

**MODELING AND OPTIMIZATION OF CENTRAL RING
TRANSPORTATION SYSTEM (CRTS) IN TURKISH
LAND FORCES**

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL
ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF
MASTER OF SCIENCE

By

Hamdi Ünal Akmeşe

JULY,2003

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Sciences.

Assoc.Prof. Osman Oğuz (Principal Advisor)

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Sciences.

Assoc.Prof. Mustafa Pınar

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Sciences.

Asst.Prof. Oya Ekin Kardeşan

Approved for the Institute of Engineering and Sciences

Prof. Mehmet Baray

Director of Institute of Engineering and Sciences

ABSTRACT

MODELING AND OPTIMIZATION OF CENTRAL RING TRANSPORTATION SYSTEM (CRTS) IN TURKISH LAND FORCES

Hamdi Ünal Akmeşe

M.S. in Industrial Engineering

Advisor: Assoc. Prof. Osman Oğuz

July 2003

This thesis shows how Turkish Land Forces can optimally meet delivery and pick-up demands of its units via Central Ring Transportation System. A mixed integer programming model is proposed, and for the implementation of the model, mathematical modeling software GAMS is used. The model is implemented for three different fleet sizes of vehicles (4-vehicle, 5-vehicle, 6-vehicle) with taking eight-week data of 2002 into account. How transportation costs are affected by the number of vehicles is investigated, and an ideal number of vehicles and the optimal routes to be followed are proposed.

Keywords : Mixed Integer Programming, Capacitated Vehicle Routing, Time Windows, Transportation Costs.

ÖZET

KARA KUVVETLERİ MERKEZİ RİNG TAŞIMACILIĞI SİSTEMİNİN MODELLENMESİ VE OPTİMİZASYONU

Hamdi Ünal Akmeşe

Endüstri Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Doç. Dr. Osman Oğuz

Temmuz 2003

Bu çalışma Türk Kara Kuvvetlerinin kendi birliklerinin sevkiyat ve toplama taleplerini Merkezi Ring Taşımacılığı Sistemi vasıtası ile en uygun olarak nasıl karşılayabileceğini göstermektedir. Bunun için tamsayı programlama modeli önerilmiş ve bu modelin uygulanması için matematiksel modelleme yazılımı GAMS kullanılmıştır. 2002 yılına ait sekiz haftalık bilgiler gözönüne alınarak, model üç farklı sayıdaki araç filoları (4-araç, 5-araç, 6-araç) için çalıştırılmıştır. Taşıma maliyetlerinin araç sayısına bağlı olarak nasıl etkilendiği sorusuna yanıt bulunmaya çalışılmış ve ideal araç sayısı ile bunların izleyeceği ideal rotalar önerilmiştir.

Anahtar Kelimeler : Tamsayı Programlama, Kapasite Sınırlı Araç Güzergahı Bulma, Zaman Kısıtları, Taşıma Maliyetleri.

To my parents

ACKNOWLEDGEMENT

I would like to express my gratitude to Assoc.Prof. Osman Oğuz for his guidance, attention, interest and patience throughout our study.

I am indebted to the readers Assoc. Prof. Mustafa Pınar and Asst.Prof. Oya Ekin Karaşan for their effort, kindness and time.

I would like to thank Captain Ümit Ata Narin for his support to attend a MS program, guidance and understanding during the years in Bolu.

I am very thankful to Savas Çevik, Ömer Selvi, Güneş Erdoğan, Çağatay Kepek, Halil Şekerci, Duygu Pekbey for their assistance, support and friendship during the graduate study. I am also indebted to all Industrial Engineering Dept. staff.

I am grateful to Turkish Land Forces for giving the opportunity to continue my education in Bilkent University.

Finally, I am very thankful to my family for their love, tolerance and patience.

CONTENTS

List of Figures	X
List of Tables	XI
English-Turkish Meanings of Some Military Terms	XII
1.INTRODUCTION	1
1.1. Central Ring Transportation System (CRTS)	2
1.1.1. Provision Types of Land Forces	2
1.1.2. Current and Proposed CRTS	3
2. LITERATURE REVIEW	7
2.1. The TSP and <i>m</i>-TSP	7
2.1.1. Mathematical Formulations of <i>m</i> -TSP	8
2.1.2. Solution Methods for <i>m</i> -TSP	10
2.1.2.1.Exact Solution Methods for <i>m</i> -TSP	11
2.1.2.2.Heuristic Solution Methods for <i>m</i> -TSP	12
2.1.2.2.1.Tour Construction Heuristics	12
2.1.2.2.2.Tour Improvement Heuristics	13
2.1.2.3.Transformation to TSP	15
2.2.Vehicle Routing Problems (VRP)	15
2.2.1. Mathematical Formulations of VRP	16
2.2.2. Solution Methods for VRP	21
2.2.2.1.Exact Solution Methods for VRP	22
2.2.2.2.Heuristic Solution Methods for VRP	25

3.ANALYSIS OF THE PROBLEM and MATHEMATICAL FORMULATION	29
3.1. Data File	29
3.1.1. Main Depot and Garage	29
3.1.2. Points	30
3.1.3. Vehicles	30
3.1.4. Items	31
3.2. Assumptions	32
3.2.1. All Demands Must be Satisfied	32
3.2.2. All Demands are Fixed and Known in Advance	32
3.2.3. The Load/Unload Times are Fixed and Known in Advance	33
3.2.4. Time Windows are Fixed and Known in Advance	33
3.2.5. Vehicle Speed is Fixed and Known in Advance	33
3.2.6. The Number of Vehicle is Fixed and Known in Advance	33
3.3. Formulation	34
3.3.1. Indices	34
3.3.2. Initial Data and Parameters	34
3.3.3. Variables	35
3.3.4. Constraints	35
3.3.4.1. Assignment Constraints	35
3.3.4.2. Capacity Constraints	36
3.3.4.3. Load Feasibility Constraints	36
3.3.4.4. Time Feasibility Constraints	37
3.3.4.5. Non-Negativeness and Binary Variables	37
3.3.5. Objective Function	38
3.4. Experimentation	39

3.5. Results	41
3.5.1. Case 1: Fleets of Five Vehicles are Employed	42
3.5.2. Case 2: Fleets of Four Vehicles are Employed	43
3.5.3. Case 3: Fleets of Six Vehicles are Employed	44
4.CONCLUSION	48
4.1. Future Research Topics	49
Bibliography	50
Appendix	57
A. Data File	57
B. GAMS Code	66

List of Figures

Figure 1.1. Routes of the Current System.	4
Figure 1.2. Routes of Sample Problem Solved by Proposed Model.	5
Figure 2.1. Crossover.	27
Figure 2.2. Customer Relocation.	27
Figure 2.3. Customer Exchange.	27
Figure 2.4. Transformation to Four Depots.	28
Figure 3.1. Comparison of The Current System with The Proposed System in Distance for Weeks in Case 1.	42
Figure 3.2. Comparison of The Current System with The Proposed System in Cost for Weeks in Case 1.	43
Figure 3.3. Comparison of The Current System with The Proposed System in Distance for Weeks in Case 2.	44
Figure 3.4. Comparison of The Current System with The Proposed System in Cost for Weeks in Case 2.	45
Figure 3.5. Comparison of The Current System with The Proposed System in Distance for Weeks in Case 3.	46
Figure 3.6. Comparison of The Current System with The Proposed System in Cost for Weeks in Case 3.	46
Figure 3.7. Comparison of Cases in Distance for Weeks	46
Figure 3.8. Comparison of Cases in Cost for Weeks.	47

List of Tables

Table 1.1. Demanding Points of the Sample Problem.	5
Table 3.1. Capacities of The Vehicles	31
Table 3.2. Required CPU Times, Number of Iterations and Number of Nodes When Five Vehicles are Employed in the Model.	40
Table 3.3. Required CPU Times, Number of Iterations and Number of Nodes When Four Vehicles are Employed in the Model.	41
Table 3.4. Required CPU Times, Number of Iterations and Number of Nodes When Six Vehicles are Employed in the Model.	41
Table 3.5. Total Distances, Costs and Savings for Weeks in Case1.	42
Table 3.6. Total Distances, Costs and Savings for Weeks in Case 2.	43
Table 3.7. Total Distances, Costs and Savings for Weeks in Case 3.	45
Table A.1. Demand (pickup/delivery) Quantities of Points for Week 1.	57
Table A.2. Demand (pickup/delivery) Quantities of Points for Week 2.	58
Table A.3. Demand (pickup/delivery) Quantities of Points for Week 3.	59
Table A.4. Demand (pickup/delivery) Quantities of Points for Week 4.	60
Table A.5. Demand (pickup/delivery) Quantities of Points for Week 5.	61
Table A.6. Demand (pickup/delivery) Quantities of Points for Week 6.	62
Table A.7. Demand (pickup/delivery) Quantities of Points for Week 7.	63
Table A.8. Demand (pickup/delivery) Quantities of Points for Week 8.	64
Table A.9. Distance Between Points	65

ENGLISH-TURKISH MEANINGS OF SOME MILITARY TERMS

Turkish Land Forces: Türk Kara Kuvvetleri

Corps: Kolordu

Division: Tümen

Brigade: Tugay

Branch: Askeri sınıf (Ordudonatım, muhabere, vb.)

