Routing of the Medical Waste Collection Vehicles for the Istanbul Metropolitan Municipality

by

Müge Güçlü

A Thesis Submitted to the

Graduate School of Engineering

in Partial Fulfillment of the Requirements for

the Degree of

Master of Science

in

Industrial Engineering

Koç University

September 2012

Koc University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Müge Güçlü

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

____

Asst. Prof. Deniz Aksen (Advisor)

____

Asst. Prof. Onur Kaya

____

Prof. Necati Aras

Date: ____________________

…to my parents

in love and gratitude…

…may the force be with you…

ABSTRACT

Municipal solid waste production has increased in recent years parallel with the economic growth and technological improvements. This increase generated a need for efficient waste management. Since waste management has a tremendous share of the municipal's budget, efficiency in waste collection, routing, recycling and disposal is a must to manage resources.

Municipals are responsible for medical waste collection also. Since medical waste can convey contagious diseases, the medical waste collection and routing must be done more carefully and on the time. For public health medical waste should be collected and disposed of periodically and precisely.

In this thesis, we focus on waste collection in Istanbul Municipality. Istanbul Municipality has two disposal facilities, one is placed in Anatolian side of Istanbul and the other is placed on the European side. The city is divided into two regions therefore two sub-problems are generated to be solved. Both the European side and the Anatolian side of Istanbul are examined and two different routing schemes are generated by using a hashing inserted reactive TABU Search (RTSH) with two different initial solution generation methods.

ÖZETÇE

Kentsel katı atık üretimi son dönemlerde ekonomik büyüme ve teknolojik gelişmelere de parallel olarak artış gösterdi. Bu artış atıkların verimli yönetimi konusunda bir ihtiyaç doğurdu. Atık yönetimi belediye bütçesinin büyük bir kısmına sahip olduğundan ötürü, kaynakların doğru kullanımı açısından atık toplama, taşıma, geri-dönüşüm ve yakılma işlemlerinin verimli yapılması zorunlu hale gelmiştir.

Belediyeler aynı zamanda tıbbi atıkların toplanmasından da sorumlu. Tıbbi atıkların salgın hastalıklara neden olabileceginden dolayı, tıbbi atık toplama ve taşıma işlemleri dikkatli ve zamanında yapılmalıdır. Halk sağlığı açısından bu atıkların toplanma ve yakılma işlemleri periyodik ve düzgün olarak yapılmalıdır.

Bu tez ile İstanbul Büyükşehir Belediyesi tıbbi atık toplama ve bu atıkların yönetimine odaklanılmıştır. İstanbul Büyükşehir Belediyesi bir tanesi Anadolu yakasında, diğeri Avrupa yakasında olmak üzere iki adet tıbbi atık yakma merkezine sahiptir. Şehir Anadolu ve Avrupa olmak üzere ikiye ayrılarak, iki adet alt problem yaratılmış ve çözümlenmiştir. Anadolu ve Avrupa yakaları için TABU arama metodu kullanılarak iki farklı araç rotalama sistemi geliştirilmiştir.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Asst. Prof. Deniz Aksen for the continuous support of my graduate study, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me to accomplish this research project. It has been privilege for me to work two years with him.

I would like to thank the members of my thesis committee Asst. Prof. Onur Kaya and Prof. Necati Aras for critical reading of this thesis and for their valuable comments.

I am grateful to Filiz Sayın, Buşra Ürek, Abdullah Bilgin, Umut Arıtürk, Görkem Sarıyer, Nihal Bahtiyar, Duygu Tutal, Ezel Ezgi Budak, Ferda Tangüner and Özlem Yalçın not only for being wonderful friends, but also for being a family to me. Without their presence and close friendship, the last three years would not be bearable. I would also like to thank all my friends at Koç University for all the fun and good times we shared together.

Finally, I owe special thanks to my brother Fatih, for being my role model, to my sister-in-law Hilal, for the fun she brings to our life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Environmental issues are placed in today's top topics. Long since pollution has become an emerging problem that affects human health. Developments in technology results increment on waste, but this development is also used in the solution of pollution problems.

Municipal solid waste production has been increased in recent years parallel with the economic growth and technological improvements. This increase generated a need in efficient waste management. Since waste management has a tremendous share of the municipal's budget, efficiency in waste collection, routing, recycling and disposal is a must to manage resources.

Municipals are also responsible for medical waste collection. Since medical waste can convey contagious diseases, the medical waste collection must be done more carefully and on the time. For public health, medical waste should be collected and disposed of periodically and precisely.

Today, many developed countries have legal provisions with regard to the proper management of health-care waste. Turkey, like most of the economically developing countries, has not emphasized the proper handling and disposal of health-care waste.

In this thesis, we focus on waste collection in Istanbul Municipality. Istanbul Municipality has two disposal facilities, one is placed on the Anatolian side of Istanbul and the other is placed on the European side. The city is divided into two regions;

therefore two sub-problems are to be solved. Both for the European side and the Anatolian side of Istanbul, two different routing schemes are generated.

The vehicle routing problem is a classical problem in operations research, where the objective is to design least cost routes for a fleet of identical capacitated vehicles to service geographically scattered customers. VRP is characterized as follows; from a starting point (depot) goods must be picked-up and delivered in given quantities to the given customers in a sequence. A number of vehicles are available for the transportation of the goods, each having a certain capacity. Every vehicle that is used in the solution must cover a route, starting from and ending at the depot, on which goods are delivered to one or more customers that are given.

The problem is to determine the routes and the allocation of the customers among the routes, the sequence in which the customers shall be visited on a route, and which vehicle that shall cover a route. The objective is to find a solution which minimizes the total transportation costs in general. Furthermore, the solution must satisfy the restrictions that every customer is visited exactly once, where at the same time satisfying the demanded quantities and the total demand on every route must be within the vehicle's capacity.

VRP considers the cost of driving from any point to any other point and the fixed cost of using a vehicle as transportation costs. These costs are not necessarily identical in the two directions between two given points; costs can be asymmetric since the shortest distances between two points is different for each direction.

In this application main objective is always to minimize total cost (minimize total distance covered by vehicles), with this main objective total number of vehicles used and total CO_2 emission are also be decreased.

The organization of this thesis is as follows. The next part contains relevant studies in the literature. Chapter 3 focuses on the problem definition and concept of this real-life problem. In Chapter 4 we characterize the optimal solution strategy for this problem and in Chapter 5 we give computational experiments and the numerical results. Finally in Chapter 6 we give a general summary of the thesis and provide some directions for future research.

Chapter 2

LITERATURE REVIEW

2.1. Introduction

The literature on Vehicle Routing Problems is very extensive. The studies are based on varying approaches and comprise many different disciplines. This multi-disciplinary research is ranging from Operations Research, through Environmental Engineering and other mid-disciplines to Computer Sciences. Although there are various researches in VRP, there exist rare sources on the specific type of VRP which is Vehicle Routing Problem with intermediate facilities (VRP-IF). And there exist very few sources on a sub problem of VRP-IF which is Asymmetric Capacitated Vehicle Routing Problem with intermediate facilities (ACVRP-IF).

This study tackles a vehicle routing problem with intermediate facilities that is specified for waste management. The problem also considers the total duration of the routes as time window. In order to handle this real-life problem, generation of accurate solutions is important, for accurate solutions traveling distances must be considered as asymmetric. In the literature there exists many studies for this specific problem; nevertheless an extensive literature search shows that a significant number of studies conducted cannot cover asymmetric traveling distances, intermediate facility and time windows in one problem. In this thesis, regarding to intermediate facility and asymmetric distances we suggest a modified Clarke & Wright algorithm as to generate starting solution, and modified move operators for Local Post Optimization (LPO) stage. Further, we present a new ensemble methodology. The relevant streams of literature for this problem are enumerated below:

- ✓ Vehicle Routing Problem with extensions
- ✓ Waste Management studies
- ✓ Related Tabu Search and Hybrid Algorithms

2.1.1. Vehicle Routing Problem with Extensions

The Vehicle Routing Problems with intermediate facilities belong to the most intensively studied problems, Vehicle Routing Problems, in combinatorial optimization. A comprehensive survey on VRP with all sides and versions of the problem gives the brief history of the problem (Toth & Vigo, 2002) . More recent survey of (Golden, Raghavan, & Wasil, 2008) is another study that gives insight on the VRP and solution methods evolution.

An extension for Periodic Vehicle Routing Problem is studied by (Angelelli & Speranza, 2002). In this study Intermediate Facilities are considered, each vehicle has to visit intermediate facilities. Intermediate facilities are assumed as the warehouses and in a delivery problem each vehicle must visit first the warehouse to fill the vehicle then deliver those goods to each customer regarding to each point's demand. In collection problem, different from the delivery problem, vehicles start the route empty from depot and collect goods from each node then before returning back to depot each vehicle visit the intermediate facility to emptying.

(Gröer, Golden, & Wasil, 2010) includes a literature review for the local search heuristics specified on vehicle routing problem. The study provides an open source software library of heuristics for generating solution instances of VRP. They describe several applications done by using the library then provide computational results for benchmark problems.

2.1.2. Waste Management Studies

Waste collection routing problems are divided into three main categories. (Golden, Assad, & Wasil, 2001) First, the commercial collection contemplates waste collection from business and organizations. Second is the residential collection of waste, from households. And the third is named Rollon-Rolloff which involves large containers, at the disposal facilities filled containers are replaced by empty ones.(Bodin, Mingozzi, Baldacci, & Ball, 2000)

Waste management is studied very extensively, in a case study from Hanoi, Vietnam, a heuristic procedure is designed to solve the waste collection problem in Hanoi. An improvement of 4.6% is stated to be achieved when compared with the current situation. (Tung & Pinnoi, 2000)

Another case study for waste management studied for Italy and Belgium, in this study a model proposed that fits three different waste collection systems to estimate the operational cost of each system. (Angelelli & Speranza, 2002)

Waste collection is done by private organizations in many countries, since the organization has the motivation to make profit; the operational costs have to be minimized to maximize the profit. A system developed called WasteRoute by (Sahoo, Kim, Kim, Krass, & Popov, 2005) to reduce operational costs, in this study the example result as reducing total number of routes from 10 to 9 which means 1 less vehicle, its fixed and variable costs, and less fuel usage therefore less CO_2 emission.

A scenario based waste management problem designed previously by (Alagöz & Kocasoy, 2008). In this study they focused on health waste collection in İstanbul, and they used a commercial vehicle routing package to consider a number of scenarios relating to the type of facility for waste disposal.

A set of waste disposal facilities assumed as intermediate facilities in (Benjamin & Beasley, 2010), in this VRP-IF study drivers' rest time and driver's working hours are considered also. Vehicles start route empty from the depot and collect waste from customers then after the last customer each vehicle visits one disposal facility to drop the waste collected. Tabu Search and Variable Neighborhood Search is applied after an initial solution generated. Also a hybrid of TS and VNS is applied, each variable neighborhood is searched by TS logic.

(El-Salam, 2010) focused on hospital waste management system in El-Beheria, Egypt. In this study eight randomly selected hospitals are taken and total daily waste generation rate is determined. Through this information an appropriate waste management system is proposed.

In (Bdour, Altrabsheh, Hadadin, & Al-Shareif, 2007; Coker, Sangodoyin, Sridhar, Booth, Olomolaiye, & Hammond, 2009) stated that improper clinical waste management is increasing both health hazards and environmental pollution. And this improper management of clinical waste directly impacts healthcare staffs, patients and hospitals environment. Diseases like cholera, dysentery, skin infection, infectious hepatitis can spread epidemic way due to mismanagement of clinical solid waste. For this reason, determining appropriate methods for safe management of clinical waste is urgent (Sawalem, Selic, & Herbell, 2009; Tamplin, Davidson, Powis, & O'Leary, 2005).

(Kim, Kim, & Sahoo, 2006) focused on a real life waste collection vehicle routing problem with the consideration of multiple disposal trips and drivers' lunch time. Solomon's insertion algorithm used in this study with extensions. A cluster-based waste collection vehicle routing problem with time windows algorithm is developed to ensure route compactness and workload balancing.

(Shih & Chang, 2001) discussed the waste management system in Taiwan. The solution is proposed in two phases; the first phase solves standard VRP and the second phase uses a mixed integer programming method to assign routes to particular days of the week. A user-friendly visual basic application is proposed to this periodic vehicle routing problem. An application is proposed for Tainan City having 348 hospitals.

(Nuortio, Kytojoki, Niska, & Braysy, 2006) focused on the waste management system in Eastern Finland. In the solution phase they used variable neighborhood thresholding algorithm. The study demonstrates that significant cost reductions can be obtained compared with the current practice.

(Belien, De Boeck, & Van Ackere, 2011) prepared a literature review on municipal solid waste collection. This study includes the waste problem types in the literature and an overview on solution algorithms for each type of problem.

(Benjamin & Beasley, 2012) extended the scope of their previous study (Benjamin & Beasley, 2010). The study considers multiple disposal facilities and in addition their positioning. The problem considered in this study contains single depot, 2,092 customers and multiple disposal facilities. The driver lunch times are also considered. The generated a disposal facility positioning procedure (DFP) and used a hybrid metaheuristics structure based on TS and VNS and the combination of these.

(Buhrkal, Larsen, & Ropke, 2012) studied waste collection vehicle routing problem, considering time windows and drivers rest periods. An adaptive large neighborhood search is proposed and an application is provided to a Danish garbage company.

2.1.3. Related Tabu Search and Hybrid Algorithms

This part introduces sample solution methodologies for VRP, CVRP, ACVRP and CVRP-IF. Finally it represents hybrid algorithms that combine probabilistic and deterministic heuristics together.

A waste collection problem in two regions of Eastern Finland is considered in (Nuortio, Kytojoki, Niska, & Braysy, 2006). The problem includes time windows and they solved the problem using Guided Variable Neighborhood Thresholding (GVNT). GVNT is proposed by (Kytojoki, Nuortio, Niska, Braysy, & Gendreau, 2007) this method includes a number of implementation guidelines for designing efficient solution methods for very large-scale practical routing problems.

