of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

April 2011

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Sally Kassem**

Entitled: **Integrated Forward and Reverse Logistics Network Design**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Mechanical Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. A. Ben Hamza ______________________________ Chair

Dr. T. ElMekkawy______________________________ External Examiner

Dr. Y. Zeng ______________________________ External to Program

Dr. A. Akgunduz ______________________________ Examiner

Dr. A. Bulgak ______________________________ Examiner

Dr. M. Chen ______________________________ Thesis Supervisor

Approved by Dr. W-F. Xie____

Graduate Program Director

April 4, 2011 Dr. Robin A.L. Drew____

Dean of Faculty of Engineering & Computer Science

ABSTRACT

Integrated Forward and Reverse Logistics Network Design

Sally S. Kassem, Ph.D.
Concordia University, 2011

Many manufacturers are moving towards green manufacturing. One of the actions for environment friendly manufacturing is collection of end-of-life products (EOL). EOL products are transported to the proper facilities for reprocessing or proper disposal. Movement of collected products is performed through reverse logistics networks. Reverse logistics networks may be designed independent of forward logistics networks, or as integrated networks, known as integrated forward and reverse logistics (IFRL) networks. Recent research shows that IFRL networks are more efficient than independent networks. In this work, we study a number of IFRL networks. We present a comprehensive mathematical model to represent an assignment and location-routing IFRL network. Afterwards, this model is decomposed into a number of sub-models that represent different IFRL networks. For each network we develop a solution methodology to solve practical size problems.

Two sub-models based on the comprehensive model are presented to design two IFRL location-routing networks. The first network considers decision on the location to establish a disassembly plant. The second network considers decisions on the location to establish a manufacturing facility. For both networks, routing decisions are assigning customers to vehicles, and establishing vehicles' routes. We develop two heuristic methods to solve the models. The heuristics are able to reach optimal or near optimal solutions in reasonable computational times.

The vehicle routing problem with simultaneous pickup and delivery and time windows (VRPSPD-TW) is studied in this work. We use a sub-model of the comprehensive model to represent the problem. Classic heuristics and intelligent optimization or metaheuristics are widely used to solve similar problems. Therefore, we develop a heuristic method to solve the VRPSPD-TW. Results of the heuristic serve as initial solutions for a simulated annealing (SA) approach. For most tested problems, the SA approach is able to improve the heuristic solutions, and reach optimal solutions. Computational times are reasonable for the heuristic and SA.

We also study the multi-depot vehicle routing problem with simultaneous pickup and delivery and time windows (MDVRPSPS-TW). A sub-model of the comprehensive model represents the problem. The network considers assignment of customers and vehicles to depots, assignment of customers to vehicles and routing of vehicles within customers' time windows. We develop a 2-phase heuristic and a SA approach to solve the problem. Heuristic solutions serve as initial solutions for the SA approach. SA is able to reach optimum or near optimum solutions. Computational times are reasonable for the heuristic and SA.

ACKNOWLEDGEMENTS

I am very pleased to express my sincere thanks and gratitude to Professor Mingyuan Chen for his continuous guidance, advice, and valuable suggestions which enabled me to carry out this research. I am very thankful for his understanding and support throughout the course of the research. I would like to thank him for his positive and quick responses that eased all the hardships that faced me from the beginning till the end of this work.

Special thanks are due to my dear parents whose help, support, and encouragement were very significant throughout the years it took me to complete this work. I am very grateful for the time and effort they spent with me here in Montreal, and for their endless support that was always available for me and for my family. It would not have been possible for me to reach where I am now without them.

Special thanks to my dear husband who supported me with all his power to pursue my research. I sincerely appreciate his understanding, patience, and the time he used to spend helping me with my studies and research. I will always remember his encouraging spirit at hard times. I am very grateful for his personal and academic help, and for carrying extra family responsibilities on himself to leave me the time to work.

I would like also to thank all my other family members, my friends and my colleagues who supported me and were always available in times of need.

I am thankful for the financial support provided by the Egyptian Government, NSERC of Canada through Dr. Chen’s Discovery Grant and the GSSP program in the Faculty of Engineering and Computer Science, for my Ph.D. study and research at Concordia University.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

VRP	-	Vehicle routing problem
EOL	-	End of life
IFRL	-	Integrated forward and reverse logistics
LRP	-	Location routing problem
MDVRP	-	Multi-depot vehicle routing problem
MDVRPSPD-TW	-	Multi-depot vehicle routing problem with simultaneous pickup and delivery and time windows
MILP	-	Mixed integer linear programming
SA	-	Simulated annealing
VRPB	-	Vehicle routing problem with backhauls
VRPM	-	Vehicle routing problem with mixed loads
VRPSPD	-	Vehicle routing problem with simultaneous pickup and delivery
VRPTW	-	Vehicle routing problem with time windows
VRPSPD-TW	-	Vehicle routing problem with simultaneous pickup and delivery and time windows

CHAPTER 1

INTRODUCTION

1.1 Introduction

In recent years, environmentally friendly manufacturing has gained much attention amongst researchers and practitioners. Manufacturers are required to take a number of actions towards green manufacturing. One of these actions is product recovery. The term product recovery refers to all the required procedures in the reuse of the product after being used by the consumer. Examples of product recovery activities include collection of used products, inspection and reuse of the collected products and/or product parts, proper disposal of unusable parts, and redistribution of the remanufactured products. Product recovery has resulted in many manufacturers changing their traditional roles as suppliers of manufactured products to consumers. Many of them have played another important role in taking back from customers their products at their end of life (EOL). The EOL products may be refurbished, remanufactured, recycled or trashed depending on many conditions such as their quality, materials, values, etc. This new dimension of manufacturing activities caused manufacturers to re-design their logistics systems in a two way manner. They typically include both forward logistics and reverse logistics networks, either independent or integrated. To ensure an efficient product distribution and recovery system, considerable research is required for the design of integrated forward and reverse logistics (IFRL) networks.

1.2 Drivers for Product Recovery

The importance of the typical forward logistics activities is widely recognized. On the other hand, product recovery activities, including reverse logistics networks, have been comparatively recent. Therefore, we would like to shed some light on the drivers for product recovery.

Product recovery is motivated by the following three main drivers:

1- Legislation: As the consumption of the natural resources and the disposal of solid waste are increasing, many governments impose legislations on manufacturers to collect their products for recovery. Examples of these legislations are available in Ginter and Starling (1978) and Fleischmann *et. al.* (1997).

2- Economic Issues: Many manufacturers recognize the economical advantages of product recovery. Gungor and Gupta (1999) mention that automobile recycling in the USA, and precious material recovery from electronic products, are examples of the industries that benefit economically from product recovery.

3- Customer Expectation: According to the work of Vandermerwe and Oliff (1990), firms strongly acknowledge the importance of establishing a green image to attract and keep customers. Product recovery and proper waste disposal have been among the major issues that reflect a green image of the manufacturer.

1.3 Issues to Consider for Product Recovery

Due to the recognized importance of product recovery, a large number of manufacturers are willing to consider product recovery issues within their manufacturing plans. Fleischmann *et. al.* (1997) and Gungor and Gupta (1999) address the common issues that manufacturers need to consider for product recovery. These issues are:

1- Reverse distribution planning: Reverse distribution planning is mainly concerned with the movement of used products at their EOL. Movement of EOL products involves collection of used products from customers, returning them to the manufacturer, and then moving the product from the manufacturer to either redistribution or disposal. Reverse distribution planning is best represented by reverse logistics networks or IFRL networks.

2- Disassembly: After the EOL products have been collected from customers and transferred to the manufacturer, these products become candidates for either reuse and/or proper disposal. The collected products are disassembled, and undergo proper examination and cleaning. Disassembled parts may be reused or used as spare parts. Parts that are not reusable are properly disposed of. Disassembly has been the subject of a lot of research recently, for example, the work of Gupta and Taleb (1994) and the work of Jovane *et. al.* (1993) address many details of the issue.

3- Production planning: According to Fleischmann *et. al.* (1997) production planning of returned products could be classified into the following categories:

   a. Direct reuse: where the returned product can be reused directly ‘as is’ after necessary cleaning and/or minor repair.

   b. Material recycling: which involves several steps for material recovery. Examples of this category are transportation packages, pallets, boxes or bottles.

c. Refurbishing, remanufacturing, and/or cannibalization: where extensive required testing, disassembly, and repair are required to bring the returned product to ‘as new’ condition.

4- Inventory Control: One of the challenges that face manufacturers when considering product recovery is inventory control. The complexity of inventory control within the product recovery environment arises from the uncertainties involved. The manufacturer encounters uncertainties in the quantity, quality, and timing of incoming products and parts. The work of DeCroix and Zipkin (2005) and Kiesmuk and Van Der Laan (2001) are examples of research on inventory management within the product recovery environment.

From our review of the published work on product recovery, we noticed that reverse distribution planning is the least researched issue among the four issues of product recovery. We also noticed that the importance of reverse distribution planning has been recognized recently. In addition, many researchers emphasized on the advantages of IFRL networks, and the body of research related to designing the IFRL networks is growing.

1.4 Integrated Forward and Reverse Logistics Networks

1.4.1 Definitions

Most manufacturing companies need to construct and operate a forward supply chain in providing new products to customers. When these companies consider reverse distribution planning, they may also need to construct and operate a reverse supply chain to collect, reprocess, and redistribute EOL products. Reverse supply chains represent reverse distribution planning. Published research work refers to reverse supply chains either by “reverse logistics networks” or “product recovery networks”. Reverse logistics networks may be designed as separate networks, or may be integrated with a forward logistics network, known as integrated forward and reverse logistics networks (IFRL networks). Akcali *et. al.* (2009) define product recovery networks when designed independently as “*networks consist of the reverse channel only, i.e., the reverse activities and reverse flows. The set of reverse activities includes collection, inspection, sorting, disassembly, reprocessing/recycling, and disposal operations, and the set of reverse flows pertains to the flows among these reverse activities*” (Akcali *et. al.* (2009), p. 232). IFRL networks, also known as closed loop supply chain networks, are defined by Akcali *et. al.* (2009) as networks that “*include the forward channel, i.e., forward activities and forward flows, along with the reverse channel. The set of forward activities includes manufacturing and distribution operations, and the set of flows includes the forward flows among the forward activities as well as the flows among the forward and reverse activities*” (Akcali *et. al.* (2009), p. 232). The advantages of IFRL networks over separate forward and reverse logistics networks have been recognized by many researchers. This resulted in a growing body of research for the design of IFRL networks. In this research we consider the design of IFRL networks.

1.4.2 Difference between Forward and Reverse Logistics Networks

Reverse logistics networks differ from forward logistics networks in many characteristics. According to Fleishmann (2001) the major differences between forward and reverse logistics networks are:

1- In forward logistics networks supply is typically an endogenous variable, while in reverse logistics networks supply is exogenous.

2- Forward logistics networks usually do not include inspection point(s) as that of the reverse logistics networks, therefore reverse logistics networks are usually more complex than forward networks.

3- The interaction between the collection and distribution/redistribution of used products is another characteristic of reverse networks.

4- The number of sources of used products tends to be much larger in reverse logistics networks than those for forward networks (see Figure 1.1).

5- Demand uncertainty is a characteristic to reverse logistics networks.

Figure 1.1 shows the difference between the structures of forward and reverse logistics networks.

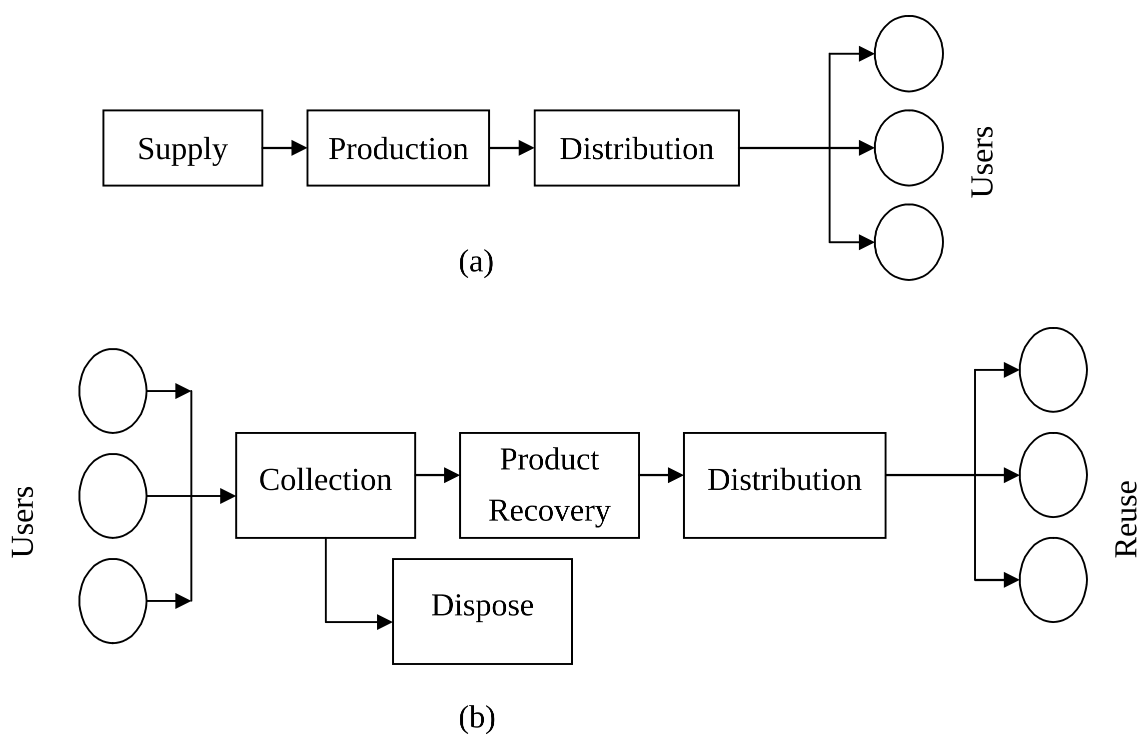

Figure 1.1: (a) Forward Logistics Networks, (b) Reverse Logistics Networks

1.4.3 Design of IFRL networks

Most of the work that has been done to design IFRL networks uses mixed integer linear programming models (MILP) or mixed integer non-linear programming models. In either case, most IFRL networks are designed to solve plant location problems or vehicle routing problems.

As mentioned in Section 1.2, product recovery involves new issues to consider, for example, disassembly and inventory control. These new issues require many manufacturers to consider building new facilities to accommodate the new manufacturing requirements. The same applies to the IFRL networks. Hence, plant location problems for reverse logistics are widely researched, and consequently for the integrated networks. On the other hand, the first step in reverse distribution planning is collection of EOL products. Collection is the opposite of distribution. In distribution, manufactured products start from a single source, that is, the manufacturing facility or warehouse. These products are then brought to multiple destinations, which are the end users. The distribution process is done through forward logistics networks. The collection process starts at multiple sources, namely, the end users, and end at a single source, which is the remanufacturing facility. The collection process is carried out through reverse logistics networks (see Fleischmann *et. al.* (1997)). Collection of EOL products has gained a lot of attention recently. Collection, and combined collection and distribution, are represented within reverse logistics networks using extensions of the vehicle routing problem models.

Many researchers recognize the various uncertainties within reverse logistics. Examples of these uncertainties include the quantity and quality of collected EOL products. On the other hand, many researchers nowadays believe that such uncertainties are due to the issue being recently recognized. For example Fleischmann (2001) recognizes that the issue of uncertainty within reverse logistics is expected to be resolved after the manufacturers start implementing it. He expects that forecasting the systems parameters will be easy after sometime of application based on historical data. This is expected to help reverse logistics planners to solve easier deterministic problems rather than stochastic ones. Since many researchers consider deterministic problems for the forward logistics networks, it is possible to consider deterministic integrated forward and reverse logistics problems.

1.5 Research Objectives

Based on our review on design of IFRL networks, we noticed that the body of research considering plant location models is extensive. More recently, research on the combined distribution and collection issues has been growing. Combined distribution and collection of EOL products has been researched by extending the typical vehicle routing problem (VRP) model. Such extensions include VRP with mixed loads, and the vehicle routing problem with simultaneous pickup and delivery (VRPSPD). It has been noticed that solving the combined distribution and collection problem through VRPSPD models is very efficient. One of the very important and realistic extensions to the typical VRP is the vehicle routing problem with time windows (VRPTW). Research that combines VRPSPD and VRPTW has been very limited and recent.

Combining location and routing problems in the design of logistics networks has been recognized for its importance. Nevertheless, combined location and routing has not been considered yet within the reverse logistics context, neither within the IFRL context.

The Multi-depot vehicle routing problem (MDVRP) with time windows is another important class of problems within logistics systems. This class of problems has gained little attention within the IFRL framework.

The VRP and its extensions, the combined location-routing problems, and the MDVRP and its extensions have been well established to be NP-hard. Therefore, solving real size problems requires heuristic algorithms to reach efficient solutions during reasonable computational times.

The objectives of this research are to design IFRL networks that consider location and routing issues, and the MDVRP. In addition, routing issues are considered with the realistic extension of time windows constraints. In order to solve real life problems, heuristics and metaheuristics are developed.

Therefore, the objectives of this research may be summarized as follows:

1- Design IFRL networks that combine plant location and vehicle routing problems.
2- Design IFRL networks that consider solving the combined problem of VRPSPD and VRPTW.
3- Consider the MDVRP in addition to the extension of time windows constraints within the IFRL frame.
4- Develop efficient solution techniques to solve real size problems represented by the above mentioned networks.

1.6 Research Approach

To achieve the research objectives mentioned above, the research approach consists of the following steps:

1- Explore the different approaches to design IFRL networks through reviewing published work about the issue.
2- Develop a generic mathematical programming model to represent:
    a. Combined location routing issues within IFRL context.
    b. Combined VRPSPD and VRPTW.
    c. MDVRP with time windows within IFRL context.
3- Generate test instances to solve the mathematical models mentioned in step 2.
4- Test the generated instances using a commercial optimization solver, when possible.
5- Develop a solution technique to solve the mathematical models in step 2.
6- Develop a computer code of the solution technique.
7- Validate the efficiency of the developed solution technique.
8- Draw conclusions and suggest directions for future work.

1.7 Research Contribution and Publications/Submitted Papers

In this thesis we develop a comprehensive mathematical model that represents a combined location-routing IFRL network. The model also represents the multi-depot vehicle routing problem within the IFRL context. In addition, the time windows variant of the vehicle routing problem is considered within the model. To practically represent and solve IFRL network problems, sub-models of the comprehensive model are used throughout the thesis.

We introduce for the first time IFRL networks that combine plant location and vehicle routing problems. We propose two mathematical models to represent two location-routing IFRL networks. For each network we develop a heuristic method to solve practical size problems that have various real life applications. The developed heuristics are tested on a number of instances and are able to reach optimum and near optimum solutions in reasonable times. The two IFRL networks and the associated heuristics are compiled in the following two publications:

1. Kassem S. S. and Chen M., “Reverse Logistics Network Design”, 2008, Proceedings of the 9th Cairo University International Conference on Mechanical Design and Production (MDP-9), Cairo, Egypt, January 8-10 2008.
2. Kassem S. S. and Chen M., “Reverse Logistics Network Design with Location and Vehicle Routing”, to be submitted to the International Journal of Industrial and Systems Engineering.

We consider the vehicle routing problem with simultaneous pickup and delivery and time windows. Literature on the problem is very recent and limited. We present the problem with a set of assumptions and constraints that differ from those available in the literature. We propose a mathematical model to represent the problem and a heuristic method to solve it. The heuristic method reaches optimum or near optimum solutions within reasonable computational times. The mathematical model and the developed heuristic results are summarized in the following publication:

3. Kassem S. S. and Chen M., “Heuristics for Solving Reverse Logistics Vehicle Routing Problems with Time Windows”, under review at the International Journal of Industrial and Systems Engineering.

We use simulated annealing, for the first time, to solve the vehicle routing problem with simultaneous pickup and delivery and time windows. The simulated annealing procedure uses the results of the developed heuristic mentioned above as initial solutions. Results of the simulated annealing are similar to, or close to optimum and computational times are reasonable. Simulated annealing results are compiled in the following publication:

4. Kassem S. S. and Chen M., "Reverse Logistics Vehicle Routing Problems with Time Windows", under review at the International Journal of Advanced Manufacturing Technology.

We present for the first time the multi-depot vehicle routing problem with simultaneous pickup and delivery and time windows. A mathematical model is developed to represent the problem. We develop a 2-phase heuristic method and a simulated annealing approach to solve the problem. The proposed solution method is able to reach optimum or near optimum solutions in reasonable computation times. The presented problem and the mathematical model, together with the solution method and results are compiled in the following publication:

5. Kassem S. S. and Chen M., "Multi-Depot Vehicle Routing Problem with simultaneous pickup and Delivery and Time Windows", to be submitted to the 1st International Conference on Logistics, Informatics, and Service Science, LISS 2011, Beijing Jiaotong University, China, June 8-11, 2011 .

1.8 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 provides a review of the literature related to combined location routing problems, integrated forward and reverse logistics networks, and variants of multi-depot vehicle routing problems. In Chapter 3, we provide a comprehensive mathematical model that represents location, routing, and assignment issues within the integrated forward and reverse logistics frame. The mathematical model presented in Chapter 3 is used thereafter within the thesis to design different IFRL networks. Chapter 4 provides mathematical models and solution methodologies for two combined location routing problems. In Chapter 5, the vehicle routing problem with simultaneous pickup and delivery and time windows is studied. A mathematical model of the problem is presented. The problem is solved using a classical heuristic method and a simulated annealing approach. In Chapter 6, we present a multi-depot vehicle routing problem with simultaneous pickup and delivery and time windows. The problem is represented by a mathematical model. A 2-pahse heuristic method and a simulated annealing approach are used to solve the problem. Summary, Conclusion, and future research are given in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Research on integrated forward and reverse logistics networks (IFRL) has grown recently. There are many published articles that solve location problems within reverse logistics networks. On the other hand, research on solving the vehicle routing problem and its extensions for IFRL is rather limited. This chapter presents a review on the research related to plant location and vehicle routing problems within the context of IFRL. Recent and early publications that provide mathematical models and efficient solution methodologies are considered. These publications provide a general guideline for designing the IFRL networks considered in our work. We also present research on combined location routing problems and variants of multi-depot vehicle routing problems, since they are studied in this research.

2.2 Combined Location Routing Problems

The combined location routing problem (LRP) is not a well defined problem, as mentioned in Nagi and Salhi (2007). Therefore, we start by giving a brief explanation of the problem. The combined location routing problem can be defined as follows: Given a set of potential facility locations, and a set of customer locations where each customer has a determined amount of demand, it is required to make decisions on: the location(s) to open one or more facilities, the number of vehicles needed at each facility and the route of each vehicle to customer locations such that the demand of all customers is satisfied. The objective function of the problem is usually minimizing a combination of location costs and routing costs. The combined location routing problem dates back to the early 60's. The work of Von Boventer (1961) is one of the first articles to acknowledge the relationship between location and transportation problems. In the work of Maranzana (1964), it has been recognized that "*the location of factories, warehouses and supply points in general to serve customers distributed over a network of cities, is often influenced by transport costs.*", (Maranzana (1964), p.261). The work of Salhi and Rand (1989) shows that ignoring tours when locating facilities in a location-allocation problem, may lead to increased distribution cost. In addition, Simultaneous location and routing decisions have been addressed as one of the important issues in logistics system modeling and analysis, as pointed out in Min *et. al.* (1998).

Location decisions are strategic, while routing decisions are tactical. To investigate the validation of combining the strategic location problem with the tactical routing problem, Salhi and Nagi (1999b) studies this issue. . In their work, it is found that the location-routing methods they propose are not affected by small changes in customer demands. Their work also shows that the approaches considering the combined location-routing models can produce better solutions than the approaches considering them separately. Based on these findings, a considerable amount of research has been conducted on the combined location routing problem, despite the difference in planning horizon.

The LRP defined earlier is solved under different sets of constraints and assumptions. For example, the work of Srivastava (1993) solves LRP with location cost of opening a facility, and routing costs of vehicles' routes to serve customers. The problem considers vehicles' capacity constraints. Three heuristics are proposed to solve the problem. The three heuristics are proved efficient and their performance is comparable with each other. The same problem is solved in the work of Tuzun and Burke (1999) using a two-phase Tabu search algorithm. The authors test the Tabu search solution approach on a number of problems. They compare their results with the results of the solution heuristics developed in Srivastava (1993). The tabu search algorithm is shown to obtain significantly better results. The same problem is considered in Barreto *et. al.* (2007), with the added constraints of depots' capacity. The authors present clustering techniques to solve the problem.