Ordnance Corps: Ordudonatım sınıfı

Signal Corps: Muhabere sınıfı

Engineers Corps: İstihkam sınıfı

Quartermaster Corps: Levazım sınıfı

Mid-Level Auto Transportation Battalion: Orta Ulaştırma Oto Taburu

CHAPTER 1

INTRODUCTION

Turkish Land Forces have implemented many modifications in its structure to catch up with technological development in recent years. In the context of these modifications, Turkish Land Forces have started projects about logistics that plays an important role for the Land Forces to perform its duty written in the Constitution. Establishing a unit managing all logistic efforts was one of these projects started. For that purpose, Logistics Command was established in 1988.

Logistics, which stems from the word “*logisticus*” belonging to Medieval Latin, means “the planning and control of the flow of goods and materials through an organization”[19]. A logistic system can only be achieved on the basis of a functional, economical and effective transportation system. Transportation is a technical service, which arranges the movements and replacements of the units, personnel and items regarding to requirements of the management and control of the operations and logistics. For this purpose, Transportation Command was formed with the order of Logistics Command, which is in charge of logistics on behalf of Turkish Land Forces. It is the mission of Transportation Command to meet the transportation demands of the units of the Land Forces in time with economical and secure means.

Transportation Command uses three modes of transportation to perform its task. These are Airway Transportation, Railway Transportation and Ring (Highway) Transportation. A modern transportation concept is based on the principle of delivering goods from an address to demanding address or picking up goods from demanding address to another address. Recently, Railway Transportation and Ring Transportation have been in preference among the transportation systems in Land Forces since they are more economical and secure.

Ring Transportation system is composed of Central Ring Transportation System (CRTS) and Regional Ring Transportation System (RRTS). CRTS satisfies transportation demands through Logistics Command's depots, military factories and armies whereas RRTS meets transportation demands inside of Armies' and Corps' own responsibility areas. In order to run CRTS, a Mid-Level Auto Transportation Battalion Command was formed in Yenikent, Ankara in 2001. On the other hand, RRTS is run locally by its own Light-Level Auto Transportation units of Army/Corps/Division/Brigade.

1.1 CENTRAL RING TRANSPORTATION SYSTEM (CRTS)

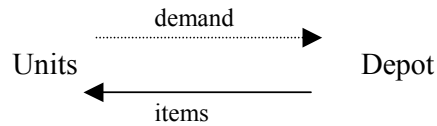
CRTS's mission is to deliver items from Main Depot to points (units, depots, factories) and to pick up items from points to bring to Main Depot. Delivering to points is called as Provision. The existing types of provision are explained in Section 1.1.1. Picking up depends on the demand points requiring those items in their own areas be sent to Main Repair Factories in Ankara through Main Depot, i.e., to send a defective machine, weapon and ammunition for repair.

As mentioned above, CRTS is performed by Mid-Level Auto Transportation Battalion Command in Yenikent, where Main Depot of Logistics Command is also situated. Related to The Joint Support Concept of Land Forces; ordnance, signal, engineer and quartermaster corps are combined under one unit to provide cooperative purchase, sharing depot resources and combining transportation efforts. So, Main Depot was formed in Yenikent in addition to depots in Army/Corps regions.

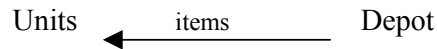
1.1.1 Provision Types of Land Forces

There are mainly two types of provision in military.

Pull System: Provision is provided when a demand occurs.



Push System: Provision is provided periodically. Logistics Command makes plans to send items to the units periodically. There is no need for the units to make requests for items.



In order to prevent system faults in both systems and to control the provision, Material Management System cooperating with Depot Managements in Logistics Command was established to manage all branches of items in Main Depot and other depots.

1.1.2 The Current and The proposed CRTS

In the current CRTS, five fixed routes are performed every week. The points representing the units, depots and factories on routes were selected by Transportation Command regarding size , importance, function and previous demands. All demands (pick-up/delivery) must be satisfied in three days following the vehicles' departure.

CRTS delivers to the points in two occasions. First, points can request any item from Main Depot at any time and inform Mid-Level Auto Transportation Battalion Command and Main Depot Management (Pull System). Then, CRTS delivers to these points. Secondly, Points are delivered periodically. Main Depot knows when to deliver to the points without being informed (Push System). The word “delivery demand” will stand for two occasions in the following parts of our study. The points having delivery demand can be named as Linehaul points.

CRTS also picks up items from the points. The points make “pick-up demand” when they have any item to send to Main Depot or to the factories in Ankara.

In the current system, points have a tendency not to inform Mid-Level Auto Transportation Battalion Command and Main Depot Management about their pick-up demands since points are sure to be visited. This effects vehicle utilization and creates problems in transportation. Figure 1.1 shows the current CRTS. The points having pick-up demand can be named as Backhaul points.

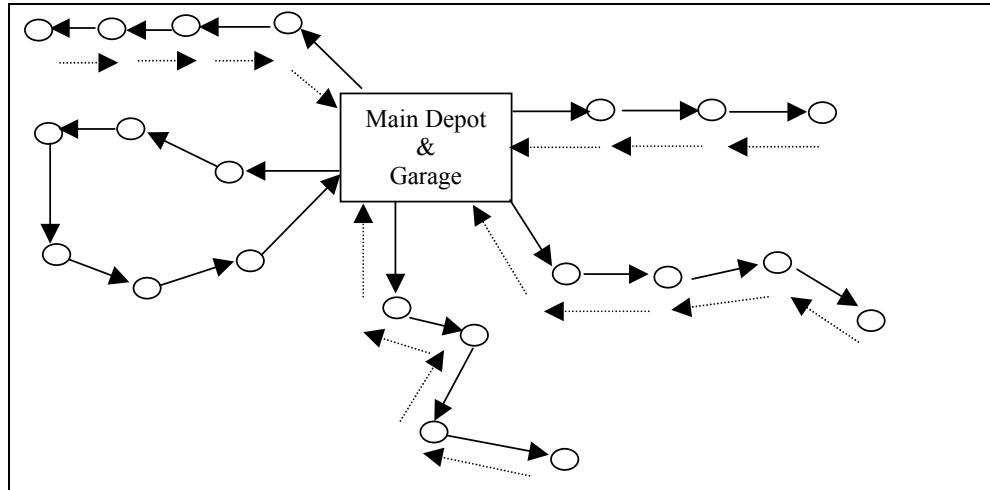


Figure 1.1 Routes of the Current System

Each vehicle is assigned to a tour and then loaded in Main Depot after Mid-Level Auto Transportation Battalion Command and Main Depot Management have gotten information of delivery demand quantities of the linehaul points on vehicles' routes. The vehicle fleet loaded with delivery demands come back to Garage with the load that they picked up from the backhaul points after completing the routes.

As seen in Fig.1.1, one of the routes is a cycle and others are two-way trips on a route. Fifteen points included on the two-way routes are visited twice a week while six points included by the cyclic route are visited once. This application brings fixed travel distance and fixed cost for each week since five fixed routes are performed even if some points on the routes have zero demand. It is clear that this fixed travel distance (cost) includes unnecessary distance (cost) because of re-visiting some points and ignoring non-demanding points.

Our aim is to develop a mathematical programming model to provide a better operational system for CRTS. In our mathematical model, only demand points are visited and these points are visited only once a route. To give an example, let us assume that fourteen points have demands (pick-up/delivery) and these points inform Mid-Level Auto Transportation Battalion Command and Main Depot Management about their demand quantities. The other characteristics of the problem (volume&weight quantities, time windows, vehicle capacities, speed, load/unload times) are explained in details in Chapter 3. We only aim to show the improvement in travel distance (cost) in this part. Table 1.1 shows these demanding (pick-up/delivery) points.

	1	2	3	4	5	6	7	8	9	10	11
Delivery	+		+		+	+	+	+	+		
Pick-up	+		+		+	+	+	+	+		

	12	13	14	15	16	17	18	19	20	21	
Delivery	+	+	+		+			+	+	+	
Pick-up	+	+	+		+			+	+	+	

Table 1.1 Demanding Points of the Sample Problem

Optimal solution of our Mixed-Integer Programming Model is shown in Figure 1.2. The points in black are demanding points. To comply with the current system, we employed five vehicles.

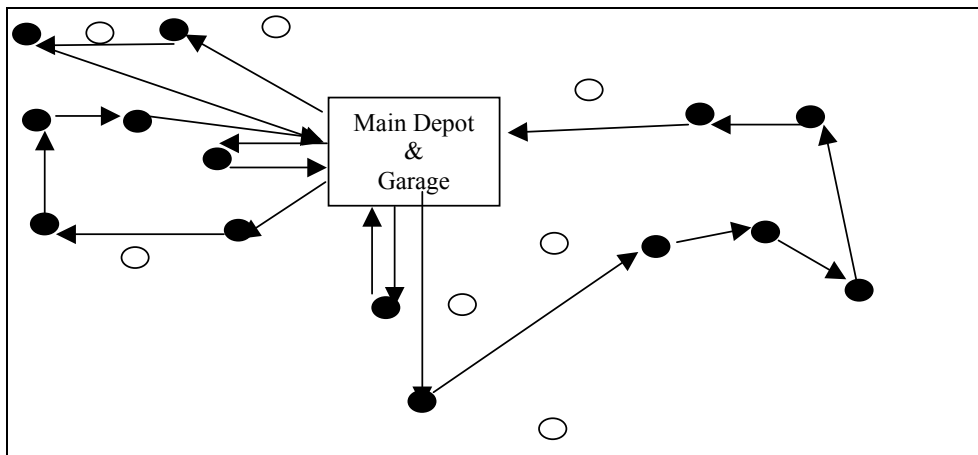


Figure 1.2 Routes of Sample Problem Solved by The proposed Model.