The most relevant study with our study is (Du & He, 2012). In this study large-scale VRP is solved by using a hybrid algorithm that combines nearest neighbor search with tabu search algorithm. The initial solution for the Tabu search is constructed by nearest neighbor search in the first phase, then Tabu search is utilized to optimize the intra-route and inter-route in the second phase. The combined algorithm is applied to a real life problem in China, a tobacco seller problem (in the same manner of TSP) with 6772 tobacco customers. Homberger's 400 customer dataset is taken as a benchmark dataset. (Homberger & Gehring, 2005)

A hybrid algorithm of simulated annealing and tabu search algorithm that specializes on vehicle routing problem is designed by (Osman, 1993). In this study tabu search algorithm properties are inserted into simulated annealing. Tabu list component of Tabu search algorithm is used in simulated annealing cooling schedule. (Sigauke, 1994) modified Osman's simulated annealing and Tabu search combination algorithm and applied on a CVRP. The objective of the study is to determine feasible routing patterns for an organization in the alcoholic beverage industry. The total traveling time results are sharply better than the manual schedule's result.

Another relevant study is proposed by (Wassan, 2006). In this study a reactive Tabu search is proposed for capacitated VRP problem. The initial solution is generated by using Clarke and Wright savings algorithm then improved by some improvement heuristics. Tabu search is the second phase of the solution structure, by using the constructed solution Tabu algorithm improves the solution with 1-exchange and 2-exchange neighborhood search procedures. The whole algorithm aims to start Tabu search with a high quality solution and then improve it within given Tabu conditions and termination criteria. Since the algorithm gives reaction to the infeasible solutions by increasing and decreasing the penalty rate for infeasible solutions, the algorithm is a reactive search algorithm.

Chapter 3

MODELING APPROACH

3.1. Introduction

In this chapter, we introduce our problem with all sides and propose our modeling approach that is based on the real-life constraints.

3.2. Problem Definition

We model the problem as a classical single-depot capacitated vehicle routing problem with a maximum tour duration constraint. In addition, we also enforce each vehicle route to pass through a landfill facility (LF) just prior to returning to the depot and also asymmetric distances. This is, each vehicle must stop by the LF before coming back to the depot. In this special VRP with a maximum tour duration constraint, we resort to the Lifted Miller-Tucker-Zemlin (Lifted-MTZ) inequalities to enforce subtour elimination in the model.

3.2.1. Index sets of (MTZ)-VRP

We present the ACVRP-IF model formulation to minimize the total cost (in a manner total distance) by determining the optimal number of vehicles and routes of these vehicles. The model parameters and decision variables are defined as follows;

There are three sets in the model:

$\boldsymbol{IC} = \{\mathbf{1},\ldots,\boldsymbol{n}\}$: the set of n supply nodes (hospitals and clinics),

$\boldsymbol{IY} = \{\boldsymbol{n}+\mathbf{1},\ldots,\boldsymbol{n}+\boldsymbol{k}\}$: the set of d landfill facilities (LFs),

$\boldsymbol{I} = \boldsymbol{IC} \cup \boldsymbol{IY} \cup \{\mathbf{0}\}$ **:** the set of all nodes inclusive of the depot.

We assume that there are as many as k landfill facilities at either of which each vehicle route must stop by before returning to the depot.

3.2.2. Parameters of (MTZ)-VRP

The parameters in the model are as follows.

d : number of vehicles available for use.

k : number of LFs.

v : operating cost per vehicle.

c_{ij} : vehicle traveling cost from node i to node j (i and $j \in \boldsymbol{I}$).

t_{ij} : vehicle traveling time from node i to node j (i and $j \in \boldsymbol{I}$).

s_i : service time at node i (i and $j \in \boldsymbol{IC} \cup \boldsymbol{ID}$).

TT : maximum duration of a vehicle trip which ends at the depot after stopping by the LF.

M : the Big-M number. It is set to $\max_{(i,j)\in \boldsymbol{I}\setminus\{0\}} \{TT + s_i + t_{ij}\}$ to avoid a scaling problem in the model.

Q : the capacity of each vehicle expressed as the maximum amount of medical waste carried.

q_i : the amount of waste accumulated at supply node i ($i \in \boldsymbol{IC}$).

We assume that $q_i = 0$ for $i \in \boldsymbol{IY} \cup \mathbf{\{0\}}$. Furthermore, we also assume that all available vehicle fleet is homogeneous in speed and capacity.

3.2.3. Decision variables of (MTZ)-VRP

U_i : the load of a vehicle after visiting node i where $i \in \boldsymbol{IC}$. This variable is used in the lifted Miller-Tucker-Zemlin (MTZ) inequalities serving as subtour elimination constraints (SEC).

X_{ij} : binary variable indicating whether or not arc (i, j) is traversed at least once by a vehicle.

Y_i : integer variable indicating the number of leaving (entering) arcs at each LF where $i \in \boldsymbol{IY}$.

A_i : arrival time of a vehicle at node i ($i \in \boldsymbol{IC} \cup \boldsymbol{IY}$). Each vehicle departs from the depot at time $t = 0$.

X_{ij} and Y_{ij} are defined mutually exclusively to ensure that the last visited node on each route is a landfill facility.

3.2.4. Objective function of (MTZ)-VRP

$$\text{minimize } z = \sum_{i\in \boldsymbol{IC}} (v + c_{0i}) X_{0i} + \sum_{i\in \boldsymbol{IC}} \sum_{\substack{j\in \boldsymbol{IC}\cup \boldsymbol{IY} \\ i\neq j}} c_{ij} X_{ij} + \sum_{i\in \boldsymbol{IY}} c_{i0} Y_i \qquad (0.1)$$

3.3. Default constraints of (MTZ)-VRP

subject to:
$$\sum_{i\in \boldsymbol{IC}} X_{0i} \leq d \qquad (3.2)$$

$$Q \sum_{i\in \boldsymbol{IC}} X_{0i} \geq \sum_{i\in \boldsymbol{IC}} q_i \qquad (3.3)$$

$$\sum_{\substack{j\in \boldsymbol{IC}\cup\{0\} \\ i\neq j}} X_{ji} = 1 \qquad \forall i \in \boldsymbol{IC} \qquad (3.4)$$

$$\sum_{\substack{j\in \boldsymbol{I}\setminus\{0\} \\ i\neq j}} X_{ij} = 1 \qquad \forall i \in \boldsymbol{IC} \qquad (3.5)$$

$$\sum_{j\in \boldsymbol{IC}} X_{ji} = Y_i \qquad \forall i \in \boldsymbol{IY} \qquad (3.6)$$

$$\sum_{i\in \boldsymbol{IC}} X_{0i} = \sum_{j\in \boldsymbol{IY}} Y_j \qquad (3.7)$$

$$\sum_{i\in \boldsymbol{IC}} X_{i0} = 0 \qquad (3.8)$$

$$\sum_{i\in \boldsymbol{IY}} \sum_{j\in \boldsymbol{IC}} X_{ij} = 0 \qquad (3.9)$$

$$X_{i0} \geq \frac{1}{d} Y_i \qquad \forall i \in \boldsymbol{IY} \qquad (3.10)$$

$$X_{i0} \leq Y_i \qquad \forall i \in \boldsymbol{IY} \qquad (3.11)$$

$$U_i - U_j + Q X_{ij} + (Q - q_i - q_j) X_{ji} \leq Q - q_j \qquad \forall (i,j) \in \boldsymbol{IC},\ i \neq j \qquad (3.12)$$

$$q_i \leq U_i \leq Q \qquad \forall i \in \boldsymbol{IC} \qquad (3.13)$$

$$A_j \leq A_i + s_i + t_{ij} + (1 - X_{ij}) M \qquad \forall i \in \boldsymbol{IC} \cup \{0\}, \forall j \in \boldsymbol{IC} \cup \{0\},\ i \neq j \qquad (3.14)$$

$$A_j \geq A_i + s_i + t_{ij} - (1 - X_{ij}) M \qquad \forall i \in \boldsymbol{IC} \cup \{0\},\ \forall j \in \boldsymbol{I} \setminus \{0\},\ i \neq j \qquad (3.15)$$

$$A_i + s_i + t_{i0} X_{i0} - (1 - X_{i0}) M \leq TT \qquad \forall i \in \boldsymbol{IY} \qquad (3.16)$$

$$A_0 = 0 \qquad (3.17)$$

$$X_{ij} \in \{0,1\} \qquad \forall i \in \boldsymbol{I},\ \forall j \in \boldsymbol{I} \qquad (3.18)$$

$$Y_i \in \mathbf{Z}^+ \qquad \forall i \in \boldsymbol{IY} \qquad (3.19)$$

$$A_i \geq 0,\ U_i \geq 0 \qquad \forall\ i \in \boldsymbol{IC} \qquad (3.20)$$

3.3.1. Valid inequalities for (MTZ)-VRP

- Arrival time at the first supply node on a given route

$$A_i \geq t_{0i} X_{0i} \qquad \forall i \in \boldsymbol{IC} \qquad (3.21)$$

- Trivial subtour elimination constraints

$$X_{ji} + X_{ij} \leq 1 \qquad \forall (i,j) \in \boldsymbol{IC},\ i \neq j \qquad (3.22)$$

- Alternative lower bounds on the variables ***U***

$$q_i + \sum_{\substack{j \in \boldsymbol{IC} \\ j \neq i}} q_j X_{ji} \leq U_i \qquad \forall i \in \boldsymbol{IC} \qquad (3.23)$$

- Alternative upper bounds on the variables ***U***

$$U_i \leq Q - \sum_{\substack{j \in \boldsymbol{IC} \\ j \neq i}} q_j X_{ij} \qquad \forall i \in \boldsymbol{IC} \qquad (3.24)$$

- Additional upper bounds on the variables ***U*** for the first supply nodes on a route

$$U_i \leq Q - (Q - q_i) X_{0i} \qquad \forall i \in \boldsymbol{IC} \qquad (3.25)$$

- Additional tight upper bounds on the variables ***U*** as a replacement for (3.24) and (3.25)

$$U_i \le Q - (Q - \max_{\substack{j \in \boldsymbol{IC} \\ j \neq i}}\{q_j\} - q_i) X_{0i} - \sum_{\substack{j \in \boldsymbol{IC} \\ j \neq i}} q_j X_{ij} \quad \forall i \in \boldsymbol{IC} \tag{3.26}$$

The objective function in (3.1) of the above formulation shows the total traveling and vehicle operating cost. The traveling costs of the arcs going from the depot to the supply nodes have been augmented by the unit vehicle operating cost *v*. This way, we automatically capture the total vehicle operating cost which is proportional to the number of vehicles that are actually dispatched from the depot.

The first constraint in (3.2) is a trivial upper bound on the number of vehicles used. The second constraint in (3.3) ensures that the cumulative capacity of the vehicles dispatched from the depot is sufficient to collect the total amount of medical waste accumulating at the supply nodes.

Constraints in (3.4) state that at each supply node there must be exactly one entering (incoming) arc which originates either from the depot or from the other supply nodes. Likewise, constraints in (3.5) state that there must be exactly one leaving (outgoing) arc at each supply node which goes either to another supply node or to an LF. These two sets of constraints provide the degree balance for supply nodes in ***IC***.

Constraints in (3.6) count at each LF node the number of incoming arcs from the supply nodes. These constraints and the single constraint in (3.7) establish the degree balance at each LF node and at the depot between the incoming and outgoing arcs. The constraint in (3.8) implies that no vehicle can proceed from the supply nodes to the depot without passing through the LFs. Likewise, the constraint in (3.9) implies that no vehicle can return from an LF node back to the supply nodes. This trivial constraint can also be written for each LF node separately, in which case we would have to replace

(3.9) with $\sum_{j \in \boldsymbol{IC}} X_{ij} = 0 \quad \forall i \in \boldsymbol{IY}$. Constraints (3.10) and (3.11) set the value of the binary routing variable X_{i0} for the i^{th} LF in accordance with the number of arcs leaving that LF for the depot, namely Y_i.

The next two sets of constraints in (3.12) and (3.13) are lifted Miller-Tucker-Zemlin (MTZ) subtour elimination constraints for the VRP first proposed by Desrochers and Laporte (1991), corrected later by Kara et al. (2005). In our model (MTZ)-VRP they only apply to supply nodes and also account for the vehicle capacity limit. The definition of arrival time variables and the maximum tour duration constraint are given in (3.14)–(3.16). Time begins to elapse with the synchronized departure of vehicles from the depot; hence the arrival time of the depot is set equal to zero. Notice that the upper bound constraints on the arrival times are written exclusively for the supply node pairs $(i, j) \in (\boldsymbol{IC} \times \boldsymbol{IC})$ while the lower bound constraints apply also to the pairs of a supply node and an LF node $(i, j) \in (\boldsymbol{IC} \times \boldsymbol{IY})$. We must not constrain the arrival time at an LF from above. More than one arrival time calculation may be necessary for each LF node, since it is very likely that multiple vehicles enter the same LF from different supply nodes at different times. In the validation of the maximum tour duration constraint, one has to take into account the latest vehicle arrival time at each LF node. Thus, we do not impose upper bounds on the arrival times at LF nodes.

Finally, the last three constraints pertain to the binary restrictions on X_{ij}'s, to the integrality of Y_i's, and to the nonnegativity of A_i's and U_i's. The formulation in (3.1)−(3.26) has exactly n^2 MTZ inequalities in lieu of the exponential number of traditional subtour elimination or connectivity constraints.

3.4. Computational Complexity of the Problem

VRP-TD is a specific version of general VRP, with total tour duration and capacitated vehicles. The VRP is well known to be an NP-Hard problem because it includes TSP as a special case. Since one of its sub problems is NP-Hard, VRP-TD is also an NP-Hard problem.

Even for very small values of *n,* the solution of the problem is very difficult. As the number of nodes increases, number of solutions in the solution space increases sharply. The solution space cardinality can be computed as follows. Assume there are *n* hospitals to collect medical waste. If we consider the problem as a simple TSP problem the solution space cardinality should be *n!*. Since our problem is a specific version of VRP, the total number of solutions including both feasible and infeasible solutions is *n!*, but as each vehicle has a capacity of Q, the problem is also a partitioning problem and it gets more complexity.

Because of the computational difficulty, we develop a heuristic algorithm to find high quality, near optimal solutions in a reasonable amount of time.