The components of the cost minimization objective function vary among different problems. In the work of Chien (1993) LRP is presented, where location facilities are uncapacitated, and vehicles are capacitated. The incurred costs are fixed costs associated with opening a facility, and a per unit throughput for each open facility. The routing costs are a per vehicle dispatched cost, and per unit distance travel costs. The author solves the problem using a number of heuristics, and recommends three of them for their efficiency. A slightly different cost function is given in Melechovshy *et. al.* (2005). A non linear cost component per open depot is introduced. This non linear cost increases with the depot demand. The problem has capacity constraints for the depots and the vehicles. A tabu search algorithm is developed to solve the problem. The work of Lin *et. al.* (2002) solves the location routing loading problem for a bill delivery services company in Hong Kong. The problem solved in this work requires decisions on the locations to open new facilities among candidate locations, the number of vehicles to employ and the vehicles' routes.

Loading times are considered, and the maximum working hours must not be exceeded. The problem is solved using a metaheuristic approach based on threshold accepting and simulated annealing. Nagi and Salhi (2007) provide an extensive survey on recent research conducted in this area. Their work points out the NP-hard nature of the LRPs. Therefore, to solve practical size problems that have real life applications, heuristics techniques are developed.

Based on the literature review that we have performed, it is noticed that the LRP has been solved for the forward logistics network only. It is also shown that the problem has many practical applications, and has been the subject of a considerable amount of research. To the best of our knowledge, solving LRPs within reverse logistics or IFRL has not been studied previously.

2.3 Integrated Forward and Reverse Logistics Networks

Throughout this thesis we study different variants of location and routing issues for the IFRL networks. In addition, we also consider MDVRP with time windows within IFRL frame. Therefore, in the following subsections we will focus on the work done in location and vehicle routing within the IFRL context. We also consider the time windows extension of the problem.

2.3.1 Plant Location

Plant location models were among the first models developed to deign IFRL networks and reverse logistics networks. Most of the mathematical formulations developed to express the location problem were mixed integer programming formulations. For example, Marin and Pelegrin (1998) proposed a mixed integer linear programming

(MILP) location model for returned products. The authors considered deterministic demands to decide optimal forward and backward flows simultaneously. The model was solved by a heuristic algorithm and also by exact solution methods. Krikke *et. al.* (1999) presented an un-capacitated plant location model for multi-echelon reverse logistic network design for durable products. The model was solved using LINDO. Jayaraman *et. al.* (1999) also developed a 0-1 MILP model for closed loop logistics network. The model solution provided optimal decisions for remanufacturing/distribution facility locations, production, material and product transportation, and inventory level for remanufactured products. The model was solved by an exact method. Spengler *et. al.* (1997) developed a model for multi-level warehouse locations. The model used a piecewise linear cost function and decided the locations of recycling facilities with two practical applications. They solved the problem by GAMS/OSL software. Fleischmann (2001) introduced a MILP recovery network design model. It was a generic model and simultaneously decides the forward and backward flows in the network. It was based on the classical multilevel warehouse location problem and might have closed loop or open loop flows. This generic model was extended in Salema *et. al.* (2006). The modified model considered additional issues such as capacity limits, multiproduct management, and demand/returns uncertainties. Another capacitated location problem was presented by Lu and Bostel (2007). The authors developed a 0-1 MILP model. Among a number of potential locations, the model considered decisions on the locations to establish production facility(ies), remanufacturing center(s), and intermediate center(s). The amount of material flow within the network participants is treated as a decision variable of the model. The authors developed an algorithm that is based on Lagrangian relaxation to solve the problem. Demirel and Gökçen (2008) studied the problem of IFRL network through location and remanufacturing decisions. They represent the problem through a 0-1 mixed integer programming model. The model provides values of production quantities of manufactured and remanufactured products. In addition the model solves the location problem of disassembly, collection, and distribution centers. Material flow within network members is also considered among the decision variables. The proposed model is solved to optimality using commercial software. Easwaran and Uster (2010) solved the capacitated multi-product plant location problem for IFRL networks using exact solution approaches based on Bender's cuts. To the best of our knowledge, using metaheuristic approaches to solve different plant location problems for IFRL is recent. Wang and Hsu (2010) used genetic algorithms that are based on spanning tree methods to solve the capacitated location problem. Pishvaee *et. al.* (2010) developed an algorithm that is based on a combination of local search and genetic algorithms to solve a capacitated multi-stage IFRL location problem. Ding (2010) provided a location model that considers locating new facilities and expanding existing ones. The author developed a differential evolution algorithm to solve the problem.

2.3.2 Vehicle Routing

The vehicle routing problem (VRP) considering collection issues is a variant of IFRL networks. Collecting products from customers may be done in different ways. For example, pickups can be performed after all deliveries have been performed, which is known as the vehicle routing problem with backhauls (VRPB). The other version is VRP with mixed loads (VRPM), where, pickups and deliveries are allowed in any sequence, under the condition that customers are divided into pure delivery customers and pure pickup customers. The vehicle routing problem with simultaneous pickup and delivery (VRPSPD) is another approach for vehicle routing and reverse logistics. In VRPSPD vehicles perform both delivery and pickup from each customer simultaneously at each stop. VRPM may be considered as a special case of the VRPSPD where customers have either demand or pickup equals zero. Nagy and Salhi (2005) provide more details on different VRPs with collection strategies.

In many practical applications, vehicle routing problems are solved requiring that the vehicles must visit each customer within a specified time interval, the "time window". Such problems are called vehicle routing problem with time windows (VRPTW). The vehicles start at the depot and visit the customers within their associated time windows. A vehicle arriving earlier than the required time can wait until the beginning of the time window but may not be allowed to arrive after the end of the time window. This problem is referred to as hard time windows problems. In some problems arriving outside the time window interval is allowed. In that case a penalty cost is added to the total routing cost. This case is called soft time windows.

We present some of the work that has been done on VRPSPD and VRPTW.

Literature on VRPSPD

VRPSPD has recently been recognized for its importance, especially in reverse logistics applications. Some applications for VRPSPD are in soft drink industries where delivery of full bottles and collection of empty ones are typical. It also found application in printer manufacturing where full ink toners and cartridges are delivered and empty ones are collected. Another typical application is in photocopier manufacturing industry where manufacturers are required to take back or properly dispose of EOL products. The early work introducing VRPSPD can be found in Min (1989). The work in Min (1989) was concerned with a library situation where delivery and pickup of books were required. The reported problem consisted of 22 customers and a heuristic approach was developed to solve the problem. Salhi and Nagi (1999a) considered a similar problem and developed an insertion-based heuristic to solve the problem for different cases with single and multiple depots. Dethloff (2001) proposed a mathematical model to formulate the VRPSPD and a heuristic algorithm (RESCAP-RS) to solve the problem. Dethloff (2002) compared the RESCAP-RS algorithm with other algorithms originally developed to solve VRPB or VRPM. The comparison showed that RESCAP-RS can produce superior results. Gajpal and Abad (2010) solved the same problem using savings algorithms. The savings algorithms yielded high quality solutions. Alshamrani *et. al*. (2004) developed a mathematical model for VRPSPD requiring certain amount of material to be picked up at each stop. The model assumes penalty costs for materials which cannot be picked up due to vehicle capacity limit. Alshamrani *et. al.* (2007) extended this model to include decision variables for both forward and backward flows. The problems discussed in Alshamrani *et. al.* (2004) and Alshamrani *et. al*. (2007) considered a single vehicle and probabilistic data. Subramanian *et. al.* (2010a) presented a parallel algorithm to solve the VRPSPD. The algorithm used the technology of multiple core possessors to solve large size problems. The parallel algorithm is embedded with a multi-start heuristic .Several researchers used meta-heuristics to solve VRPSPD. For example, Crispim and Brandao (2005), Montane and Galvao (2006) and Wassan *et. al.* (2008) used Tabu search to solve the problem. Zachariadis *et. al.* (2010) used a tabu search technique that is embedded within an adaptive memory framework. Zachariadis and Kiranoudis (2011) were able to find new better solutions for some of the large size benchmark problems. The authors used a combination of tabu search and a local search approach, where the proposed local search approach can efficiently examine rich neighbor solutions. Chen and Wu (2006) proposed a heuristic method based on record-to-record travel, Tabu lists, and route improvement procedures. Cao and Lai (2007) used an improved differential evolution algorithm. In addition, the ant system and ant colony methods were used by Haijun *et. al.* (2007), and Chen *et. al*. (2007), respectively, to solve VRPSPD problems. Catay (2010) used an ant colony optimization technique, incorporating a saving approach to solve the problem.

Research on lower bounds and exact algorithms for the VRPSPD is very limited. Dell'Amico *et. al.* (2006) presented for the first time exact algorithms to solve the VRPSPD. They introduced new ideas to improve dynamic programming algorithms used to solve the pricing problem. The authors examined dynamic programming and state space relaxation. The authors concluded that the branch and price algorithm is more suitable for small to medium sized VRPSPD. Optimal solutions were found for problems with up to 40 customers. Subramanian *et. al.* (2010b) presented MILP formulations for the VRPSPD with directed and undirected two-commodity flow. The authors tested their formulations and another formulation from the literature using branch and cut algorithms. The undirected formulation obtained better results.

Literature on VRPTW

Comparing to literature on VRPSPD, literature on VRPTW is abundant, due to its early and well recognized importance. Extensive literature reviews on VRPTW problems and solution methodologies can be found in Bräysy and Gendreau (2005a) and Kallehauge

(2008). Since VRPTW is NP-hard in nature, methods based on various meta-heuristics have been widely used in solving the problem. Bräysy and Gendreau (2005b) provided a literature review on using meta-heuristics for solving VRPTW problems before 2005. More recent development includes, for example, Lin *et. al.* (2006) and de Oliveira *et. al.* (2007) where simulated annealing was used. Ming-Yao *et. al.* (2008) used tabu search, Wenfeng *et. al.* (2008) used genetic algorithms, and Jiang *et. al.* (2009) used a hybrid particle swarm and evolution algorithm for solving this type of problems.

Literature on Vehicle Routing Problem with Simultaneous Pickup and Delivery and Time Windows (VRPSPD-TW)

The VRPSPD-TW is a combination of the standard versions of VRPSPD and VRPTW. The problem considers simultaneous pickup and delivery at each customer such that a customer is visited only once within the specified time window and without violating the vehicle capacity constraints. One of the earlier works discussing this type of problem can be found in Angelelli and Mansini (2002). They presented two mathematical models to formulate the problem. The first one is a generalized mixed integer programming model and the second one is a set covering formulation. The authors solved several testing problems with up to 20 customers using column generation based on the set covering formulation. Cao and Lai (2007) discussed VRPSPD-TW problems and formulated a mathematical programming model of the problem. They developed a genetic algorithm method to solve the problem. Yun and Guorui (2008) presented VRPSDP-TW with soft time windows, heterogeneous fleet of vehicles, and drivers' costs. The authors solve the problem by combining a heuristic algorithm and genetic algorithms. Chun-Hua *et. al.* (2009) further investigated the VRPSPD-TW. They considered the VRPSPD-TW with penalty values for tardiness. In the more recent work of Mingyong and Erbao (2010) the VRPSPD-TW was solved by genetic algorithms considering maximum route length constraints and hard time windows.

2.4 Variants of the Multi-depot Vehicle Routing Problem

The multi-depot vehicle routing problem considers a number of established depots in different locations. A set of customers at different locations require demand from the depots. It is required to assign each customer to a depot, and simultaneously route vehicles from each depot to serve the corresponding customers. There are different variants of the MDVRPs in the literature. For example, Salhi and Nagi (1999a) presented the MDVRP with simultaneous pickup and delivery and MDVRP with mixed loads. To solve the problem, the authors proposed a number of insertion based heuristics that incorporate border line customers' criterion. Nagi and Salhi (2005) developed another set of heuristics to solve the same problems. The new heuristics outperformed the heuristics proposed in Salhi and Nagi (1999a). Chunyu *et. al.* (2009) present the MDVRP with backhauls. Their work studies the homogeneous and heterogeneous fleet of vehicles. The presented problems were solved using genetic algorithms. Wang *et. al.* (2009) studied a similar problem and developed a hybrid genetic algorithms and tabu search procedure to solve the problem. Irnich (2000) studied a special type of MDVRP with pickup and delivery. In the presented problem, it was assumed that all deliveries or pickups were performed at one central location, and that each vehicle can serve only one or very few customers. Therefore, vehicle routing was not the main decision in this problem; instead, assignment of requests for pickup or delivery was the main decision variable.

Heterogeneous vehicles' fleet was considered. The authors developed an algorithm to solve the problem.

The MDVRP with time windows have been studied by many researchers. Cordeau *et. al.* (2001) introduced the problems of MDVRP with time windows and the periodic VRP with time windows. The authors proposed a tabu search heuristics to solve the problems. To check the quality of the proposed heuristic the authors developed a number of instances for the two problems. Polacek *et. al.* (2001) considered the MDVRP with time windows. They used a variable neighborhood search combined with tabu search to solve the problem. Chiu *et. al.* (2006) proposed a heuristic approach and a tabu search metaheuristic to solve the same problem. They considered minimizing waiting times by adding them to the objective function. Dondo and Cerda (2007) studied the MDVRP with time windows and heterogeneous fleet of vehicles. They developed a novel three phase heuristic/algorithmic method to solve the problem. The proposed approach aimed at integrating a clustering heuristic into an optimization framework. Hadjar and Soumis (2009) proposed a branch and price algorithm to solve the MDVRP with time windows. They incorporated a procedure to speed up the classical branch and cut algorithm. Dondo and Cerda (2009) introduced the MDVRP with soft time windows. Customer nodes were either delivery or pickup. The problem was solved using a local search improvement algorithm that explored a large neighborhood of the current solution. The paper also presented a decomposition scheme to reduce the problem size. Zhen and Zhang (2009) solved the same problem using ant colony and local search improvement algorithms.

CHAPTER 3

MATHEMATICAL MODEL

3.1 Introduction

Design of integrated forward and reverse logistics (IFRL) networks requires considering a number of factors and conditions. These factors and conditions may not be similar to those of the typical forward logistics networks. Therefore, to design IFRL networks efficiently, we need to consider new models that address the relevant factors and conditions. Mathematical programming has been widely used to design different types of logistics networks, such as forward logistics networks, reverse logistics networks, and more recently, IFRL networks. As noticed from the literature review in Chapter 2, most of the mathematical programming models that represent IFRL networks are mixed integer linear programming (MILP) models. In this chapter we present a MILP model to represent an IFRL network. The proposed model represents location, routing, and assignment issues within the integrated forward and reverse frame. It is based on minimizing a combination of location and routing costs, under capacity and flow constraints, within a time restricted environment. Decision variables that represent various routing and location decisions are introduced, in addition to some auxiliary variables. The proposed mathematical model may represent different problem types by selecting the appropriate objective function terms, decision variables, and sets of constraints. Hence, relevant elements of the MILP model presented in this chapter are used throughout the thesis to represent different IFRL networks' problems.

3.2 Problem Definition and Mathematical Model

In this section we present the problem definition, as well as, details of the generic mathematical model.

3.2.1 Problem Definition

The problem presented in this chapter describes an IFRL network. The network is to design a combined location, routing, and assignment problem. The presented problem is solved under hard time windows constraints.

The combined location, routing, and assignment problem considers a number of candidate locations to establish a manufacturing facility and a disassembly plant. Each candidate location has an associated fixed cost for establishing the corresponding unit. A manufacturing facility is responsible for producing products to meet customers' demands. It has limited production capacity. The capacity of the manufacturing facility may not be exceeded. A disassembly plant is responsible for examining the collected end-of-life (EOL) products, and separating reusable parts and/or products from scrap. In the presented problem, a disassembly plant is assumed to be uncapacitated. A number of customers at determined locations are present. Each customer has a predefined amount of delivery and pickup products. Moreover, customers have time windows within which the service must be completed. Each customer is assigned to exactly one manufacturing facility and one vehicle. Vehicles serve customers by delivery of the customers' demands, and simultaneously, pickup of EOL products. Delivery and pickup for a customer must be performed completely at one stop using exactly one vehicle, i.e. no splitting of delivery or pickup is allowed. This restriction is recognized by many researchers, since customers usually prefer to be visited only once for their convenience.

The sequence of visiting customers is the routing decision considered in this problem. If a vehicle arrives earlier than the beginning of a customer's time window, the vehicle waits until the beginning of the time window to start the service, resulting in wait time. Arriving after the time window is not allowed. Vehicles' capacities may not be exceeded at any stop. A vehicle is assigned to exactly one manufacturing facility. Vehicles start their trips from the assigned manufacturing facility. After serving all assigned customers, vehicles visit a disassembly plant. At the disassembly plant, vehicles deliver the EOL products collected from customers. At the same time vehicles collect from the disassembly plant reusable parts and/or products in addition to scrap. Afterwards, vehicles visit a waste disposal site, where they dispose of scrap. Finally, each vehicle ends the trip at the manufacturing facility where it has started the trip. Vehicles deliver reusable parts and/or products to its assigned manufacturing facility. The IFRL network defined here is modeled mathematically considering the following assumptions:

- A homogeneous fleet of vehicles is available at each manufacturing facility.
- Unlimited number of vehicles at each manufacturing facility.
- Uncapacitated disassembly plants.
- Deterministic data.
- Manufacturing facilities are not linked.
- Manufacturing facilities, disassembly plants, and the waste disposal sites have no time window constraints.
- Single commodity for delivery and pickup.

- The amount of scrap materials and the total quantities of the reusable parts and/or products from the disassembly plant are smaller than the total available vehicle capacity.

These assumptions are to define the problem. However, many of them may be relaxed without much difficulty. For example adding time windows to manufacturing facilities, and/or disassembly units would require mild modifications to the time windows constraints ranges.

3.2.2 Mathematical Model

Before presenting the mathematical model, we present the notations and symbols that will be used throughout the model.

Parameters:

N_D : Number of (candidate) manufacturing facility locations;

N_C : Number of customers;

N : Total number of nodes;

K : Number of vehicles to employ;

V_C : Vehicle capacity;

Dc_j : Capacity of Manufacturing facility at node $j = 1,2,...N_D$;

C_{ij} : Transportation cost between node i and node j with $C_{ii} = 0$ and $i, j = 1,2,...N_D$, $N_D + 1,..., N_C + N_D$, $N_C + N_D + 1,..., N - 1, N$, where, nodes $N_D + 1,..., N_D + N_C$ representing all the customers nodes,

nodes $N_C + N_D + 1, ..., N-1$ are candidate locations for the disassembly site, and node N representing the disposal site;

T_{ij} : Travel time between node i and node j, $i, j = 1,2,...N$ and $T_{ii} = 0$;

E_j : Earliest allowed start time of service for the customer at node j, $j = N_D + 1, ... N_D + N_C$;

L_j : Latest allowed start time of service for the customer at node j, $j = N_D + 1, ... N_D + N_C$;

Sr_j : Service time for the customer at node j, $j = N_D + 1, ... N_D + N_C$;

D_j : Amount of new products to deliver to the customer at node j, $j = N_D + 1, ..., N_D + N_C$;

P_j : Amount of EOL products to collect from the customer at node j, $j = N_D + 1, ..., N_D + N_C$;

md : The number of manufacturing facilities to open;

dp : The number of disassembly plants to open;

F_d : Fixed cost of setting up a manufacturing facility at location d, $d = 1, ..., N_D$;

F_e : Fixed cost of setting up a disassembly plant at location e, $e = N_C + N_D + 1, ..., N-1$;

M : A positive large number

Decision Variables:

l_v^o : Load of vehicle v when it leaves the manufacturing facility, $v = 1,...,K$

l_j : Load of a vehicle after it serves the customer at node j, $j = N_D + 1,..., N_D + N_C$;

ld_j : Load at manufacturing facility in location j, $j = 1,..., N_D$;

π_j : Auxiliary variable to prohibit subtours, $j = N_D + 1,..., N_D + N_C$;

st_j : Service starting time for the customer at node j, $j = N_D + 1,...N_D + N_C$;

$$u_{ij} = \begin{cases} 1, & \text{if customer at node } i \text{ is assigned to manufacturing facility } j, \\ 0, & \text{otherwise,} \end{cases}$$

$i = N_D + 1,..., N_C + N_D$, $j = 1,..., N_D$

$$x_{ijv} = \begin{cases} 1, & \text{if vehicle } v \text{ travels from node } i \text{ to node } j, \\ 0, & \text{otherwise,} \end{cases} \quad i, j = 1,..., N, v = 1,...K;$$

$$y_d = \begin{cases} 1, & \text{if a manufacturing facility is established at location } d, \\ 0, & \text{otherwise,} \end{cases}$$

$d = 1,..., N_d$.

$$y_e = \begin{cases} 1, & \text{if a disassembly plant is established at location } e, \\ 0, & \text{otherwise,} \end{cases}$$

$e = N_C + N_D + 1,..., N - 1$.

$$z_{vj} = \begin{cases} 1, & \text{if vehicle } v \text{ is assigned to manufacturing facility } j, \\ 0, & \text{otherwise,} \end{cases}$$

$v = 1,..., k$, $j = 1,..., N_D$

Objective Function and Constraints:

The objective function of the model is to minimize the total cost encountered. The total cost consists of:

a. Total vehicle traveling cost

b. Cost of opening a manufacturing facility

c. Cost of opening a disassembly plant

Mathematical representation of the objective function is:

Minimize $z =$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{v=1}^{K} C_{ij}.x_{ijv} + \tag{3.1a}$$

$$\sum_{d=1}^{N_D} F_d.y_d + \tag{3.1b}$$

$$\sum_{e=N_C+N_D+1}^{N-1} F_e.y_e \tag{3.1c}$$

The above objective function will be minimized with the following constraint functions:

$$\sum_{i=1}^{N_c+N_D}\sum_{v=1}^{K} x_{ijv} = 1, \; j = N_D+1,\ldots,N_C+N_D \tag{3.2}$$

$$\sum_{i=1}^{N} x_{isv} = \sum_{j=1}^{N} x_{sjv}, \; s = N_D+1,\ldots,N, \; v = 1,\ldots,K \tag{3.3}$$

$$z_{vj} = \sum_{i=N_D+1}^{N_C+N_D} x_{jiv}, \; j = 1,\ldots,N_D, \; v = 1,\ldots,K \tag{3.4}$$

$$u_{ij} \geq \left(\left(z_{vj} + \sum_{q=1}^{N-1} x_{iqv}\right) - 1\right), \quad i = N_D+1,\ldots,N_C+N_D, \; j = 1,\ldots,N_D, v = 1,\ldots,K \tag{3.5}$$

$$\sum_{j=1}^{N_D} u_{ij} = 1 \, , \; i = N_D + 1, \ldots, N_C + N_D \tag{3.6}$$

$$ld_j = \sum_{i=N_D+1}^{N_C+N_D} u_{ij} D_i \quad , \quad j = 1, \ldots, N_D \tag{3.7}$$

$$ld_j \leq Dc_j \; , \; j = 1, \ldots, N_D \tag{3.8}$$

$$l_v^o = \sum_{i=1}^{N_C+N_D} \sum_{j=N_D+1}^{N_C+N_D} D_j x_{ijv} \; , \; v = 1, \ldots, K \tag{3.9}$$

$$l_j \geq l_v^o - D_j + P_j - M(1 - \sum_{i=1}^{N_D} x_{ijv}) \, , \; j = N_D + 1, \ldots, N_C + N_D, v = 1, \ldots, K \tag{3.10}$$

$$l_j \geq l_i - D_j + P_j - M(1 - \sum_{v=1}^{K} x_{ijv}) \;\; i, j = N_D + 1, \ldots, N_C + N_D, \;\; j \neq i \tag{3.11}$$

$$l_v^o \leq V_C \; , v = 1, \ldots, k \tag{3.12}$$

$$l_j \leq V_C \, , \; j = N_D + 1, \ldots, N_C + N_D \tag{3.13}$$

$$\sum_{i=1}^{N_D} \sum_{j=N_D+1}^{N_D+N_C} x_{ijv} = 1 \, , \; v = 1, \ldots, K \tag{3.14}$$

$$x_{djv} \leq y_d \, , \;\; d = 1, \ldots, N_D, j = N_D + 1, \ldots, N_C + N_D, v = 1, \ldots, K \tag{3.15}$$

$$\sum_{d=1}^{N_D} y_d = mf \tag{3.16}$$

$$\sum_{i=N_D+1}^{N_C+N_D} \sum_{e=N_C+N_D+1}^{N-1} x_{ie\,v} = 1 \, , \; v = 1, \ldots, K \tag{3.17}$$

$$x_{iev} \leq y_e \, , \;\; i = N_D + 1, \ldots, N_C + N_D, e = N_C + N_D + 1, \ldots, N - 1, v = 1, \ldots, K \tag{3.18}$$

$$\sum_{e=N_C+N_D+1}^{N-1} y_e = dp \, , \tag{3.19}$$

$$\sum_{e=N_D+N_C+1}^{N-1} x_{eNv} = 1\,,\; v = 1,...,K \tag{3.20}$$

$$x_{eNv} \le y_e\,,\; e = N_C + N_D + 1,..., N-1, v = 1,...,K \tag{3.21}$$

$$x_{Niv} = 1\,,\; i = 1,..., N_D, v = 1,...,K \tag{3.22}$$

$$\sum_{j=N_D+1}^{N_C+N_D} x_{ijv} = \sum_{j=N_D+1}^{N_C+N_D} x_{jiv}\;,\; i = 1,..., N_D,\; v = 1,...,K \tag{3.23}$$

$$\pi_j = \pi_i + 1 - N_c(1 - \sum_{v=1}^{K} x_{ijv})\,,\; i, j = N_D + 1,..., N_C + N_D, i \ne j \tag{3.24}$$

$$st_i + T_{ij} + Sr_i - M \times (1 - x_{ijv}) \le st_j$$

$$i = 1,..., N_C + N_D\,,\;\; j = 1,..., N_C + N_D\,,\;\; v = 1,..., K \tag{3.25}$$

$$E_j \le st_j\,,\;\; j = N_D + 1,..., N_C + N_D + 1 \tag{3.26}$$

$$L_j \ge st_j\,,\;\; j = N_D + 1,..., N_C + N_D + 1 \tag{3.27}$$

$$\pi_j \ge 0\,,\; j = N_D + 1,...N_C + N_D \tag{3.28}$$

$$x_{ijv} = \{0,1\}\,,\; i, j = 1,..., N_C, v = 1,..., K \tag{3.29}$$

$$y_d = \{0,1\}\,,\; d = 1,..., N_D \tag{3.30}$$

$$y_e = \{0,1\}\,,\; e = N_C + 1,..., N - 1 \tag{3.31}$$

$$z_{vj} = \{0,1\}\;,\; v = 1,..., k,\; j = 1,..., N_D \tag{3.32}$$

$$u_{ij} = \{0,1\}\;,\;\; i = N_D + 1,..., N_C + N_D\;,\;\; j = 1,..., N_D \tag{3.33}$$