Vehicle fleet visits only demanding customers and just once. This application enables us to get rid of unnecessary distance (cost). It can easily be seen that less than five vehicles are sufficient to meet the demand in the sample problem. So, instead of employing fleets of five vehicles every week, we can also determine the vehicle fleet size giving the optimal solution by considering the number of demanding points.

In this thesis, our goal is to investigate how the total distance (cost) is affected when the proposed system for CRTS is employed. We have developed a mixed-integer programming model for solving this problem. The model determines which routes will be followed, and assigns demands (pick-up/delivery) to vehicles. Its objective is to minimize distance and transportation costs. The model is solved for eight-week data for three cases. To compare the proposed system with the current system, the model is solved for a fleet of five vehicles in Case 1. In Case 2 and Case 3, the model is solved for fleets of four and six vehicles fleet, respectively. To determine the most suitable vehicle size to satisfy the demands of points belonging to these eight-week data, cases are compared according to the total distance and cost.

Assumptions and details of the model are given in Chapter 3.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we give a brief review of the literature on problem related to vehicle routing problem, starting with the traveling salesman problem (TSP) and its variations.

2.1 The TSP and m -TSP

The Traveling Salesman Problem (TSP) is one of the most widely studied combinatorial optimization problems. TSP has interested many mathematicians and computer scientists specifically, because it is so easy to describe and so difficult to solve. Its statement is deceptively simple and yet it remains one of the most challenging problems in Operational Research. Hundreds of articles have been written on TSP. The reader is directed to Lawler et al [41] and Burkard [9], for a comprehensive survey of major research studies up to now.

The first use of the term “traveling salesman problem” in mathematical circles goes back to 1832. In that year, a book was printed in Germany entitled “*Der Handlungsreisende, wie er sein soll und was er zu Thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiss zu sein. Von einem alten Kommiss-Voyageur*” (“The Traveling Salesman , how he should be and what he should do to get Commissions and to be Successful in his Business. By a veteran Traveling Salesman”). Although devoted for the most part to other issues, the book reaches the essence of the TSP in its last chapter: ‘By a proper choice and scheduling of the tour, one can often gain so much time that we have to make some suggestion... The most important aspect is to cover as many locations as possible without visiting a location twice...’(Voigt, 59). We don’t know who brought the name TSP into mathematical circles, but there is no question that Merrill Flood [23] is most responsible for publicizing it within that community and the operations research fraternity as well. In 1948, Flood

popularized the TSP at the RAND Corporation, at least partly motivated by the purpose of creating intellectual challenges for models outside the theory of games. A prize was offered for a significant theorem bearing on TSP.

Another reason for the popularity of the problem was its intimate connection with prominent topics in combinatorial problems arising in the new subject of linear programming. As mentioned before, the TSP was like those other problems, but apparently harder to solve, and the challenge became an intriguing issue.

In its most basic form, TSP is the problem of finding the shortest route in a given graph which starts and ends at the same node, called the origin, with all the other nodes visited exactly once.

A generalized version of the well-known TSP is the Multiple Traveling Salesman Problem (m -TSP). Furthermore, the characteristics of the m -TSP seem more appropriate for real life applications where there is a need for more than one salesman.

2.1.1 Mathematical Formulation of the m -TSP

The m -TSP is the problem of finding a set of routes for m salesmen, such that the total distance traveled or the cost is minimized. Consequently, in a feasible solution, there will be m distinct tours all containing the origin such that all the nodes are covered.

Let $G = (V, A)$ be a graph where $V = \{1, 2, \dots, n\}$ is the set of nodes (vertices) and $A = \{\{i, j\} : i, j \in V, i \neq j\}$ is the set of edges, and let $C = (c_{ij})$ be a distance (cost) matrix associated with A . The m -TSP, then, is the problem of finding the minimum cost circuits in G . (Circuit is a graph cycle (i.e., closed loop) through a graph that visits each node exactly once.) The distance (cost) matrix mentioned above can be of different types. It will be useful to distinguish between the cases where C is symmetric, i.e. when $c_{ij} = c_{ji}$ for all $i, j \in V$, and

the case where it is asymmetric, i.e. when $c_{ij} \neq c_{ji}$ for some $i, j \in V$. C is said to satisfy the triangle inequality if and only if $c_{ij} + c_{jk} \geq c_{ik}$ for all $i, j, k \in V$. This occurs in Euclidean Problems, i.e. when V is a set of points in R^2 and c_{ij} is the straight-line distance between j and i .

The m -TSP is generally formulated as an integer linear program (ILP) based on the assignment model as follows. Let,

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i, j) \text{ is in the tour} \\ 0, & \text{otherwise} \end{cases}$$

x_{ij} is a 0-1 variable indicating whether or not a salesman goes directly from node i to city j . Then we would like to find x_{ij} which are to become 1, i.e. finding the edges that salesman should go through, for the distance traveled or cost to be minimized.

Miller, Tucker and Zemlin's Formulation

The first Integer Linear Programming formulation (ILP) for the m -TSP was given by Miller, Tucker and Zemlin (MTZ) [47]. In the formulation of MTZ, the salesman turns back to the origin, denoted by 0, t times.

$$\text{Minimize } \sum_{i=0}^n \sum_{j=0, i \neq j}^n c_{ij} x_{ij}$$

$$\text{Subject to } \sum_{i=1}^n x_{i0} = t \tag{2.1}$$

$$\sum_{i=0}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \quad i \neq j \tag{2.2}$$

$$\sum_{j=0}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad j \neq i \tag{2.3}$$

$$u_i - u_j + px_{ij} \leq p - 1 \quad 1 \leq i \neq j \leq n \quad (2.4)$$

$$u_i \in R \quad \forall i = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j$$

The constraint (2.1) forces the salesman to turn back to the origin exactly t times. The constraints (2.2) and (2.3) are the usual degree constraints of an assignment problem. They proposed constraint (2.4) by using extra variable in order to reduce the number of exponentially growing subtours, which do not include the origin. These constraints are generally called subtour elimination constraints (SEC). In constraint (2.4), p is the maximum number of nodes that a salesman is allowed to visit and u_i are arbitrary real numbers.

2.1.2 Solution Methods of m -TSP

The TSP and m -TSP problems belong to the class of combinatorial optimization problems, known as NP-complete. Specifically; if one can find an efficient algorithm (i.e., an algorithm that will be guaranteed to find the optimal solution in a polynomial number of steps) for the TSP and m -TSP problems, then efficient algorithms could be found for all other problems in the NP-complete class. To date, however, no one has found a polynomial-time algorithm for the m -TSP. Exact methods to find an optimum solution require too much computation time, whereas heuristic approaches need much more less computational effort, but do not guarantee optimality.

The solution methods found in the literature for the m -TSP can be discussed in three categories. These are exact solution methods including branch-and-bound approach, branch-and-cut approach and cutting plane algorithm, heuristics including extended TSP heuristics and neural networks, and transformations. Although m -TSP is an important extension of TSP there is not much work done on the problem. For a detailed review of these solution methods, the reader is directed to Bodin et al [6], Laporte [37] and Lawler et al [41].

2.1.2.1 Exact Solution Methods for m -TSP

A number of exact solution methods proposed for the m -TSP are based on Integer Linear Programs (ILPs) formulations and algorithms. The most promising exact methods seem to be branch-and-bound approach, branch-and-cut approach and cutting plane algorithms.

The branch-and-bound algorithm can formally be stated as relaxing some of the constraints initially and at each iteration either finding a different lower bound for the value of the optimal solution for the current subproblem or “fathoming” the current subproblem by showing that it does not contain a feasible solution. If the subproblem cannot be fathomed, it is further split into other subproblems containing the feasible solution of original problem. Svestka and Huckfelt [56] employed their formulation in a branch-and-bound framework developed by Bellmore and Malone [6] with a new set of SECs. Gavish and Srikanth [26] employed branch-and-bound algorithm to solve m -TSP by relaxing one of the constraints given below :

Degree Constraints: (They require each node to be visited only once)

SEC: (They eliminate tours not including the origin)

Integrality Constraints: (They force all variables in the optimal solution to be integer)

The algorithm basically starts with solving the associated assignment problem (AP). However the solution of the corresponding AP requires the augmentation of the original distance matrix. If there exists any subtours in the solution, a subset of subtour elimination constraints are introduced and the resulting AP is solved as an integer programming again. The procedure repeats itself until there are no subtours and the solution satisfies all the constraints of the ILP formulation.

The cutting plane technique is based on relaxing the original problem and continuously tightening the lower bound by introducing extra cuts, thus getting a

relaxation value close to the optimal integer value. The most commonly used inequalities for cuts are SECs, comb inequality, clique tree inequality, path, crowns and ladder inequality. For detailed information the reader is directed to Jünger et al [32]. The first example of cutting plane method was due to Dantzig, Fulkerson and Johnson [15]. They proposed a method for solving TSP and showed the power of this method by solving 49-city TSP. Laporte and Nobert [39] introduce a cutting plane algorithm for the m -TSP similar that of Svestka and Huckfeld, without modifying the original distance matrix. The algorithm can be extended to have the number of salesman fixed or variable, or the objective function to include a fixed traveling cost.

Branch-and-cut approach uses the cutting planes in a branch-and-bound scheme to obtain better relaxations. It means that for each partition of the solution set of LP relaxation, several cuts are added to the current formulation to tighten the problem. That is in bounding step, instead of solving one relaxation; a sequence of relaxation is solved each time adding an inequality that is violated by the current fractional solution. Various branch-and-cut techniques have been proposed for solving m -TSP. Some of these can be found in the papers of Hong in 1972, Miliotis in 1976, Padberg and Hong in 1980. Applegate et al [2] uses a new branch and cut type method to obtain a solution of 13,509-city instance.