Chapter 4

SOLUTION METHODOLOGIES

4.1. Introduction

In this section all of the considered solution methodologies are discussed. The section starts with exact approaches, continues with heuristics part and then finishes with metaheuristics part.

4.2. Exact Approaches

Exact approaches propose to compute every possible solution until the best is reached. Since the exact approaches compute every possible alternative solution, these approaches requires huge computation time for large-scale problems.

One of the most successful exact approaches for CVRP is the K-tree method of () that succeeded in solving a problem with 71 customers. However, there are smaller instances that have not been exactly solved yet. To treat large-scale instances, or to compute solutions faster, heuristic methods must be used. This exact approach uses a Branch and Bound procedure which the problem is partitioned by fixing the edge incidence of selected subsets of clustered customers. The side constraints are dualized to obtain a Langrangian relaxation that is a minimum degree-constrained K-tree problem. The K-tree approach can be extended to accommodate realistic variations, such as asymmetric costs/distances, time Windows, and non uniform fleets.

A branch and bound algorithm uses a divide and conquer strategy to partition the solution space into sub problems and then optimizes individually over each sub problem. Using branch and bound, we initially examine the entire solution space S. In the processing or bounding phase, we relax the problem. In so doing, we admit solutions that are not in the feasible set S. Solving this relaxation yields a lower bound on the value of an optimal solution. If the solution to this relaxation is a member of S or has cost equal to that of $\grave{s}$ (where $\grave{s} \in S$), then the procedure is done, either the new solution or $\grave{s}$ is optimal.

Otherwise we identify n subsets $S_1, \ldots, S_n$ of S, such that $\cup_{i=1}^{n} S_i = S$. Each of these subsets is called a sub problem. The procedure adds each subset of S to the list of candidate sub problems (those which await processing). This is the branching part of the procedure. To continue the algorithm, one of the sub problems is selected and processed.

There are four possible results for each branch;

1. If a feasible and better solution than $\grave{s}$ is found then the best solution is updated with the new one.
2. The selected sub problem may have no solutions; in this case prune procedure is applied to the related branch.
3. Otherwise, the lower bound of the sub problem is compared with the global upper bound, given by the value of the best feasible solution encountered thus far. If it is greater than or equal to the current upper bound then the related branch again may be pruned.
4. If a sub problem branch cannot be pruned then the children (sub branches) of this sub problem must be generated and added to the solution list. Branching

continues this way until the list of candidate sub problems is empty, at this point current best solution is the optimal solution.

4.3. Heuristics

In this part two constructive heuristics are discussed. Constructive heuristics aim to build a feasible solution while keeping an eye on solution cost, but these methods do not include an improvement phase in their procedures.

4.3.1. Nearest Insertion

Since exact algorithms cannot solve large-scale problems, heuristics are used to construct feasible solutions. Nearest Insertion is one of the most known constructive heuristics.

The procedure starts with calculating the cost of insertion for each node. Then the algorithm starts building a feasible solution by inserting the nodes with lowest costs iteratively. When the capacity prevents insertion then a new route is created and insertion is done to that new route. The algorithm inserts all of the nodes until the node list is empty.

Since the algorithm has no improvement phase in it, algorithm never considers backward insertion, which means considering older routes. Algorithm proceeds iteratively forward thus the main aim is to generate a good feasible solution not to find the optimal solution.

The pseudo-code of the algorithm is as follows;

Algorithm 4.1. Nearest Insertion

1. $Routes \leftarrow \emptyset$
2. generate insertion values for the hospitals in the empty route
3. **while** $hList \neq \emptyset$
4.   **do** find hospital node $h1$ with lowest insertion value
5.   **if** $h1$ should be inserted into a new route
6.   **then** create a new route $r1$ and insert $h1$
7. $Routes \leftarrow Routes \cup \{r1\}$
8.   **else** insert $h1$ to the wished route
9. $hList \leftarrow hList \setminus \{h1\}$
10.   generate new insertion values for $h1$
11. **return** $Routes$

Figure 4.1: Nearest Insertion pseudo-code

4.3.2. Clarke & Wright Parallel Savings Algorithm

The Clarke and Wright savings algorithm is one of the most known heuristics for VRP. It was developed on () and it applies to problems for which the number of vehicles is not fixed (number of vehicles is a decision variable), and it works equally well for both directed and undirected problems. When two routes $(0, \dots, i, 0)$ and $(0, j, \dots, 0)$ can feasibly be merged into a single route $0, \dots, i, j, \dots, 0)$, a distance saving generated is;

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}$$

The heuristic proceeds by merging every pair of routes in descending order of s_{ij}, as long as the merge operation does not violate these conditions;

1. s_{ij} is greater than or equal to 0
2. Merging does not violate capacity constraint
3. Vertex i is the first of its route
4. Vertex j is the last of its route

5. Vertices i and j do not belong the same route

   Even if saving is 0, merging can decrease the total number of vehicles by one, so only the negative savings are forbidden. As savings between vertices does not change and a merge between two vertices i and j can only be done once, the need is to initially calculate savings and sort them for one time only.

Clarke and Wright Savings algorithm can split into 2 steps; (the first step is same for both parallel and sequential savings algorithms)

Step1. Savings Computation

- Compute the savings $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ for $i, j = 1, \dots, n$ and $i \neq j$
- Create n vehicle routes $(0, i, 0)$ for $i = 1, \dots, n$
- Order the savings non-increasingly

Step2. Best Feasible Merge (Parallel Savings Algorithm)

- Given a savings s_{ij} determine whether there exists two routes can feasibly be merged:
  - One starting with $(0, j)$
  - One ending with $(i, 0)$
- Combine these two routes by deleting $(0, j)$ and $(i, 0)$, and introducing (i, j)

Step2. Route Extension (Sequential Savings Algorithm)

- Consider in turn each route $(0, i, \dots, j, 0)$
- Determine the first savings s_{ki} or s_{jl}that can feasibly be used to merge the current route with another route ending with $(k, 0)$ and starting with $(0, l)$
- Implement the merge and repeat this operation to the current route

- If not feasible merge exists, consider the next route and reapply the same operations
- Stop when no merge is feasible

In this study Clarke & Wright Parallel savings algorithm is used to construct the initial solution for the second phase, the tabu search algorithm.

Algorithm 4.2. Clarke & Wright Parallel Savings Algorithm

```
next ⟵ [], prev ⟵ []
first ⟵ [], last ⟵ []
load ⟵ [], savings ⟵ []
for all v ∈ V \ depot do // initialization
  next[v] ⟵ prev[v] ⟵ depot
  first[v] ⟵ last[v] ⟵ v
  load[v] ⟵ demand(v)
  for all w ∈ V depot do
    c ⟵ cost(v, depot) + cost(depot, w) − cost(v, w)
    savings.append((c, v, w))
  end for
end for
sort(savings) // sort by s_ij in descending order
for all (c, v, w) ∈ savings do // visit savings in descending order, merging routes
  if s_vw ≥ 0 then
    if next[v] = depot ∧ prev[w] = depot ∧ last[w] ≠ v ∧ first[v] ≠ w then
      if load[v] + load[w] ≤ capacity then
        next[v] ⟵ w
        prev[w] ⟵ v
        first[last[w]] ⟵ first[v]
        last[first[v]] ⟵ last[w]
        load[first[v]] ⟵ load[last[w]] ⟵ load[v] + load[w]
      end if
    end if
  else
    break
  end if
end for
for all v ∈ V do // route construction
  if v ≠ depot ∧ prev[v] = depot then
    route ⟵ [depot]
    while v ≠ depot do
      route.append(v)
      v ⟵ next[v]
    end while
    routes.append(route)
  end if
end for
return routes
```

Figure 4.2: Clarke & Wright Parallel Savings Algorithm pseudo-code

Asymmetric Clarke & Wright Application

Clarke and Wright parallel savings algorithm is used in this study to construct initial solution. Since the considered problem is ACVRP-IF then some modifications are proposed due the asymmetric property and intermediate facilities.

In the original C&W the saving values are calculated for every route at the beginning and updated easily by adding up the savings value.

Route1:

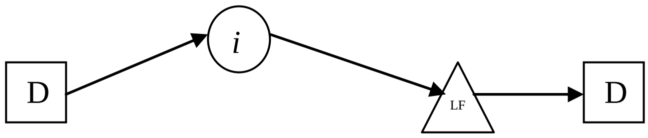

Figure 4.3: C&W algorithm visualization - route1

Route2:

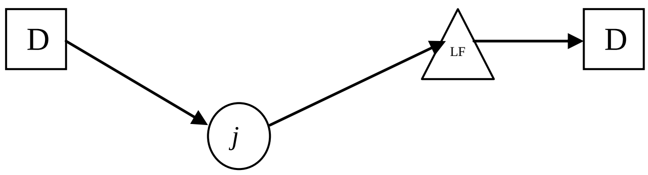

Figure 4.4: C&W algorithm visualization - route1

If the distances are symmetric then no matter in what order the merge is applied, the savings are not changed. In this study the distances are asymmetric and must be considered.

Merged Route 1:

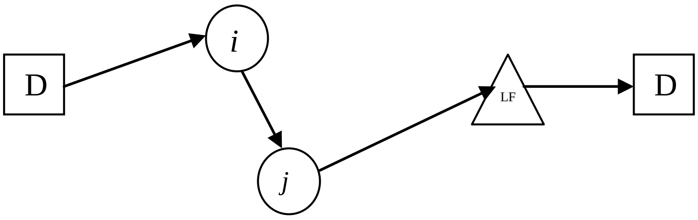

Figure 4.5: C&W output – Merged Route1

Merged Route 2:

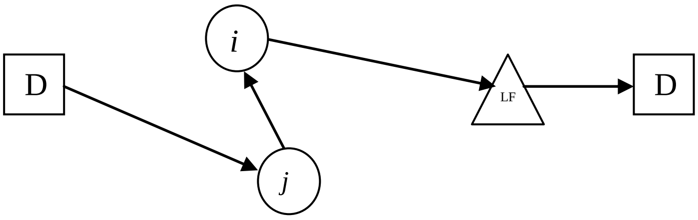

Figure 4.6: C&W output – Merged Route2

$$S_1 = c_{i,LF} + c_{D,j} + c_{LF,D} - c_{i,j}$$

$$S_2 = c_{j,LF} + c_{D,i} + c_{LF,D} - c_{j,i}$$

Since the travelling distances are asymmetric, the distances from i to j and from j to i are different therefore $S_1 \neq S_2$. For this differentiation $subcase$ concept is generated. With this $subcase$ concept the order of i and j after merge operation is stored. If merge operation is applied in $i - j$ sequence then $subcase = 1$ else $subcase = 2$ is assigned.

Another modification for asymmetric distances is the merge operation in reverse order of a part of a route.

Route1:

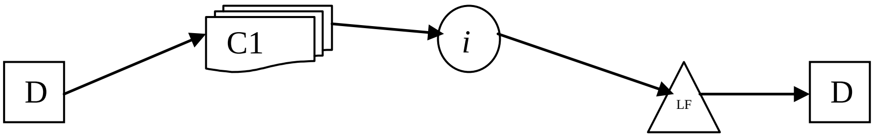

Figure 4.7: Sample Route1

Route2:

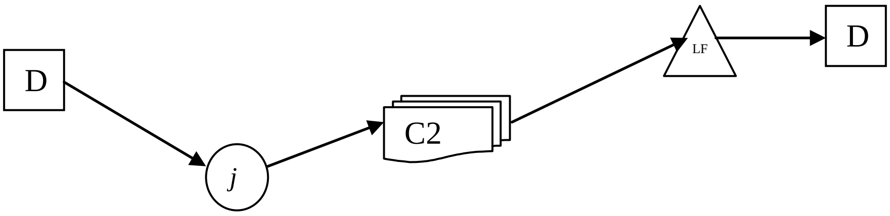

Figure 4.8: Sample Route2

Four different route1-route2 matches can occur with respect to the place of i and j. C abbreviation here is used to define a set of nodes (node cloud). For these four different match, four different savings calculation must be done, within each two subcases.

Merged Route1:

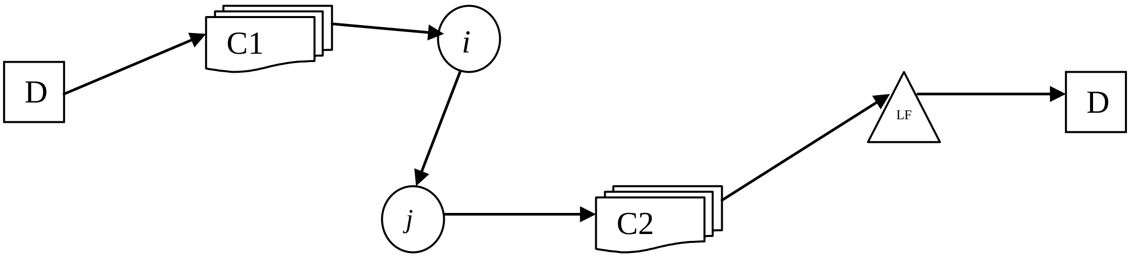

Figure 4.9: Sample Merged Route1

Merged Route2:

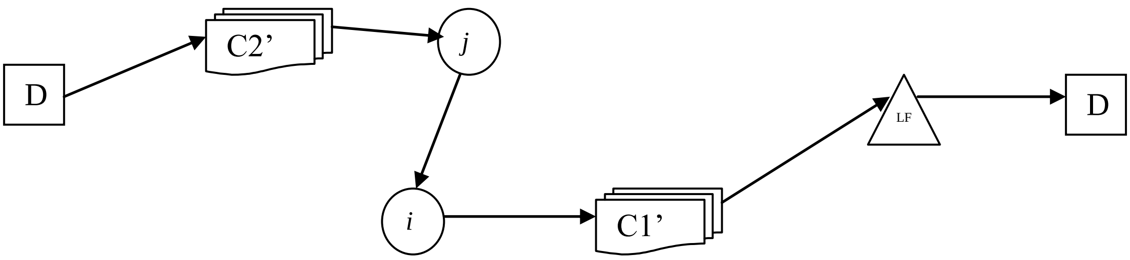

Figure 4.10: Sample Merged Route2

The first merge has no reversing so the calculation of saving is simple, but in the second merge alternative $C1$ and $C2$ are reversed and because of asymmetry in distances the total distance of the final clouds may be different. So in the saving calculation of the second subcase the reverse distance summation of $C1$ and $C2$ are also considered.