In the above mentioned model, Constraint set (3.2) ensures that each customer is served exactly once. Constraint set (3.3) defines the flow conservation constraints. Constraints (3.4) are the vehicle to manufacturing facility assignment, where each vehicle is assigned to exactly one manufacturing facility. Constraint sets (3.5) and (3.6), together assign each customer to exactly one manufacturing facility. Constraint sets (3.7) and (3.8) ensure that a manufacturing facility capacity limit is not exceeded, with constraint set (3.7) calculating the load at each manufacturing facility, and constraint set (3.8) limiting the maximum load to the manufacturing facility capacity limit. Constraint sets (3.9), (3.10) and (3.11) calculate the vehicles' loads. Constrains (3.9) calculate the vehicles' loads at their assigned manufacturing facilities, while constraint set (3.10) calculates vehicles' capacities for the customers directly after the manufacturing facilities. Constraint set (3.11) calculates vehicles capacities en route. We notice that constraint sets (3.9) – (3.11) calculate the vehicles' capacities at the manufacturing facilities and customers, and not for the disassembly plant. This is due to the assumption mentioned previously that the amount of scrap materials and the total quantities of the reusable parts and/or products from the disassembly plant are smaller than the total available vehicle capacity. Constraint sets (3.12) and (3.13) ensure that the vehicles' capacities are not exceeded at any time. Constraint sets (3.14) – (3.23) define the sequence of vehicles' trips, and the associated logical conditions. Specifically, constraints (3.14) ensure that a vehicle must start the trip from a manufacturing facility, and constraints (3.15) guarantees this facility is established at the corresponding location. Constraint (3.16) limits the number of manufacturing facilities to open. Constraints (3.17) ensure that after serving all its assigned customers, a vehicle visits a disassembly plant, and constraints (3.18) guarantees that a disassembly plant is established at the corresponding location. Constraint (3.19) limits the number of disassembly plants to open. Constraint set (3.20) allows vehicles to visit the waste disposal location after visiting the disassembly plant, and constraint set (3.21) guarantees that vehicles visit the waste disposal site from an established disassembly plant. Constraint set (3.22) is for vehicles to end their trips at the manufacturing facility after visiting the waste disposal site. Constraint set (3.23) is for vehicles to start and end their trip at the same manufacturing facility. Constraints (3.24) are to eliminate any subtours. Time window constraints are defined by constraint sets (3.25) – (3.27). Constraint set (3.25) calculates the start of service at customers. Constraint set (3.26) guarantee that serving customers must begin on or after the beginning of the time window. Constraint set (3.27) guarantee that service must begin no later than the end of the time window. We notice that the time window constraint sets (3.25)-(3.27) are defined only for customers. This is resulting from the assumption that manufacturing facilities, disassembly plants, and the waste disposal site have no time window constraints. Constraint set (3.28) is non negativity constraint. Finally constraint sets (3.29)-(3.33) are binary conditions for some of the decision variables. Figure 3.1 illustrates briefly the network structure represented by the mathematical model.

Location routing problems (LRP) available in the literature consider only the forward logistics network (Min *el.al.* (1998) is an example). IFRL networks consider either location problems, for example as in Ding (2010), or routing problems, for example as in Catay (2010). The model presented in this chapter considers adding a pickup quantity at each customer in addition to establishing disassembly plant units for a LRP. Therefore, this model introduces for the first time the LRP for an IFRL network. Also the model presents a vehicle routing problem with simultaneous pickup and delivery and time windows with different assumptions than those available in the literature, see Mingyong and Erbao (2010) for a comparison. In in Dondo and Cerda (2009) the multi-depot vehicle routing problem has been studied with soft time windows and customers were allowed either pickup or delivery. In this model we present for the first time the multi-depot vehicle routing problem with hard time windows and simultaneous pickup and delivery at customers' nodes.

The presented comprehensive mathematical model is used to represent different versions of IFRL networks in the following chapters.

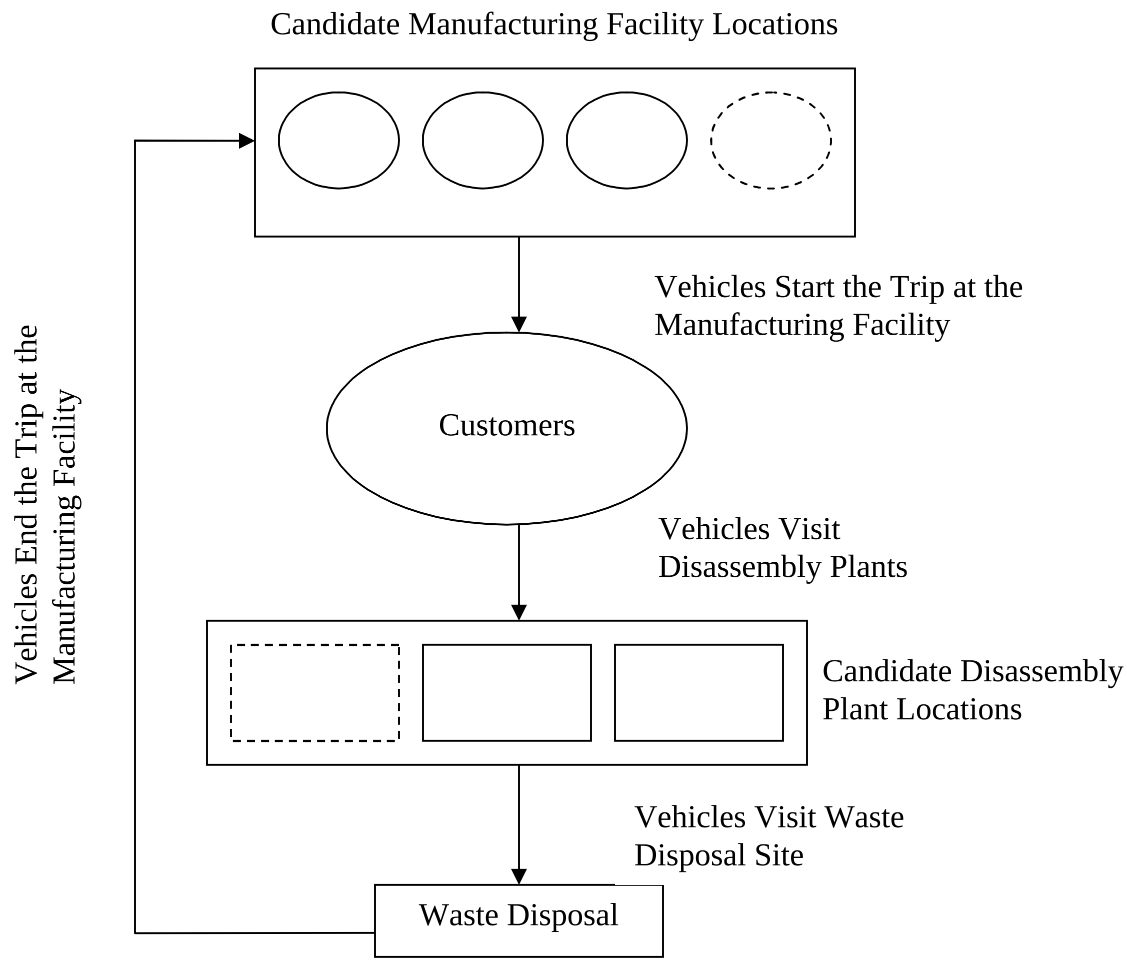

Figure 3.1: IFRL Network Structure Represented by the Proposed Model

CHAPTER 4

COMBINED LOCATION ROUTING

4.1 Introduction

In this chapter, integrated forward and reverse logistics (IFRL) network design problems are discussed with site location and routing issues. Two mixed integer programming models are developed for optimal network design. The models presented in this chapter are sub-models of the mathematical model presented in Chapter 3. The first model provides decisions for disassembly plant locations as well as vehicle routing. The second model provides decisions for manufacturing facility locations as well as vehicle routing. The two models consider new product delivery to customers, in addition to end-of-life (EOL) product collection. Since the developed models involve location and vehicle routing decisions, solving the models is of NP-hard nature. In order to solve problems represented by these models efficiently, heuristic solution methods are developed and tested in this research. Numerical examples are provided to illustrate the developed models and solution methodologies.

4.2 Disassembly Plant Location-Routing Problem

In this section, we present a combined location routing problem (LRP) for IFRL network design. Details of the problem and the MILP model representing it are given in the following sub-sections.

4.2.1 Problem Definition

In the considered IFRL network design problem, one specific location will be selected among several candidate locations to open a disassembly plant. Each location has an associated set-up (fixed) cost. The collected EOL products will be delivered to the disassembly plant for disassembly, inspection, repair, and/or disposal. The candidate locations are associated with a network connected to customer sites. Vehicles will start from a given manufacturing facility and are routed through the network for delivery and pickup from customers at the nodes of the network. Each link in the network connecting two customers is associated with a travel cost. A vehicle will return to the manufacturing facility after it visits the customers, the disassembly plant and the disposal site. Figure 4.1 shows the network structure and the IFRL process. As shown in Figure 4.1, vehicles start their trips from the manufacturing facility. Each vehicle is loaded with new products to be delivered to the customer sites. The load of each vehicle is to satisfy the demands of customers assigned to that vehicle. The vehicles travel to the assigned customers to deliver new products and to collect EOL products from the assigned customers. Vehicles, after serving their assigned customers, travel to the disassembly plant to deliver the collected EOL products. In addition, they will load the refurbished products or components completed at the disassembly plant. The vehicles will also load the scrap parts and materials from the disassembly plant and transport them to the waste disposal site where the waste is being disposed of. The vehicles will end their trips back at the manufacturing facility with refurbished components and parts from the disassembly plant. The cost components considered in formulating the model are fixed cost for setting up the disassembly plant and transportation costs for transporting new and EOL products to and from customers, EOL products to the disassembly plant, scrap to the waste disposal location, and refurbished components back to the manufacturing facility. In developing the mathematical programming model, we assume that the demand from the customers in terms of new product delivery and EOL product collections are based on aggregate data, and hence deterministic. Vehicles traveling to customer sites perform both delivery and pickup simultaneously. We also assume that the amount of scrap materials and the total quantities of the refurbished products from the disassembly plant are smaller than the total available vehicle capacity. The manufacturing facility and the disassembly plant have unlimited capacity. In developing the model, we consider homogeneous vehicle fleet and single commodity.

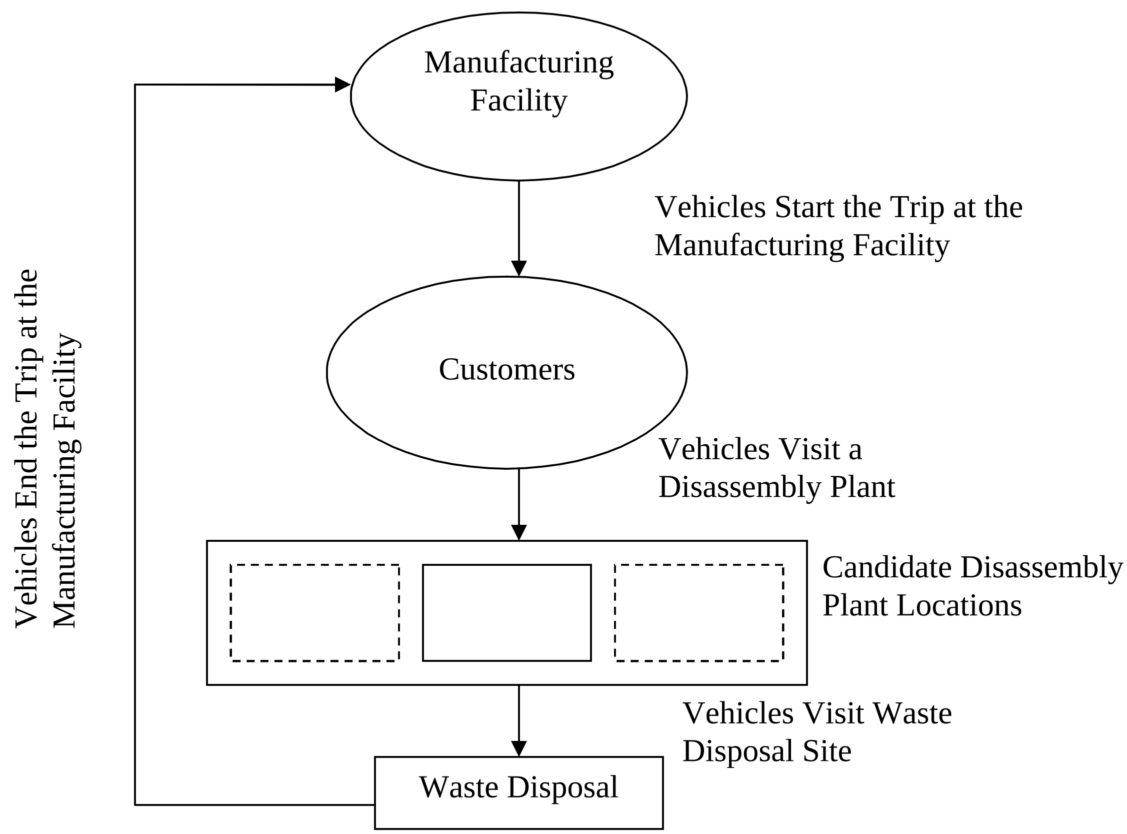

Figure 4.1: Network Structure of the Disassembly Plant LRP

4.2.2 Mathematical Model

The mathematical model representing the problem defined in Section 4.2.1 is a sub-model of the general model developed in Chapter 3. We use the same notations given in Chapter 3. We select the parameters, decision variables, objective function terms and constraints that represent the problem in this section.

Parameters:

The following parameters are considered for the mathematical model that represents the Disassembly Location-Routing Problem:

N_D, N_C, N , K, V_C, C_{ij}, D_j, P_j, dp, F_e, M .

In this mathematical model we consider the following:

$N_D = 1$

C_{ij} : Transportation cost between node i and node j with $C_{ii} = 0$ and $i, j = 1,2,...N_C, N_C + 1,..., N - 1, N$, where node 1 representing the site of the manufacturing facility, node N representing the disposal site and $N_C + 1,..., N - 1$ are candidate locations for the disassembly site;

$dp = 1$;

All the other parameters are defined similar to Chapter 3.

Decision Variables:

The following decision variables are considered for the mathematical model that represents the disassembly plant location routing Problem:

l_v^o , l_j , π_j , x_{ijv} , y_e .

These decision variables are defined similar to Chapter 3.

Objective Function and Constraints:

The objective function of the model is to minimize the total cost. The total cost consists of:

a. Total vehicle traveling cost

b. Cost of establishing the disassembly plant

Mathematical representation of the objective function is by considering the terms (3.1a) and (3.1c) of the mathematical model presented in Chapter 3.

The above objective function will be minimized under the following constraints from the mathematical model in Chapter 3:

(3.2), (3.3), (3.9)-(3.14), (3.17)-(3.22), (3.24), (3.28), (3.29) and (3.31).

4.2.3 Solution Methodology

As pointed out in Chapter 2, LRPs are NP-hard in nature. Therefore, to solve practical size problems of the IFRL problem presented, we use a heuristic approach. Sequential route construction algorithms are among the early and widely used heuristics to solve vehicle routing problems and their variants. In this part of research we use a solution methodology that is based on a sequential route construction method and an upper bound technique. The proposed solution method is based on the insertion procedure introduced in Dethloff (2001) for solving vehicle routing problems with forward and reverse flow. This insertion procedure was modified for solving the LRP discussed in this Section. The modification considers an upper bound approach that increases the efficiency of the procedure for the LRP. In developing the solution procedure, we assume that the vehicles have enough capacity to make delivery and pickup for at least one customer so the considered problem always has feasible solutions. We also assume that the number of vehicles is not limited. The detailed steps of the heuristic method are presented below.

Detailed Steps of the Proposed Heuristic Method

In the following steps we name the manufacturing facility location node 0.

Step 1. Sort the disassembly plant sites by their fixed costs F_e. Without loss of generality, assume that $F_{N_c+1} \le F_{N_c+2} \le \cdots \le F_{N-1}$. Let $e = N_c + 1$ and the initial upper bound $UB = \infty$.

Step 2. Let the set SC contain all customer sites.

Step 3. Let $k = 1$, and let the set URC contain all customer sites. Select a customer site $S_k \in SC$ and let $SC = SC - \{S_k\}$.

Step 4. Create an initial route R, $R = \{0, e, N, 0\}$ connecting the manufacturing facility (Node 0), the disassembly plant e, the waste disposal site N, and back to the manufacturing facility, in that sequence. Insert customer site S_k between Node 0 and plant e to establish a tentative route. Let $URC = URC - \{S_k\}$ and $S_1^* = S_k$. The tentative route is: $R_{new} = \{0, S_1^*, e, N, 0\}$. Let $PR(k) = R_{new} = \{0, S_1^*, e, N, 0\}$. Let $Z = 2$, the number of positions in route R for possible insertion of other customer sites. Calculate the corresponding travel cost $C_T(SC) = \sum_k \sum_{st \in PR(k)} C_{st}$, the total travel and disassembly fixed cost $C(e)$, where $C(e) = F_e + C_T(SC)$. If $C(e) \geq UB$, go to Step 7; otherwise, continue.

Step 5.

Step 5.1. If $URC = \phi$, go to Step 6; otherwise, continue.

Step 5.2. For $p = 1,...,Z$ and all $S_i, S_i \in URC$, check if a feasible route can be established by inserting S_i at position p in route $R = \{0, S_1^*, S_2^*,..., S_{Z-1}^*, e, N, 0\}$, where position 1 is between 0 and S_1^*, position 2 is between S_1^* and S_2^*, …, position Z is between S_{Z-1}^* and site e. If an insertion is feasible, let $I(S_i, p) = 1$; otherwise, let $I(S_i, p) = 0$.

Step 5.3. If $I(S_i, p) = 0$, for all $p = 1,...,Z$ and all $S_i, S_i \in URC$,

then:

the current $R = \{0, S_1^*, S_2^*,..., S_{Z-1}^*, e, N, 0\}$ is a complete route established for a vehicle to travel in the network;

let $PR(k) = R = \{0, S_1^*, S_2^*,..., S_{Z-1}^*, e, N, 0\}$, $k = k + 1$;

select a customer site $S_k \in URC$ and go to Step 4.

otherwise, continue.

Step 5.4. For all $S_i, S_i \in URC$ and $p = 1,...,Z$ with corresponding $I(S_i, p) = 1$, calculate $RCRS(S_i, p)$ following the procedure given in Dethloff (2001). Let $(S_i^*, p^*) = \arg_{S_i, P}[\min RCRS(S_i, p)]$. Insert S_i^* in $R = \{0, S_1^*, S_2^*,..., S_{Z-1}^*, e, N, 0\}$ at position p^*. The updated tentative route is: $R_{new} = \{0, S_1^*, S_2^*,...,.S_i^*,..., S_{Z-1}^*, e, N, 0\}$. If necessary, renumber the selected customers in the tentative route after the new insertion. Calculate $C(e) = F_e + C_T(SC)$. If $C(e) \geq UB$, go to Step 7; otherwise, let $URC = URC - \{S_i\}$, $R = R_{new}$, $Z = Z + 1$, go to Step 5.1.

Step 6. Output all established routes $PR(k), k = 1,2,...$ and calculate the corresponding travel cost $C_T(SC) = \sum_k \sum_{st \in PR(k)} C_{st}$.

Step 7. If $SC \neq \phi$, go to Step 3; otherwise, let $C_T^* = \min(C_T(SC))$, and $C(e) = F_e + C_T^*$. If $C(e) < UB$, then let $UB = C(e)$; otherwise, continue.

Step 8. Let $e = e + 1$. If $e = N$, go to Step 9; otherwise, go to Step 2.

Step 9. Select the site $e^*, e^* = \min(C(e)),\ e = N_c + 1,..., N - 1$ with its corresponding routes $PR(k), k = 1,2,...$ as the solution of the problem. The procedure is complete.

In the above procedure, we break any tie arbitrarily. In Steps 4 and 5.4, the current upper bound is used to eliminate unnecessary routing operations. In other words, for specific S_k and e, if the best insertion causes the total fixed and travel costs to exceed the current upper bound, then subsequent insertions associated with these S_k and e are terminated. This approach should help to eliminate significant amount of calculations in creating vehicle routes which often needs extensive calculation. In Step 5.2, the following calculation is performed to check the feasibility of an insertion. For any node q in the network, let $IMP(q)$ is the immediate predecessor and $IMS(q)$ is the immediate successor of node q. Calculate:

$$RD(0) = V_C - \sum_s D_s \tag{4.1}$$

$$RD(q) = \min\left\{RD(IMP(q)), V_C - l_q\right\} \tag{4.2}$$

$$RP(PRI(0)) = V_C - \sum_s P_s \tag{4.3}$$

$$RP(q) = \min\left\{ RP(IMS(q)), V_C - l_q \right\} \tag{4.4}$$

Where: $l_0 = \sum_s D_s$, $l_q = l_{q-1} - D_q + P_q$

An insertion of customer q in a certain route is feasible if $D_q \le RD(IMP(q))$ and $P_q \le RP(IMS(q))$. It should be noted that the values of RD and RP in (4.1), (4.2), (4.3) and (4.4) are calculated only for customers and the manufacturing facility, not for the disassembly plant site or waste disposal site. The insertion feasibility check process is the same as given in Dethloff (2001). In Step 5.4 of the algorithm, the $RCRS$ values are calculated based on the following procedure, also given in Dethloff (2001). Using the values obtained in (4.1), (4.2), (4.3), and (4.4), calculate:

$$RDT = \left[\sum_q RD(q) \times CD(SUI(q)) \right] / \left[\sum_q CD(SUI(q)) \right] \tag{4.5}$$

$$RPT = \left[\sum_q RP(q) \times CP(q) \right] / \left[\sum_q CP(q) \right] \tag{4.6}$$

$$TC = \left[DU / \sum_s D_s \right] (1 - RDT / V_C) + \left[PU / \sum_s P_s \right] (1 - RPT / V_C) \tag{4.7}$$

$$RCRS(S_i, p) = TD + \lambda \times TC(2 \times C_{\max} - C_{\min}) - \gamma \times (C_{0k} + C_{k0}) \tag{4.8}$$

Where $TD = C_{ik} + C_{kj} - C_{ij}$ and customer k is inserted between nodes i and j. C_{ik}, C_{kj} and C_{ij} are distances of the links connecting nodes (i,k), (k,j) and (i,j), respectively. In calculating RDT in (4.5) and RPT in (4.6), $SUI(q)$ denote the set of immediate successors of node q. $CD(q)$ denotes the travel distance along the route from the manufacturing facility to site q. $CP(q)$ denotes the distance along the route from site q

to the disassembly plant. A detailed step by step illustration of the solution procedure is given in Section 4.2.4. In (4.8), λ and γ are parameters in the range of $[0,1]$. In the process of incorporating Dethloff (2001) heuristic algorithm, we investigated the impact of these two parameters on the quality of the solutions and the results are summarized in Section 4.3.3.

4.2.4 Illustrative Example of the Proposed Solution Method

Figure 4.2 shows the network with 2 candidate disassembly plant locations at node 6 and node 7. Waste disposal site is at node 8. In the network, customer demands and set-up costs are shown next to the nodes. Customers' pickup quantities are all 20 units, except for customer 4, which is 50 units. Travel costs are also shown along the links. All data values of the example are randomly generated.