2.1.2.2 Heuristic Solution Methods for m -TSP

In order to obtain a “near-optimal” solution in a reasonable amount of time instead of finding the optimal solution in more expensive way, many heuristic solution techniques have been introduced for the TSP and m -TSP. Many heuristics developed for TSP are applied to m -TSP by transforming it to standard TSP. Heuristic solutions can be grouped in two. The First is Tour Construction Heuristics and the second is Tour Improvement Heuristics.

2.1.2.2.1 Tour Construction Heuristics

These types of heuristics build an approximately optimal tour starting from the original distance matrix according to a construction procedure. The procedure stops without trying to improve it, once it has found a feasible solution. In the tour construction procedures the following three components serve as key ingredients: The choice of an *initial tour*, the *selection* criterion, the *insertion* criterion.

Christofides [10] developed a construction heuristic on the base of spanning trees and minimum weight matching. Johnson and McGeoch [31] proved that this procedure has a computation time of $O(n^3)$. But later on, Reinelt [51] showed that computation time can be reduced to $O(n^2)$.

In Nearest Neighbor Heuristics, the simplest of all the heuristics, the procedure starts from an initial node arbitrarily and each step moves to the next nearest node, which was not visited yet. After all the nodes are visited, the tour is finally accomplished by linking the last visited node to the initial starting node. By considering all the n vertices as the starting point, the computation time of this procedure can grow up to $O(n^3)$ from $O(n^2)$ (Laporte, 37).

Insertion Algorithm starts with a small subset of nodes and enlarge this subset by inserting the nodes, which are not included in the subset. While inserting, the procedure uses insertion criterions. The computation time of this procedure can grow up to $O(n^3)$ from $O(n \cdot \log n)$ depending on the insertion criterion (Laporte, 37).

Another tour construction heuristic is The Savings Algorithm having derived from the vehicle routing algorithm proposed by Clarke and Wright [13]. This heuristic will be explained in detail in Section 2.2.2. The computation time of this procedure is $O(n^2 \cdot \log n)$, but the storage space for savings require a space of $O(n^2)$ Reinelt [51].

2.1.2.2.2 Tour Improvement Heuristics

Tour Improvement Heuristics try to improve a feasible tour by simple tour modifications after an initial tour is obtained by the use of tour construction heuristics. Tour improvement heuristics are performed according to a specified order of operations until a tour for which no operation yields a better tour is reached. These specified order of operations could be assumed as a local search method because better tours obtained are only local optimum tours.

Lin [44] proposed the r -opt algorithm where r edges in a feasible tour are exchanged for r edges not in that optimal solution as long as the result remains a tour and the length of that tour is less than the length of the previous tour. An improvement to Lin's r -opt algorithm is due to Lin and Kerningham [45], where the value of r changes dynamically during the algorithm.

Simulated Annealing Heuristics remove the disadvantage of r -opt algorithm, which can get stuck at local optima by moving from a given initial solution to a minimum –cost solution by changing the initial solution gradually. However, sometimes, the initial solution is substituted by the new solution although the new solution is more costly. This increases the probability of getting closer to the global optimum. Many authors have proposed the application of simulated annealing to the TSP. Some of these are due to Rossier et al. [53] and Nahar et al. [49].

Tabu search heuristics are similar to the simulated annealing in the way of prevention from getting stuck at local optima. These kinds of heuristics have become popular for the last decades. The solutions which have already been examined are stored in 'tabu list' to prevent cycling. The tabu search has been applied to the TSP with numerous successful results. Some of these are due to Knox [34], Malek [46] and Fiechter [20].

Genetic algorithms (GA) have been recent approach to the combinatorial optimization type problems. GAs are actually based on human genetics, trying to imitate the natural evolution scheme and known to find near-optimal solutions to highly complex problems. For the case of TSP, the chromosomes are used to

represent the tours are generally coded as the order of visited vertices or edges in the graph. Detailed discussions on the subject can be found in Grefenstette al [29].

2.1.2.3.Transformations to the TSP

Third group for the solution methods of m -TSP is the transformation of the problem to a single TSP. By transformation, m -TSP can be solved by any algorithm developed for TSP. Orloff [50], work of whose demonstrates that solving any m -TSP is equivalent to a TSP solution on a transformed graph.He transformed m vehicle General Routing Problem (m -GRP) into a single vehicle GRP on the same graph.

Bellmore and Hong [4] showed that the asymmetric case of the m -TSP with m salesman and n nodes could be converted into a standard asymmetric TSP containing $(n+m-1)$ nodes. Hong and Padberg [30] proposed a similar work to that of Bellmore and Hong where symmetric m -TSP with m salesman and n nodes is transformed into a standard symmetric TSP containing $(n+m-4)$, considering the fixed charges as well.

2.2.VEHICLE ROUTING PROBLEMS (VRP)

Vehicle Routing, which is called “one of the great success stories of operations research in the last decade”, is distinguished by a highly successful interplay between algorithmic techniques and the development of effective routing systems for industry. On the one hand, operations researchers with academic affiliations have gone beyond the design and development of algorithms to play an important role in the implementation of routing systems. On the other hand, developments in computer hardware and software and their growing integration into operational activities of commercial concerns have created a high degree of awareness of potential benefits of vehicle routing. The interested reader may consult a number of useful surveys of the field including Golden and Assad [28], Bodin et al [6], Bott and Ballaou [7],Christofides [11].

The Vehicle Routing Problem (VRP) is a distribution problem in which a set of customers with known locations and requirements for some commodity is to be supplied from a single depot using a set of delivery vehicles each with a prescribed capacity. From a graph theoretical point of view the VRP may be stated as follows: Let $G(V, A)$ be a complete graph with node set $V = \{0, 1, \dots, n\}$ and arc set $A\{(i, j) : i, j \in V, i \neq j\}$. In this graph model, node “0” is the depot and the other nodes are the customers to be served. Each node except from the depot is associated with known demands d_i . For every arc, there is an associated cost $c_{ij}, (i \neq j)$, representing the travel cost (distance, time) between nodes i and j . There are m identical vehicles based at the depot, having same capacity Q . The VRP consists of designing a set of least-cost vehicles in such a way that each node except the depot is visited exactly once by exactly one vehicle to satisfy its demand; all vehicle routes start and end at the depot, vehicle capacities are not exceeded and some other side constraints are satisfied.

The VRP can have different aspects that form the characteristics of the problem. Some of these are: Nature of demand (pure pick-ups or pure deliveries, pick-ups or deliveries with backhaul option, single or multiple commodities, priorities for customers), information on demand (known in advance, changeable by time), vehicle fleet (fixed or variable fleet size, one or more than one) homogeneous fleet or multiple vehicle types), depot (single or multiple), scheduling requirements (time windows for pick-up/delivery (soft, hard), load/unload times).

2.2.1 Mathematical Formulations of VRP

Most of the VRP problems with underlying characteristics listed above are well defined mathematically, and they are extremely difficult in NP-hard class. This difficulty in the problem induced researchers to develop heuristic solution procedures in addition to optimization algorithms. We present general models where a set of customers is served by m non-identical vehicles based at a depot with a set of routes whose total distance (travel cost) is minimized for

Capacitated Vehicle Routing Problem (CVRP), Vehicle Routing Problem with Backhaul (VRPB) and Vehicle Routing Problem with time windows (VRPTW).

Dantzig and Ramser Formulation

Dantzig and Ramser (DR) [14], who proposed a formulation for single depot VRP with vehicle with capacity and maximum cost (time or distance) constraints, considered the VRP for the first time in the literature. The formulation is given as follows:

$$\text{Minimize } \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N c_{ij} x_{ijk}$$

$$\text{Subject to } \sum_{i=1}^N \sum_{k=1}^N x_{ijk} = 1, \quad j = 1, 2, \dots, N-1 \quad (2.20)$$

$$\sum_{j=1}^N \sum_{k=1}^N x_{ijk} = 1, \quad i = 1, 2, \dots, N-1 \quad (2.21)$$

$$\sum_{i=1}^N x_{ihk} - \sum_{j=1}^N x_{hjk} = 0, \quad k = 1, 2, \dots, V, \quad h = 1, 2, \dots, N \quad (2.22)$$

$$\sum_{i=1}^N Q_i \sum_{j=1}^N x_{ijk} \leq P_k \quad k = 1, 2, \dots, V \quad (2.23)$$

$$\sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ijk} \leq T_k \quad k = 1, 2, \dots, V \quad (2.24)$$

$$\sum_{j=1}^{N-1} x_{Njk} \leq 1 \quad k = 1, 2, \dots, V \quad (2.25)$$

$$\sum_{j=1}^{N-1} x_{iNk} \leq 1 \quad k = 1, 2, \dots, V \quad (2.26)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j, k$$

$$\begin{aligned}
y_i - y_j + Nx_{ijk} &\leq N-1 & 1 \leq i \neq j \leq N-1 \\
& & 1 \leq k \leq V
\end{aligned} \tag{2.27}$$

In the formulation, V vehicles, say $\{1, \dots, V\}$ based at a depot (node N) with capacities P_k respectively visit the customers whose demands are Q_i . T_k stands for maximum cost allowed for a route of vehicle k . x_{ijh} is 0-1 variable and $x_{ijk}=1$ if pair i,j is in the route of vehicle k , 0 otherwise. Constraints (2.20) and (2.21) are usual assignment constraints. Route continuity is represented by constraint (2.22). Constraint (2.23) represents the vehicle capacity constraints. Constraint (2.24) represent the total route constraint. Constraints (2.25) and (2.26) make certain that vehicle availability is not exceeded. Finally constraint (2.27) is SEC. In this model, it has been assumed that whenever a customer is serviced, his requirements are satisfied.