Merging alternatives

Merging operation is applied if merging two routes results a saving to the current situation. Saving calculation is explained above with asymmetric distances modifications.

In this study merging operation has 4 cases with 2 subcases in each. The cases are listed below;

CASE1:

Route1:

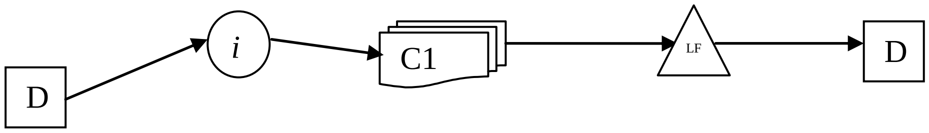

Figure 4.11: Case1 - Route1

Route2:

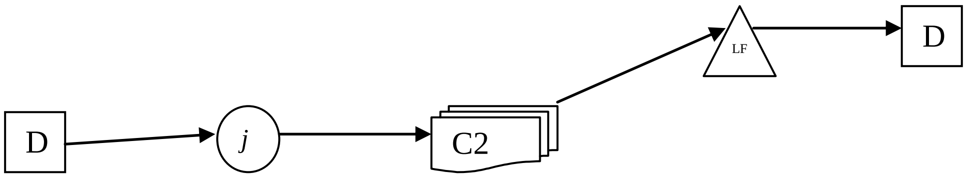

Figure 4.12: Case1 - Route2

Merged Route1:

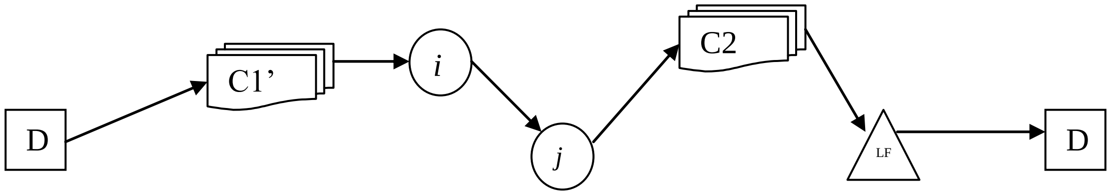

Figure 4.13: Case1 - Merged Route1

Merged Route2:

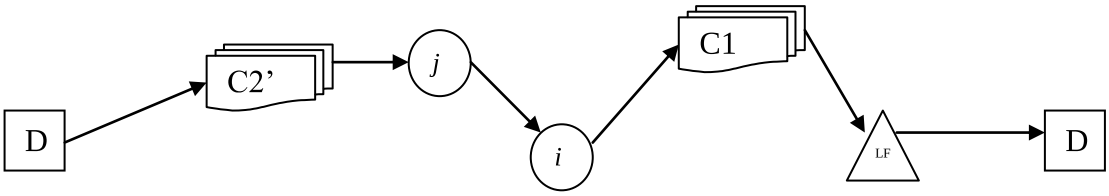

Figure 4.14: Case1 - Merged Route2

In the first alternative route1 is reversed and attached to the beginning of the route2. And in the second alternative route2 is reversed and attached to the beginning of route1.

If we assume the first and last nodes of the clouds as;

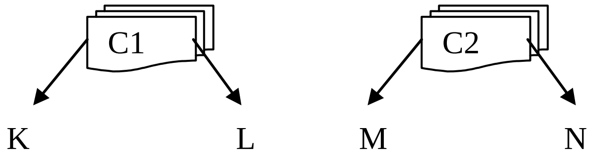

Figure 4.15: Cloud Structures

The savings for each is as;

$$S_1 = c_{D,i} + c_{D,j} - c_{D,L} + c_{L,LF} + c_{LF,D} + \overline{C1}$$

$$S_2 = c_{D,j} + c_{D,i} - c_{D,N} + c_{N,LF} + c_{LF,D} + \overline{C2}$$

CASE2:

Route1:

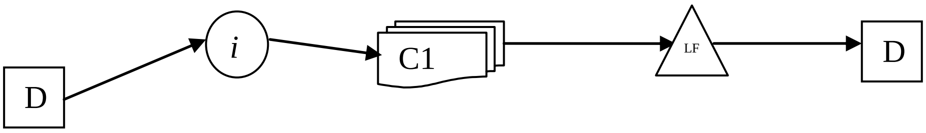

Figure 4.16: Case2 - Route1

Route2:

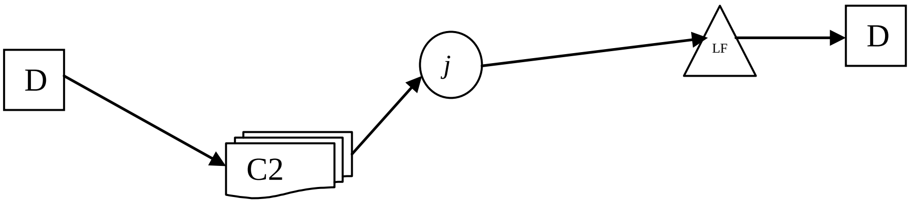

Figure 4.17: Case2 – Route2

Merged Route1:

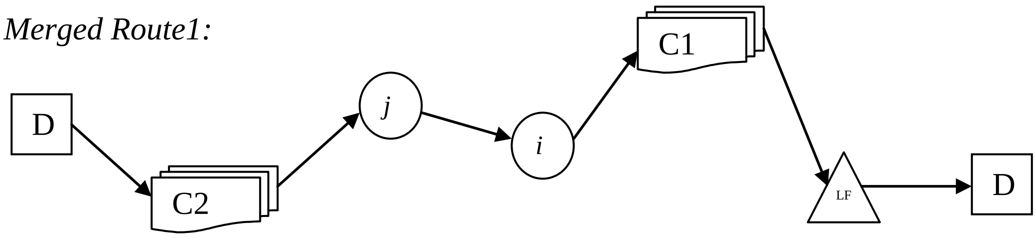

Figure 4.18: Case2 – Merged Route1

Merged Route2:

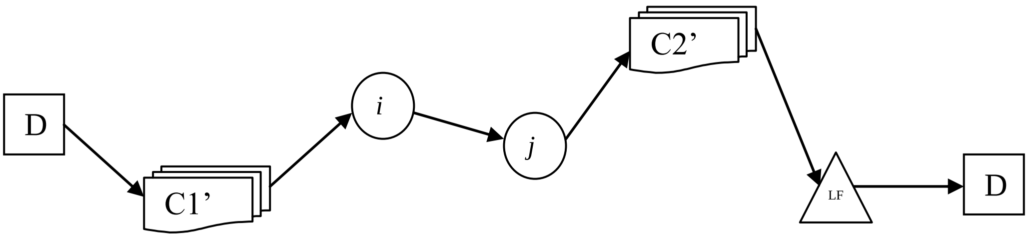

Figure 4.19: Case2 – Merged Route2

In the first alternative route1 is attached to the beginning of the route2, none of the routes is reversed. And in the second alternative route1 and route2 are both reversed and route1 is attached to the beginning of route2.

$$S_1 = c_{D,i} + c_{j,LF} + c_{LF,D} - c_{D,M} + c_{L,LF} - c_{j,i}$$

$$S_2 = c_{D,i} + c_{j,LF} + c_{D,M} + c_{L,LF} + c_{LF,D} - c_{D,L} - c_{i,j} - c_{M,LF} + \overline{C1} + \overline{C2}$$

CASE3:

Route1:

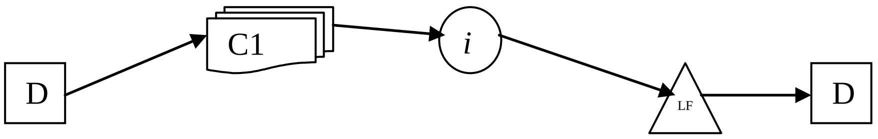

Figure 4.20: Case3 – Route1

Route2:

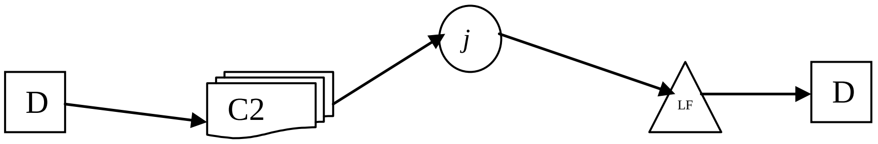

Figure 4.21: Case3 – Route2

Merged Route1:

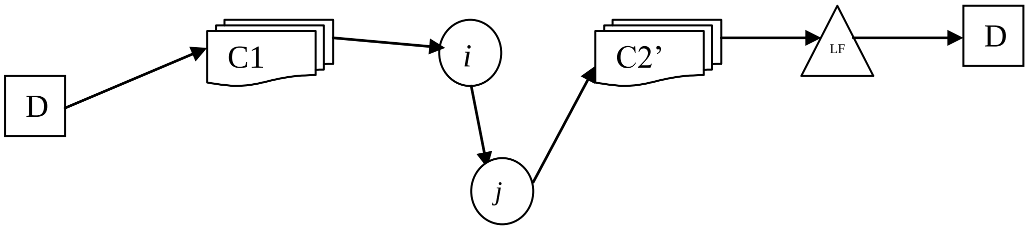

Figure 4.22: Case3 – Merged Route1

Merged Route2:

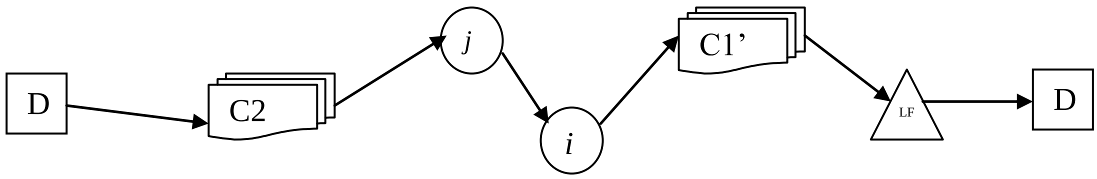

Figure 4.23: Case3 – Merged Route2

In the first alternative route2 is reversed and attached to the beginning of the route1. And in the second alternative route1 is reversed and attached to the end of route2.

$$S_1 = c_{i,LF} + c_{j,LF} + c_{LF,D} + c_{D,M} + c_{M,LF} - c_{i,j} + \overline{C2}$$

$$S_2 = c_{D,K} + c_{i,LF} + c_{j,LF} + c_{LF,D} - c_{K,LF} - c_{j,i} + \overline{C1}$$

CASE4:

Case4 is visualized at the beginning of the section, in the first alternative route2 is attached to the end of route1; none of the routes is reversed. In the second alternative both routes are reversed and route1 is attached to the end of route2.

The savings for both merge alternatives are;

$$S_1 = c_{i,LF} + c_{D,j} + c_{LF,D} - c_{i,j}$$

$$S_2 = c_{j,LF} + c_{D,i} + c_{LF,D} - c_{j,i}$$

4.3.3. Enhancements of Clarke and Wright Parallel Savings Algorithm

4.3.3.1. Gaskell and Yellow Enhancement

This enhancement adds the route shape parameter in the savings calculation to improve the performance of Clarke and Wright savings algorithm.

$$s_{ij} = c_{i0} + c_{j0} - \lambda c_{ij}$$

In this study, since the distances are not symmetric the savings formula is different for each case and subcase. In the implementation phase this change in formulas are considered and the enhancement is applied by adding the route shape parameter (λ) as a multiplier for the additional arc (c_{ij} or c_{ji}).

4.3.3.2. Paessens' Enhancement

With this enhancement in addition to Gaskell and Yellow's enhancement a weighted parameter (μ) is inserted to the savings calculation formula. With this parameter the use of asymmetry between customers i and j with respect to their distances to the depot is enabled.

$$s_{ij} = c_{i0} + c_{j0} - \lambda c_{ij} + \mu \left| c_{0i} - c_{j0} \right|$$

Since this study considers asymmetric distances and an intermediate facility before returning to the depot, the enhancement implemented in a very different way with respect to the original formulation. In each case and subcase the savings algorithm formulation changes therefore the Paessens' enhancement is implemented as replacing c_{0i} and c_{j0} by the newly added arcs differing from case to case.

4.3.3.3. Altınel and Öncan Enhancement

This enhancement is the newest enhancement on Clarke&Wright savings algorithm. The difference with this enhancement is the demand consideration of the respective nodes.

$$s_{ij} = c_{i0} + c_{j0} - \lambda c_{ij} + \mu |c_{0i} - c_{j0}| + \nu \frac{d_i + d_j}{\bar{d}}$$

Here the last term is the ratio of the summation of demands i and j to the average demand of all nodes. ν is the multiplier of the normalized demand addition. With this enhancement demand side of the problem is considered (Bin Packing Problem) and with the multiplier the nodes with higher demand are tried to put together. (put first larger items)

In this study the implementation of this enhancement is straight forward than the previous enhancements. Demand consideration is directly added in the savings calculation formula.

4.3.4. Local Post Optimization Operators

These operators are the building blocks of the Local Search. They are used to travel from one solution to another. The choice of operators controls the richness of the solution neighborhood. The number of operators used in the algorithm increases the total number of solutions to be visited during each iteration. On the other hand the more operators used and more solution generated at each iteration, the more time it takes for each iteration. Relevant works in the literature uses single operator or maximum three operators.

The operators used are divided into two groups; intra-route and inter-route operators. Intra-route operators travel to a solution by making change in a same route. Inter-route operators can have an effect on nodes of multiple routes. There are several reasons for this distinction; the first reason is the limiting the amount of routes being evaluated at once reduces the amount of nodes taken into consideration. Second reason is the differentiation of the stages that these two distinct operators are used. Mainly in the construction phase intra-route versions are used in the literature. Third reason for distinction is intra-route moves have no effect on demand constraints, which makes it easier to process.

4.3.4.1. 1-0 Move

This move is applied as removing one node from a route and then inserting that node to another place in the same route or a different route. The insertion is made when the best solution is found in the solution space.