It is required to choose one location to open a disassembly plant. Starting from the manufacturing facility (node 0 in Figure 4.2); vehicles will deliver the required demand to customers, visit a disassembly plant and the waste disposal site, and return to the manufacturing facility, with the quantities picked up from customers and the disassembly plant. Each vehicle has a capacity of 80 units. Applying the steps of the solution methodology:

$N = 9$, $N_C = 5$.

1. $F_7 \leq F_6$, then, rename F_6 to become F_7, and F_7 to become F_6 accordingly, $F_6 \leq F_7$. $e = 6$, as shown in Figure 4.3. $UB = +\infty$.

2. $SC = \{1,2,3,4,5\}$.

3. $k = 1$. $URC = \{1,2,3,4,5\}$. $S_1 = 1$, and $SC = \{2,3,4,5\}$.

4. $R = \{0,6,8,0\}$. $URC = \{2,3,4,5\}$ and $S_1^* = S_1 = 1$. $PR(1) = R_{new} = \{0,1,6,8,0\}$. $z = 2$.

$C_T(SC) = 14$. $C(e) = 20 + 14 = 34 < UB$, then continue

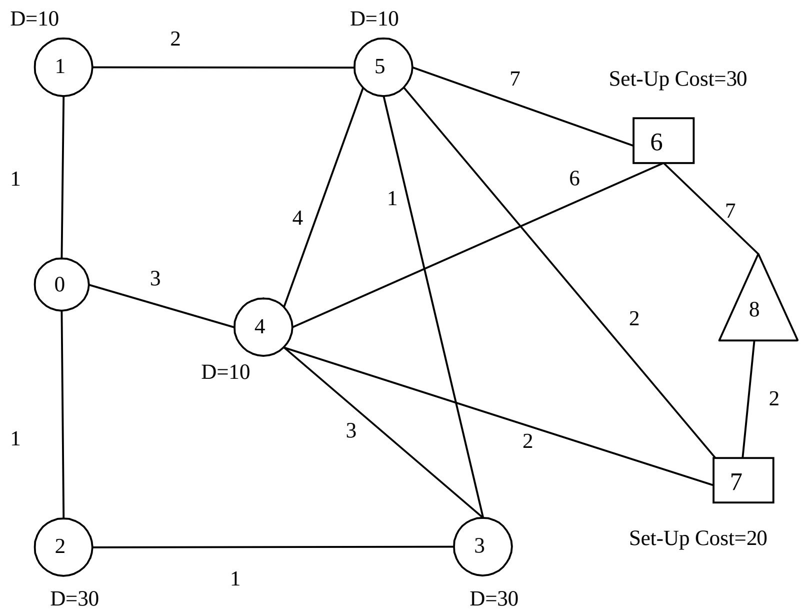

Figure 4.2: Routing Network for 2 Candidate Disassembly Plant Locations

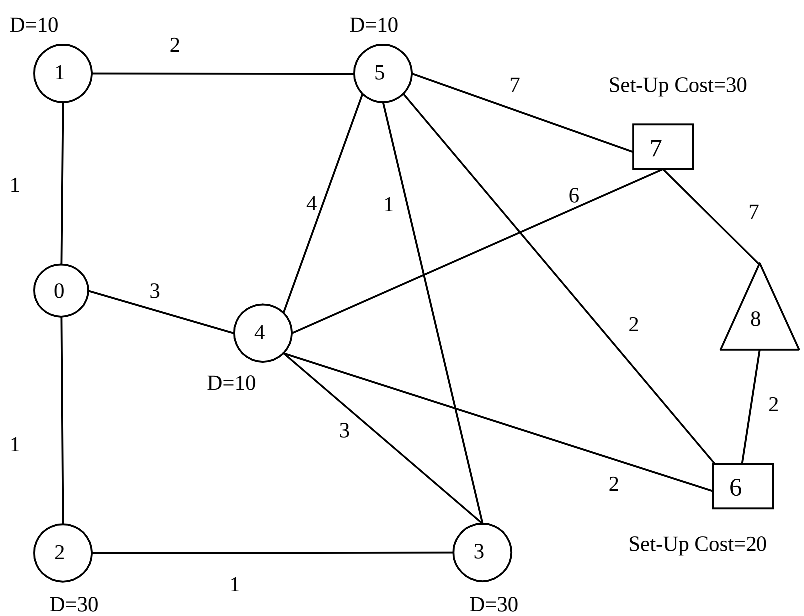

Figure 4.3: Renaming Candidate Disassembly Plant Locations (Step 1)

5.

5.1. $URC \neq \phi$, then continue.

5.2. $p = 1,2$. $URC = \{2,3,4,5\}$. Check the feasibility of inserting customer site 2 (S_1) after the depot $(p = 1)$ to establish $R = \{0,2,1,6,8,0\}$ and after customer 1 to establish $R = \{0,1,2,6,8,0\}$, i.e. check weather $I(S_1,1) = 0, or\ 1$, and $I(S_1,2) = 0, or\ 1$. For $PR(1) = \{0,1,6,8,0\}$:

calculate $RD(0) = 80 - 10 = 70$, $RD(1) = \min(70,80 - 20) = 60$, $D_2 < RD(0)$, $D_2 < RD(1)$. $RP(1) = 80 - 20 = 60$, $RP(0) = \min(60,80 - 10) = 60$. $P_2 < RP(0)$, $P_2 < RP(1)$. Thus it is feasible to insert customer 2 before or after customer 1, i.e. $I(2,1) = 1$, and $I(2,2) = 1$. Similarly, for all $S_i = 3,4,5$, $I(S_i,1) = I(S_i,2) = 1$.

5.3. Since at least one value of $I(S_i, p) = 1$, then continue.

5.4. Calculate $RCRS(S_i, p)$ for all (S_i, p), since all $I(S_i, p) = 1$. As an example, consider $RCRS(S_1,2)$ corresponding to the current tentative route $R = \{0,1,2,6,8,0\}$. Following the calculation procedure:

$$RDT = [RD(0) \times c_{01} + RD(1) \times (c_{01} + c_{12}) + RD(2) \times (c_{01} + c_{12} + c_{26})]/[c_{01} + (c_{01} + c_{12}) + (c_{01} + c_{12} + c_{26})]$$
$$= [40 \times 1 + 30 \times (1 + 2) + 30 \times (1 + 2 + 4)]/11 = 30.9 .$$

$$RPT = [RP(2) \times c_{62} + RP(1) \times (c_{62} + c_{21}) + RP(0) \times (c_{62} + c_{21} + c_{10})]/[c_{62} + (c_{62} + c_{21}) + (c_{62} + c_{21} + c_{10})]$$
$$= [40 \times 4 + 30 \times (4 + 2) + 30 \times (4 + 2 + 1)]/17 = 32.3$$

$$TC = [70/40] \times [1 - 30.9/80] + [90/40] \times [1 - 32.3/80] = 2.42 .$$

Using the parameter values of $\lambda = 1$, and $\gamma = 0$, we have:

$$RCRS(2,2) = c_{12} + c_{26} - c_{16} + 0 \times (TC) \times (2 \times (c_{\max} - c_{\min})) - 1 \times (c_{02} + c_{20}) = 0 .$$

Similarly, all the $RCRS$ values are calculated and summarized in the Table 4.1.

Table 4.1: Different Values of $RCRS$

$RCRS(S_i, p)$	$S_i = 2$	$S_i = 3$	$S_i = 4$	$S_i = 5$
$p = 1$	0	0	0	-2
$p = 2$	0	-2	-4	-6

Since $RCRS(5,2) = -6$ is the smallest among all the values of $RCRS(S_i, p)$, insert $S_i = 5$ in position $p = 1$ and the new route is $R_{new} = \{0,1,5,6,8,0\}$. Since $C(6) = 14 + 20 = 34 < UB$, then continue. Update $URC = \{2,3,4\}$, $R = \{0,1,5,6,8,0\}$ and $Z = 3$. Go to Step 5.1. Repeat the process until all $I(S_i, p) = 0$. The first route is $PR(1) = R = \{0,1,3,5,6,8,0\}$. Step 5.3: Update $k = 2$ and select, $S_2 = 2 \in URC = \{2,4\}$ and go to Step 4. Repeat the process to generate a new route $R_{new} = \{0,2,4,6,8,0\}$. Update $C(6) = 20 + 32 = 52 < UB$. Since $URC = \phi$, go to Step 6.

6. $PR(1) = \{0,1,3,5,6,8,0\}$, $PR(2) = \{0,2,4,6,8,0\}$. $C_T(4) = 32$.

7. $SC \neq \phi$, go to step 3. At step 3: $k = 1$. $URC = \{1,2,3,4,5\}$. $S_1 = 2$, and $SC = \{3,4,5\}$. Continue as mentioned above until step 6. At step 6: $C_T(3) = 32$, similarly $C_T(2) = 32$ $C_T(1) = 32$ $C_T(0) = 32$. When $SC = \phi$, $C^* = 32$, and $C(6) = 20 + 32 = 52$. $C(6) < UB$, then $UB = C(6) = 52$.

8. $e = 7 < N$. Go to Step 2 and follow the same procedure. At Step 5.4, route $PR(1) = \{0,1,7,8,0\}$ is generated with travel cost $C_T(4) = 33$ and total cost $C(7) = 30 + 33 = 63 > UB$. Similarly, all $C_T(SC)$ will yield $C(e) > UB$ and the upper bound remains unchanged. When $SC = \phi$, go to Step 9.

9. $e^* = 6$, with corresponding routes $PR(1) = \{0,2,1,5,6,8,0\}$, $PR(2) = \{0,3,4,6,8,0\}$ and total cost of $C(6) = 20 + 32 = 52$ is the solution of the problem.

4.2.5 Example Problems and Numerical Results

Since there are no available benchmark problems for the problem presented in Section 4.2.2, the above heuristic is tested on a set of randomly generated instances. Two sets of instances are used, a set of small size problems and a set of larger size problems. For the small size problems, each problem has 3 candidate locations for disassembly plant and 6 customer sites. The larger size problems have 4 candidate locations for disassembly plant and 50 customer sites. The smaller problems were also solved by exact methods to determine the effectiveness of the heuristic method. Details of these testing problems and calculation results are presented below.

Problem Features

The structure and data of the test problems are generated randomly. The coordinates forming the networks are uniformly distributed in [0,100]. Distances between pairs of nodes are calculated using Euclidean distance. Customer demands are between 0 and 100 and amounts for pickup are between 0 and 120, both uniformly distributed. For the small size problems, the fixed cost to establish disassembly plants is uniformly distributed between 20,000 and 30,000, and vehicle capacities are set to the total customer demand. More than 1 vehicle may be needed since the pickup quantity for each customer is not a fraction of the demand, i.e., it is possible that some of the pickup amounts can be larger than the demand. For the larger problems, the fixed cost to establish disassembly plants is uniformly distributed between 150,000 and 300,000, and vehicle capacities are set to half the total demand of all customers. Location decisions are long term strategic decisions; while routing decisions are short term and tactical ones. To incorporate both decisions in one model, scaling is required to bring the two decisions to a common planning horizon. In solving these test problems, the scaling is based on the forecasted demand and pickup quantities over a period of 10 years. To simplify cost calculations, the traveling cost is assumed to be 1 money unit per unit distance, and vehicles make the required tours on weekly basis, i.e. the distance matrix expresses the travel distance per week.

Computational Results and Analysis

The heuristic algorithm was coded in Matlab, without utilizing the MATLAB parallel computing facility. Runs were performed on a Pentium Q9300 PC, with 2.5 GHZ processor and 8 GB RAM. To test the effectiveness of the heuristic method developed in this work, we compared the heuristic solutions with the optimal solutions of some of the test problems. To obtain optimal solutions we used LINGO software (LINDO Systems, 2002) on the same PC computer. It took several seconds to solve the small size problems using the heuristic algorithm and using the LINGO solver. It took 231 seconds in average to solve the larger problems using the heuristic method. The size of the integer programming model of the larger problems exceeds the capacity of our LINDO solver. For the small problems, the computational results from the heuristic method were compared with optimal solutions generated by LINGO. The values of λ and γ were incremented by 0.1 each in the reduced ranges of $0 \le \lambda \le 0.5$ and $0.3 \le \gamma \le 1$ as discussed in Section 4.3.3. When deviations between heuristic and optimal solutions exist, the values of λ and γ were further tested over the whole range of [0, 1] to examine if other better solutions may be found. It was found that expanding the range of λ and γ to [0,1] did not yield any better solutions than the ones obtained in the recommended range in Section 4.3.3. The best results for the 10 small test problems are shown in Table 4.2. Optimal solutions recorded in Table 4.2 are based on iteration on the number of vehicles. We obtained optimum solutions for the fleet size provided by the heuristic method, 1 and 2 vehicles less, and 1 and 2 vehicles more than that fleet size.

Table 4.2: Comparison between Optimal and Heuristic Solutions

Problem	Minimum Cost	Cost by Heuristic Method	Difference	Number of Vehicles Required
1	155084	155084	0	1
2	273674	276274	1%	2
3	211462	218742	3.4%	2
4	164242	164242	0	1
5	168296	168296	0	1
6	235571	243371	3.3%	2
7	138908	138908	0	1
8	226347	226347	0	1
9	303552	314472	3.6%	2
10	209259	209259	0	2

As shown in Table 4.2, solutions obtained using the heuristic method for 6 out of the 10 problems are similar to optimal solutions. On the other hand, the maximum and average deviations from optimal for the remaining 4 instances are 3.6%, and 2.8%, respectively. It is noted that the number of vehicles required using the heuristic approach and the exact method are the same.

For the larger size problems with 50 customer sites and 4 candidate disassembly plant sites, all the solutions were obtained with both λ and γ set to 0.5. The number of required vehicles for the larger size problems was 3 vehicles. Computational times for solving these larger problems are shown in Table 4.3.

Table 4.3: Computing Results for Larger Size Problems

Problem	Computational Time (sec.)
1	268
2	251
3	204
4	227
5	207

4.3 Manufacturing Facility Location-Routing Problem

In this section, we present another combined location routing problem for IFRL network design. Details of the problem and the MILP model representing it are given in the following sub-sections.

4.3.1 Problem Definition

In the combined location routing problem presented here, location decisions concern choosing a location to establish a manufacturing facility. A number of potential manufacturing facility locations are available, out of which one location will be chosen to establish the manufacturing facility. Each location has an associated fixed cost. The manufacturing facility is to serve a number of customers, at predefined locations. Customers have a predefined amount of products to be delivered from the manufacturing facility. In addition, customers have a predefined amount of EOL products to be collected. The collected EOL products are returned back to the manufacturing facility, where they are further processed and separated into reusable parts and/or products, or scrap. Hence, the manufacturing facility serves for manufacturing new products to satisfy customers' demands, and simultaneously serve as a collection center for collected EOL products. A set of homogeneous vehicles at the established manufacturing facility are available to serve customers. The number of required vehicles is determined according to the demand, pickup, and vehicle capacity. Routing decisions concern assigning customers to vehicles and providing the sequence of customers' visits by vehicles. A customer is served exactly once, using exactly one vehicle. Reverse logistics are represented in the form of EOL commodities pickup from customers. The selection criterion for choosing the manufacturing facility location and the routing decisions are based on cost minimization. The total cost consists of the fixed cost of establishing a manufacturing facility and the vehicles' traveling cost.

Vehicles start their trips from the manufacturing facility. After serving all assigned customers, vehicles end the trip at the manufacturing facility. Figure 4.4 represents the network structure defined and modeled in this section.

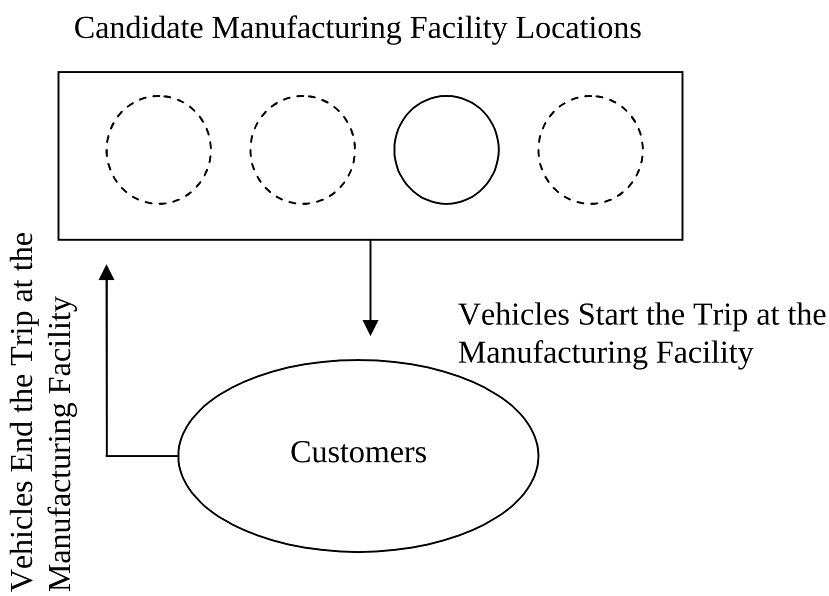

Figure 4.4: Network Structure of the Manufacturing Facility LRP

In the presented problem we assume uncapacitated manufacturing facility, a homogeneous fleet of vehicles, unlimited number of vehicles and a single commodity model. The output of the proposed model is the location to establish the manufacturing facility, the vehicles' routes, and the number of vehicles required.

4.3.2 Mathematical Model

The mathematical model representing the problem defined in Section 4.3.1 is a sub model of the general model developed in Chapter 3. We use the same notations given in Chapter 3. We select the parameters, decision variables, objective function terms and constraints that represent the problem in this section.

Parameters:

The following parameters are considered for the mathematical model that represents the Manufacturing Facility Location-Routing Problem:

N_D, N_C, N , K, V_C, Dc_j, C_{ij}, D_j, P_j, md , F_d, M .

In this mathematical model we consider the following:

$Dc_j = \infty$;

C_{ij}: Transportation cost between node i and node j with $C_{ii} = 0$ and $i, j = 1,2,...N_D$, $N_D + 1,..., N$, where, nodes $N_D + 1,..., N$ representing all the customers nodes, and node $N = N_D + N_C$;

$md = 1$;

All the other parameters are defined similar to Chapter 3.

Decision Variables:

The following decision variables are considered for the mathematical model that represents the Manufacturing Facility Location-Routing Problem:

$$l_v^o,\ l_j,\ \pi_j,\ x_{ijv},\ y_d\ .$$

These decision variables are defined similar to Chapter 3.

Objective Function and Constraints:

The objective function of the model is to minimize the total cost consisting of:

d. Total vehicle traveling cost

e. Cost of establishing the manufacturing facility

Mathematical representation of the objective function is by considering the terms (3.1a) and (3.1b) of the mathematical model presented in Chapter 3.

The above objective function will be minimized under the following constraints from the mathematical model in Chapter 3:

(3.2), (3.3), (3.9)-(3.16), (3.24), (3.28)-(3.30).

4.3.3 Solution Methodology

As mentioned previously, the presented problem is NP-hard. Therefore, we propose a heuristic approach to solve practical size problems. The proposed approach is based on sequential route building and upper bound methods. The concept of the solution method is similar to that given in Section 4.2.3, with modifications to solve the manufacturing facility location routing problem. The solution methodology is briefly explained next.

Step 1. Sort the manufacturing facilities' sites by their fixed costs F_d. Without loss of generality, assume that $F_1 \le F_2 \le \cdots \le F_{N_D}$. Let $d = 1$ and the initial upper bound $UB = \infty$.

Step 2. Let the set SC contain all customer sites.

Step 3. Let $k = 1$, and let the set URC contain all customer sites. Select a customer site $S_k \in SC$ and let $SC = SC - \{S_k\}$.

Step 4. Create an initial route R, $R = \{d, S_k, d\}$ connecting the manufacturing facility d to customer site S_k and back to the manufacturing facility d. Let $URC = URC - \{S_k\}$ and $S_1^* = S_k$. The tentative route is: $R_{new} = \{d, S_1^*, d\}$. Let $PR(k) = R_{new} = \{d, S_1^*, d\}$. Let $Z = 2$, the number of positions in route R for possible insertion of other customer sites. Calculate the corresponding travel cost $C_T(SC) = \sum_k \sum_{st \in PR(k)} C_{st}$, the total travel and manufacturing facility fixed cost $C(d)$, where $C(d) = F_d + C_T(SC)$. If $C(d) \ge UB$, go to Step 7; otherwise, continue.

Step 5.

Step 5.1. If $URC = \phi$, go to Step 6; otherwise, continue.

Step 5.2. For $p = 1, ..., Z$ and all $S_i, S_i \in URC$, check if a feasible route can be established by inserting S_i at position p in route $R = \{d, S_1^*, S_2^*, ..., S_{Z-1}^*, d\}$, where position 1 is between d and S_1^*, position 2 is between S_1^* and S_2^*, …, position Z is between S_{Z-1}^* and site d. If an insertion is feasible, let $I(S_i, p) = 1$; otherwise, let $I(S_i, p) = 0$.

Step 5.3. If $I(S_i, p) = 0$, for all $p = 1,...,Z$ and all $S_i, S_i \in URC$,

then:

the current $R = \{d, S_1^*, S_2^*, ..., S_{Z-1}^*, d\}$ is a complete route established for a vehicle to travel in the network;

let $PR(k) = R = \{d, S_1^*, S_2^*, ..., S_{Z-1}^*, d\}$, $k = k + 1$;

select a customer site $S_k \in URC$ and go to Step 4.

otherwise, continue.

Step 5.4. For all $S_i, S_i \in URC$ and $p = 1,...,Z$ with corresponding $I(S_i, p) = 1$, calculate $RCRS(S_i, p)$ following the procedure given in Dethloff (2001). Let $(S_i^*, p^*) = \arg_{S_i,P}[\min RCRS(S_i, p)]$. Insert S_i^* in $R = \{d, S_1^*, S_2^*, ..., S_{Z-1}^*, d\}$ at position p^*. The updated tentative route is: $R_{new} = \{d, S_1^*, S_2^*, ...,.S_i^*, ..., S_{Z-1}^*, d\}$. If necessary, renumber the selected customers in the tentative route after the new insertion. Calculate $C(d) = F_d + C_T(SC)$. If $C(d) \geq UB$, go to Step 7; otherwise, let $URC = URC - \{S_i\}$, $R = R_{new}$, $Z = Z + 1$, go to Step 5.1.

Step 6. Output all established routes $PR(k), k = 1,2,...$ and calculate the corresponding travel cost $C_T(SC) = \sum_k \sum_{st \in PR(k)} C_{st}$.

Step 7. If $SC \neq \phi$, go to Step 3; otherwise, let $C_T^* = \min(C_T(SC))$, and $C(d) = F_d + C_T^*$. If $C(d) < UB$, then let $UB = C(d)$; otherwise, continue.

Step 8. Let $d = d + 1$. If $d = N_D + 1$, go to Step 9; otherwise, go to Step 2.

Step 9. Select the site $d^*, d^* = \min(C(d)),\ d = 1,...,N_D$ with its corresponding routes $PR(k), k = 1,2,...$ as the solution of the problem. The procedure is complete.

The procedure for feasibility check and the steps to calculate the $RCRS$ values are similar to the procedure in Section 4.2.3. In calculating RDT in (4.5), $CD(q)$ denotes the travel distance along the route from the manufacturing facility to site q. In calculating RPT in (4.6), $CP(q)$ denote the travel distance along the route from the manufacturing facility to site q. The solution methodology is illustrated using the following example:

Figure 4.5 shows two routing networks, one for each of 2 candidate manufacturing facility sites (manufacturing facility nodes are represented as node 0 and node 1). On the networks, manufacturing facilities' set-up costs are shown next to each manufacturing facility node, while customers' demands are shown next to each customer node. Customers' pickup quantities are fixed at 20 units per customer, except for customer 5, whose pickup quantity is 50 units. It is required to choose one location to establish a manufacturing facility. Starting from that manufacturing, vehicle(s) will deliver the required demand to all customers, and return back to the manufacturing facility, with the customers' pickup quantities. Each vehicle has a capacity of 130 units.

According to the steps explained above we start by considering site 0 (least set-up cost). Arbitrarily choose customer 2 as the first seed customer. Build route (0-2-0). The remaining unrouted customers 3 to 6 are considered for expanding the route. The insertion criterion presented in Section 4.3.3 is evaluated for customers 3 to 6 if inserted after the manufacturing facility, and after customer 2. Customer 3 after the manufacturing facility has the minimum $RCRS$; hence the route is expanded to be (0-3-2-0).