Desrochers, Savelsberg, Lenstra and Soumis Formulation

$$\text{Minimize } \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$\text{Subject to } \sum_{j \in N} x_{ij} = 1, \quad \text{for } i \in N \tag{2.28}$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = 0, \quad \text{for } i \in N \tag{2.29}$$

$$D_i + t_{ij} \leq D_j \quad \text{for } (i,j) \in I \tag{2.30}$$

$$e_i \leq D_i \leq l_i \quad \text{for } i \in N \tag{2.31}$$

$$y_i + q_{ij} \leq y_j \quad \text{for } (i,j) \in I \tag{2.32}$$

$$0 \leq y_i \leq Q \quad \text{for } i \in N \tag{2.33}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j$$

In the formulation of Desrochers, Savelsberg, Lenstra and Soumis [16] for VRPTW, x_{ij} is a 0-1 variable and $x_{ij}=1$ if a vehicle visits customer j immediately after customer i , and 0 otherwise. Q stands for the capacity of the vehicle, c_{ij} represents the travel cost between customers i and j . t_{ij} is the travel time between customer i and j . For each customer, there is a demand q_i and time window $[e_i, l_i]$ which identifies the earliest and latest time to be visited, respectively. We note that the number of vehicles is not bounded in the formulation.

Constraints (2.28) and (2.29) are usual assignment constraints. Constraints (2.30) and (2.31) guarantee feasibility of the schedule and constraints (2.32) and (2.33) ensure feasibility of the loads.

Toth and Vigo Formulation

Toth and Vigo [58] proposed an integer linear programming model for VRPB in asymmetric distance matrix. They grouped the vertices as Linehaul (L) and Backhaul (B). Let Λ (respectively B) be the family of all subsets of vertices in L (respectively B), and let $\Phi = \Lambda \cup B$. For each $S \in \Phi$, let $\sigma(S)$ be the minimum number of vehicles needed to serve all the costumers in S . For each i , $\Gamma_i^+ = \{j : (i, j) \in A\}$ and $\Gamma_i^- = \{j : (j, i) \in A\}$ where A is the arc set. In the formulation K is the number of vehicles and depot is node 0.

$$\text{Minimize } \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$\text{Subject to } \sum_{i \in \Gamma_j^-} x_{ij} = 1 \quad j \in V \quad (2.34)$$

$$\sum_{i \in \Gamma_i^+} x_{ij} = 1 \quad i \in V \quad (2.35)$$

$$\sum_{i \in \Gamma_0^-} x_{i0} = K \quad (2.36)$$

$$\sum_{j \in \Gamma_0^+} x_{0j} = K \quad (2.37)$$

$$\sum_{j \in S} \sum_{i \in \Gamma_j^- \setminus S} x_{ij} \geq \sigma(S) \quad (2.38)$$

$$\sum_{j \in S} \sum_{i \in \Gamma_j^+ \setminus S} x_{ij} \geq \sigma(S) \quad (2.39)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j$$

Constraints (2.34), (2.36) and (2.35),(2.37) impose indegree and outdegree constraints for the customer and the depot vertices, respectively. Constraints (2.38) and (2.39) impose both connectivity and the capacity constraints.

Kulkarni and Bhawe Formulation

A simple formulation based on that of DR was proposed by Kulkarni and Bhawe [36]. In the formulation L is the maximum number of nodes which a vehicle can visit and Q_i is the demand of node i . If P is the capacity of all the vehicles and T is the maximum cost allowed for a route for the vehicle, then for a single depot VRP formulation is as follows:

$$\text{Minimize } \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij}$$

$$\text{Subject to } \sum_{i=1}^N x_{ij} = 1, \quad j = 1, 2, \dots, N-1 \quad (2.40)$$

$$\sum_{j=1}^N x_{ijk} = 1, \quad i = 1, 2, \dots, N-1 \quad (2.41)$$

$$\sum_{j=1}^N x_{Nj} \leq V, \quad (2.42)$$

$$\sum_{i=1}^N x_{iN} \leq V, \quad (2.43)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j$$

$$y_i - y_j + Lx_{ij} \leq L-1 \quad 1 \leq i \neq j \leq N-1 \quad (2.44)$$

$$u_i - u_j + Px_{ij} \leq P - Q_i \quad 1 \leq i \neq j \leq N-1 \quad (2.45)$$

$$v_i - v_j + Tx_{ij} \leq T-1 \quad 1 \leq i \neq j \leq N-1 \quad (2.46)$$

In the above formulation, constraints (2.40)-(2.43) ensure that each node is being served only once and that all the V vehicles are being used. Constraints (2.44)-(2.46) are the subtour breaking constraints, which also represent the node constraints, capacity constraints and cost constraints respectively. In words these equations ensure that all the tours are starting and ending at the Depot N and further every route serves at the most L nodes and the load and the cost on every route are less than or equal to the vehicle capacity P and the maximum allowable cost T respectively.

But, later Achuthan and Caccetta [1] proved that formulation above fails since constraints (2.46) do not ensure that the maximum allowable cost for a vehicle tour is at most T . They redefined constraints (2.46) as:

$$v_i - v_j + (c_{ij} + T)x_{ij} \leq T \quad (2.47)$$

In 1988, Brodie and Waters [8] redefined constraints (2.39) in Kulkarni and Bhave Formulation as:

$$v_i - v_j + T x_{ij} \leq T - c_{ij} \quad (2.48)$$

2.2.2 Solutions Methods of VRP

Almost all problems of vehicle routing are NP-hard. (Lenstra&Rinnoy Kan 43). Computational complexity theory has provided strong evidence that any

optimization algorithm for their solution is likely to perform very poorly on some occasions which means that its worst case running time is likely to grow exponentially with problem size. The solution methods of VRP may be classified as exact algorithms and heuristic algorithms.

2.2.2.1 Exact Algorithms for VRP

The usual solution methods in integer programming and combinatorial optimization are applied in the design of exact solution algorithms to the VRP. Exact methods are based on the formulations that were given before. Exact algorithms for the VRP can be classified into three broad categories: direct tree search methods, dynamic programming and integer linear programming.

Laporte, Mercure and Nobert [40] proposed an algorithm by exploiting the relationship between the VRP and one of its relaxation, the m -TSP. Given an upper bound m_u on m , the m -TSP can be transformed into a 1-TSP. (Lenstra and Rinnooy Kan 42). Then, the problem is solved through a branch-and-bound process in which subproblems are assignment problems. Using this methodology, Laporte, Mercure and Nobert [40] have solved to optimality randomly generated asymmetrical CVRP involving up to 260 vertices by partitioning the current infeasible subproblem.

Christofides, Mingozzi and Toth [12] have developed an algorithm for symmetrical VRPs. It is based on the k -degree center tree relaxation of the m -TSP where m is fixed. In any feasible solution the set E of edges are partitioned into four subsets which are (i) edges not belonging to the solution, (ii) edges forming k -degree tree center, (iii) y edges incident of vertex 1, which is depot, (iiii) $m-y$ edges not incident to vertex 1. The core of the algorithm is to embed the lower bound on the solution in a branch-and-bound scheme. They have successfully solved VRPs ranging in size from 10 to 25 vertices.

Kolen, Rinnooy Kan and Trienekens [35] extended the state space relaxation presented by Christofides, Mingozzi and Toth. This state space relaxation is

based in a mapping from the original state space to a space of smaller cardinality. Kolen, Rinnooy Kan and Trienekens used a two-level state space relaxation for VRPTW. At the first level, a lower bound on the costs of a time-constrained path from the depot to vertex j with load q is computed. The states are of the form (t, q, j) , where q is the load of a shortest path arriving at vertex j no later than time t . At the second level, a lower bound on the costs of m routes with total load $\sum_{i \in N} q_i$ and different destination vertices is computed. The states are of the form (k, q, j) , where q is the total load of the first k routes and j is the destination vertex of route k . Problems with up to fifteen customers are solved.

Fisher [21] proposed an algorithm using minimum k -trees. Given a graph with $n+1$ nodes a K -tree is defined to be a set of $n+K$ edges that span the graph. He showed that the vehicle routing problem could be modeled as the problem of finding a minimum cost K -tree with K edges incident on the depot and subject to some side constraints that impose vehicle capacity and the requirement that each customer be visited exactly once. The side constraints are dualized to obtain a Lagrangian problem that provides lower bounds in a branch-and-bound algorithm. This algorithm has produced proven optimal solutions for a number of difficult problems, including a well-known problem with 100 customers and several real problems with 25-71 customers.