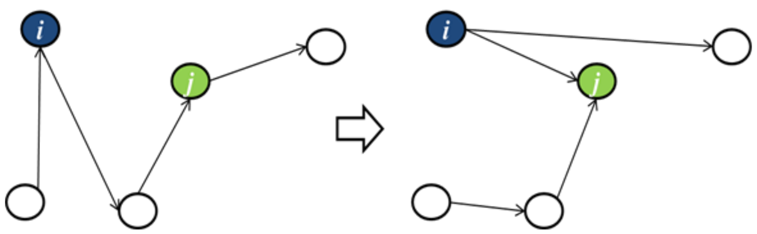

Figure 4.24: Simple visualization of 1-0 move on same route

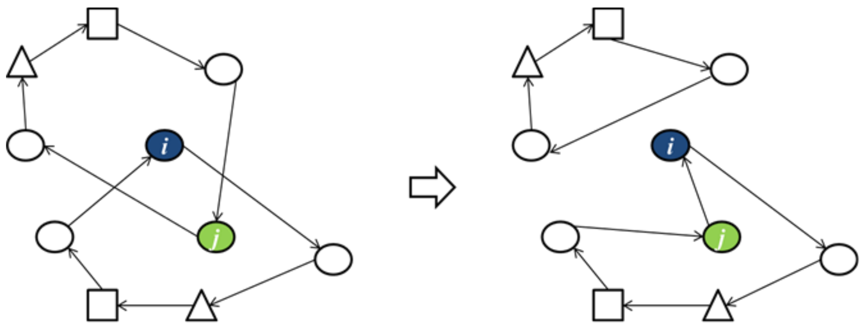

Figure 4.25: Simple visualization of 1-0 move on different routes

4.3.4.2. 1-1 Exchange

This move is applied as exchanging two different nodes from same route or different routes to construct a better feasible solution.

Figure 4.26: Simple visualization of 1-1 exchange in same route

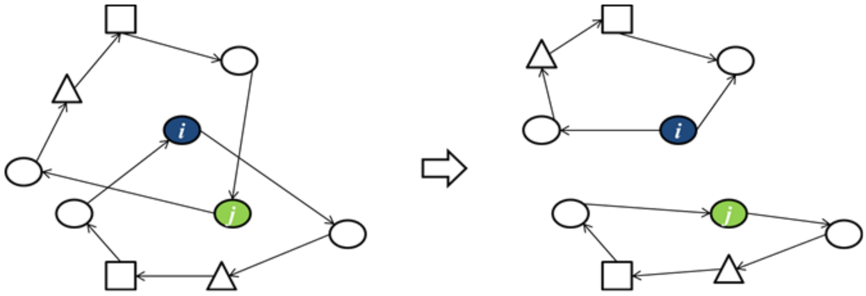

Figure 4.27: Simple visualizayion of 1-1 exchange - different routes

4.3.4.3. 1-1-1 Rotation

This move is designed for three different routes (x,y,z), from these routes three nodes are selected (i,j,k). An interchange of these three nodes in the given procedure is applied at each move to improve current solution.

The procedure for move:

- $k \rightarrow i$
- $i \rightarrow j$
- $j \rightarrow k$

The algorithm has a n^3complexity, as searching for the best j and k respectively for a node i. Three nested loops are needed to perform this search. The move is performed when the best i, j, k nodes are found and the solution generated is feasible. The algorithm decides this from the overall decrease in the total distance of the solution.

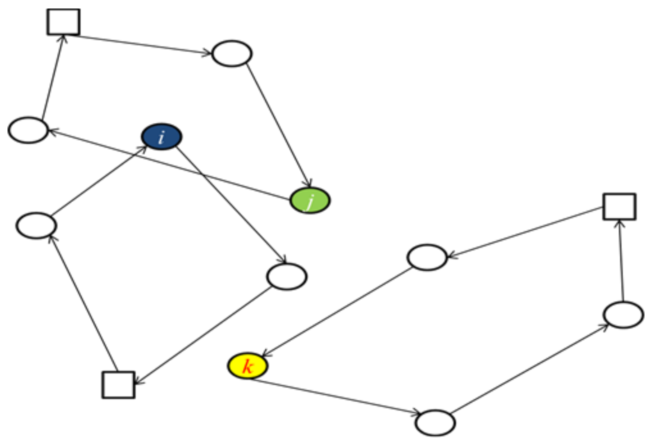

Figure 4.28: 1-1-1 rotation - before the move

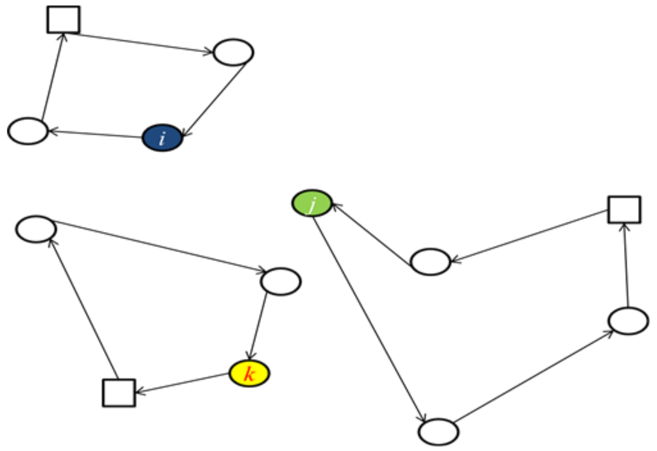

Figure 4.29: 1-1-1 rotation – after the move

4.3.4.4. 2-Opt

Intra-route

2-Opt is applied as selecting two nodes from same route then change the traveling order to prevent crossing on the route. Visualization is given below;

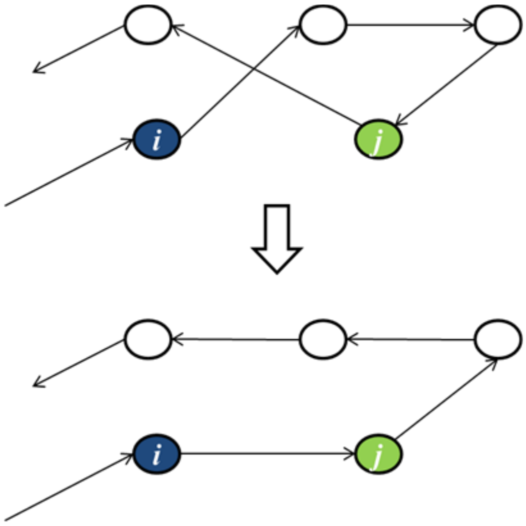

Figure 4.30: Simple visualization for 2-Opt intra-route

Inter-route

The inter-route version of 2-opt is different than the intra-route version, since we deal with the asymmetric distances different cases are generated. Cases are visualized below, in these cases C_n represents node cloud n.

Route1:

$$Depot \rightarrow C1 \rightarrow i \rightarrow C2 \rightarrow LF \rightarrow Depot$$

Route2:

$$Depot \rightarrow C3 \rightarrow j \rightarrow C4 \rightarrow LF \rightarrow Depot$$

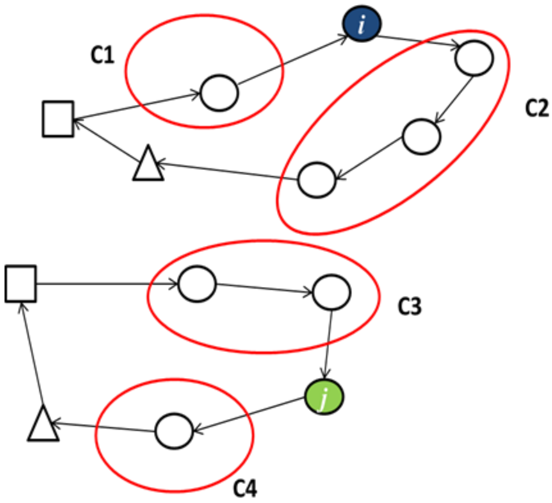

Figure 4.31: Initial status of two different routes

Case1:

NewRoute1:

$$Depot \rightarrow C1 \rightarrow i \rightarrow j + 1 \rightarrow C4 \rightarrow LF \rightarrow Depot$$

NewRoute2:

$$Depot \rightarrow C3 \rightarrow j \rightarrow i + 1 \rightarrow C2 \rightarrow LF \rightarrow Depot$$

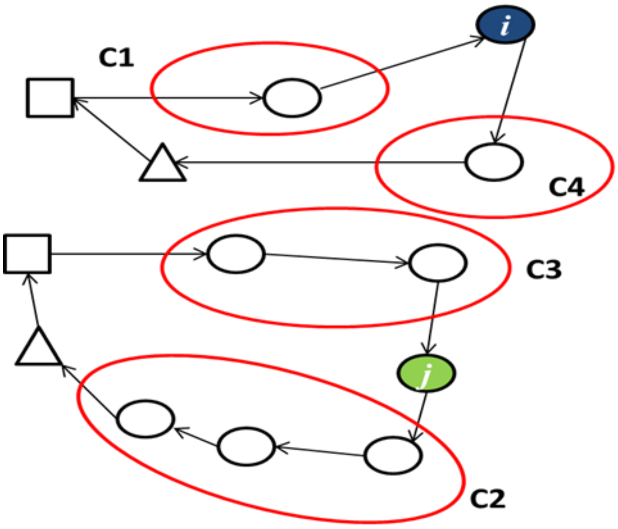

Figure 4.32: New routes after 2-Opt Case1

Case2:

NewRoute1:

$$Depot \rightarrow C1 \rightarrow i \rightarrow j \rightarrow \overline{C3} \rightarrow LF \rightarrow Depot$$

NewRoute2:

$$Depot \rightarrow \overline{C2} \rightarrow i+1 \rightarrow j+1 \rightarrow C4 \rightarrow LF \rightarrow Depot$$

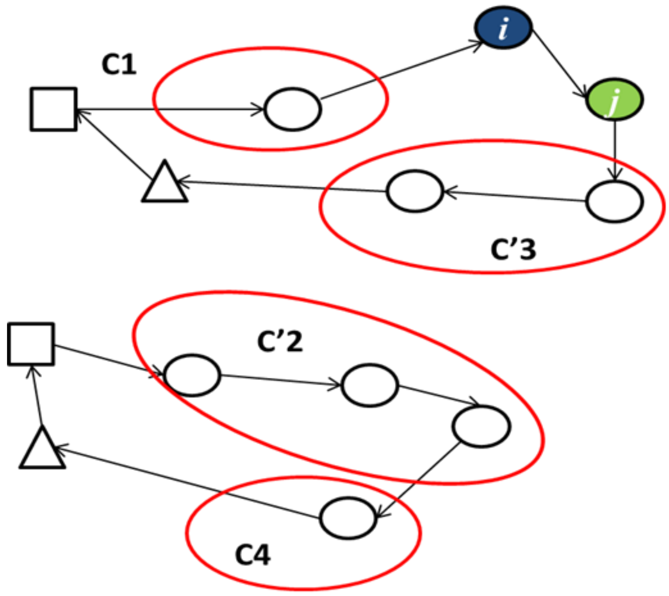

Figure 4.33: New routes after 2-opt Case2

Case3:

NewRoute1:

$$Depot \rightarrow C3 \rightarrow j \rightarrow i \rightarrow \overline{C1} \rightarrow LF \rightarrow Depot$$

NewRoute2:

$$Depot \rightarrow \overline{C2} \rightarrow i+1 \rightarrow j+1 \rightarrow C4 \rightarrow LF \rightarrow Depot$$

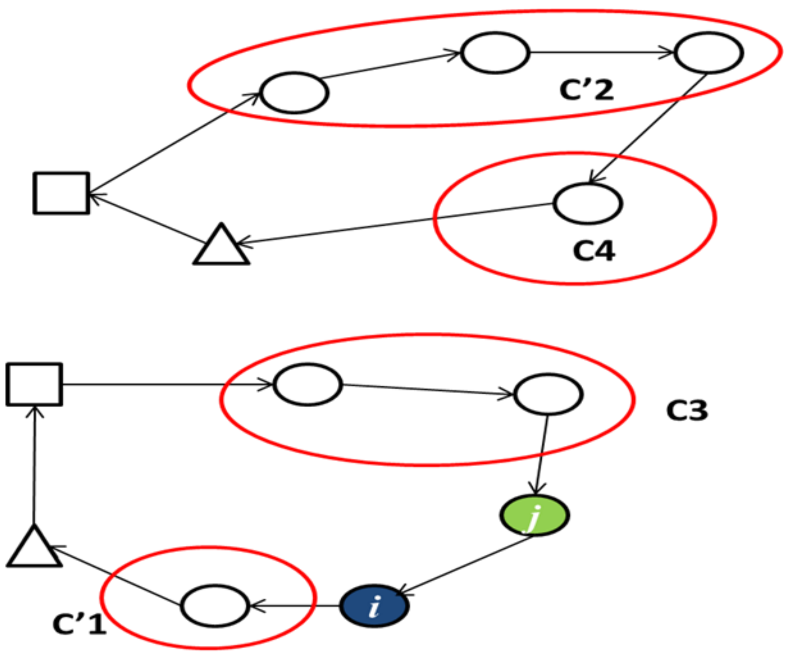

Figure 4.34: New routes after 2-Opt Case3

Case4:

NewRoute1:

$$Depot \rightarrow C1 \rightarrow i \rightarrow j \rightarrow \overline{C3} \rightarrow LF \rightarrow Depot$$

NewRoute2:

$$Depot \rightarrow \overline{C4} \rightarrow j+1 \rightarrow i+1 \rightarrow C2 \rightarrow LF \rightarrow Depot$$

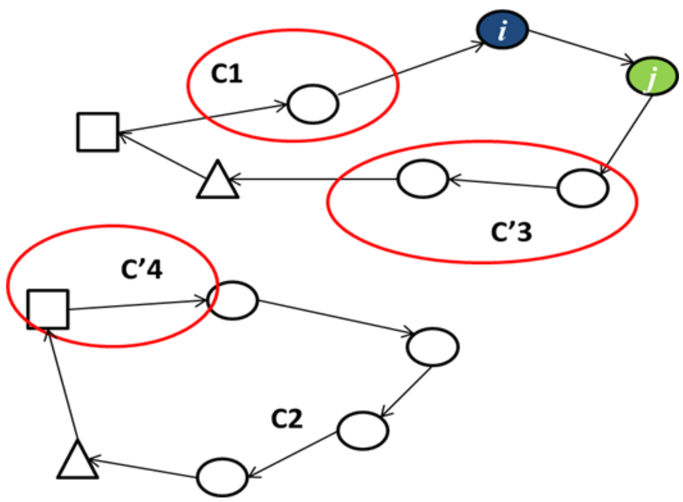

Figure 4.35: New routes after 2-Opt Case4

4.3.4.5. N Successive 1-0 Move

In this part two or more successive 1-0 move is performed. These additional move trials can turn a bad move into a good one when all procedure is done. For example, if first 1-0 moves does not decrease the total distance, the following 1-0 moves can have such revenue that covers all additional distances and overall total distance can be decreased. In this project, the version N=2 is used.