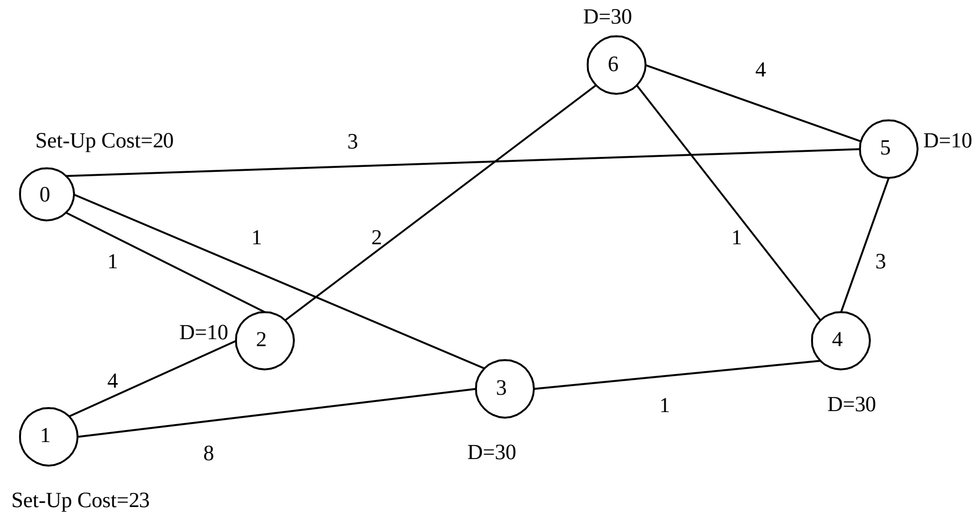

Figure 4.5: Routing Network for 2 Candidate Manufacturing Facility Locations

Customers 4 to 6 are considered for the evaluation criteria in the following positions: directly after the manufacturing facility, directly after customer 3, and directly after customer 2. At this stage, customer 4 directly after the manufacturing facility is chosen to expand the route, which becomes now (0-4-3-2-0). The same procedure is repeated for customers 5 and 6. Customer 5 cannot be added to this route due to the vehicle capacity constraint, and hence a new route is built for customer 5. The minimum $RCRS$ value is obtained when customer 6 is inserted directly after the manufacturing facility. The resulting 2 routes are: (0-6-4-3-2-0) and (0-5-0), with total cost of 34. The upper bound is now set at 34. The above steps are repeated with customers 3 to 6 each serving as a first seed customer. When customer 3 is chosen as the first seed customer, the total cost is 32, and the upper bound is lowered to 32. The corresponding routes are (0-2-6-4-3-0), and (0-5-0). The next step is to consider establishing a manufacturing facility at site 1.

Arbitrarily choose customer 3 as the first seed customer. Build route (1-3-1) with total cost of 39. Since this cost exceeds the upper bound, the insertion process terminates. Consider customer 2 as the first seed customer. Build route (1-2-1), with total cost of 31. It is noticed that expanding the existing the route (1-2-1) will always result in a cost that exceeds the upper bound. Any other customer serving as seed customer will result in a total cost that is higher than the upper bound. Therefore, based on the proposed solution methodology, site 0 with routes (0-2-6-4-3-0) and (0-5-0), results in the minimum total cost and the solution to the problem.

Impact of λ and γ on Routing Solutions

As pointed out in Dethloff (2001) there are no general guidelines of using different λ and γ values in the heuristic algorithm for the VRPSPD. In this research, we used several different instance scenarios and incremental λ and γ values to examine the effect of these two parameters on the quality of vehicle routing solutions. We considered 3 scenarios: 1) customer sites are scattered randomly; 2) customer sites are clustered; and 3) customer sites form a combination of scattered and clustered. We used 4 different instance sets for each scenario. Detailed information on these instance sets are given in Table 4.4 and Table 4.5. In total, 90 test instances were examined and 80 of the best results were obtained with $0 \le \lambda \le 0.5$ and $0.1 \le \gamma \le 1$. In the heuristic algorithm, larger λ tends to let those customers with more delivery and less pickup amounts to be inserted earlier against the shortest distances. One would expect that smaller λ values will lead to better results if the customer sites are more clustered. This was clearly reflected in our test runs as all the best results of the tested problems of the second scenario had

$0 \le \lambda \le 0.5$. The results also confirm that the parameter γ is effective as all the best results of the tested problems correspond to $\gamma \ge 0.1$. Based on the observations on the 90 instances tested in this work, we suggest that the reduced ranges of $0 \le \lambda \le 0.5$ and $0.3 \le \gamma \le 1$ be used in the heuristic routing algorithm. In particular, smaller λ should be considered if the customer sites are more clustered.

Features of the different instances used to examine the values of λ and γ are given in Tables 4.4 and 4.5. For all the instances: The pickup quantity P_i of customer i is calculated using: $P_i = [0.5 + r] \times D_i$, where r is a random number in the range [0,1], and D_i is the demand of customer i. λ and γ are incremented by 0.1.

Table 4.4: Ranges of Randomly Generated Data

Scenario 1	
Set 1	x - y coordinates in [0-100], D_i in [20-60]
Set 2 and 3	5 instances derived from problem R101 in Solomon (1987) where each successive 20 customers forming an instance. Other 5 instances: x - y coordinates in [0-100], D_i in [0-50]
Set 4	x - y coordinates in [0-100], D_i in [20-50]
Scenario 2	
Set 1	x - y coordinates of the depot and 5 customers in [0-30] x - y coordinates of other 5 customers in [40-70], D_i in [20-60]
Sets 2 and 3	Derived from problems C101, C102 in Solomon (1987) where each successive 20 customers forming an instance.
Set 4	x - y coordinates of the depot and 15 customers in [0-60] x - y coordinates of other 20 customers in [60-100], D_i in [20-60]
Scenario 3	
Set 1	x - y coordinates of the depot and 5 customers in [10, 30] x - y coordinates other 5 customers in [0, 70], D_i in [20-60]
Sets 2 and 3	10 different combinations of the data of Set 2 problems of the first and the second scenarios
Set 4	x - y coordinates of the depot and 15 customers in [0,40] x - y coordinates of other 20 customers in [0-100], D_i in [20, 60]

Table 4.5: Common Instance Features for the 3 Considered Scenarios

	Set 1	Set 2	Set 3	Set 4
Number of Instances	5	10	10	5
Number of Customers	10	20	20	35
Min. Number of Vehicles Required	3	3	5	7

4.3.4 Test Problems

Since there are no available benchmark problems for the problem presented by the model in Section 4.3.2, the heuristic is tested on a set of instances based on real data, in addition to another set of randomly generated instances. The set of instances based on real life data consists of five small sized instances, and one medium sized instance, details of which are as follows:

The traveling time (in minutes) between each pair of nodes, and the customers' demands are taken as in Lin *et. al.* (2002). The routing cost is considered 60 money units per hour. The fixed set-up costs of establishing a manufacturing facility is randomly generated between 250,000 and 300,000 money units. We consider non equal set-up costs of different manufacturing facility locations, contrary to the equal set-up costs of manufacturing facility locations in Lin *et. al.* (2002). This is because in real life problems there is a trade off between location and routing decisions where manufacturing facility set-up costs vary. We also choose to consider only manufacturing facilities set-up costs and routing costs, and ignore the other variable costs for the sake of simplification, since the aim is to examine the efficiency of the heuristic approach. To bring strategic location decisions and tactical routing decisions to the same planning horizon, we consider the available demand over a period of 10 years, and calculate the routing costs over that period. To be more consistent with real life expectations, 2 of the small sized problems are considered with a doubled demand, as to examine the situation when future demand increases. The pickup quantity of each customer is randomly generated as a fraction of the customer's demand. Utilized vehicles' capacities are set at 6000 units each for the real demand and 9000 units each when demand is doubled.

Randomly generated instances are larger in size than the above mentioned instances, they are considered medium to large sized, where 34, 40, 46, and 50 customers are considered, and 3 manufacturing facility potential sites for the first 2 instances, and 4 manufacturing facility potential sites for the last 2 instances. The four instances are generated such that:

The coordinates of manufacturing facility potential locations and customers are uniformly distributed in the interval [0, 100]. Distances between every pair of nodes are measured using the Euclidean metric. It is assumed that every unit distance requires 1 min of traveling time, thus the generated distance matrix corresponds to a traveling time matrix in minutes (similar to the data for the six real life data instances). The fixed set-up costs of establishing a manufacturing facility is randomly generated between 200,000 and 350,000 money units. Customers' demands are uniformly distributed between 400 and 3000 units per customer. The pickup quantity for each customer is a random fraction of the customer's demand. The minimum number of vehicles required to serve customers is pre-determined, based on which, vehicles' capacities are calculated as the total demand divided by the pre-set minimum number of vehicles. For the set of randomly generated instances, the minimum number of vehicles required is set to 4. All other parameters and conditions are similar to those of the set of real life data instances, mentioned above. For the small size problems, the values of λ and γ were incremented by 0.1 each in the reduced ranges of $0 \leq \lambda \leq 0.5$ and $0.3 \leq \gamma \leq 1$ as discussed in Section 4.3.3. For the larger size problems the values of λ and γ were set at 0.5.

4.3.5 Results and Analysis

The heuristic algorithm was coded in Matlab, without utilizing the MATLAB parallel computing facility. Runs were performed on a Pentium Q9300 PC, with 2.5 GHZ processor and 8 GB RAM. Run times for the 5 small and moderate test problems were in the order of a few seconds. For the medium/large instances the run times ranged between 72 seconds (34 customers and 3 depots) to 350 seconds (50 customers, 4 depots). To estimate the quality of the proposed heuristic, the 5 small test instances are solved using LINGO software (LINDO Systems, 2002). Optimal solutions are obtained and are compared with solutions obtained from the proposed heuristic. Details of the comparison are shown in Table 4.6.

Table 4.6: Comparison between Optimal and Heuristic Solutions

Problem	Solution		Relative Error (%)	CPU Time (Sec.)	
	Optimum	Heuristic		Optimum	Heuristic
nodes 5-14	594090	650290	9.5	5	5
nodes 12-23	659090	716810	8.8	2403	10
nodes 22-31	578490	612463	5.9	93	12
nodes 5-14 double demand	621130	660000	6.3	132	6
nodes 12-23 double demand	671050	741076	10.4	6911	11

As shown in Table 4.6 the deviation from the optimal solution when using the proposed heuristic ranges between 5.9 % and 10.4%. The trade off between obtaining an optimal solution and a maximum of 10.4 % inferior solution is obvious in the time required to solve the problem. It is noticed that increasing the number of customers from 10 to 12 increases the solution time significantly to obtain an optimum solution, which is expected for NP-hard problems. It is not possible to solve the medium and medium/large problems using LINGO. Table 4.7 shows the time elapsed when solving different problem sizes using the proposed heuristic, in addition to the total cost encountered, and the number of vehicles utilized.

Table 4.7: Results for Larger Problems

Problem	Total Cost Over 10 yrs	Number of Utilized Vehicles	Solution Time (Seconds)
nodes 5-31*	1265236	7	< 15
34 customers, 3 depots	779456	5	72
40 customers, 3 depots	798523	5	80
46 customers, 4 depots	887956	5	270
50 customers, 4 depots	787960	5	350

* Obtained from the real life problem available in Lin *et. al.* (2002)

CHAPTER 5

VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP AND DELIVERY AND TIME WINDOWS

5.1 Introduction

One of the challenging problems in integrated forward and reverse logistics (IFRL) is the vehicle routing problem with simultaneous pickup and delivery and time windows (VRPSPD-TW). The importance of this type of problems lies in its wide applications. For example, one vehicle routing problem with simultaneous pickup and delivery (VRPSPD) application is in the soft drink industries where delivery of full bottles and collection of empty ones may be performed by the same vehicle. In printing material supply, a vehicle may deliver full ink tonners or cartridges and collect empty ones. Another typical application is in electronics manufacturing where manufacturers may be required to take back and properly dispose of end-of-life (EOL) products. In many practical applications, solving vehicle routing problems (VRPs) requires visiting customers during a specific time interval. These types of problems are known as the vehicle routing problem with time windows (VRPTW). In this chapter, we study the VRP with simultaneous pickup and delivery and time windows (VRPSPD-TW), a combination of VRPSPD and VRPTW. We present a mathematical model for the problem. We develop a new heuristic solution to solve practical size problems that have real life applications. The solution obtained using the heuristic method is further improved using simulated annealing (SA) approach. In addition we use Lagrangian relaxation to obtain lower bounds for some problems. Numerical examples are provided to illustrate the developed model and solution methodology.

5.2 Problem Details

In this section, we present a VRPSPD-TW for IFRL network design. Details of the problem and the mixed integer linear (MILP) model representing it are given in the following sub-sections.

5.2.1 Problem Definition

We consider a connected network where at each node a customer requires certain amount of new products to be delivered from a central depot (or manufacturing facility) in the network. In addition, certain amount of EOL products will be collected from the customer site and returned to the central depot. A fleet of vehicles start their trips at the central depot, such that, each vehicle serves a group of customers. Each customer must be served by a vehicle within a given time interval, or a time window. Vehicles perform simultaneous deliveries and pickups at the customer sites and end their trips back at the central depot. To simplify the problem, we assume that the depot does not have time window constraints without loss of generality. Vehicles' capacities must not be exceeded at any stop, and each customer is to be visited only once. The solution of the problem is to determine the number of vehicles needed to serve all customers, to assign groups of customers to be served by each vehicle, and to build the route for each vehicle. In solving this problem we also assume an uncapacitated central depot, single commodity and deterministic data. In addition, it is assumed that the required number of vehicles to serve all customers is always available and the vehicles are homogenous. Furthermore, we assume that each vehicle has sufficient capacity to make delivery and pickup for at least one customer so the considered problem always has feasible solutions. The mathematical programming model of the problem is given next.

5.2.2 Mathematical Model

The mathematical model representing the problem defined in Section 5.2.1 is a sub model of the general model developed in Chapter 3. We use the same notations given in Chapter 3. We select the parameters, decision variables, objective function terms and constraints that represent the problem in this section.

Parameters:

The following parameters are considered for the mathematical model that represents the VRPSPD-TW:

N_C, N , K, V_C, C_{ij}, D_j, P_j, T_{ij}, E_j , L_j , Sr_j , M .

In this mathematical model we consider the following:

C_{ij} : Transportation cost between node i and node j, where node 1 represents the site of the manufacturing facility; $i, j = 1,2,...N$ and $C_{ii} = 0$;

$N = N_C + 1$;

$N_D = 1$;

All the other parameters are defined similar to Chapter 3.

Decision Variables:

The following decision variables are considered for the mathematical model representing the VRPSPD-TW:

l_v^o , l_j , st_j , x_{ijv} .

These decision variables are defined similar to Chapter 3.

Objective Function and Constraints:

The objective function of the model is to minimize the total cost. The total cost is the total vehicle traveling cost.

Mathematical representation of the objective function is by considering the term (3.1a) of the mathematical model presented in Chapter 3.

The above objective function will be minimized under the following constraints from the mathematical model in Chapter 3:

(3.2), (3.3), (3.9)-(3.14), (3.25)-(3.27), (3.29).

5.3 Heuristic Solution

The complexities of VRPTW and VRPSPD are NP-hard (Lenstra and Kan (1981); Dethloff (2001)). Based on these considerations, we propose a heuristic solution method to solve the VRPSPD-TW problem considered in this Chapter.

5.3.1 Details of the Heuristic Method for VRPSPD-TW

Using classical heuristics for solving VRPSPD and similar problems has many advantages. It requires much less computing time than those based on metaheuristics. In addition, it will provide high quality initial solutions for metaheuristic methods (Gajpal and Abad (2010)). Sequential route construction algorithms have been used to solve VRPSPD problems, yielding high quality solutions as noticed in Dethloff (2001). In addition, sequential route construction algorithms have been widely used to solve VRPTW problems as shown in Bräysy and Gendreau (2005a). In this research we use a sequential route construction technique to solve the VRPSPD-TW. The algorithm begins by selecting a seed customer to build a route from the depot to that seed customer and back to the depot. Then, based on predefined insertion criteria, another customer is chosen to be inserted into the route. The insertion process continues until, either all customers are routed or it is not feasible to insert new customers into the current route. In the later case, a new seed customer from the set of unrouted customers (URC) is chosen and the new route building process starts again. The same procedure is repeated until all customers are routed. We use 3 different criteria to choose the seed customer. The route building and insertion are performed for each of the seed customer selection criterion and the best result is recorded. The seed customer selection criteria are based on:

1) Maximum latest allowable service time;

2) Minimum amount of total pickup and delivery;

3) Combination of 1) and 2).

The first criterion allows a large number of customers to be inserted in the route, before and after the seed customer, considering time window constraints. Based on this consideration, for all $j \in URC$, select the seed customer with the maximum L_j i.e.:

$$LAT = \max(L_j) \tag{5.1}$$

Breaking ties arbitrarily.

The second criterion allows a large number of customers to be inserted in the route considering vehicles capacity constraints. Based on this consideration, we calculate, for all $j \in URC$:

$$DP_j = D_j + P_j \tag{5.2}$$

and use the minimum DP_j to select the seed customer.

The third criterion allows a large number of customers to be inserted in the route considering a combination of vehicle capacity and time window constraints. Based on this consideration, we calculate, for all $j \in URC$:

$$LDP_j = L_j - (D_j + P_j) \tag{5.3}$$

and use the maximum LDP_j to select the seed customer.

After the route with the seed customer is established, subsequent customers are selected to be inserted in the current route s. The criterion for choosing a customer and the position to insert the customer in the route is a weighted average of 4 index values. These indices are important to generate a high quality vehicle routes for the problem considered in this work. These 4 indices, used to calculate the selection and insertion (SAI) criterion are:

- minimum additional transportation cost
- minimum reduction to the remaining vehicle capacity
- minimum total waiting time of vehicles on the route
- minimum total number of vehicles to serve all customers

The SAI values for each unrouted customer k to be inserted at position p in the current route s are calculated by:

$$SAI_{kps} = ATC_{kps} + \lambda \times IOI_{kps} \times (C_{\max} - C_{\min}) - \beta \times RDRS_k + \delta \times WT_{kps} \tag{5.4}$$

and the customer with the minimum SAI value will be selected. In the above Eq.(5.4), ATC_{ks} is the additional transportation cost due to the insertion of customer k in position p in route s; IOI_{kps} is the factor for residual capacity with $C_{\max}$ and $C_{\min}$ being the maximum and minimum values in the transportation cost matrix, respectively; $RDRS$ represents the distance from the depot to customer k; and WT_{kps} is the total waiting time within the current route s. Also in Eq. (5.4), λ, β and δ are weights for the corresponding indices. They vary in the range of $[0,1]$. Before presenting the detailed steps of the heuristic method, the specific calculations of the 4 indices are given below. Let S denote the set of customers in the current route s.

Additional Transportation Cost, ATC_{kps}. For all $k \in URC$ and $k \notin S$:

$$ATC_{kps} = C_{ik} + C_{kj} - C_{ij} \qquad \text{for all } i, j \in S$$

where: C_{ik}, C_{kj} and C_{ij} are transportation costs along links (i,k), (k,j) and (i,j), respectively.

Remaining Vehicle Capacity, IOI_{kps}. For all $i \in S$, define:

PRI_i - the immediate predecessor of node i;

SUI_i - the immediate successor of node i;

CD_i - transportation cost from the depot to node i along s;

CP_i - transportation cost from node i to the depot along s;

Insert customer k in all possible position p between each pair of nodes in route s and calculate:

$$l_0 = \sum_{j \in S} D_j, \;\; l_i = l_{PRI_i} - D_i + P_i, i \in S$$

$$RD_0 = V_C - \sum_{j \in S} D_j$$

$$RP_{PRI_0} = V_C - \sum_{j \in S} P_j$$

$$RD_i = \min\left\{RD_{PRI_i}, V_C - l_i\right\}$$

$$RP_i = \min\left\{RP_{SUI_i}, V_C - l_i\right\}$$

$$RDT = \left[\sum_{i \in S} RD_i \times CD_{SUI_i}\right] / \left[\sum_{i \in S} CD_{SUI_i}\right]$$

$$RPT = \left[\sum_{i \in S} RP_i \times CP_i\right] / \left[\sum_{i \in S} CP_i\right]$$

Then the impact index on vehicle capacity when inserting customer k at position p in route s is calculated by:

$$IOI_{kps} = (1 - RDT\ / V_C) + (1 - RPT\ / V_C) \tag{5.5}$$

Eq. (5.5) is similar to the calculation proposed in Dethloff (2001) for generating vehicle routes without time window constraints. However, Eq.(5.5) does not consider the effect of remaining vehicle capacity on unrouted customers. This is based on the observation that considering the effect of remaining vehicle capacity on unrouted customers requires more computational time while does not improve the solution quality of the problems with time window constraints.

<u>*Minimum Number of Vehicles*</u>**,** $RDRS_k$. For all $k \in URC$ and $k \notin S$, we calculate:

$$RDRS_k = 2 \times C_{0k}$$

This is similar to that discussed in Bianchessi and Righini (2007) and Casco *et. al.* (1988); to assign a negative value of the total transportation cost required by an additional vehicle to the selection criterion. The purpose is to reduce the size of the fleet. This "bonus" value is also called radial distance and radial surcharge.

Total Waiting Time of Customers in the Route, WT_{kps}. If a vehicle arrives earlier than the earliest allowable service time of customer i, it must wait until the earliest allowable service time begins. The waiting time for customer i is calculated by:

$$wt_i = \min(0, (E_i - (st_{PRI_i} + Sr_{PRI_i} + T_{PRI_i,i}))$$

To find out the total waiting time within route s, calculate, for all $k \in URC$ and $k \notin S$:

$$WT_{kps} = \sum_{i \in s} wt_i \tag{5.6}$$

Eq. (5.6) is used to measure the impact on the total waiting time of customers in route s when customer k is inserted at position p in the route.

In the heuristic algorithm given below, we calculate the value of SAI for all feasible candidate customers and candidate insertion positions in route s. The customer in the position with minimum SAI value will be selected to insert in the corresponding position.

5.3.2 Steps of the Heuristic Insertion Method for VRPSPD-TW

Step 1. Let the set URC contain all customer sites. Let $a = 1$.

Step 2. Select a customer site $k_{sd} \in URC$ according to Eq.(5.1). Let $URC = URC - \{k_{sd}\}$.

Step 3. Create an initial route S, $S = \{0, k_{sd}, 0\}$ connecting the depot (Node 0) to customer site k_{sd} and back to the depot, in that sequence. Let $k_1^* = k_{sd}$. The tentative route is: $S_{new} = \{0, k_1^*, 0\}$. Let $PR(a) = S_{new} = \{0, k_1^*, 0\}$. Let $Z = 2$; the number of positions in route S for possible insertion of other customer sites.

Step 4.

Step 4.1. For $p = 1,..., Z$ and all $k_i, k_i \in URC$, check if a feasible route can be established by inserting k_i at position p in route $S = \{0, k_1^*, k_2^*,..., k_{Z-1}^*, 0\}$, where position 1 is between 0 and k_1^*, position 2 is between k_1^* and k_2^*, …, position Z is between k_{Z-1}^* and the depot (Node 0). If an insertion is feasible, let $I(k_i, p) = 1$; otherwise, let $I(k_i, p) = 0$.

Step 4.2. If $I(k_i, p) = 0$, for all $p = 1,..., Z$ and all $k_i, k_i \in URC$,

then the current $S = \{0, k_1^*, k_2^*,..., k_{Z-1}^*, 0\}$ is a complete route established for a vehicle to travel in the network. Let $PR(a) = S = \{0, k_1^*, k_2^*,..., k_{Z-1}^*, 0\}$, $a = a + 1$ and go to Step 2; otherwise, continue.

Step 4.3. For all $k_i, k_i \in URC$ and $p = 1,..., Z$ with corresponding $I(k_i, p) = 1$, calculate $SAI_{k_i ps}$ given in equation (5.4) above. Let $(k_i^*, p^*) = \arg_{k_i, p}[\min SAI_{k_i ps}]$. Insert k_i^* in $S = \{0, k_1^*, k_2^*,..., k_{Z-1}^*, 0\}$ at position p^*. The updated tentative route is: $S_{new} = \{0, k_1^*, k_2^*,...,.k_i^*,..., k_{Z-1}^*, 0\}$. Let $URC = URC - \{k_i^*\}$. If necessary, renumber the selected customers in the tentative route after the new insertion.

Step 5. If $URC = \phi$, output all established routes $PR(a), a = 1,2,...$ and calculate the corresponding travel cost $C = \sum_{a} \sum_{kt \in PR(a)} C_{kt}$, otherwise, go to step 4.

Step 6. Repeat Steps 1 to 5 for 2 times, except that in Step 2, the seed customers will be selected using Eqs. (5.2) and (5.3), respectively. The best results from the total 3 time runs of the algorithm will be used as the solution of the problem with the corresponding total cost value. The procedure is then complete.