Eilon, Watson-Gandy and Christofides [18] were the pioneers to propose dynamic programming for VRP with a fixed number of m vehicles. Let $c(S)$ denote the cost (length) of a vehicle route through vertex 1 –the depot– and all vertices of a subset S of $V \setminus \{1\}$. Let $f_k(U)$ be the minimum cost achievable using k vehicles and delivering to subset U of $V \setminus \{1\}$. Then the minimum cost can be determined through the following recursion:

$$f_k(U) = \begin{cases} c(U) & (k = 1) \\ \min_{U^* \subset U \subseteq V \setminus \{1\}} [f_{k-1}(U \setminus U^*) + c(U^*)] & (k > 1) \end{cases} \quad (2.49)$$

The solution cost is equal to $f_m(V \setminus \{1\})$ and the optimal solution corresponds to the optimizing subsets U^* in (2.49)

Balinski and Quandt [3] were among the first to propose a set partitioning formulation for VRPs. Consider J , the set of all feasible routes j and a_{ij} be a binary coefficient equal to 1 if and only if vertex $i > 1$ appears on route j . Let c_j^* be the optimal cost of route j and x_j , binary variable equal to 1 if and only if route j is used in the optimal solution. The problem can then be formulated as follows:

$$\begin{aligned}
 & \text{Minimize } \sum_{j \in J} c_j^* x_j \\
 & \text{Subject to } \sum_{j \in J} a_{ij} x_j = 1 \quad (i \in V \setminus \{1\}) \\
 & \quad \quad \quad x_j \in \{0,1\} \quad (j \in J)
 \end{aligned} \tag{2.50}$$

There is at least one exact procedure for VRPB developed by Yano et al [60]. This procedure uses set covering to find an optimal set of routes. Each route can at most have four points. All such routes are generated and the optimal route sequencing is found by complete enumeration.

Mingozi, Giorgi and Baldacci [48] described a procedure for VRPB that computes a valid lower bound to the optimal solution cost by combining different heuristic methods for solving the dual of the LP-relaxation of the exact formulation.

In the Toth and Vigo's [60] new integer linear programming model mentioned in Chapter 2.2.1, a Lagrangian lower bound, which is strengthened in a cutting plane fashion for VRPB in symmetric and asymmetric versions is proposed. The model is used to derive a Lagrangian lower bound, based on projection of the

shortest spanning arborescence with fixed degree at the depot vertex and the optimal solution of min-cost flow problems.

Desrosiers, Soumis and Desrochers [17] apply branching on flow variables when time windows are very tight for VRPTW. They solved problems up to 150 customers. Sörensen [54] suggests the use of Lagrangean decomposition for VRPTW. The two resulting sub problems are the shortest path problem with time window and the generalized assignment problem. But no computational results were reported.

2.2.2.2. Heuristic Algorithms for VRP

Heuristic algorithms for the VRP can often be derived from procedures derived from the TSP. (Laporte, 38). Several families of heuristics proposed for the VRP can be classified into two main classes: classical heuristics developed mostly between 1960 and 1990, and metaheuristics whose growth has occurred in the last decade. Most standard construction and improvement procedures in use today belong to the first class. These classical heuristics perform limited exploration of search space. Nevertheless, they can produce good solutions. In metaheuristics, a deep exploration of search space is performed. Combining neighborhood search rules, memory structures, are effective in metaheuristics. They can produce higher quality solutions than classical heuristics can. But, it is clear that the price to pay is increased computing time.

Clarke and Wright's saving algorithm [13] was first proposed in 1964 to solve CVRP which has unlimited number of vehicles. The method is based on to merge two routes according to the largest savings that can be generated. The savings $s_{ij} = c_{i1} + c_{1j} - c_{ij}$ for $i, j = 2, \dots, n$ and $n-1$ vehicle routes are created. The savings are ordered in a non-increasing way. Then, routes are merged as in the following example. Let two vehicle routes contain $(i,1)$ and $(1,j)$, respectively. If $s_{ij} > 0$ these two routes are merged by introducing arc (i,j) and deleting arcs $(i,1)$ and $(1,j)$. The merge is implemented if the resulting node is

feasible. The procedure is repeated until no improvement is possible. This procedure has $O(n^2 \log n)$ operation time.

One of the classical approaches of the VRP is Fisher and Jaikumar [22] algorithm, which is also named in cluster-first, route-second algorithms. These kinds of algorithms solve a Generalized Assignment Problem (GAP) in order to form the clusters instead of using a geometric method. The Fisher and Jaikumar algorithm includes the following step.

- Choose seed points j_k in V to initialize each cluster k . (seed selection)
- Compute the cost d_{ik} of allocating each customer i to each cluster k as $d_{ij_k} = \min \{ c_{0i} + c_{ij_k} + c_{j_k 0}, c_{0j_k} + c_{j_k i} + c_{i0} \} - (c_{0j_k} + c_{j_k 0})$.
(allocation)
- Solve a GAP with costs d_{ij} , customer weights q_i and vehicle capacity Q .
(generalized assignment)
- Solve a TSP for each cluster corresponding to the GAP solution.(TSP solution)

In the algorithm, the number of vehicle routes m is fixed and relating to the customer weights the plane is parted into m cones. The seed points are dummy customers located along the rays bisecting the cones. After having determined the clusters, the algorithm solves the TSPs optimally using a constraint relaxation based approach.

The last group of classical heuristics is improvement heuristics. Improvement heuristics for VRP are performed on each separate vehicle route or on several routes at a time. Most improvement procedures for TSP are based on *r-opt* algorithm Lin-Kernighan. As explained before, r edges in a feasible tour are exchanged for r edges not in that optimal solution as long as the result remains a tour and the length of that tour is less than the length of the previous tour.

Thompson and Psaraftis [57] and Kindervater and Savelsbergh [33] propose descriptions of multi-route exchanges for the VRP. In these papers tours are not considered in isolation. Paths and customers are exchanged between different tours until no further improvements can be obtained. Figure 2.1, Figure 2.2 and Figure 2.3 show the Crossover, Customer Relocation and Customer Exchange operations, respectively.

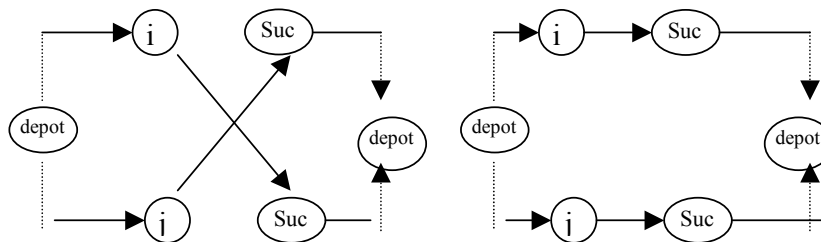


Figure 2.1 Crossover

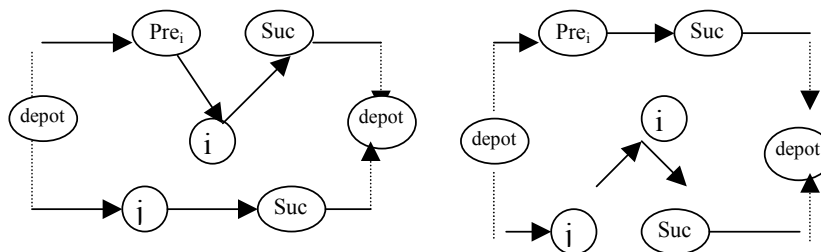


Figure 2.2 Customer Relocation

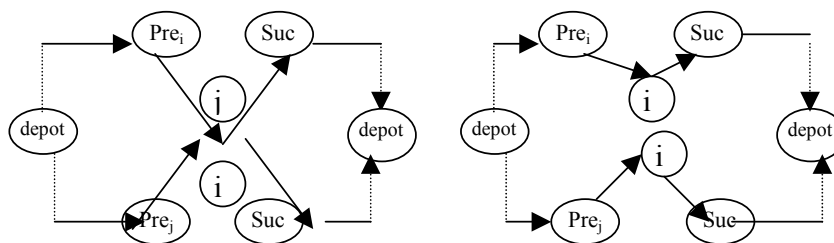


Figure 2.3 Customer Exchange

One group of the metaheuristics is tabu search heuristics. It starts from an initial solution x_1 and moves at each iteration t from x_t to its best neighbour x_{t+1} , until a stopping criteria 's satisfied. To avoid cycling, solutions that were recently examined are forbidden, or *tabu*. In order to reduce time and memory requirements, it is generally accepted to record an attribute of tabu solutions

rather than solutions themselves. Rochat and Taillard [52] proposed Tabu search with set-partition based heuristic where an adaptive memory is kept as a pool of good solutions.

Gendreau, Hertz and Laporte [27] present a GENI procedure related to the tabu. When applied to classical VRPs described in the literature, the proposed heuristic may be one of the best ever developed for the VRP regarding the results they found.

A new metaheuristic approach inspired by foraging behavior of real colonies of ants is Ant Colony Optimization (ACO). The basic ACO idea is that a large number of simple artificial agents are able to build good solutions to hard combinatorial optimization problems via low-level based communications. Real ants cooperate in their search for food by depositing chemical traces on the floor. An artificial ant colony simulates this behavior. Artificial ants cooperate by using a common memory that corresponds to the pheromone deposited by real ants. This artificial pheromone is one of the most important components of ant colony optimization and is used for constructing new solutions. Gambardella, Taillard and Agazzi [24] used ant colony system to solve VRP. In their paper, VRP is transformed into a TSP by adding $m-1$ new depots. A sample is pictured in Figure 2.4. Ants are used to compute complete feasible tours. Local search that exchange paths and customers is executed.

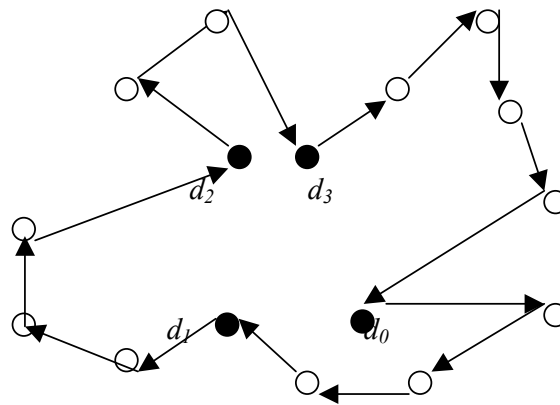


Figure 2.4 Transformation to Four Depots

CHAPTER 3
ANALYSIS OF THE PROBLEM
and
MATHEMATICAL FORMULATION

In this thesis, our objective is to develop and solve a model that determines the optimal routes, which vehicle fleet will follow and provides a method for assigning demands (pick-up/delivery) to vehicles. Best fleet size will be determined by comparing the optimal solutions for alternative sizes.