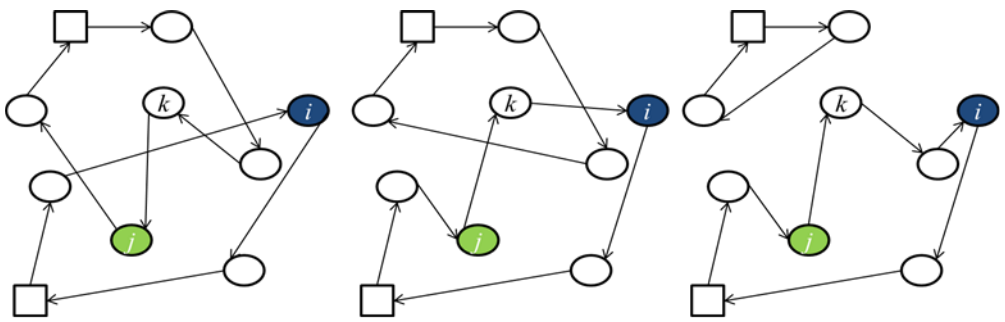

Figure 4.36: Simple visualization for 2 Successive 1-0 Move

4.3.4.6. 3-Opt

3-opt algorithm selects three edge to work on, it deletes three edges from a route and reconnects three remaining parts in some other way. In this study only the intra-route version of 3-opt algorithm is implemented. Inter-route version is not implemented since it is not likely to pay off.

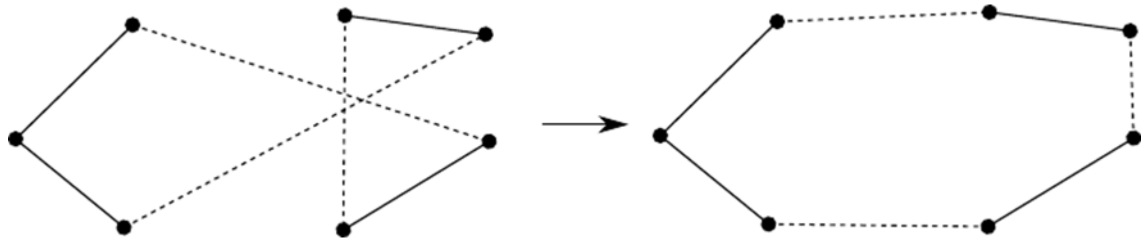

Figure 4.37. 3-Opt sample move

4.3.5. Osman's Algorithm

Osman's algorithm is used to reduce the size of evaluated neighborhood. With this algorithm the set of moves can be limited to only those that are likely to lead a better solution. Two matrices are generated; the first matrix $Dist\_Order$ contains the ordinal ranking of nodes by distance to one node. Meaning;

$$Dist\_Order[3][2] : contains\ the\ 2^{nd}\ nearest\ node\ to\ 3^{rd}\ node$$

The second matrix is generated to set an indicator that is filled with YES till a given number $\eta * \#ofnodes$. η is the input of Osman's algorithm, it is used to limit the number of nodes, as a cutting point.

$$Binary\_Indicator\_Nearest[2][6]: \begin{cases} \textit{Contains "YES" if } 6^{th} \textit{ node is in the nearest node set of } 2^{nd} \textit{ node} \\ \textit{else contains "NO"} \end{cases}$$

By checking the indicator in $Binary\_Indicator\_Nearest$ for the related nodes, set of moves are limited for given η value.

The LPOs are tested with the Osman's Algorithm inserted versions of each, the results are;

Table 4.1: CMT Problem set with Osman's Algorithm inserted Tabu Search

Instance*	Problem	Laporte-Semet C&W	C&W	C&W+LPO	Osman's Algorithm Added RTSH
p01	E051 - 05e	584.64	584.64	560.46	560.46
p02	E076 - 10e	900.26	907.39	896.3	896.3
p03	E101 - 08e	886.83	886.83	879.21	879.21
p04	D121 - 11c	1133.43	1133.43	1121.96	1121.96
p05	D101 - 11c	1395.74	1395.74	1383.37	1383.37
p06	E101 - 10c	618.4	618.39	609.58	609.87
p07	E121 - 07c	975.46	975.46	967.52	967.52
p08	E151 - 12c	973.94	973.94	964.25	967.42
p09	E200 - 17c	1287.64	1287.64	1281.16	1282.49
p10	D051 - 06c	1538.66	1538.66	1516.54	1516.54
p11	D200 - 18c	1071.07	1071.07	1046.93	1046.57
p12	D151 - 14c	833.51	833.51	821.29	821.29
p13	D076 - 11c	1596.72	1592.26	1577.53	1579.13
p14	D101 - 09c	875.75	875.75	867.76	868.84

*CMT Problem Set

Here we see many of CMT problems are solved as same with or without Osman's Algorithm in quality. Since we are dealing a large-sized real life problem computation time is important, therefore Osman's Algorithm inserted versions of each LPO is used in the application phase.

4.4. Metaheuristics

4.4.1. Proposed Tabu Search Algorithm

4.4.1.1. Neighborhood Structure

The size of neighborhood is directly related with the used operators. In this study Tabu Search algorithm uses 1-0 move, 1-1 Exchange and 3-opt operators for the generation of neighbor solutions in each iteration. The inter-route and intra-route versions for 1-0 and 1-1 are both used in the iterations, and 3-opt used as only intra-route version.

4.4.1.2. Tabu Definition

In this study a different Tabu listing is proposed. Storing solutions with their hash values is explained in 4.4.2 section. The general TS logic includes forbidding solutions to be visited by listing these solutions in the Tabu List. Tabu definition and the list size is important here. If a potential solution is has been previously visited within a certain short term period or if it has violated the tabu rule then that solution is called the tabu solution.

When two (or three with respect to move type) nodes, named as node i and j, are involved in obtaining a new solution, the $ith\ and\ jth$ node positions cannot be changed again for a certain number of iterations such as the size of tabu tenure. Using dissimilar neighborhood structures result in keeping tabu restrictions in different ways. Using hash function to generate each solution a unique key enables the algorithm store tabu list in one simple way, without depending on the move method.

Recency-based memory is used in Tabu Search algorithm which forbids moves towards the most recent visited solution. Size of tabu tenure is also selected according to the problem size.

All these general tabu structures conversion to the one that is used in this study is explaned in Hash Function section.

4.4.1.3. Aspiration Criteria

Aspiration criteria are used to cancel the status of a move and allow that solution to return back to be revisited even this move is restricted by the algorithm. The classical aspiration criteria is used to allow a move even it is tabu, if it results as a solution with better objective function value which is better than the incumbent one.

4.4.1.4. Termination Criteria

Termination criteria selection is the most difficult decisions in tabu search algorithms. Two parallel termination criteria are used in the proposed algorithm. One of them is the classical maximum number of iterations, and the other is set as the maximum number of non-improving solutions.

4.4.1.5. Strategic Oscillation

In this study Tabu Search algorithm is designed to search a very large solution space with a dynamic reactive manner. The diversification is proposed by letting moving an infeasible solution when the algorithm is stucked in a local optimum solution. By moving an infeasible solution and letting the algorithm to change the current solution structure the move operators may jump the local optimum and reach a better solution in the following iterations. In this proposed algorithm strategic oscillation is controlled by penalty assignment. Penalty assignment is not static so that makes the tabu search algorithm reactive to the direction of movements.

4.4.2. Hash Function

The iterations of Tabu Search algorithm are directed by using a computer science technique called Hashing Function Search (HFS), a pattern described in (Juliff, 1990). This hashing method generates a unique code to each solution, S_r.(Wassan, 2006) S_r is the sum of the product of the customer index x_j and the total number of customers in each corresponding route $|R_p|$:

$$S_r = \sum_{p=1}^{v} \sum_{x_{j \in R_p}} x_j |R_p|$$

$$x_j: index\ of\ customer\ j\ in\ respective\ route$$

After generating hash values a doubly-linked list is used to search whether the solution is visited before or not. Doubly-linked list has a double sided search structure so that this search requires less computational time than other search methods.

(Gröer, Golden, & Wasil, 2010) also provides a hasing structure for VRP. In this study the hashing function and the calculation algorithm designed as follows;

Algorithm 4.3. Hashing Function Calculation Algorithm

1. **for** j=1 to R **do**
2. **if** $a_j > z_j$ **then** Reverse the ordering of route *j*
3. **end if**
4. **end for**
5. Create a list L containing the resulting, possibly reversed routes, and sort this list in terms of the index of the first node visited in the route
6. With $L = \{r_1, r_2, \ldots, r_R\}$, concatenate these routes into a single list $\{m_1, m_2, \ldots, m_n\}$
7. Return $H = \bigoplus_{i=1}^{n-1} Y_{(m_i + m_{i+1})\ mod\ n}\ mod\ 2^h$

Figure 4.38. Hashing Function Calculation Algorithm Pseudo-Code

This algorithm assigns a 32-bit integer hash value to each solution. The logic behind this algorithm focuses on to generate unique hashing values for each solution to decrease the collusion rate. Mod operator in the algorithm scales each hasing function that enables the uniqueness of hashing function values.

Chapter 5

COMPUTATIONAL EXPERIMENTS

In this chapter, computational experiments are examined to analyze the performance of proposed meta-heuristics in terms of the solution quality and computation time. In section (5.1) benchmark problems and performance studies are listed, in section (5.2) application data generation and parameter settings are explained, finally in section (5.3) application results are listed.

5.1. Data Preparation and Parameter Settings

The study requires an input of distance matrices of the hospitals for both Anatolian and European sides of Istanbul. The distances are generated by using maps application of Google. By modifying an application programming interface (API) coded in PHP an automatic distance calculated is proposed. The application is put on a virtual PHP server to get the result matrices. The distance and time matrices both generated from Google Maps via the proposed API.

Since parameter setting controls the balance between diversification and intensification in the searched solution set, it is one of the most difficult problems in heuristic implementations. This hybrid algorithm proposed in this study contains many parameters in the construction phase and also in RTSH. In the construction phase Osman's parameter limits total solutions to be visited, for this parameter, different values are analyzed by using CMT-14 problems and selected the most effective one for the application phase (Analysis results are shown in the following figure). The other construction phase parameters are used in the enhancements of Clarke & Wright algorithm. There are many versions for this tuning phase but the search started from the best known parameter settings in the literature.

In RTSH phase of this study to balance diversification operators search area is important. The larger search area is the best for diversification. And for intensification hash function is a component. Hash values for each solution are stored and this enables the algorithm to intensify on a specific range of solutions.

Vehicle capacity and route duration are not given so a set of different problems are generated. Vehicles are classified in three groups, 1500, 200 and 2300 kg of capacity. And route tour durations are taken as 7, 8 and 9 hours. So in total 9 problems for each side of İstanbul are generated.

In the cost analysis part the estimated cost of collecting 1 ton of medical waste given by European Commission – Life Project is taken (European Commission, 2011). The cost is given as 400 dollars per ton. Also the driver's and collectors' daily cost is estimated and inserted in the cost analysis phase. There are two collectors and one driver in a medical waste vehicle.

Table 5.1 and 5.2 shows the test results done for parameter settings. Table 5.1 is the first stage of the settings. In this stage Nearest Insertion (NI), Nearest Insertion with LPO (NI+LPO), Clarke & Wright Algorithm (C&W) and Clarke & Wright Algorithm with LPO (C&W+LPO) are compared. The initial solution generation method is decided at this stage. The LPOs use Osman's algorithm therefore a η value is needed to be generated. The larger the η value the more different solutions the algorithm visits. Therefore a fine tuning for η is important to generate a high quality initial solution for RTSH algorithm in shorter time with respect to $\eta = 1$.

Table 5.1: CMT problem tests for parameter settings (1st stage)

CMT Problem	NI	NI + LPO		C&W	C&W + LPO	
		$\eta = 0.4$	$\eta = 0.6$		$\eta = 0.4$	$\eta = 0.6$
p01	663.21	607.39	607.39	584.64	568.21	560.46
p02	983.27	976.14	976.14	907.39	891.85	887.71
p03	905.38	903.27	903.27	886.83	867.21	867.21
p04	1325.34	1310.41	1310.41	1133.43	1078.34	1078.34
p05	1562.76	1521.23	1514.87	1395.74	1304.56	1304.56
p06	754.24	708.79	708.79	618.39	578.94	578.94
p07	1149.81	1093.22	1086.65	975.46	947.81	947.81
p08	1219.66	1167.34	1167.34	973.94	948.73	948.73
p09	1314.95	1282.27	1282.27	1287.64	1273.98	1245.64
p10	1775.11	1713.86	1713.86	1538.66	1454.37	1427.83
p11	1265.94	1207.92	1207.92	1071.07	1046.57	1046.57
p12	1053.74	986.61	964.33	833.51	821.11	821.11
p13	1781.85	1687.96	1646.21	1592.26	1602.56	1568.76
p14	1035.17	984.29	984.29	875.75	871.55	871.55

Table 5.2: CMT-14 problem tests for parameter settings (2nd stage)

CMT Problem	NI + RTSH	NI + LPO + RTSH		C&W + RTSH	C&W + LPO + RTSH	
		$\eta = 0.4$	$\eta = 0.6$		$\eta = 0.4$	$\eta = 0.6$
p01	595.39	580.46	580.46	584.64	**548.21**	**548.21**
p02	934.36	896.73	896.73	907.39	**870.44**	**870.44**
p03	872.41	871.39	871.39	886.83	**854.78**	**854.78**
p04	1286.63	1211.82	1211.82	1133.43	1082.21	**1078.34**
p05	1451.82	1419.93	1419.93	1395.74	**1304.56**	**1304.56**
p06	670.16	607.89	607.89	618.39	**578.94**	**578.94**
p07	924.76	987.14	987.14	975.46	963.96	**947.81**
p08	1108.73	1087.51	1087.51	973.94	**948.73**	**948.73**
p09	1263.34	1258.37	1258.37	1287.64	1257.45	**1245.64**
p10	1687.65	1648.77	1648.77	1538.66	**1427.83**	**1427.83**
p11	1153.69	1107.02	1107.02	1071.07	**1046.57**	**1046.57**
p12	949.56	906.91	906.91	833.51	**821.11**	**821.11**
p13	1542.27	1597.26	1597.26	1592.26	1565.84	**1563.57**
p14	965.48	921.39	921.39	875.75	**867.34**	**867.34**

Two main algorithms are tested; Nearest Insertion (NI) and Clarke & Wright Parallel Savings Algorithm (C&W). C&W preforms better than NI for all test problems in the first stage. In the second stage RTSH and LPO versions are tested. To decide the η parameter, two values are tested individually; these two η values (0.4 and 0.6) are the most used η values in the literature. The η having the value of 0.6 performs better in the proposed RTSH algorithm. The European and the Anatolian side waste collection problems are solved with RTSH and the initial solution is generated by C&W+LPO algorithm.