In the above procedure, we break any tie arbitrarily. In Step 4.1, inserting customer k in position p in route s is feasible if the insertion satisfies the following conditions:

$$D_i \leq RD_{PRI_i} \qquad \text{for all } i \in \{S \cup k\}$$

$$P_i \leq RP_{PRI_i} \qquad \text{for all } i \in \{S \cup k\}$$

$$st_{PRI_i} + Sr_{PRI_i} + T_{PRI_i,i} \leq L_i \qquad \text{for all } i \in \{S \cup k\}$$

5.3.3 Numerical Examples

To examine the proposed insertion heuristic approach, we used 2 types of test problems. The first type of problems has customer nodes in clear clusters, while the second type of problems has customer nodes scattered randomly. For the clustered problems, we used several problems from the sets of problems C1 and C2 of the well known 56 Solomon benchmark problems (Solomon (1987)). For the randomly scattered problems we used some problems from the R1 and R2 sets of the same source. Details of the problem data can be found at http://neo.lcc.uma.es/radi-aeb/WebVRP. In the original Solomon problems, C1 and R1 sets of problems have tight time windows and small vehicle capacities, allowing smaller number of customers to be served by each vehicle. Correspondingly, the fleet sizes to serve all customers will be larger. On the other hand, problems in C2 and R2 sets have loose time windows and larger vehicle capacities. The corresponding fleet size will be smaller. In solving these problems, a vehicle capacity of 200 units was used for problems in C1 and R1 sets. Vehicle capacities of 700 units and 1,000 units were used for C2 and R2 problems, respectively. In solving the test problems in our work, we added pickup amounts for each customer from the original Solomon problems. The pickup quantity p_i at customer i was generated by:

$$p_i = (0.5 + r) \times D_i$$

where r is a random number in the range of (0,1) and D_i is the demand quantity of customer i. Each of the original Solomon problems has one depot and 100 customers. Since we need to test the methods developed in this work for different problem sizes, the original problems were reduced to having 10, 15 and 50 customers. One may note that all the solved VRPSPD-TW problems as reported in the literature have 40 or less customers. To generate the 10, 15 and 50 customer problems, we used data of the first 10, 15 and 50 customers, respectively, from the corresponding Solomon problems. Only the integer values of the cost figures from the original problems were used in solving the reduced testing problems in this work. In presenting our testing results, the names of the original problems were revised to reflect the sizes and the added pickup quantity of these modified problems. For example, problem P15-C101 is based on the original Solomon problem C101 with first 15 customers and pickup quantities generated for each customer. In our test problems, we used a vehicle capacity of 100 units for the 10 and 15 customer problems. For the 50 customer problems, we used vehicle capacities similar to those of the original Solomon problems. The heuristic method developed in this work was coded in Matlab R2007a, without utilizing the MATLAB parallel computing facility. Runs were performed on a Pentium Q9300 PC, with 2.5 GHZ processor and 8 GB RAM.

5.3.4 Results and Analysis

We followed the procedure in the Heuristic Insertion Method for VRPSPD-TW presented in Sections 5.3.1 and 5.3.2 to generate the solutions of the test problems with parameters λ and β incremented by 0.05 each time from 0.0 to 1.0 for the 10 and 15 customer problem. For the 50 customer problems, λ and β were incremented by 0.2 from 0.0 to 1.0. In reporting the results, we report the best solution corresponding to a certain combination of the parameters λ and β. The parameter δ was set at $\lambda/4$ since it tends to generate the best results as observed from our preliminary trials in solving these problems. To test the effectiveness of the heuristic method developed in this work, we compared the heuristic solutions with the optimal solutions of the test problems. We were able to obtain optimal solutions for all the 10 customer and 15 customer problems using LINGO software (LINDO Systems, 2002) on the same PC computer. For the 50 customer problems, we were not able to obtain optimal solutions for only 2 of the 5 cases after more than 20 hours of computation. Comparisons of the total cost values corresponding to the solutions by the heuristic method and optimal solutions by LINGO for the tested 10 customer problems are shown in Table 5.1. Both the proposed heuristic method and the used exact method of LINGO use 2 to 5 seconds of computational times to solve the 10 customer problems. As Table 5.1 shows, the proposed heuristic method found optimal solutions for 6 out of the tested 14 problems. For the other 8 problems, the relative errors in terms of the cost values are from 2.6% to 16.9% with the average being 4.7% for all the 14 tested problems. Similar comparisons for 8 tested problems with 15 customers are presented in Table 5.2. As can be seen from Table 5.2, the heuristic method found optimal solution for 1 of the 8 tested problems. The relative errors are from 1.3% to 11.6% for the near-optimal solutions. The average deviation is 6.8% considering all the 8 tested problems. Table 5.2 also shows the computational times used by LINGO to reach the optimal solution and those used by the heuristic methods which are much shorter.

Table 5.1: Solution Comparison for 10-Customer Problems

Problem	Proposed Heuristic	Solution by LINGO	Relative Error (%)
P10-C107	86	86	0
P10-C109	85	85	0
P10-C201	147	147	0
P10-C202	147	147	0
P10-R101	295	266	10.9
P10-R102	243	226	7.5
P10-R105	275	250	10
P10-R110	210	210	0
P10-R112	200	195	2.6
P10-R203	228	195	16.9
P10-R206	202	195	3.6
P10-R207	192	192	0
P10-R208	207	192	7.8
P10-R209	207	195	6.1
Average			4.7

Table 5.2: Solution Comparison for 15-Customer Problems

Problem	Proposed Method		Solution by LINGO		Relative Error (%)
	Solution	CPU Time (sec.)	Solution	CPU Time (sec.)	
P15-C101	205	0.3	205	42	0
P15-C104	201	1.2	194	2,647	3.6
P15-C203	238	1.8	235	908	1.3
P15-C207	254	1.5	235	1804	8.1
P15-R102	366	0.3	328	7204	11.6
P15-R107	342	0.5	311	1329	9.9
P15-R209	309	1.1	277	32	11.6
P15-R210	317	1.4	290	110	9.3
Average					6.9

Computational results of 5 tested problems with 50 customers are presented in Table 5.3. As can be seen from this table, optimal solutions of 3 out of the 5 problems of this size were obtained using LINGO. Table 5.3 also presents the lower bounds of the cost function and cost values corresponding to the best feasible solutions found by LINGO after 20 hours of computation for problems P50-C107 and P50-R105. No feasible solutions were found by LINGO with 20 hours of computation for other tested 50-custormer problems generated from the Solomon benchmark problems. From Table 5.3, it can be seen that the proposed heuristic method found optimal solution for problem P50-C201. It found near-optimal solution for problems P50-C105 and P50-C106 with errors of 9.3% and 8.6%, respectively. The heuristic method used 1.1 seconds to find a near-optimal solution for problem P50-C107. The near optimal solution deviates by 1.8% from the best feasible solution and by 6.2% from the lower bound found by LINGO after 20 hours of computation. Table 5.3 also presents similar information in solving problem P50-R105, where the near optimal solution is superior by 2.6% to the best feasible solution obtained by LINGO, and deviates by 10.6% from the lower bound. Optimal solutions recorded in Tables 5.1 to 5.3 are based on iteration on the number of vehicles. We obtained optimum solutions for the fleet size provided by the heuristic method, 1 and 2 vehicles less, and 1 and 2 vehicles more than that fleet size.

Table 5.3: Solution Comparison for 50-Customer Problems

Problem	Proposed Method		Solution by LINGO			Relative Error from Optimal or Lower Bound (%)
	Solution	CPU Time (sec.)	Lower Bound	Best Solution	CPU Time (sec.)	
P50-C105	386	1.1	353	353	15	9.3
P50-C106	444	0.95	409	409	59,315	8.6
P50-C107	394	1.1	371	387	(20 Hours)	6.2
P50-C201	430	2.5	430	430	10	0
P50-R105	960	0.73	868	986	(20 Hours)	10.6
Average						6.9

In analyzing the computational results by the developed heuristic method, we notice that the quality of the heuristic solutions is, in general, independent of the problem sizes. The average relative errors from optimum or lower bounds for different types of problems are extracted from Tables 5.1, 5.2 and 5.3. They are summarized in Table 5.4. The average errors of Group 1 (C1 and R1) problems and Group 2 (C2 and R2) problems are comparable. It means that the quality of the heuristic solutions is not related to the tightness of the time windows or the vehicle capacity of these problems. On the other hand, we notice that the errors are significantly different between the clustered type and random type of problems. As can be seen from Table 5.4, the proposed heuristic method performs better in solving clustered type of problems than in solving random type problems. This could be due to that it is easier to assign a cluster of customers to the same vehicle and hence leads to higher quality solutions.

Table 5.4: Average Relative Errors for Different Types of Problems

Problem Group	Average Relative Errors (%)
C1	4
C2	1.9
R1	7.7
R2	7.9
Clustered (C1, C2)	3
Random (R1, R2)	7.8
Group 1 (C1, R1)	5.9
Group 2 (C2, R2)	4.9

5.4 Simulated Annealing

Simulated annealing is a probabilistic local search technique, which imitates the concept of physical annealing of metals. It has been widely and successfully used for solving various combinatorial optimization problems. In general, the SA search process starts with an initial feasible solution and searches for better solutions in the neighborhood of the current solution. It accepts a “bad” neighborhood solution with a certain probability in order to avoid local optima. Simulated annealing has yielded promising solutions to the VRPTW as shown, for example, in Lin *et. al.* (2006) and de Oliveira *et. al.* (2007). A review in Bräysy and Gendreau (2005b) shows that simulated annealing is an effective metaheuristic approach for solving the VRPTW problems efficiently. Hence, we adopt a simulated annealing approach to improve the initial solution obtained by the heuristic insertion method presented in Section 5.3.

5.4.1 Simulated Annealing Procedure

A typical SA procedure in its simple version requires an initial temperature T_{init}, freezing temperature T_0 and a cooling rate r. The typical SA steps to be followed while searching for the optimal solution of any combinatorial optimization problem are:

Step 1. Let the initial solution be the current trial solution f_c and $Z_c = Z_{init}$, where Z_{init} is the value of the objective function to be minimized, corresponding to the initial solution. Set temperature value $T_c = T_{init}$.

Step 2. If stopping criteria are met, stop; otherwise, find a feasible solution f_n in the neighborhood of f_c as the candidate for the next trial solution. Calculate the corresponding objective function value Z_n.

Step 3. If $Z_n \leq Z_c$, let $f_c = f_n$, $Z_c = Z_n$ and $T = T_0 + r \times T_c$, go to Step 2;

otherwise, generate a random number *R* in (0,1):

if $R < P = \exp[(Z_n - Z_c)/T]$, let $f_c = f_n$, $Z_c = Z_n$ and $T = T_0 + r \times T_c$,

go to Step 2;

otherwise: let $T = T_0 + T_c$, go to Step 2.

As mentioned earlier, the SA search procedure evaluates large number of feasible solutions of the problem. In this research, we generate these feasible solutions using the neighborhood search methods discussed below.

5.4.2 Neighborhood Search

In this research, customer exchange operator, edge exchange operator and insertion operator are used to obtain neighborhood solutions. These operators are similar to some of those discussed in Li and Lim (2003), Lin *et. al.* (2006), and Taillard *et. al.* (1997).

1. Customer exchange operator. Two non-empty routes are randomly selected. One customer is arbitrarily selected from each route. These two arbitrarily selected customers exchange their positions. Figure 5.1 illustrates the operation of the customer exchange operator.

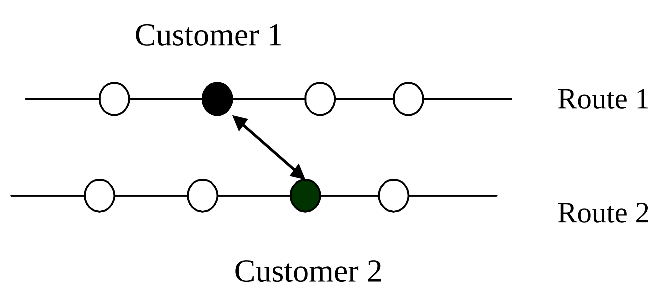

Figure 5.1: Customer Exchange Operator

2. Edge exchange operator. Two non-empty routes are randomly selected. From each route randomly choose a group of consecutive customers, called an edge. The two identified edges will be exchanged. There is no limit on the number of customers per edge. Figure 5.2 illustrates the operation of the edge exchange operator.

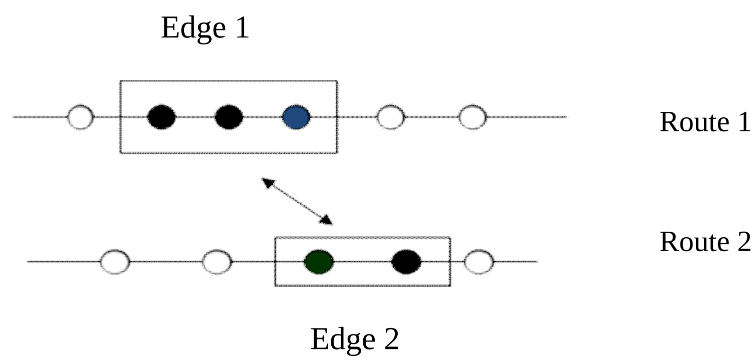

Figure 5.2: Edge Exchange Operator

3. Insertion operator. This operator removes a random customer on an arbitrarily selected route. It then inserts the customer in another randomly selected route. The position of the insertion is also random. Figure 5.3 illustrates the operation of the insertion operator.

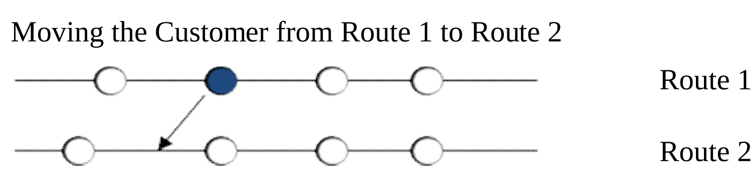

Figure 5.3: Insertion Operator

For customer exchange and edge exchange operators, when choosing 2 routes randomly, the same route may be chosen twice. This is allowed, and the result will be that: two customers on the same route will exchange places in the case of customer exchange operator, while in the case of edge exchange operator, two groups of successive customers, on the same route, will exchange places. In search for neighborhood solutions, one of the operators mentioned above is utilized. The utilized operator is chosen according to a certain predefined probability. If the new (neighbor) solution is infeasible in terms of vehicle capacity or time windows constraints, it is simply rejected. The solution procedure (INST-SA) consisting of the insertion heuristic (Sub-Sections 5.3.1 and 5.3.2), and the simulated annealing procedure (Sub-Sections 5.4.1 and 5.4.2) are tested using a number of numerical examples.

5.4.3 Numerical Examples

To test the proposed insertion heuristic and the simulated annealing approach we used the well known 56 Solomon problems (Solomon (1987)). Details for problems in groups C1, C2, R1, and R2 are similar to those in Sub-Section 5.3.3. We also used problems in groups RC1 and RC2, where, some customers are clustered, while others are scattered. For problems in groups RC1 and RC2, we used customers 1 to 25 and customers 76 to 100. For problems in group RC1, each vehicle has a capacity of 200 units. For problems in group RC2, each vehicle has a capacity of 1000 units. Pickup quantities at each customer, and instances' naming are the same as in Sub-Section 5.3.3. To obtain the initial solution of the test problems, we set the values of λ and β in the insertion heuristic method to 0.0, 0.5 and 1.0 for trial runs, and δ at the level of $\lambda/4$. We conducted several test runs in order to set better values of the parameters in the SA algorithm. Based on the results of the trail runs, $T_{init} = 100, T_f = 1$ and $r = 0.95$ with cooling equation $T = T_0 + r \times T_c$, were selected for the SA search. The procedure would run 1500 iterations before reducing the temperature. The probability of choosing the customer exchange operator was 60%. The probability of choosing the edge exchange operator was 10%. The probability of choosing the insertion operator was 30%. We run the SA for 10 times in solving each of the test problems, and the best result is reported. The heuristic INST-SA procedure developed was coded in Matlab R2007a without using the Matlab parallel computing facilities. Runs were performed on a PC with Pentium Q9300 processor of 2.5 GHZ and 8 GB RAM.

5.4.4 Results and Analysis

We tested the effectiveness of the proposed solution procedure by comparing the heuristic and simulated annealing solutions with optimal solutions of several testing problems of smaller sizes. Due to the size of most of the testing problems and limited computing resources, we are able to obtain optimal solutions using LINGO (LINDO Systems, 2002) for 14 problems among the tested modified 56 Solomon problems as shown in Table 5.5. For these 14 problems, the proposed INST-SA procedure is able to find optimal solutions. As can be seen from Table 5.5, when problem size is relatively small with 10 customers, it took comparable computational time for INST-SA and LINGO to reach the optimal solutions. However, for problems of larger sizes, the advantage of using INST-SA becomes apparent as it requires much less computing time comparing to the brute force based optimization procedure. For problems with 15 customers, INST-SA was able to find optimal solutions in much shorter computational time with one exception recorded. It has been noted that computational time for group 1 problems is less than that for other problems. This is due to the tighter time windows and smaller vehicle capacities which permits fewer customers in one vehicle, and hence a fewer insertion options to select. For problems with 50 customers, optimal solution was obtained for only three of the testing problems by LINGO while INST-SA obtained the same solution much faster. Using LINGO, we were not able to find optimal solutions for other 50 customer problems after 20 hours of computation. Table 5.5 also shows lower bounds and best solutions determined by LINGO after 20 hours of computation for problems P15-RC204, P50-C107, and P50-R105. The heuristic INST-SA procedure generated similar or better solutions than LINGO with much shorter computational time. The maximum errors of the solutions from the lower bounds set by LINGO in terms of the objective function value were 14.5% for P15-RC204, 4.3% for P50-C107 and 5.9% for P50-R105, respectively. Tables 5.6 to 5.11 show the results of the 56 modified Solomon problems in the 6 groups, respectively. Each of these tables shows the transportation costs of the initial solutions, the improvement by the SA search and the required number of vehicles corresponding to the initial solutions and the improved solutions. Results in these tables show that the SA procedure is able to improve the initial solutions for all the testing problems, except for P50-C201. The range of the improvements is from 1.8% to 38.5%. The required number of vehicles for group 1 problems (P50-C1, P50-R1 and P50-RC1) is generally more than those for group 2 problems. This is because in group 1 problems, time windows and vehicle capacities are tighter to allow for small number of customers to be served by each vehicle. Group 2 problems have more relaxed time windows and larger vehicle capacities. The results in Tables 5.6 to 5.11 also show that the number of vehicles in 6 of the group 1 problems was reduced by the SA search process while there is no vehicle reduction by the SA search for group 2 problems. This could also be due to the tight vehicle capacities and time windows for group 1 problems. Table 5.12 shows the average improvements from the SA search for the 6 sets testing problems. From Table 5.12, it can be seen that the average improvement by the SA for all the problems is 19%. We also notice that the improvements for group P50-R2 problems are more significant than those for group P50-R1 problems. A similar phenomenon is observed between group P50-RC2 problems and group P50-RC1 problems. This could also be due to the differences in vehicle capacities and time window limits between these groups of problems. Optimal solutions recorded in Table 5.5 follow the same iteration procedure for optimal solutions in Tables 5.1 to 5.3.

Table 5.5: Solution and Computational Time Comparison

Problem	Vehicle Capacity	Solution by INST-SA		Solution by LINGO		
		Best Solution	CPU Time (sec.)	Lower Bound	Best Solution	CPU Time (sec.)
P10-C101	100	88	3	88	88	2
P10-C202		147	4	147	147	22
P10-R101		266	2.6	266	266	1
P10-R205		213	10	213	213	3
P10-RC103		232	3	232	232	233
P10-RC201		275	1.6	275	275	58
P15-C104	100	194	5.3	194	194	2,647
P15-C203		235	8.2	235	235	908
P15-R102		328	4.6	328	328	7204
P15-R209		277	5.4	277	277	32
P15-RC106		320	3.7	320	320	5
P15-RC204		283	4.8	247	302*	(20 Hours)
P50-C105	200	353	68	353	353	15
P50-C106	200	409	64	409	409	59,315
P50-C107	200	387	70	371	387*	(20 Hours)
P50-C201	200	430	95	430	430	10
P50-C204	700	345	190	NA†	NA†	(20 Hours)
P50-R105	200	919	47	868	986*	(20 Hours)
P50-R210	1000	657	143	NA†	NA†	(20 Hours)
P50-RC102	200	757	54	NA†	NA†	(20 Hours)
P50-RC203	1000	586	170	NA†	NA†	(20 Hours)

*Best feasible solution found by LINGO for 20 hours of computation.
†No feasible solution found by LINGO for 20 hours of computation.

Table 5.6: Results of Modified Solomon C1 Problems

Problem	Total Cost			Number of Required Vehicles		
	Initial Solution	SA Solution	Improvement (%)	Initial Solution	SA Solution	Difference
P50-C101	493	445	9.7	6	6	0
P50-C102	557	452	18.9	6	6	0
P50-C103	581	437	24.8	6	6	0
P50-C104	476	378	20.6	5	5	0
P50-C105	388	353	9.0	5	5	0
P50-C106	444	409	7.7	6	6	0
P50-C107	394	387	1.8	5	5	0
P50-C108	505	390	22.8	6	5	1
P50-C109	526	440	16.3	6	6	0

Table 5.7: Results of Modified Solomon C2 Problems

Problem	Total Cost			Number of Required Vehicles		
	Initial Solution	SA Solution	Improvement (%)	Initial Solution	SA Solution	Difference
P50-C201	430	430	0.0	2	2	0
P50-C202	466	407	12.7	2	2	0
P50-C203	515	405	21.4	2	2	0
P50-C204	447	345	22.8	2	2	0
P50-C205	485	426	12.2	2	2	0
P50-C206	469	409	12.8	2	2	0
P50-C207	465	408	12.3	2	2	0
P50-C208	387	337	12.9	2	2	0

Table 5.8: Results of Modified Solomon R1 Problems

Problem	Total Cost			Number of Required Vehicles		
	Initial Solution	SA Solution	Improvement (%)	Initial Solution	SA Solution	Difference
P50-R101	1154	1021	11.5	12	11	1
P50-R102	1024	837	18.3	11	8	3
P50-R103	910	768	15.6	9	9	0
P50-R104	760	643	15.4	7	7	0
P50-R105	1035	919	11.2	9	9	0
P50-R106	908	783	13.8	8	8	0
P50-R107	824	724	12.1	8	7	1
P50-R108	732	615	16.0	6	6	0
P50-R109	929	776	16.5	8	8	0
P50-R110	842	717	14.8	8	8	0
P50-R111	850	712	16.2	7	7	0
P50-R112	694	644	7.2	6	6	0

Table 5.9: Results of Modified Solomon R2 Problems

Problem	Total Cost			Number of Required Vehicles		
	Initial Solution	SA Solution	Improvement (%)	Initial Solution	SA Solution	Difference
P50-R201	1130	869	23.1	3	3	0
P50-R202	966	736	23.8	3	3	0
P50-R203	824	660	19.9	2	2	0
P50-R204	664	498	25.0	2	2	0
P50-R205	1026	714	30.4	3	3	0
P50-R206	841	667	20.7	2	2	0
P50-R207	761	590	22.4	2	2	0
P50-R208	621	506	18.5	2	2	0
P50-R209	782	674	13.8	2	2	0
P50-R210	922	657	28.7	2	2	0
P50-R211	721	577	20.0	2	2	0

Table 5.10: Results of Modified Solomon RC1 Problems

Problem	Total Cost			Number of Required Vehicles		
	Initial Solution	SA Solution	Improvement (%)	Initial Solution	SA Solution	Difference
P50-RC101	1020	869	14.8	10	10	0
P50-RC102	964	757	21.5	8	8	0
P50-RC103	898	691	23.1	8	6	2
P50-RC104	855	649	24.1	8	6	2
P50-RC105	1089	896	17.7	10	10	0
P50-RC106	890	752	15.5	8	8	0
P50-RC107	860	699	18.7	8	8	0
P50-RC108	737	665	9.8	7	7	0

Table 5.11: Results of Modified Solomon RC2 Problems

Problem	Total Cost			Number of Required Vehicles		
	Initial Solution	SA Solution	Improvement (%)	Initial Solution	SA Solution	Difference
P50-RC201	1085	796	26.6	3	3	0
P50-RC202	923	648	29.8	3	3	0
P50-RC203	847	586	30.8	2	2	0
P50-RC204	701	512	27.0	2	2	0
P50-RC205	1017	667	34.4	3	3	0
P50-RC206	844	644	23.7	3	3	0
P50-RC207	936	576	38.5	3	3	0
P50-RC208	642	478	25.5	2	2	0

Table 5.12: Average Costs of the Initial Solution and the Improved Solution by SA

Problem Group	Average Total Cost		Average Improvement (%)
	Initial Solution	SA Solution	
P50-C1	484.9	410.2	15.4
P50-C2	458.0	395.9	13.8
P50-R1	888.5	763.3	14.1
P50-R2	841.6	649.8	22.8
P50-RC1	914.1	747.3	18.2
P50-RC2	874.4	613.4	29.8
Average	743.6	596.7	19

5.5 Lower Bound

As mentioned previously, the VRPSPD-TW is NP-hard in nature. Reaching optimal solutions for practical size problems in most cases is either not possible or time consuming. In this research, we used LINGO (LINDO Systems, 2002) to obtain optimal solutions. The problems we attempted to solve were some of the 50-customer modified Solomon problems mentioned in Sections 5.3.3 and 5.4.3. We tried to solve all problems belonging to C1, C2, R1, and R2 problems, and some of problems belonging to the RC1 and RC2 groups. Nevertheless, we were able to obtain optimal solutions for only 3 problems, namely, problems P50-C105, P50-C106, and P50-C201.