It is the aim of the Ring Services System to deliver items to points and to pick up items from points within three days if there is a demand (pick-up/delivery). The fleet is loaded regarding to points' demands in the Main Depot, where all kind of items is stored. After having performed the routes, the fleet loaded with pick-ups comes back to the Garage, which is near the Main Depot.

This process is repeated every week. As explained in Section 1.1, five fixed routes are performed in the current system regardless of demands (pick-up/delivery). Although a point doesn't have any demand (pick-up/delivery), one vehicle of the fleet must visit this point anyway, according to the current system. So, the fleet makes a lot of unnecessary mileage. For each week, Transportation Command has a fixed distance and cost of these five fixed routes. Provided that Transportation Command knows in advance which of the points have demands (pick-up/deliver), the fleet visits only these points and can do away with the unproductive mileage. It is clear that the fixed cost for each week will be reduced by this method.

3.1. DATA FILE

Mid-Level Auto Transportation Battalion Command, which is responsible for performing Central Ring Transportation System (CRTS) as a subordinate unit

of Transportation Command, has a very detailed documentation of past years. Since the CRTS started to be employed in 2001, Mid-Level Auto Transportation Battalion Command has filed the information for all weeks, i.e., weights and volumes they delivered and picked up, types of vehicles performing the routes. We used an eight-week data belonging to 2002.

3.1.1. Main Depot and Garage

Logistic Command wants to keep its safety stocks only in Main Depot. This depot is large enough to store Ordnance, Signals, Quartermaster and Engineers Corps items. Although ordnance, signal and engineers corps have their own depots as fourth level depots in the Corps region, provision except for fourth level is provided from Main Depot. There exists a strong communication network including Material Management Centers established to arrange each of the branches in Logistic Command, points, Mid-Level Auto Transportation Battalion Command and Main Depot management. Garage is located in Mid-Level Auto Transportation Battalion near Main Depot. A large number of trucks with different capacities are available here.

3.1.2. Points

The military units, factories and depots are situated throughout the country. Nearly, in all of the cities in the country a military unit or a factory or a depot is located. But regarding size, importance and function of those, twenty-one points were selected for demand (pick-up/ delivery) points. The demands (pick-up/ delivery) of military units, factories and depots out of these twenty-one points are provided by the Regional Ring Transportation System (RRTS). This issue is not related to our study. The shortest distance table is shown in Appendix A, page 65.

3.1.3.Vehicles

Mid-Level Auto Transportation Battalion Command has a vehicle fleet in different capacities and characteristics to perform its task. We used the same fleet of five vehicles (Case 1) as in the routes performed in eight-week data of 2002. When solving the model for fleet of four vehicles (Case 2), we excluded the Type 5 and used the first four types of vehicles, while one more Type 1 vehicle is added to fleet when fleet of six vehicles are employed (Case 3). Capacities of vehicles are shown in Table 3.1

	TYPE 1	TYPE 2	TYPE 3	TYPE 4	TYPE 5
WEIGHT (1000KG)	35	23	16	20	25
VOLUME (M ³)	55	46	53	40	54

Table 3.1. Capacities of Vehicles

Since vehicles can be refueled on their way whenever they need, it doesn't make sense to consider how long a vehicle can travel with a full depot.

3.1.4. Items

All kind of items related to ordnance, signal, quartermaster and engineer corps are delivered and picked up with these vehicles. The volumes and weights of the items are important in the view of volume and weight capacity of vehicles. In the documentation belonging to 2002, although we know the volumes and weights of the items, we cannot get any information about contribution of specific items to these volumes and weights. Properties (volume and weight) of the items demanded (picked up/delivered) are shown in the Appendix A, pages 57-64.

3.2 ASSUMPTIONS

In order to simplify the problem and make it solvable, we make some assumptions. These are:

3.2.1. All demands (pick-up/delivery) must be satisfied

We assume that the Main Depot must meet all demands (pick-up/delivery). For delivery, it is not true in reality. Vendors and military factories supply Main Depot and Main Depot supplies the points. Main Depot may not provide every item whenever needed. As mentioned in Chapter 1, Land Forces have two types of Provision, i.e., Pull System and Push System. In real life, the situation that demands (delivery) are not satisfied may occur in Pull System, since the points may demand an item from Main Depot at any time when Main Depot does not have this item. However, because Provision is provided periodically in Push System, the probability that demands (delivery) are not satisfied in Push System is less than the Pull system. There is a schedule determining when and how much to provide units in the Push System. Items are stored in Main Depot according to this schedule and all demands (delivery) are satisfied in Push System. But in our study, we do not consider the type of Provision. It is assumed that all demands (delivery) are satisfied. Pick-up demands are independent of the type of Provision. All points can have pick-up demands whenever they have any item to send to Main Depot or factories in Ankara..

3.2.2. Demands (pick-up/delivery) are fixed and known in advance

For delivery, in Pull System, points inform Main Depot Management, Material Management Centers, Mid-Level Auto Transportation Battalion Command about their demands (delivery). In Push System, the same trio knows how much to deliver to points according to schedule. For pick-up demands, points inform Main Depot Management, Material Management Centers, Mid-Level Auto

Transportation Battalion Command about the quantities of items to be sent to Main Depot.

3.2.3. The load/unload times are fixed and known in advance.

Time to load/unload the vehicle in the points is known in advance. In real life, weather conditions, the number of personnel in loading/unloading points and disorders in loading/unloading machines may affect the length of the time.

3.2.4. Time windows are fixed and known in advance.

Each point has time intervals, i.e., earliest and latest time to be visited. Earliest time of the points are not important but all the points demanding (pick-up/delivery) must be visited within three days following the vehicles' departure. The fleets leave the Main Depot on Mondays and all the points on the routes must be visited by Thursday.

3.2.5. Vehicle speed is fixed and known in advance.

Each vehicle's speed is fixed with 70 km/h during the travel. In spite of the fact that vehicles use freeways, vehicles cannot exceed the speed limit for a secure trip.

3.2.6. The number of vehicles is fixed and known in advance.

The number of vehicles is fixed. Battalion Commands has the initiative to decide on the number of vehicle fleet size at the beginning of the tours. First, we used fleets of five vehicles to compare the current system with the proposed system. We also used four and six vehicles of fleet to see if there is any improvement in total distance (cost) for different sizes of fleet.

3.3.FORMULATION

In the literature survey, we come across many ILP formulations of m -TSP and VRP. Each one has different features (different vehicle types, pickups&deliveries, time windows, load&unload times, variable speed) and different objectives (minimize distance, minimize travel time, minimize number of vehicles, minimize total cost). We used some of the constraints in the formulations complying with our model.

3.3.1. Indices

- N : Set of all the points including Main Depot and Garage.
Main Depot,2,3...20,21,22, Garage.
- V : Set of vehicles.
Type1, Type2...Type6.

3.3.2.Initial Data and Parameters

- C_{ij} : Distance from i to j
- T_{ij} : Travel time between i and j (C_{ij}/h)
- T_{\min}^i : Earliest time of point i to be visited.
- T_{\max}^i : Latest time of point i to be visited.
- L_i : Time of loading/unloading in point i .
- Q_v : Weight capacity of vehicle v .(1000 kg)
- B_v : Volume capacity of vehicle v .(m^3)
- d_i : Quantity to be delivered to point i .(in 1000 kgs)
- p_i : Quantity to be picked up from point i .(in 1000 kgs)
- f_i : Quantity to be delivered to point i .(m^3)
- g_i : Quantity to be picked up from point i .(m^3)
- h : Speed of the vehicles (70 km/h)
- q : The number of the vehicles

3.3.3. Variables

X_{ijv} : Indicator of visit of vehicle v to point j immediately after point i .

S_i : Time of arriving at point i .

M_i : Load (weight) of a vehicle arriving at i before delivering or picking up at i in terms of weight (kg).

Y_i : Load (volume) of a vehicle arriving at i before delivering or picking up at i in terms of volume (m^3).