Benchmark Problems

In this stage the quality of the proposed algorithm RTSH is tested on the benchmark problems set CMT-14. This is one of the most widely used symmetric VRP test bed. The datasets are available at the website of VRP Web with the best known solutions. (VRP Web)

This problem set has symmetric distances, and the general definition of these problems is CVRP. The only difference with our real world problem is the distances symmetric property.

Table 5.3: Computational results for CMT-14 problems

Instance*	Problem	Laporte-Semet C&W	C&W	C&W+LPO	RTS	RTSH
p01	E051 - 05e	584.64	584.64	560.46	560.46	548.21
p02	E076 - 10e	900.26	907.39	896.3	887.71	870.44
p03	E101 - 08e	886.83	886.83	879.21	878.43	854.78
p04	D121 - 11c	1133.43	1133.43	1121.96	1121.96	1078.34
p05	D101 - 11c	1395.74	1395.74	1383.37	1383.37	1304.56
p06	E101 - 10c	618.4	618.39	609.58	608.87	578.94
p07	E121 - 07c	975.46	975.46	967.52	967.52	947.81
p08	E151 - 12c	973.94	973.94	964.25	961.42	948.73
p09	E200 - 17c	1287.64	1287.64	1281.16	1280.49	1245.64
p10	D051 - 06c	1538.66	1538.66	1516.54	1516.54	1427.83
p11	D200 - 18c	1071.07	1071.07	1046.93	1046.57	1046.57
p12	D151 - 14c	833.51	833.51	821.29	821.11	821.11
p13	D076 - 11c	1596.72	1592.26	1577.53	1571.13	1563.57
p14	D101 - 09c	875.75	875.75	867.76	867.34	867.34

*CMT Problem Set

The results are not very impressing since the algorithm proposed asymmetric distances the merge function executes in different order when considered with symmetric version. This problem set is designed with symmetric distances therefore the algorithm proposed is not extremely good in these problems.

Same route 3-Opt algorithm’s performance is tested on the CMT-14 benchmark problems. 3-Opt performs better on the large sized problems rather than small or medium sized problems. The results of benchmarks are proposed below.

Table 5.4: 3-Opt (same route) computational results for CMT-14 problems

Instance*	Problem	Laporte-Semet C&W	C&W	C&W+LPO	RTS	3-Opt Added RTSH
p01	E051 - 05e	584.64	584.64	560.46	560.46	560.46
p02	E076 - 10e	900.26	907.39	896.3	887.71	887.71
p03	E101 - 08e	886.83	886.83	879.21	878.43	854.78
p04	D121 - 11c	1133.43	1133.43	1121.96	1121.96	1078.34
p05	D101 - 11c	1395.74	1395.74	1383.37	1383.37	1304.56
p06	E101 - 10c	618.4	618.39	609.58	608.87	578.94
p07	E121 - 07c	975.46	975.46	967.52	967.52	947.81
p08	E151 - 12c	973.94	973.94	964.25	961.42	948.73
p09	E200 - 17c	1287.64	1287.64	1281.16	1280.49	1245.64
p10	D051 - 06c	1538.66	1538.66	1516.54	1516.54	1516.54
p11	D200 - 18c	1071.07	1071.07	1046.93	1046.57	1046.57
p12	D151 - 14c	833.51	833.51	821.29	821.11	821.11
p13	D076 - 11c	1596.72	1592.26	1577.53	1571.13	1568.76
p14	D101 - 09c	875.75	875.75	867.76	867.34	867.34

*CMT Problem Set

Another benchmark study is proposed for 1-1-1 Rotation operator. This operator selects three different nodes from three different routes then rotates them. The algorithm’s performance on CMT-14 benchmark problems are proposed below.

Table 5.5: 1-1-1 computational results for CMT-14 problems

Instance	Problem	Laporte-Semet C&W	C&W	C&W+LPO	RTS	1-1-1 Inserted RTSH
p01	E051 - 05e	584.64	584.64	560.46	560.46	560.46
p02	E076 - 10e	900.26	907.39	896.3	887.71	887.71
p03	E101 - 08e	886.83	886.83	879.21	878.43	878.43
p04	D121 - 11c	1133.43	1133.43	1121.96	1121.96	1121.96
p05	D101 - 11c	1395.74	1395.74	1383.37	1383.37	1364.74
p06	E101 - 10c	618.4	618.39	609.58	608.87	608.87
p07	E121 - 07c	975.46	975.46	967.52	967.52	967.52
p08	E151 - 12c	973.94	973.94	964.25	961.42	956.21
p09	E200 - 17c	1287.64	1287.64	1281.16	1280.49	1267.28
p10	D051 - 06c	1538.66	1538.66	1516.54	1516.54	1504.81
p11	D200 - 18c	1071.07	1071.07	1046.93	1046.57	1046.57
p12	D151 - 14c	833.51	833.51	821.29	821.11	821.11
p13	D076 - 11c	1596.72	1592.26	1577.53	1571.13	1571.13
p14	D101 - 09c	875.75	875.75	867.76	867.34	867.34

*CMT Problem Set

Multi successive 1-0 move is coded to maximize the savings by forcing route deletion. When a route is deleted because of LF effect the saving is generally larger than a normal 1-0 move. The benchmark study is proposed for CMT-14 test instances and the results are shown below.

Table 5.6: Multi Successive 1-0 move computational results for CMT-14 problems

Instance	Problem	Laporte-Semet C&W	C&W	C&W+LPO	RTS	Multi Succ. 1-0 Move Added RTSH
p01	E051 - 05e	584.64	584.64	560.46	560.46	548.21
p02	E076 - 10e	900.26	907.39	896.3	887.71	887.71
p03	E101 - 08e	886.83	886.83	879.21	878.43	854.78
p04	D121 - 11c	1133.43	1133.43	1121.96	1121.96	1078.34
p05	D101 - 11c	1395.74	1395.74	1383.37	1383.37	1304.56
p06	E101 - 10c	618.4	618.39	609.58	608.87	578.94
p07	E121 - 07c	975.46	975.46	967.52	967.52	947.81
p08	E151 - 12c	973.94	973.94	964.25	961.42	948.73
p09	E200 - 17c	1287.64	1287.64	1281.16	1280.49	1245.64
p10	D051 - 06c	1538.66	1538.66	1516.54	1516.54	1427.83
p11	D200 - 18c	1071.07	1071.07	1046.93	1046.57	1046.57
p12	D151 - 14c	833.51	833.51	821.29	821.11	821.11
p13	D076 - 11c	1596.72	1592.26	1577.53	1571.13	1563.57
p14	D101 - 09c	875.75	875.75	867.76	867.34	867.34

*CMT Problem Set

GAMS Test Runs for European and Anatolian Side

In this section we present the test results of the heuristics developed for the CVRP-IF, and the comparison of GAMS and the algorithm proposed. The codes of the algorithms were written in C and built in Microsoft Visual Studio 2008. Run times were measured on a workstation which has 64-bit Windows 7 Professional operating system. The workstation is equipped with one Intel Xeon W3690 3.46 GHz Hexa-Core processor and 24 GB DDR3 ECC memory.

Both problems are converted to GAMS codes by combining the land-fill facility and the final destination, garage as one node. The aim is to code general VRP and solve the problems by changing only the structures of the problems. Table 5.7 and 5.8 are the comparison tables, all the CPU times have been reported in seconds.

Table 5.7: GAMS Test Results – European Side

Veh. Type	Shift Length	UB_{CPLEX}	LB_{CPLEX}	CPLEX Gap (%)	RTSH	CPLEX – RTSH Gap (%)	CPU_{CPLEX} (s)	CPU_{RTSH} (s)
1	7 hrs	1987.6	1336.1	32.8	1949.1	2.8	14400	3.21
	8 hrs	1729.8	1162.7	32.8	1599.3	11.2	14400	3.25
	9 hrs	1621.3	1041.9	35.8	1439.5	17.4	14400	3.15
2	7 hrs	2119.4	967.2	54.4	1907.9	23.0	14400	3.24
	8 hrs	1746.4	965.3	44.8	1620.8	15.3	14400	3.22
	9 hrs	1499.9	964.2	35.8	1400.2	13.4	14400	3.24
3	7 hrs	2151.3	882.5	58.9	1901.6	34.5	14400	3.19
	8 hrs	1721.4	881.3	48.8	1523.7	25.3	14400	3.21
	9 hrs	1464.1	881.4	39.8	1391.2	11.3	14400	3.22

Veh. Type 1: 1500 kg capacity
Veh. Type 2: 2000 kg capacity
Veh. Type 3: 2300 kg capacity

For European Side problem, the results are given in the Table 5.7. The GAMS/CPLEX results and the RTSH results for each problem are proposed with the improvement rates.

$$\text{CPLEX} - \text{RTSH Gap} = \frac{UB_{CPLEX} - RTSH\ Result}{LB_{CPLEX}}\ x\ 100\%$$

Table 5.8: GAMS Test Results – Anatolian Side

Veh. Type	Shift Length	UB_{CPLEX}	LB_{CPLEX}	CPLEX Gap (%)	RTSH	CPLEX – RTSH Gap (%)	CPU_{CPLEX} (s)	CPU_{RTSH} (s)
	7 hrs	1962.5	947.2	51.8	1955.4	0.7	14400	2.41
1	8 hrs	1698.2	945.4	44.3	1621.5	8.1	14400	2.43
	9 hrs	1408.9	941.2	33.2	1461.7	5.0	14400	2.41
	7 hrs	1754.2	781.3	55.5	1907.9	6.0	14400	2.46
2	8 hrs	1677.8	752.5	55.1	1620.8	7.5	14400	2.42
	9 hrs	1524.3	745.8	51.1	1400.2	16.6	14400	2.42
	7 hrs	1685.5	726.9	56.8	1901.6	11.5	14400	3.43
3	8 hrs	1608.1	668.9	58.4	1523.7	12.6	14400	2.41
	9 hrs	1504.2	674.7	55.1	1391.2	16.7	14400	2.42

Veh. Type 1: 1500 kg capacity
Veh. Type 2: 2000 kg capacity
Veh. Type 3: 2300 kg capacity

For Anatolian Side problem, the results are given in the Table 5.8. The GAMS/CPLEX results and the RTSH algorithm results for each problem are proposed with the improvement rates.

$$\text{CPLEX} - \text{RTSH Gap} = \frac{UB_{CPLEX} - RTSH\ Result}{LB_{CPLEX}}\ x\ 100\%$$

In the Table 5.7 GAMS/CPLEX and RTSH comparison is proposed for European side sub-problem. RTSH performs better for all CMT-14 problems. And the execution times for each problem sharply differ. While RTSH algorithm can solve the problems in a couple of seconds GAMS/CPLEX needs hours to get close to RTSH solution.

5.2. Application Results

In this part of the study the application results for Istanbul Metropolitan Municipality is examined.

İstanbul Metropolitan Municipality collects medical waste from only the hospitals those have minimum 20 bed capacity. The average daily waste collection amount are stated changing in a large range, but in this project the actual amounts of 2009 are used to estimate daily waste amounts and resulted a total daily waste collection of about 13 tons for Anatolian side and 23 tons for European side.

European Commission has a Life Program that considers waste management in the regarding city. A handbook is designed within this project to state the current situation and to create new business opportunities. The medical waste for bed in İstanbul is given as 0.6 kg per day, results as about 5,000 of waste bags a day in İstanbul. Burning and reforming waste is the final work after collection, medical waste is transformed into compost enables the re-usage of the medical waste. Waste volume is decreased 95% and mass decrease of waste is 75% after transformation. The collection, transportation and burning phase costs about 400 dollars for one ton of medical waste in İstanbul. This cost does not include the vehicle fixed costs therefore the vehicle fixed costs are considered in the application phase of this study. (European Commission, 2011)

Since the distance matrix is not symmetric whole study is based on asymmetric distances. The problem is divided into two sub-problems, the Anatolian side and the European side. This chapter firstly gives the results of both two sub-problems then analyzes the results in economic scale.

Each route designed to start from garage collect waste from the hospitals, visit and drop the waste at the landfill then go back to the garage at the end of the day. So *"going back to garage"* is considered in each route as for time consuming extra travel before the shift ends.

Both problems are solved for different type of vehicles, the fleets are homogeneous but the type of vehicles is changed and the results are analyzed.

5.2.1. Anatolian Side Sub-Problem

The Anatolian side sub-problem has 88 hospitals, 1 garage and 1 landfill. Anatolian side sub-problem results are given below;

Table 5.9 demonstrates the results for the entire sub problems generated. For 3 different vehicle type and shift length each solution shown in distances, # of vehicles used and the average # of vehicles on each route.