To further test the effectiveness of the proposed heuristic and simulated annealing procedure, we tried to find lower bounds for some of the problems. We used two methods for the lower bound. The first method was running LINGO for 20 hours and using the lower bound provided by the software. Lower bounds were found for three problems, P15-RC204, P50-C107, and P50-R105, as shown in Tables 5.3 and 5.5.

In another attempt to obtain more lower bounds we used Lagrangian relaxation. We implemented the traditional subgradient optimization method to relax constraint set (3.2) in Chapter 3. After applying Lagrangian relaxation to constraint set (3.2), the objective function of the new problem becomes:

$$\text{Minimize} \quad \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{v=1}^{K} C_{ij}.x_{ijv} + \sum_{j=2}^{N} \lambda_j (x_{ijv} - 1) \tag{5.7}$$

Where, λ_j is the Lagrange multiplier associated with each customer.

Objective function (5.7) is minimized subject to constraints (3.3), (3.9)-(3.14), (3.25)-(3.27), (3.29) from Chapter 3.

Constraint set (3.2) is the only constraint linking vehicles together. Therefore, the relaxed model decomposes to a number of similar sub-problems; a sub-problem for each vehicle.

Using the relaxed model, we tried solving the modified 50-customer problems in C1, C2, R1, and R2 problems (details are in Section 5.3.3). In addition we solved 17 different problems of sizes 10 and 15 customers to test the gap between the Lagrange lower bounds and the optimum solutions. We used the solution obtained from INST-SA as the initial solution for the Lagrangian procedure. Table 5.13 shows a comparison between the optimal solutions and the Lagrangian lower bounds for the 10 and 15 customer problems.

The difference between the Lagrange lower bound and the optimum solution for the 17 small problems (10 and 15 customers) in Table 5.13 ranges between 0.5% and 7.7%, with an average of 2.6%.

As mentioned earlier, we tried to solve the modified 50-customer problems in C1, C2, R1, and R2 problems using the relaxed model. We were able to obtain Lagrange lower bounds for 6 problems. These problems are P50-C105, P50-C106, P50-C107, P50-C201, P50-R101 and P50-R105. For 3 out of the 6 problems, we obtained optimal solutions. The three problems are P50-C105, P50-C106 and P50-C201. Comparison between the

Lagrangian lower bounds and optimal solutions for these three problems are reported in Table 5.13.

Table 5.13: Optimal and Lagrange Lower Bounds

Problem	Optimal Solution	Lagrange Lower Bound	Deviation (%)
P10-C107	86	84	2.3
P10-C109	85	83	2.3
P10-C201	147	146	0.7
P10-C202	147	146	0.7
P10-R101	266	255	4.1
P10-R102	226	222	1.8
P10-R110	210	206	1.9
P10-R112	195	193	1
P10-R203	195	194	0.5
P10-R206	195	191	2.1
P10-R207	192	190	1
P10-R208	192	191	0.5
P10-R209	195	191	2.1
P15-C101	205	194	5.4
P15-C104	194	179	7.7
P15-C203	235	229	2.6
P15-C207	235	219	6.8
P50-C105	353	350	0.8
P50-C106	409	382	6.6
P50-C201	430	395	8.1

We notice from Table 5.13 that the Lagrangian lower bounds of the 50-customer problems deviate from optimal in the range of 0.8% to 8.1%. According to the above results we may conclude that the percentage deviation between the Lagrangian lower bounds and the optimal are not dependant on the problem size. The problems presented in Table 5.13 were solved using our proposed solution methodology INST-SA, and it was able to obtain the optimum solutions for all the problems.

Lower bounds for problems P50-C107, P50-R101 and P50-R105 are presented in Table 5.14. For problems P50-C107 and P50-R105, we have two lower bounds, one through running LINGO for 20 hours and the Lagrange lower bound. The lower bounds from

LINGO are also shown in Table 5.14. Table 5.14 shows that the lower bounds obtained by LINGO are closer to optimum than the Lagrangian lower bounds. INST-SA solution for problem P50-C107 deviates by 4.3% from LINGO lower bound, and by 7% from the Lagrange lower bound. Regarding problem P50-R105, INST-SA solution deviates by 5.9% from LINGO lower bound, and by 6% from the Lagrange lower bound. INST-SA solution for problem P50-R101 deviates 3% from the Lagrange lower bound.

Table 5.14: Lagrange and LINGO Lower Bounds

Problem	Lagrange Lower Bound	LINGO Lower Bound
P50-C107	360	371
P50-R101	990	-
P50-R105	863	868

The reason we were not able to obtain more lower bounds using the simple Lagrangian relaxation presented in this section is as follows:

When using Lagrangian relaxation to decompose the VRPTW, the problem decomposes to a number of elementary shortest path problems with time windows and capacity constraints, denoted by ESPPTWCC (Kohl and Madsen (1997)). According to Kallehauge *et. al.* (2005) the ESPPTWCC is NP-hard in nature. On the other hand, the relaxed model represented by objective function (5.7), and constraints (3.3), (3.9)-(3.14), (3.25)-(3.27), (3.29) is NP-hard also. To prove this, we follow the same approach presented in Dethloff (2001): For any sub-problem represented by the relaxed model, if $P_j \le D_j$ or, $P_j = 0$, the sub-problem reduces to the ESPPTWCC, which is NP-hard in nature. Therefore, a sub-problem represented by the relaxed model in this section is NP-hard as well.

According to Kohl and Madsen (1997), permitting service of customers more than once, i.e. permitting cycling slightly relaxes ESPPTWCC. The new sub-problems become the shortest path problem with time windows and capacity constraints (SPPTWCC). SPPTWCC is solvable in most cases using Dynamic programming. The bounds obtained by solving the SPPTWCC for the VRPTW are of excellent quality, as mentioned in Desrochers *et. al.* (1992). Relaxing the VRPSPD-TW to the SPPTWCC with simultaneous pickup and delivery may result in more and better lower bounds for the original problem. However, this is not discussed in this research and is left for future research.

CHAPTER 6

MULTI-DEPOT VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP AND DELIVERY AND TIME WINDOWS

6.1 Introduction

The multi-depot vehicle routing problem has been studied by many researchers. Different variants of the problem have been acknowledged and studied, as shown in Chapter 2. In this chapter we study the MDVRP within the IFRL context. The presented problem is a variant of the MDVRP; the multi-depot vehicle routing problem with simultaneous pickup and delivery and time windows (MDVRPSPD-TW). According to the literature review that we have performed, this work introduces the problem for the first time. We realize the importance of the problem since the applications of the VRPSPD-TW, mentioned in Chapter 5, are very likely to operate with the multi-depot case. In that case, these applications may benefit from the work presented in this chapter. We present for the first time a MILP model to represent the problem. Since the problem requires solving a number of assignment and vehicle routing sub-problems, the MDVRPSPD-TW is expected to be NP-hard. Therefore, we develop a heuristic technique, followed by simulated annealing procedure to solve practical size problems that have real life applications.

6.2 Problem Details

In this section, we present a MDVRPSPD-TW for IFRL network design. Details of the problem and the MILP model representing it are given in the following sub-sections.

6.2.1 Problem Definition

We consider a number of established manufacturing facilities or depots. Each depot has a limited capacity. A connected network is considered between each depot and all the customers. Customers require an amount of new products to be delivered from a depot. In addition, customers require a certain amount of EOL products to be collected. Each customer must be served by one vehicle within a given time window. A homogeneous fleet of vehicles is available at each depot. The number of vehicles at a depot is not restricted. Each vehicle starts the trip from the associated depot, such that, each vehicle serves a group of customers. A vehicle performs simultaneous delivery and pickup at the customer sites, and ends the trip at the same depot. Vehicles' capacities must not be exceeded at any stop, and each customer must be visited only once. Vehicles are not allowed to travel between depots. Since customers must be served by one vehicle, and travel between depots is prohibited, then any customer must be assigned to one depot and one vehicle. The demand for a group of customers assigned to a depot must not exceed the capacity of that depot. We do not assume depot capacity restrictions on the amount of pickup products. To simplify the problem, we assume that depots do not have time window constraints without loss of generality. In addition, we assume a single commodity model and deterministic data. The required number of vehicles to serve all customers is always available and each vehicle has sufficient capacity to make delivery and pickup for at least one customer. Hence, the considered problem always has feasible solutions. The solution of the problem is to assign a group of customers for each depot, determine the number of vehicles required at each depot to serve the assigned customers. In addition, assign groups of customers to be served by each vehicle, and to build the route for each vehicle. The solution of the problem may result in one or more depots being idle; i.e. no customers are assigned to that depot.

6.2.2 Mathematical Model

The mathematical model representing the problem defined in Section 6.2.1 is a sub model of the general model developed in Chapter 3. We use the same notations given in Chapter 3. We select the parameters, decision variables, objective function terms and constraints that represent the problem in this section.

Parameters:

The following parameters are considered for the mathematical model that represents the MDVRPSPD-TW:

$N_D, N_C, N, K, V_C, Dc_j, C_{ij}, D_j, P_j, T_{ij}, E_j, L_j, Sr_j, M$.

In this mathematical model we consider the following:

C_{ij}: Transportation cost between node i and node j with $C_{ii} = 0$ and $i, j = 1,2,...N_D, N_D + 1,..., N$, where, nodes $N_D + 1,..., N$ representing all the customers nodes.

$N = N_D + N_C$.

All the other parameters are defined similar to Chapter 3.

Decision Variables:

The following decision variables are considered for the mathematical model representing the MDVRPSPD-TW:

$$l_v^o,\ l_{jv},\ ld_j,\ st_j\ ,\ u_{ij},\ x_{ijv},\ z_{vj}$$

These decision variables are defined similar to Chapter 3.

Objective Function and Constraints:

The objective function of the model is to minimize the total cost. The total cost is the total vehicle traveling cost.

Mathematical representation of the objective function is by considering the term (3.1a) of the mathematical model presented in Chapter 3.

The above objective function will be minimized under the following constraints from the mathematical model in Chapter 3:

(3.2)-(3.14), (3.23), (3.25)-(3.27), (3.29), (3.32)-(3.33).

6.3 Solution Methodology

To solve the problem defined in the previous section, we propose a 2-phase heuristic solution method followed by a simulated annealing procedure to improve the results of the heuristic solution.

6.3.1 Heuristic Method for MDVRPSPD-TW

Heuristic methods have been used by many researchers to solve different variants of MDVRP, as seen in Chapter 2. To solve the problem explained in Section 6.2, we use the sequential route construction heuristic presented in Chapter 5, Section 5.3. However, some modifications are necessary for the heuristic to solve the MDVRPSPD-TW. The heuristic for solving the problem with the multi-depot case consists of two phases. The first phase is to assign customers to depots, and the second phase is to solve the VRPSPDTW for each depot with its assigned customers. The first phase of assigning customers to depots is straight forward. For each customer h_i at site i, find its nearest depot NR_j at site j (by checking the cost matrix), breaking ties arbitrarily, where depot capacity permits. Consider assigning the customer at site i to its nearest depot at site j, and perform the following depot capacity check:

Calculate the remaining capacity RC_j of the depot NR_j using the following equation:

$$RC_j = Dc_j - \sum_{l \in T_j} D_l - D_i \qquad (6.1)$$

Where: T_j is the set of customers already assigned to the depot NR_j. As defined in Chapter 3, DC_j is the capacity of the depot at site j, and D_i is the demand amount for the customer at site i.

If $RC_j \geq 0$, then assign customer at site h_i to the depot NR_j,otherwise, check the cost matrix to find the next nearest depot for customer h_i. Perform the above feasibility check. Repeat the assignment and feasibility check until customer h_i is assigned to a depot. Repeat the procedure until every customer is assigned to one depot.

The first phase terminates when all customers are assigned to depots. The second phase is solving the VRPSPD-TW for each depot and its assigned customers. To solve the VRPSPD-TW for the depots we follow the heuristic procedure explained in Chapter 5, Sections, 5.3.1 and 5.3.2.

Steps of the 2-phase heuristic to solve the MDVRPSPD_TW are given next:

Phase 1:

Step 1. Let the set UAC contain all customer sites. Let $b = 1$.

Step 2. Let the sets $T_1,..., T_{N_D} = \phi$.

Step 3. Select a customer $h_i \in UAC$. Let $UAC = UAC - \{h_i\}$.

Step 4. Determine the distance from customer h_i to depots at site j, $j = 1,.., N_D$.

Let the set NR contain all the depot sites such that $NR(b)$ is the depot at site j nearest to customer h_i , $NR(b+1)$ be the depot at another site j that is second nearest to customer h_i , and so on.

Step 5.

Step 5.1. Assign customer h_i to depot $NR(b)$ .Calculate RC_j of $NR(b)$ according to Eq. (6.1).

Step 5.2. If $RC_j < 0$, then $b = b+1$ go to Step 5.1 ; otherwise, continue.

Step 5.3. $T_{NR(b)} = T_{NR(b)} + h_i$, $b = 1$

Step 6. If $UAC \neq \phi$ go to Step 3; otherwise, continue.

Step 7. Output sets $T_1,..., T_{N_D}$.

End of phase 1.

Phase 2:

Step 1. Let $c = 1$.

Step 2. Solve the VRPSPD-TW for set T_c from phase 1 using steps 1 to 6 in Chapter 5, Section 5.3.2. Let $c = c+1$.

Step 3. If $c \leq N_D$ go to Step 2; otherwise, continue.

Step 4. Output all established routes for all sets $T_1,..., T_{N_D}$.

End of phase 2 and the heuristic procedure.

6.3.2 Simulated Annealing for MDVRPSPD-TW

Using simulated annealing for solving VRPSPD-TW in Chapter 5 improved the solutions obtained from the insertion heuristic for most of the test problems. Therefore, we used a simulated annealing approach to solve the MDVRPSPD-TW presented in this chapter. We used the solution obtained from the heuristic procedure explained in Section 6.3.1 as an initial solution for the simulated annealing approach. We used the same SA approach explained in Chapter 5 Section 5.4.1. We mentioned previously that the solutions obtained by the heuristic method may result in idle depots. Idle depots are those depots that do not have any customers assigned to them. The idle depots are considered part of the initial solution provided for the SA procedure. They are represented by an empty route. The SA procedure evaluates a large number of neighborhood solutions using the following neighborhood operators:

1- Depot exchange operator. This operator randomly chooses 2 routes, and exchanges their assigned depots. The resulting new solution is assigning the 2 chosen vehicles and their assigned customers, in the same routing order, to another depot.

2- Customer exchange operator. Where 2 routes are randomly chosen. From each route a customer is randomly chosen and the 2 chosen customers exchange their positions.

3- Insertion operator. In this operator, 2 routes are chosen randomly. A customer is chosen randomly from one of the routes and is inserted into a random position at the other route.

The customer exchange and insertion operators are similar to those used to solve the VRPSPD-TW in Chapter 5, Section 5.4.2.

For the customer exchange operator, the two chosen routes may be identical. For the insertion operator, identical routes are not permitted. For the depot exchange operator, the 2 routes must belong to two different depots. Depots that do not have any customers assigned to them (idle depots) are permitted for the depot exchange operator only. They are represented by empty routes, and this is the only case where empty routes are permitted. However, at least one of the routes chosen for the depot exchange operator must be non-empty. In the case when an idle depot (empty route) is chosen, the vehicle and customers of the non-empty route are assigned to the idle depot. This situation permits idle depots to be examined for potential better solutions.

The neighborhood search process utilizes one of the 3 operators mentioned above. The criterion for choosing an operator is based on a predefined probability. For a (new) neighbor solution obtained, a feasibility check is performed. The feasibility check is for depot capacity, vehicle capacity, and time windows constraints. If the new solution is infeasible, it is simply rejected.

The insertion heuristic and the SA procedure explained in this chapter are tested on numerical examples.

6.4 Numerical Examples

To examine the efficiency of the proposed solution method explained in Section 6.3 a number of numerical examples were used. We used a total of 12 examples of different sizes. Two of these examples were randomly generated such that: the first example consisted of 5 customers and 2 depots. The second example consisted of 9 customers and 2 depots. Distances between pairs of nodes were randomly generated in the range [0, 6]. Demand and pickup quantities were randomly generated in the range [10, 50]. Time windows for customers were randomly generated in the range [1, 10]. Service time was randomly selected at 2 time units for all customers at the two examples. The depots capacities for the 5-customer example were 70 and 100 units. The depots capacities for the 9-customer example were 100 units each. For the two examples each vehicle had a capacity of 100 units. Throughout this chapter the 5-customer example will be referred to as Example 1, and the 9-customer example will be referred to as Example 2. Distances between pairs of nodes are calculated using Euclidean measure.

Another example derived from problem R202 of the Solomon problems was used. We used nodes 2 to 51 as customers' nodes, and nodes 52 to 55 as depot nodes. Hence, the problem consisted of 50 customers and 4 depots. We added pickup amounts for each customer from the original Solomon problems. The pickup quantity p_i at customer i was generated by:

$$p_i = (0.5 + r) \times D_i \tag{6.2}$$

where r is a random number in the range of (0,1) and D_i is the demand quantity of customer i. Additional details for problem R202 are available at Chapter 5, Section 5.3.3.

Another set of examples was derived from test instances available at Cordeau (2001). Details of the problem data can be found at http://www.hec.ca/chairedistributique/data. In the original instances, nodes are generated randomly such that groups of customer nodes form clusters around depots' nodes. There are two groups of instances; each group has a number of problems with different sizes. Problems in group (a) have narrow time windows, while problems in group (b) have larger time windows. We added pickup quantities for each customer using Eq. (6.2). Most of the examples used in this work are smaller in size than the original Cordeau (2001) problems. Only the integer values of the cost figures from the original problems were used in solving the reduced testing problems in this work. In presenting our testing results, the names of the original problems were revised to reflect the sizes and the added pickup quantity of these modified problems. For example, problem P20-2a is based on the original Cordeau (2001) problem 2a with first 20 customers and pickup quantities generated for each customer. The same renaming process is applied for problem R202 mentioned above. We assumed that the number of vehicles available at each depot in unlimited. The number of depots, vehicles' capacities, and depots' capacities vary between problems, and are given in Table 6.1.

Table 6.1: Problems Details

Problem	Number of Depots	Vehicles Capacity	Depots Capacities
Example 1	2	100	70, 100
Example 2	2	100	100
P10-1b	2	100	100
P10-2b	2	100	100
P12-3b	2	100	120
P15-3a	3	150	200
P20-1a	2	200	250
P20-2a	2	200	250
P20-5a	3	200	250
P25-4a	3	200	250
P48-1a	4	200	500
P50-R202	4	200	300

To generate the 10, 12, 15, 25 and 25 customer problems, we used data of the first 10, 12, 15, 20 and 25 customers, respectively, from the corresponding Cordeau (2001). All the original Cordeau (2001) instances employed 4 depots. In the smaller test instances we used 2 or 3 depots only. Table 6.2 shows the corresponding depot nodes employed for each problem.

Table 6.2: Depot Nodes for Cordeau (2001) Modified Instances

Problem	Corresponding Depot Nodes
P10-1b	49, 50
P10-2b	49, 50
P12-3b	145, 148
P15-3a	145, 148
P20-1a	49, 50
P20-2a	49, 50
P20-5a	241, 242, 244
P25-4a	193, 194, 196
P48-1a	49, 50, 51, 52

When solving the examples using the heuristic method, the values of λ and β were incremented by 0.1 in the range [0, 1]. After performing some test runs, we set the value of δ at the level of $\lambda/4$ where the heuristic was noticed to perform best. Regarding the SA approach, we performed a number of test runs. Based on the test runs we set $T_{init} = 100$, $T_f = 1$ and $r = 0.95$ for the SA search. For the test instances with 15 customers or less, the SA procedure would run for 500 iterations before reducing the temperature. For the test instances with 20 customers or more, the SA procedure would run for 3000 iterations before reducing the temperature. The probability of choosing the depot exchange operator was 35%. The probability of choosing the customer exchange operator was also 35%, and the probability of using the insertion operator was 30%. The entire SA annealing process was run for 10 times and the best results are recorded. The heuristic procedure and the SA procedure were coded in Matlab R2007a without using the Matlab parallel computing facilities. Runs were performed on a PC with Pentium Q9300 processor of 2.5 GHZ and 8 GB RAM.

6.5 Results and Analysis

Test instances, presented in Section 6.4, were solved using the proposed 2-phase heuristic and the SA approach explained in Section 6.3. Results were compared with optimal solutions when available. We implemented the mathematical model in Subsection 6.2.2 on LINGO (LINDO Systems, 2002). We tried to obtain optimal solutions for all the problems mentioned in Table 6.1. However, optimal solutions, or LINGO lower bounds were obtained only for problems of 25 customers or less. For larger problems, LINGO software was not able to obtain a feasible solution after running for 10 hours. Comparison between optimal solutions and the SA solution are presented in Table 6.3. Optimal solutions recorded in Table 6.3 are based on iteration on the number of vehicles. We obtained optimum solutions for the fleet size provided by the simulated annealing, 1 to 3 vehicles less, and 1 to 3 vehicles more than that fleet size.

Table 6.3: Solution and Computational Time Comparison

Problem	Proposed Method		Solution by LINGO			Relative Error from Optimal or Lower Bound (%)
	Solution	CPU Time (sec.)	Lower Bound	Best Solution	CPU Time (sec.)	
Example 1	9	5	9	9	2	0
Example 2	25	6	25	25	2	0
P10-1b	327	8	327	327	3	0
P10-2b	581	11	581	581	5	0
P12-3b	458	13	458	458	1200	0
P15-3a	590	8	579	579	25	1.9
P20-1a	784	41	776	776	720	1
P20-2a	819	42	810	810	191,88	1.1
P20-5a	483	37	465	465	3	3.7
P25-4a	798	60	776	803*	(10 Hours)	2.8
P48-1a	1151	120	NA†	NA†	(10 Hours)	-
P50-R202	684	137	NA†	NA†	(10 Hours)	-

*Best feasible solution found by LINGO for 20 hours of computation.
†No feasible solution found by LINGO for 20 hours of computation.

Table 6.4: Results of Test Problems

Problem	Initial Solution	SA Solution	Improvement (%)	Number of Vehicles
Example 1	9	9	0	2
Example 2	28	25	10.7	5
P10-1b	327	327	0	2
P10-2b	599	581	3	2
P12-3b	554	458	17.3	3
P15-3a	706	590	16.4	3
P20-1a	854	784	8.2	3
P20-2a	972	819	15.7	4
P20-5a	622	483	22.3	4
P25-4a	943	798	15.4	6
P48-1a	1284	1151	10.4	7
P50-R202	776	684	11.8	5

Table 6.3 shows that the deviation from optimum ranges between a minimum of 0% and a maximum of 3.7%, with an average of 0.86%. For problem P25-4a, after running LINGO for 10 hours, the solution of the proposed method is 0.6% superior to the best feasible solution provided by LINGO, and deviates by 2.8% from LINGO lower bound. It is noticed from Table 6.3 that deviations from optimal or lower bound are not dependent on the problem size or the problem structure. As can be seen from Table 6.3, when the problem size is relatively small, with 10 customers or less, computational times for the proposed solution method and LINGO software are comparable. Nevertheless, as the problem size increases using the proposed method requires much less computational times than the brute force based optimization procedure, with the exception of problems P15-3a and 20P-5a. The noticeable difference in LINGO computational times for problems with similar sizes and features (for example P20-2a and P20-5a) may be explained by the fact that LINGO software is based on a branch and bound method. For some problems the solution space is smaller than others and hence the branches to be explored are terminated at early stages of the search, which leads to optimal solutions

quickly. Also in some cases, a strong lower bound may be found at early stages of the branch and bound method, leading to terminating a huge number of branches at these early stages, and hence obtaining the optimal solution within a short computation time.

We used the heuristic procedure presented earlier in this chapter to obtain initial solutions for the SA procedure. Comparison between the objective function values obtained by the heuristic procedure (initial solution) and the SA approach, and the number of vehicles utilized, are given in Table 6.4. Table 6.4 shows a comparison between the initial (2-pahse heuristic) solutions and the SA solutions. SA improves the initial solutions for most test problems. The improvement ranges between 0% and 22.3%, and the average improvement is 11%. It is noticed from Table 6.4 that the percentage improvement of the SA is independent of the problem size or problem structure. Table 6.4 also shows the number of vehicles required for each problem. The number of vehicles for all the test problems was similar for the initial solution and the SA procedure.