Table 5.10 gives a brief cost analysis depending on the (European Commission, 2011) study. In this study the cost of collecting 1 ton of waste is given, by adding the drivers' daily wages each solutions' cost analysis is provided.

Table 5.9: Anatolian side distance analysis

Veh. Type	Shift Length	Nearest Insertion	Veh. Used	Avg # Nodes	C&W	Veh. Used	Avg # Nodes	C&W+ LPO	Veh. Used	Avg # Nodes	RTSH	Veh. Used	Avg #Nodes
1	7 hrs	2828.7	29	3.03	2432.3	25	3.5	2386.21	25.0	3.5	2127.46	22.0	4.0
	8 hrs	2407.7	24	3.66	1966.0	20	4.4	1923.96	20.0	4.4	1583.06	16.0	5.5
	9 hrs	1954.8	19	4.63	1601.2	16	5.5	1579.01	15.0	5.8	1491.31	15.0	5.8
	AVG	**2397.1**	**24**	**3.77**	**1999.8**	**20.3**	**4.5**	**1963.06**	**20.0**	**4.6**	**1733.94**	**17.6**	**5.1**
2	7 hrs	2828.7	29	3.03	2432.3	25	3.5	2047.06	21.0	4.2	1958.16	20.0	4.4
	8 hrs	2407.7	24	3.66	1966	20	4.4	1737.41	18.0	4.9	1576.66	16.0	5.5
	9 hrs	1954.8	19	4.63	1601.2	16	5.5	1493.29	15.0	5.8	1481.19	15.0	5.8
	AVG	**2397.1**	**24**	**3.77**	**1999.8**	**20.3**	**4.5**	**1759.25**	**18.0**	**4.9**	**1672.00**	**17.0**	**5.2**
3	7 hrs	2828.7	29	3.03	2432.3	25	3.5	2384.66	25.0	3.5	1632.36	18.0	4.9
	8 hrs	2407.7	24	3.66	1966	20	4.4	1735.16	18.0	4.8	1489.99	15.0	5.8
	9 hrs	1954.8	19	4.63	1601.2	16	5.5	1493.29	15.0	5.8	1479.79	15.0	5.8
	AVG	**2397.1**	**24**	**3.77**	**1999.8**	**20.3**	**4.5**	**1871.04**	**19.3**	**4.7**	**1534.05**	**16.0**	**5.5**

Veh. Type 1: 1500 kg capacity
Veh. Type 2: 2000 kg capacity
Veh. Type 3: 2300 kg capacity

There are three types of vehicles and for each vehicle type three shift lengths are considered. Cost analysis is given below;

Table 5.10: Anatolian side cost analysis

Veh. Type	Shift Length	NI	Veh. Used	C&W	Veh. Used	C&W + LPO	Veh. Used	RTSH	Veh. Used
1	7 hrs	2720	29	2400	25.0	2400	25.0	2160	22.0
	8 hrs	2320	24	2000	20.0	2000	20.0	1680	16.0
	9 hrs	1920	19	1680	16.0	1600	15.0	1600	15.0
	AVG	**2320**	**24**	**2027**	**20.3**	**2000**	**20.0**	**1813**	**17.6**
2	7 hrs	2720	29	2400	25.0	2400	25.0	2000	20
	8 hrs	2320	24	2000	20.0	1840	18.0	1680	16
	9 hrs	1920	19	1680	16.0	1600	15.0	1600	15
	AVG	**2320**	**24**	**2027**	**20.3**	**1947**	**19.3**	**1760**	**17**
3	7 hrs	2720	29	2400	25.0	2080	21.0	1840	18
	8 hrs	2320	24	2000	20.0	1840	18.0	1600	15
	9 hrs	1920	19	1680	16.0	1600	15.0	1600	15
	AVG	**2320**	**24**	**2027**	**20.3**	**1840**	**18.0**	**1680**	**16**

Veh. Type 1: 1500 kg capacity
Veh. Type 2: 2000 kg capacity
Veh. Type 3: 2300 kg capacity

From cost perspective there is no difference between using vehicle type 3 with 8 hours of shift length or 9 hours of shift length. So vehicle type 3 can be used with both 8 and 9 hours of shift lengths.

5.2.2. European Side Sub-Problem

European side sub-problem has 158 hospitals, 1 garage and 1 landfill. The results for European side are given in Table 5.11.

Table 5.11 demonstrates the results for the entire sub problems generated. For 3 different vehicle type and shift length each solution shown in distances, # of vehicles used and the average # of vehicles on each route.

Table 5.12 gives a brief cost analysis depending on the (European Commission, 2011) study. In this study the cost of collecting 1 ton of waste is given, by adding the drivers' daily wages each solutions' cost analysis is provided.

Table 5.11: European side distance analysis

Veh. Type	Shift Length	NI + LPO	Veh. Used	Avg # Nodes	C&W	Veh. Used	Avg # Nodes	C&W+ LPO	Veh. Used	Avg # Nodes	RTSH	Veh. Used	Avg # Nodes
1	7 hrs	2311.5	28.0	5.64	1984.9	26	6.07	1955.4	26.0	6.07	1949.1	26.0	6.07
	8 hrs	1973	23.0	6.86	1653.8	21	7.52	1621.5	21.0	7.52	1599.3	21.0	7.52
	9 hrs	1812.4	21.0	7.52	1497.5	18	8.77	1461.7	18.0	8.77	1439.5	18.0	8.77
	AVG	**2032.3**	**24.0**	**6.67**	**1712.1**	**21.6**	**7.45**	**1679.5**	**21.6**	**7.45**	**1662.6**	**21.6**	**7.45**
2	7 hrs	2311.5	28.0	5.64	1984.9	26	6.07	1907.9	25.0	6.32	1896	25.0	6.32
	8 hrs	1973	23.0	6.86	1653.8	21	7.52	1620.8	21.0	7.52	1598.2	21.0	7.52
	9 hrs	1814.5	21.0	7.52	1446.0	17	9.29	1400.2	17.0	9.29	1370.1	17.0	9.29
	AVG	**2033**	**24.0**	**6.67**	**1694.9**	**21.3**	**7.62**	**1642.9**	**21.0**	**7.71**	**1621.4**	**21.0**	**7.71**
3	7 hrs	2311.5	28.0	5.64	1984.9	26	6.07	1901.6	25.0	6.32	1846.6	24.0	6.58
	8 hrs	1973	23.0	6.86	1653.8	21	7.52	1523.7	19.0	8.31	1497.9	19.0	8.31
	9 hrs	1814.5	21.0	7.52	1446.0	17	9.29	1391.2	17.0	9.29	1363.7	17.0	9.29
	AVG	**2033**	**24.0**	**6.67**	**1694.9**	**21.3**	**7.62**	**1605.5**	**20.3**	**7.97**	**1569.4**	**20.0**	**8.06**

Veh. Type 1: 1500 kg capacity
Veh. Type 2: 2000 kg capacity
Veh. Type 3: 2300 kg capacity

Cost analysis is given below;

Table 5.12: European side cost analysis

Veh. Type	Shift Length	NI+LPO	Veh. Used	C&W	Veh. Used	C&W+ LPO	Veh. Used	RTSH	Veh. Used
1	7 hrs	11040	28	10880	26.0	10880	26.0	10880	26.0
	8 hrs	10640	23	10480	21.0	10480	21.0	10480	21.0
	9 hrs	10480	21	10240	18.0	10240	18.0	10240	18.0
	AVG	**10720**	**24**	**10533**	**21.6**	**10533**	**21.6**	**10533**	**21.6**
2	7 hrs	11040	28	10880	26.0	10800	25.0	10800	25.0
	8 hrs	10640	23	10480	21.0	10480	21.0	10480	21.0
	9 hrs	10480	21	10160	17.0	10160	17.0	10160	17.0
	AVG	**10720**	**24**	**10506**	**21.3**	**10480**	**21.0**	**10480**	**21**
3	7 hrs	11040	28	10880	26.0	10800	25.0	10720	24.0
	8 hrs	10640	23	10480	21.0	10320	19.0	10320	19.0
	9 hrs	10480	21	10160	17.0	10160	17.0	10160	17.0
	AVG	**10720**	**24**	**10506**	**21.3**	**1426**	**20.3**	**10400**	**20.0**

Veh. Type 1: 1500 kg capacity
Veh. Type 2: 2000 kg capacity
Veh. Type 3: 2300 kg capacity

Since the total number of hospitals in European side doubles the Anatolian side, the number of vehicles required to collect waste from all of the hospitals in European Side is more than the number of vehicles required to collect waste from all of the hospitals in Anatolian Side. The best solution with minimum cost is the solution that uses 17 vehicles of type 3 and the shift length is 9 hours a day.

Chapter 6

CONCLUSION AND FUTURE RESAERCH

In this study the medical waste management of İstanbul is considered and for both Anatolian and European sides routing structures are generated. As conclusion the vehicle type and tour duration affects the total cost. Since the cost is changed due to the total number of vehicles used the cost does not change for different versions of problems. In this part the consideration of CO_2 emission takes the lead. Total distances covered for each problem is related with the CO_2 emission, therefore within the minimum cost solutions the one that has the least distance required completing collection must be selected.

So both for Anatolian and European sides, fleet vehicles should have the capacity of 2300 kg and if available 9 hours of shift guides us for the routing structure with minimum cost and distances covered therefore less CO_2 emission.

The future research on to this study can be the *Time Windows* addition and/or *Heterogeneous Fleet* structure.

1. BIBLIOGRAPHY

Alagöz, A. Z., & Kocasoy, G. (2008). Management methods for the health-care wastes in Istanbul. *Waste Management* , 28:1227-35.

Alagöz, A., & Kocasoy, G. (2008). Improvement and modification of the routing system for the health-care waste collection and transportation in İstanbul. *Waste Management* , 28:1461-71.

Angelelli, E., & Speranza, M. G. (2002). The application of a vehicle routing model to a waste-collection problem: two case studies. *Journal of the Operational Research Society* , 53:944-52.

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

Bdour, A., Altrabsheh, B., Hadadin, N., & Al-Shareif, M. (2007). Assestment of medical wastes management practice: a case study of the Northern part of Jordan. *Waste Management* , 27:746-759.

Belien, J., De Boeck, L., & Van Ackere, J. (2011). Municipal Solid Waste Collection Problems: A Literature Review. *HUB Research Papers* , 34.

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

Benjamin, A., & Beasley, J. (2012). Metaheuristics with disposal facility positioning for waste collection VRP with time windows . *Optimization Letters* , 40:230-243.

Bodin, L., Mingozzi, A., Baldacci, R., & Ball, M. (2000). The Rollon-Rolloff Vehicle Routing Problem. *Transportational Science* , 34(3):271-288.

Buhrkal, K., Larsen, A., & Ropke, S. (2012). The waste collection vehicle routing problem with time windows in a city logistics content. *Procedia Social and Behavioral Sciences* , 39:241-254.

Coker, A., Sangodoyin, A., Sridhar, M., Booth, C., Olomolaiye, P., & Hammond, F. (2009). Medical waste management in Ibadan, Nigeria: Obstacles and prospects. *Waste Management* , 29:804-811.

Du, L., & He, R. (2012). Combining nearest neighbor search with tabu search for large-scale vehicle routing problem. *Physics Procedia* , 25:1536-46.

El-Salam, A. M. (2010). Hospital waste management in El-Beheria governorate, Egypt. *Environmental Management* , 91:618-629.

EuropeanCommission. (2011). *Solid Waste Management.* Istanbul: European Commission - Life Project.

Golden, B. L., Assad, A. A., & Wasil, E. A. (2001). Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries. In P. Toth, & D. Vigo, *The vehicle routing problem* (pp. 245-286). Philadelphia: Society for Industrial and Applied.

Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges.* Springer Vol.43.

Gröer, C., Golden, B., & Wasil, E. (2010). A Library of Local Search Heuristics for the Vehicle Routing Problem. *Mathematical Programming Computation* , 2:79-101.

Homberger, J., & Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operations Research* , 162:220-238.

Juliff, P. (1990). Program Design. In *Program Design.* Prentice-Hall:Australia.

Kim, B., Kim, S., & Sahoo, S. (2006). Waste collection vehicle routing problemwith time windows. *Computers & Operations Research* , 33:3624-42.

Kytojoki, J., Nuortio, T., Niska, H., Braysy, O., & Gendreau, M. (2007). An efficient variable neighborhood search heuristic for very large scale vahicle routing problems. *Computers & Operations Research* , 34:2743-57.

Nourito, T., J., K., Niska, H., & Braysy, O. (2006). Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications* , 30:223-232.

Nuortio, T., Kytojoki, J., Niska, H., & Braysy, O. (2006). Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications* , 30:223-32.

Osman, H. I. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* , 41:421-451.

Sahoo, S., Kim, S., Kim, B. I., Krass, B., & Popov, A. (2005). Routing optimization for waste management. *Interfaces* , 35:24-36.

Sawalem, M., Selic, E., & Herbell, J. D. (2009). Hospital waste management in Libya. *Waste Management* , 29:1370-75.

Shih, L., & Chang, H. (2001). A routing and scheduling system for infectious waste collection. *Environmental Modeling and Assesment* , 6:261-269.

Sigauke, C. (1994). A modified Osman's simulated annealing and tabu search algorithm for the vehicle routing problem.

Tamplin, S. A., Davidson, D., Powis, B., & O'Leary, Z. (2005). Issues and option for the safe distruction and disposal of used injection materials. *Waste Management* , 25:655-665.

Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem.*

Tung, D. V., & Pinnoi, A. (2000). Vehicle routing-scheduling for waste collection in Hanoi. *European Journal of Operational Research* , 125:449-68.

Wassan, N. A. (2006). A reactive tabu search for the vehicle routing problem. *Journal of the Operations Research Society* , 57:111-116.

Wassan, N. A. (2006). A reactive tabu search for the vehicle routing problem. *Journal of Operational Research Society* , 57:111-116.

VITA

Müge Güçlü was born in Istanbul, Turkey, on June 12, 1986. She graduated from Umraniye Anatolian High School 2004. She received his B.Sc degree in Industrial Engineering from Marmara University, Istanbul 2009. Same year, she joined the M.Sc program in Industrial Engineering at Koc University as a research and teaching assistant.