CHAPTER 7

SUMMARY, CONCLUSION AND FUTURE RESEARCH

7.1 Summary and Conclusion

Environmental friendly manufacturing is one of the growing issues within manufacturing systems. Drivers for environmental friendly manufacturing include customers' expectations, legislation, and economic reasons. These drivers have caused many industries to change their manufacturing attitudes. New manufacturing attitudes include activities like collection of end-of-life (EOL) products, reuse of parts/products, and proper disposal of non-reusable parts. These activities are performed through reverse logistics networks. On the other hand, manufacturing industries usually deliver their manufactured products through typical forward logistics networks. Therefore, environmental friendly logistics include delivery and collection through forward and reverse logistics networks. Forward and reverse logistics networks may be designed separately. Nevertheless, logistics networks that integrate forward and reverse activities are more efficient, as mentioned in Pishvaee *et. al.* (2010). A number of differences exist between forward and reverse logistics networks, for example the number of resources, and the interaction between collection and distribution are some of the differences between the forward and reverse logistics networks. Accordingly, a manufacturing system requires an efficient forward and reverse logistics network to economically operate in an environmental friendly manner.

In light of the above, this research work studied integrated forward and reverse logistics networks (IFRL) for manufacturing systems. We considered the design of different types of IFRL networks, proposed mixed integer linear programming (MILP) models to represent each type of network, and provided efficient solution methodologies to solve practical size problems that have various real life applications. IFRL networks studied in this research work, and the associated solution methodologies, are summarized in the following sub-sections.

7.1.1 Comprehensive MILP Model

In this work we proposed designs for a number of different IFRL networks. We presented a comprehensive MILP model that represented a combined location-routing IFRL network. The model also represented the multi-depot vehicle routing problem within the IFRL context. In addition, the time windows variant of the vehicle routing problem was considered within the model. To practically represent and solve IFRL network problems, sub-models of the comprehensive model were used throughout the thesis. Each sub-model represented a different type of IFRL network. The IFRL networks represented were categorized as NP-hard problems. Therefore, for every network we provided a solution methodology to solve practical size problems.

7.1.2 Location-Routing IFRL Networks Modeling and Solution Methodology

According to the literature review that we had performed, we noticed that location-routing problems are recognized by many researchers for their importance. Nevertheless, studying the location-routing problem within the integrated forward and reverse logistics context has not been addressed previously. Hence, in this work we studied a location-routing problem for an IFRL network. We presented two novel network designs. The first network considered decision on a location, among a number of potential locations, to establish a disassembly plant. Routing decisions were assigning customers to vehicles, and establishing the route for each vehicle such that vehicles start at the manufacturing facility, visit the assigned customers for delivery and pick up, visit the established disassembly plant, visit a waste disposal site and end the trip at the manufacturing facility. We proposed a MILP model to represent the location-routing IFRL network. The MILP model is a sub-model of the comprehensive model discussed in Section 4.1.1, and presented in Chapter 3. The model was solved using a heuristic method. The heuristic method is based on a sequential route building algorithm and an upper bound approach. When compared with optimum, deviation of solutions of the proposed heuristic method ranged between 0% and 3.6%, with an average of 2.8%.

Furthermore, we proposed a second model that considered decision on a location, among a number of potential locations, to establish a manufacturing facility. The manufacturing facility acts also as collection centre. In this network, routing decisions were assigning customers to vehicles, and establishing the route for each vehicle such that vehicles start at the manufacturing facility, visit the assigned customers for delivery and pick up, and end the trip at the established manufacturing facility. The model was solved using a heuristic method. The heuristic method was based on a sequential route building algorithm and an upper bound approach. When compared with optimum, deviations of solutions of the proposed heuristic ranged between 5.9 % and 10.4%, with an average of 8.2%.

7.1.3 VRPSPD-TW: Modeling and Solution Methodology

The vehicle routing problem with simultaneous pickup and delivery and time windows (VRPSPD-TW) is another type of IFRL networks studied in this work. Research on this problem is recent and limited. In this we propose a MILP model to represent the problem. The presented model is a sub-model of the comprehensive MILP model discussed in Section 4.1.1, and presented in Chapter 3. Since the VRPSPD-TW is NP-hard in nature, we developed a solution methodology to solve practical size problems that have real life applications. The solution methodology consisted of a heuristic method and a simulated annealing (SA) approach. The heuristic method is based on a sequential route building method. Solutions from the heuristic method were used as an initial solution for the SA approach. When compared with optimum, the heuristic method obtained solutions with deviations that ranged from 0% to 16.9% and an average of 6.2%. When these solutions were improved using the SA approach, the resulting solutions were similar to the optimum. Computational times for the heuristic method and the SA approach were reasonable.

We tried to obtain lower bounds using Lagrangian relaxation. We obtained lower bounds for small size problems and a few of the large size problems. Results from the proposed methodology showed a gap in the range of 0.5% to 8.1% from Lagrangian lower bounds. The Lagrangian lower bounds deviated from optimum in the range of 0.5% to 8.1%.

7.1.4 MDVRPSPD-TW: Modeling and Solution Methodology

In this work, we studied the multi-depot vehicle routing problem within the IFRL context. The problem presented in this research is the multi-depot vehicle routing problem with simultaneous pickup and delivery and time windows (MDVRPSPD-TW).

We designed a network that considers the assignment of customers to manufacturing facilities (depots), the assignment of vehicles to depots, assignment of customers to vehicles, and routing of vehicles within customers' time windows constraints. We presented a MILP model to represent the problem. The model is a sub-model of the comprehensive MILP model presented in Chapter 3. The presented problem is NP-hard in Nature. Therefore, we developed a solution method to solve practical size problems. The developed solution method consists of a 2-phase heuristic technique followed by a simulated annealing approach. We used the MILP model to obtain optimum solutions for small sized problems. The results obtained using the 2-phase heuristic and the SA approach deviated from optimum or lower bound in the range of 0% to 3.7% with an average of 0.86%. The improvement of the SA procedure over the 2-pahse heuristic was in the range of 0% to 22.3%. Computational times for large size problems were reasonable.

7.2 Future Research

In this work we studied different IFRL networks. We provided solution methods to solve the networks. Some of the solution methods included heuristic approaches, others included a heuristic approach and a SA procedure. Potential future research may include extensions to the IFRL networks presented in this work. Moreover, future research may include additional solution methods.

7.2.1 Extension for IFRL Networks

The location-routing IFRL network we presented in this work, considered choosing one location to establish a disassembly plant or a manufacturing facility. We plan to extend the presented model by considering establishing more than one disassembly plant or manufacturing facility. Another extension is to establish a number of manufacturing facilities and disassembly plants within the same network. Capacity constraints for the manufacturing facility and the disassembly plant are a practical extension for the problem as well. Time windows constraints may be added to the routing constraints of the problem.

The VRPSPD-TW presented in this work may be extended by using heterogeneous vehicle fleets, and multi-commodity models. These extensions may also be considered for the location-routing problem.

Future research for the MDVRPSPD-TW includes considering heterogeneous vehicle fleets, and multi-commodity models. Another extension for future research is limiting the number of vehicles available at each depot, and setting capacity limits on pickup amounts at each depot.

7.2.2 Extension for Solution Methods

In this work, we used a number of heuristic methods based on sequential route building techniques to solve the different networks presented. Heuristics based on savings methods have been used successfully for similar problems. Therefore, heuristics that are based on saving algorithms may be considered for solving the networks presented in this research.

We also used SA to solve some the IFRL networks presented in this research. Other intelligent optimization techniques may be employed and compared with the SA approach we used. Examples of these techniques include ant systems, tabu search, and genetic algorithms. Another extension is to develop solution method(s) to solve the comprehensive mathematical model presented in Chapter 3. Since the IFRL networks studied in this work are NP-hard in nature, obtaining lower bounds to examine the efficiency of solution heuristics is preferable. We presented a Lagrangian relaxation method in this work. The presented method could be further refined and studied to obtain more powerful lower bounds for a larger number of problems.

BIBLIOGRAPHY

1- Akcali E., Cetinkaya S. and Uster H., “Network Design for Reverse and Closed-Loop Supply Chains: An Annotated Bibliography of Models and Solution Approaches”, 2009, Networks, Vol. 53, pp 231-248.

2- Alshamrani A., Mathur A. K. and Ballou H. R., “Reverse logistics: Managing Returns on a Delivery Route”, 2004, Technical Memorandum, Department of Operations, Case Western Reserve University, Cleveland, Ohio 44106, 2003.

3- Alshamrani A., Mathur A. K. and Ballou H. R., “Reverse Logistics: Simultaneous Design of Delivery Routes and Returns Strategies”, 2007, Computers and Operations Research, Vol. 39, pp 595-619.

4- Angelelli E. and Mansini R., “The Vehicle Routing Problem with Time Windows and Simultaneous Pick-up and Delivery”, 2002, In: A. Klose M.G. Speranza and Van Wassenhove L. N. (ed.) Quantitative Approaches to Distribution Logistics and Supply Chain Management, (pp 249-267), Springer-Verlag, Berlin Heidelberg 2002.

5- Barreto S., Ferreira C., Paixa J., Santos B. S., “Using Clustering Analysis in a Capacitated Location-Routing Problem”, 2007, European Journal of Operational Research, Vol. 179, pp 968–977.

6- Bianchessi N., and Righini G., “Heuristic Algorithms for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery”, 2007, Computers and Operations Research, Vol. 34, pp 578-594.

7- Bräysy O. and Gendreau M. (a), “Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms”, 2005, Transportation Science, Vol. 39, pp 104–118.

8- Bräysy O. and Gendreau M. (b), “Vehicle Routing Problem with Time Windows, Part II: Metaheuristics”, 2005, Transportation Science, Vol. 39, pp 119-139.

9- Cao E. and Lai M., “An Improved Differential Evolution Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pick-up Service”, 2007, Third International Conference on Natural Computation (ICNC 2007).

10- Casco D., Golden B., and Wasil E., “Vehicle Routing with Backhauls: Models, Algorithms, and Case Studies”, 1988, In: Golden B. L., Assad A. A. (editors) Vehicle Routing: Methods and Studies, pp 127–147. Elsevier, Amsterdam.

11- Catay B., “A New Saving-based Ant Algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery”, 2010, Expert Systems with Applications, Vol. 37, pp 6809–6817.

12- Chen P., Huang H. and Dong X., “An Ant Colony System Based Heuristic Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pickup”, 2007, Second IEEE Conference on Industrial Electronics and Applications, pp 136-141.

13- Chen J.F. and Wu T.H., “Vehicle Routing Problem with Simultaneous Deliveries and Pickups”, 2006, Journal of Operational Research Society, Vol. 57, pp 579-587.

14- Chien T. W., “Heuristic Procedures for Practical Sized-Uncapacitated Location-Capacitated Routing Problems”, 1993, Decision Sciences, Vol. 24, pp 995-1021.

15- Chiu H. N., Lee Y. S., Chang J. H., "Two approaches to Solving the Multi-depot Vehicle Routing Problem with Time Windows in a Time-based Logistics Environment", 2006, Production Planning and Control, Vol. 17, pp 480-493.

16- Chun-Hua L., Hong Z. and Jian Z. "Vehicle Routing Problem with Time Windows and Simultaneous Pickups and Deliveries". 16th International Conference on Industrial Engineering and Engineering Management 2009, pp 685-689.

17- Chunyu R., Zhendong S., xiaobo W., "Study on Single and Mixed Fleet Strategy for Multi-depot Vehicle Routing Problem with Backhauls", 2009, Proceedings of the International Conference on Computational Intelligence and Natural Computing, pp 425-428.

18- Cordeau J-F, Laporte G., Mercier A., "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows",2001, Journal of the Operational Research Society, Vol. 52, pp 928-936.

19- Crispim J. and Brandao J., "Metaheuristics Applied to Mixed and Simultaneous Extensions of Vehicle Routing Problems with Backhauls", 2005, Journal of the Operational Research Society, Vol. 56, pp 1296–1302.

20- DeCroix A. G. and Zipkin H. P., "Inventory Management for an Assembly System with Product or Component Returns", 2005, Management Science, Vol. 51, pp 1250-1265.

21- Dell'Amico M., Righini G., Salani M., "A Branch-and-Price Approach to the Vehicle Routing Problem with Simultaneous Distribution and Collection", 2006, Transportation Science, Vol. 40, pp 235–247.

22- Demirel N. O. and Gökçen H., “A Mixed Integer Programming Model for Remanufacturing in Reverse Logistics Environment”, 2008, International Journal of Advanced Manufacturing Technology, Vol. 39, pp 1197–1206.

23- Desrochers M., Desrosiers, J., Solomon N., “A new Optimization Algorithm for the Vehicle Routing Problem with Time Windows”, 1992, Operations Research, Vol. 40, pp 342-354.

24- Dethloff J., “Vehicle Routing and Reverse Logistics: The Vehicle Routing Problem with Simultaneous Delivery and Pickup”, 2001, OR Spektrum, Vol. 23, pp 79-96.

25- Dethloff J., “Relation Between Vehicle Routing Problems: An Insertion Heuristic for the Vehicle Routing Problem with Simultaneous Delivery and Pick-Up Applied to the Vehicle Routing Problem with Backhauls”, 2002, Journal of the Operational Research Society, Vol. 53, pp 115-118.

26- Ding S., “Logistics Network Design Optimization Based on Differential Evolution Algorithm”, 2010, International Conference on Logistics Systems and Intelligent Management, pp 1064-1068.

27- Dondo R. and Cerda J., “A Cluster-based Optimization Approach for the Multi-depot Heterogeneous Fleet Vehicle routing Problem with Time Windows”, 2007, European Journal of Operational Research, Vol. 176, pp 1478-1507.

28- Dondo R. G., Cerda J., “A Hybrid Local Improvement Algorithm for Large-Scale Multi-Depot Vehicle Routing Problems with Time Windows”, 2009, Computers and Chemical Engineering, Vol. 33, pp 513-530.

29- Easwaran G. and Uster, H., “A Closed-Loop Supply Chain Network Design Problem with Integrated Forward and Reverse Channel Decisions”, 2010, IIE Transactions Vol. 42, pp 779–792.

30- Fleischmann M., Bloemhof-Ruwaard J. M., Dekker R., van der Laan E., van Nunen J. A.E.E., Van Wassenhove L. N., “Quantitative Models for Reverse Logistics: A Review”, 1997, European Journal of Operational Research, Vol. 103, pp 1-17.

31- Fleischmann M., “Quantitative Models for Reverse Logistics, 1st Edition. Lecture Notes in Economics and Mathematical Systems”, 2001, Springer, Berlin.

32- Gajpal Y. and Abad P., “Saving-based Algorithms for Vehicle Routing Problem with Simultaneous Pickup and Delivery”, 2010, Journal of the Operational Research Society, Vol. 61, pp 1498-1509.

33- Ginter M. P. and Starling M. J., “Reverse Distribution Channels for Recycling”, 1978, California Management Science, Vol. 20, pp 72-82.

34- Gungor A. and Gupta M. S., “Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey”, 1999, Computers and Industrial Engineering, Vol. 36, pp 811–853.

35- Gupta S. M. and Taleb K. N., “Scheduling Disassembly”, 1994, International Journal of Production Research, Vol. 32, pp 1857-1866.

36- Hadjar A. and Soumis F., “Dynamic Window Reduction for the Multiple Depot Vehicle Scheduling Problem with Time Windows”, 2009, Computers and Operations Research, Vol. 36, pp 2160-2172.

37- Haijun M., Fei Q. U. and Xuhong L., “Integration of Forward and Reverse Logistics at the Vehicle Routing Level”, 2007, International Conference on Transportation Engineering 2007 (ICTE 2007), pp 3518-3523.

38- Irnich S., “A Multi-Depot Pickup and Delivery Problem with a Single Hub and Heterogeneous Vehicles”, 2000, European Journal of Operational Research, Vol.122, pp 310 -328.

39- Jayaraman V., Guide, J. and Drivastava, R., “A Closed Loop Logistics Model for Remanufacturing”, 1999, Journal of Operational Research Society, Vol. 50, pp 497–508.

40- Jiang W., Zhang Y. and Xie J., “A Particle Swarm Optimization Algorithm with Crossover for Vehicle Routing Problem with Time Windows”, 2009, IEEE Symposium on Computational Intelligence in Scheduling, CI-Sched 2009 - Proceedings, pp 103-106.

41- Jovane F., Alting L., Armillotta A., Eversheim W., Feldmann K., Seliger G., “A Key Issue in Product Life Cycle: Disassembly”, 1993, Annals of CIRP, Vol. 42, pp 640-672.

42- Kallehauge B., Larsen J., Madsen Oli G. B., Solomon M. M., “Vehicle Routing Problems with Time Windows”, 2005, In: Desaulniers G., Desrosiers J., Solomon M. M. (ed.), Column Generation, (pp 67-98), Springer, 2005.

43- Kallehauge B., “Formulations And Exact Algorithms For The Vehicle Routing Problem With Time Windows”, 2008, Computers & Operations Research, Vol. 35 pp 2307 – 2330.

44- Kiesmuk P. G. and van der Laan, “An Inventory Model with Dependent Product Demands and Returns”, 2001, International Journal of Production Economics, Vol. 72, pp 73-87.

45- Kohl N. and Madsen O. B. G., “An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation”, 1997, Operations Research, Vol. 45, pp 395-406.

46- Krikke H.R., van Harten A. and Schuur, P.C., “Business Case Oce: Reverse Logistic Network Re-Design for Copiers”, 1999, OR Spektrum, Vol. 21, pp 381–409.

47- Lenstra J. and Kan A. “Complexity of Vehicle Routing and Scheduling Problems”, 1981, Networks, Vol. 11, pp 221-227.

48- Li H. and Lim A. “Local Search with Annealing-Like Restarts to Solve the VRPTW”, 2003, European Journal of Operational Research, Vol. 150, pp 115–127.

49- Lin C. K. Y., Chow C. K. and Chen, A. A., “Location-Routing-Loading Problem for Bill Delivery Services”, 2002, Computers and Industrial Engineering, Vol. 43, pp 2-25.

50- Lin S.-W., Ying K.-C., Lee Z.-J., Chen H.-S., “Vehicle Routing Problems with Time Windows Using Simulated Annealing”, 2006, IEEE International Conference on Systems, Man, and Cybernetics, pp 645-650.

51- Lu Z., Bostel N., “A Facility Location Model for Logistics Systems Including Reverse Flows: The Case of Remanufacturing Activities”, 2007, Computers & Operations Research, Vol. 34, pp 299–323.

52- Maranzana F. E., “On the Location of Supply Points to Minimize Transport Costs”, 1964, Operational Research Quarterly, Vol. 15, pp 261-270.

53- Marin, A. and Pelegrin, B. “The Return Plant Location Problem: Modeling and Resolution”, 1998, European Journal of Operational Research, Vol. 104, pp 375–392.

54- Melechovsky J., Prins C. and Calvor W., “A Metaheuristic to Solve a Location-Routing Problem with Non-Linear Costs”, 2005, Journal of Heuristics, Vol. 11, pp 375-391.

55- Min H., “The Multiple Vehicle Routing Problem with Simultaneous Delivery and Pickup Points”, 1989, Transportation Research-A 23A: 377-386

56- Min H., Jayaranam V. and Srivatava R., “Combined Location-Routing Problems: A Synthesis and Future Research Directions”, 1998, European Journal of Operational Research, Vol. 108, pp 1–15.

57- Ming-Yao Q., Li-Xin M., Le Z., Hua-Yu X., “A New Tabu Search Heuristic Algorithm for the Vehicle Routing Problem with Time Windows”, 2008, International Conference on Management Science and Engineering (ICMSE), pp 1648-1653.

58- Mingyong, L. and Erbao, C., “An Improved Differential Evolution Algorithm for Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Window”, 2010, Engineering Applications of Artificial Intelligence, Vol. 23, pp. 188-195.

59- Montane F., Galvao R., "A Tabu Search Algorithm for the Vehicle Routing Problems with Simultaneous Pick-Up and Delivery Service", 2006, Computers and Operations Research. Vol. 33, pp 595-619.

60- Nagi G., Salhi S., "Location-Routing: Issues, Models and Methods", 2007, European Journal of Operational Research, Vol. 177, pp 649-672.

61- Nagi G., Salhi S., "Heuristic Algorithms for Single and Multiple Depot Vehicle Routing Problems with Pickups and Deliveries", 2005, European Journal of Operational Research, Vol. 162, pp 126-141.

62- de Oliveira H., Vasconcelos G., Alvarenga G., Mesquita R., de Souza M., "A Robust Method for the VRPTW with Multi-Start Simulated Annealing and Statistical Analysis", 2007, Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007), pp 198-205.

63- Pishvaee M. S., Farahani R. Z., Dullaert W., "A Memetic Algorithm for Bi-objective Integrated Forward/Reverse Logistics Network Design", 2010, Computers & Operations Research, Vol. 37, pp 1100–1112.

64- Polacek, M., Hartl R. F., Doerner K., "A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows", 2001, Journal of Heuristics, Vol. 10, pp 613-627.

65- Salema M. I., G., Barbosa-Povoa A.P. and Novais, A.Q., "An Optimization Model for the Design of a Capacitated Multi-Product Reverse Logistics Network with Uncertainty", 2006, European Journal of Operational Research, (Special Issue on Advances in Location Analysis).

66- Salhi S. and Nagy G., (a) “A Cluster Insertion Heuristic for Single and Multiple Depot Vehicle Routing Problems with Backhauling”, 1999, Journal of the Operational Research Society, Vol. 50, pp 1034-1042.

67- Salhi S. and Nagi G., (b) “Consistency and Robustness in Location-Routing”, 1999, Studies in Locational Analysis, Issue 13, pp 3-19.

68- Salhi S. and Rand G. K., “The Effect of Ignoring Routes When Locating Depots”, 1989, European Journal of Operational Research, Vol. 39, pp 150-156.

69- Solomon, M. “Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints”, 1987, Operations Research, Vol. 35, pp 254-265.

70- Spengler Th., Puchert H., Penkuhn T. and Rentz, O., “Environmental Integrated Production and Recycling Management”, 1997, European Journal of Operational Research, Vol. 97, pp 308–326.

71- Srivastava R., “Alternate Solution Procedures for the Location-Routing Problem”, 1993, OMEGA International Journal of Management Science, Vol. 21, pp 497-506.

72- Subramanian A., Drummond L. M. A., Bentes C., Ochi L. S., Farias R., “A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery”, 2010, Computers & Operations Research, Vol. 37, pp 1899–1911.

73- Subramanian A., Uchoa E., Ochi L. S., “New Lower Bounds for the Vehicle Routing Problem with Simultaneous Pickup and Delivery”, 2010, Experimental Algorithms, Proceedings of the 9th International Symposium, pp 276-87.

74- Taillard E., Badeau P., Gendreau M., Guertin F., Potvin J., “A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows”, 1997, Transportation Science, Vol. 31, pp 170–186.

75- Tuzun B. and Burke L. I., “A Two-Phase Tabu Search Approach to the Location Routing Problem”, 1999, European Journal of Operational Research, Vol. 116, pp 87-99.

76- Vandermerwe S. and Oliff M. D., “Customers Drive Corporations Green”, 1990, Long Range Planning, Vol. 23, pp 10-16.

77- Von Boventer E., “The Relationship Between Transportation Costs and Location Rent in Transportation Problems”, 1961, Journal of Regional Science, Vol. 3, pp 27-40.

78- Wang H., Hsu. H., “A Closed-Loop Logistic Model with a Spanning-Tree Based Genetic Algorithm”, 2010, Computers & Operations Research, Vol. 37, pp 376-389.

79- Wang X., Sun J., Ren C., “Study On Hybrid Heuristic Algorithm For Multi-Depot Vehicle Routing Problem with Hybrid Picking-Delivery Strategy”, 2009, Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, pp 1451-1456.

80- Wassan N., Wassan A. and Nagy G., “A Reactive Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pickups and Deliveries”, 2008, Journal of Combinatorial Optimization, Vol. 15, pp 368-386.

81- Wenfeng W., Zuntong W. and Fei Q., “An Improved Genetic Algorithm for Vehicle Routing Problem with Time-Window”, 2008, Proceedings - International

Symposium on Computer Science and Computational Technology, ISCSCT, Vol. 1, pp 189-194.

82- Yun Z., Guorui Z., “A Genetic Algorithm for Vehicle Routing Problem with Forward and Reverse Logistics”, 2008, 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp 1-4.

83- Zachariadis E. E. and Kiranoudis C. T., “A Local Search Metaheuristic Algorithm for the Vehicle Routing Problem with Simultaneous Pick-Ups and Deliveries”, 2011, Expert Systems with Applications, Vol. 38, pp 2717–2726.

84- Zachariadis E. E., Tarantilis C. D., Kiranoudis C. T., “An Adaptive Memory Methodology for the Vehicle Routing Problem with Simultaneous Pick-Ups and Deliveries”, 2010, European Journal of Operational Research, Vol. 202, pp 401–411.

85- Zhen T. and Zhang Q., “A Hybrid Metaheuristic Algorithm for the Multi-depot Vehicle Routing Problem with Time Windows”, 2009, International Conference on Networks Security, Wireless Communications and Trusted Computing, pp 798-801.

APPENDIX A

The Matlab code used to obtain the results for the test problems throughout the thesis is available at Concordia University Libraries.