3.3.4. Constraints

3.3.4.1. Assignment Constraints

Every point except for Garage has a successor.

$$\sum_{j=1}^n \sum_{v=1}^V X_{ijv} = 1, \quad 1 < i < n+1, i \neq j.$$

Every point except for Main Depot has an antecedent.

$$\sum_{i=2}^{n+1} \sum_{v=1}^V X_{iiv} = 1, \quad 1 < j < n+1, i \neq j.$$

The arriving and leaving vehicles of a point must be the same.

$$\sum_{i=2}^{n+1} X_{ijv} = \sum_{j=1}^n X_{jiv}, \quad 2 \leq k < n$$

Each vehicle arrives at the Garage.

$$\sum_{i=2}^n \sum_{v=1}^V X_{i1v} = q, \quad 1 = \text{Garage}$$

Each vehicle leaves the Main Depot.

$$\sum_{j=2}^n \sum_{v=1}^V X_{n+1,jv} = q, \quad n+1 = \text{Main Depot}$$

3.3.4.2. Capacity Constraints

The total weights of the items to be delivered in a vehicle at the beginning of a tour cannot exceed the vehicle's weight capacity.

$$\sum_{i=2}^n d_i \sum_{j=1}^n X_{ijv} \leq Q_v, \quad \forall i, \forall V$$

The total volumes of the items to be delivered in a vehicle at the beginning of a tour cannot exceed the vehicle's volume capacity.

$$\sum_{i=2}^n f_i \sum_{j=1}^n X_{ijv} \leq B_v, \quad \forall i, \forall V$$

3.3.4.3. Load Feasibility Constraints

Load (weight) of a vehicle arriving at i before delivering or picking up at i in terms of weight (kg) cannot exceed the vehicle's weight capacity.

$$0 \leq M_i \leq Q_v, \quad \forall i, \forall V$$

Load (volume) of a vehicle arriving at i before delivering or picking up at i in terms of volume (m^3) can not exceed the vehicle's volume capacity.

$$0 \leq Y_i \leq B_v, \quad \forall i, \forall V$$

Subtour breaking constraints, which guarantees the feasibility of loads in terms of weight and volume, where K is a large constant.

$$M_i + p_i - d_i - M_j \leq K(1 - X_{ijv}) \quad , \forall i, \forall V$$

$$Y_i + g_i - f_i - Y_j \leq K(1 - X_{ijv}) \quad , \forall i, \forall V$$

These subtour breaking constraints were first used in the constraint (2.4) of MTZ formulation. We adapted this constraint into our model. M_i (Y_i) are the loads (weigh&volume) of the vehicles before delivery or pick up in point i . If x_{ijv} equals to 1, the inequality will be true since the left and right side of the inequality will be zero. Meanwhile if x_{ijv} equals to zero, right side of the inequality will be greater than the left side and it will be true again.

3.3.4.4. Time Feasibility Constraints

Delivery and pick-up must be made in time intervals.

$$T_{\min}^i \leq S_i \quad , \forall i$$

$$T_{\max}^i \geq S_i \quad , \forall i$$

Subtour breaking constraint, which guarantee the feasibility of time, where K is a large constant.

$$S_i + L_i + T_{ij} - S_j \leq K(1 - X_{ijv}) \quad , \forall i, \forall V$$

$$1 < i \leq n+1, 1 \leq j \leq n+1, i \neq j$$

3.3.4.5. Non-negativeness and Binary Variables

$$X_{ijv} = \{0,1\} \quad \text{for all } i \in N, j \in N, v \in V$$

$$M_i, Y_i, S_i \geq 0 \quad \text{for all } i \in N$$

3.3.5. Objective Function

Our aim is to minimize the distance. Using the distance found, we calculate the total cost. The cost per kilometer is 551.400 TL (valid for the year 2002). To find the total cost this number is multiplied with total distance.

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{v=1}^V X_{ijv}$$

Sural and Bookbinder [55] describes VRPB by a scheme $\alpha / \beta / \gamma$, where α denotes the number of vehicles, β refers to backhaul service options and γ specifies the precedence order of Linehaul (demanding delivery) and Backhaul (demanding pick-up) points. Our problem is a mixed of VRPB and VRPTW. We can describe our problem 5(4)(6) / must / any, according to this scheme. The fleet consists of five (four,six) vehicles, all demanding points must be served and there is not any order of points.

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{v=1}^V X_{ijv}$$

Subject to

$$\sum_{j=1}^n \sum_{v=1}^V X_{ijv} = 1 \quad , \quad 1 < i < n+1, i \neq j$$

$$\sum_{i=2}^{n+1} \sum_{v=1}^V X_{ijv} = 1 \quad , \quad 1 < j < n+1, i \neq j$$

$$\sum_{i=2}^{n+1} X_{ijv} = \sum_{j=1}^n X_{jkv} \quad , \quad 2 \leq k < n$$

$$\sum_{i=2}^n \sum_{v=1}^V X_{i1v} = q \quad 1 = \text{Garage}$$

$$\sum_{j=2}^n \sum_{v=1}^V X_{n+1,jv} = q, \quad n+1 = \text{Main Depot}$$

$$\sum_{i=2}^n d_i \sum_{j=1}^n X_{ijv} \leq Q_v, \quad \forall i, \forall V$$

$$\sum_{i=2}^n f_i \sum_{j=1}^n X_{ijv} \leq B_v, \quad \forall i, \forall V$$

$$0 \leq M_i \leq Q_v, \quad \forall i, \forall V$$

$$0 \leq Y_i \leq B_v, \quad \forall i, \forall V$$

$$M_i + p_i - d_i - M_j \leq K(1 - X_{ijv}), \quad \forall i, \forall V$$

$$Y_i + g_i - f_i - Y_j \leq K(1 - X_{ijv}), \quad \forall i, \forall V$$

$$T_{\min}^i \leq S_i, \quad \forall i$$

$$T_{\max}^i \geq S_i, \quad \forall i$$

$$S_i + L_i + T_{ij} - S_j \leq K(1 - X_{ijv})$$

$$1 < i \leq n+1, 1 \leq j \leq n+1, i \neq j$$

$$X_{ijv} = \{0, 1\}$$

$$\text{for all } i \in N, j \in N, v \in V$$

$$M_i, Y_i, S_i \geq 0 \text{ for all } i \in N$$

3.4.EXPERIMENTATION

We have used the mathematical modeling software GAMS 2.25[25] in the implementation of the model. We solved the problem for three different vehicle fleet sizes. In the first case, fleets of five vehicles are employed to comply

with the current system, while in the second and third one four and six vehicles are employed to find the adequate vehicle fleet by comparison. As a sample, a GAMS code of the model belonging to week-7 for the case of five vehicles of fleet are shown in Appendix B, pages 65-68.

Firstly, we run the case when five vehicles are employed for eight-week data. But we have discovered that excessive amounts of CPU times are needed to solve the model in a GAMS licensed server. CPU times increase enormously as the number of demanding points increases. So, we decide to work in Unix operating system on a machine (Sun Hpc 4500) consisting of twelve 400 MHz CPU processors. Since there is no license for GAMS in this machine, we run the program for each case on a different server that has a license for GAMS in order to construct the model file including all the equations in explicit form. Then, we use the output files of GAMS to solve the model in CPLEX 7.1 at Sun Hpc 4500 and at this time we see that CPU times are shorter than those of server having GAMS license.

We found exact integer optimal solutions in Case 1 for each week without using any relative (*epgap*) or absolute (*epagap*) optimality criterions in CPLEX. *Epgap* is relative tolerance on the gap between the best integer objective and the objective of the best node remaining whereas *epagap* is the absolute tolerance on the gap between the best integer objective and the objective of the best node remaining. Following table shows number of demand points, CPU times, number of iterations and number of nodes observed.

	# of points	CPU	# of Iterations	# of nodes observed
Week-1	14	174.92	247006	20344
Week-2	16	519.97	591470	47540
Week-3	13	466.28	640031	33542
Week-4	14	122.13	132757	18507
Week-5	21	7118.54	4970274	373259
Week-6	14	43.88	33625	4214
Week-7	21	6986.41	5713782	263654
Week-8	15	855.60	1182234	122807

Table 3.2. Required CPU times, number of iterations and number of nodes observed when five vehicles are employed in the model.

We apply the same procedure when fleet size is four and six. Again, exact integer optimal solutions for each week in each case are found without using any relative or absolute gap. Following tables show number of demanding points, CPU times, number of iterations and number of nodes observed.

	# of points	CPU	# of Iterations	# of nodes counted
Week-1	14	168.20	234903	13324
Week-2	16	1084.56	1432551	107736
Week-3	13	7908.98	3545621	153161
Week-4	14	34.78	28825	3226
Week-5	21	24054.44	20299444	1434962
Week-6	14	294.61	514432	29742
Week-7	21	30687.56	24948707	1785694
Week-8	15	26468.72	38043669	3286107

Table 3.3. Required CPU times, number of iterations and number of nodes observed when four vehicles are employed in the model.

	# of points	CPU	# of Iterations	# of nodes counted
Week-1	14	511.69	806400	53614
Week-2	16	823.37	933358	94153
Week-3	13	2592.66	3146252	213960
Week-4	14	206.54	258438	21258
Week-5	21	3659.04	2647596	206223
Week-6	14	46.53	24287	4055
Week-7	21	1134.65	874358	67325
Week-8	15	381.91	409075	48316

Table 3.4. Required CPU times, number of iterations and number of nodes observed when six vehicles are employed in the model.

3.5 Results

The total distances of optimal routes, which we have found for each week in each case, are shorter than the fixed distance of the current system. Comparison of the Proposed system in three cases with the current system is evaluated in ways of distance and cost for each week. Figures and tables below show that there are vast differences between the Proposed System and the Current System for total distance and cost.

3.5.1 Case 1. Fleets of Five vehicles are employed.

Following table shows the total distance and cost of optimal routes, which we have found, and the savings, which we have made for each week when fleets of five vehicles are employed.

	Weeks							
	1	2	3	4	5	6	7	8
Current System's Distance	8346	8346	8346	8346	8346	8346	8346	8346
Optimal Distance	5595	5619	5876	5379	5605	5782	5605	5439
Current System's Cost (Million TL)	4602	4602	4602	4602	4602	4602	4602	4602
Week Cost (Million TL)	3085	3098	3240	2966	3090	3188	3090	2999
Saving (Million TL)	1517	1504	1362	1636	1512	1414	1512	1603

Table 3.5. Total distances, costs and savings for weeks in Case 1.

It can be seen more clearly on the following figures that there is great improvements in total distance and cost. The distances are shortened for %30-35. So, application of the model can save the same ratio of cost.

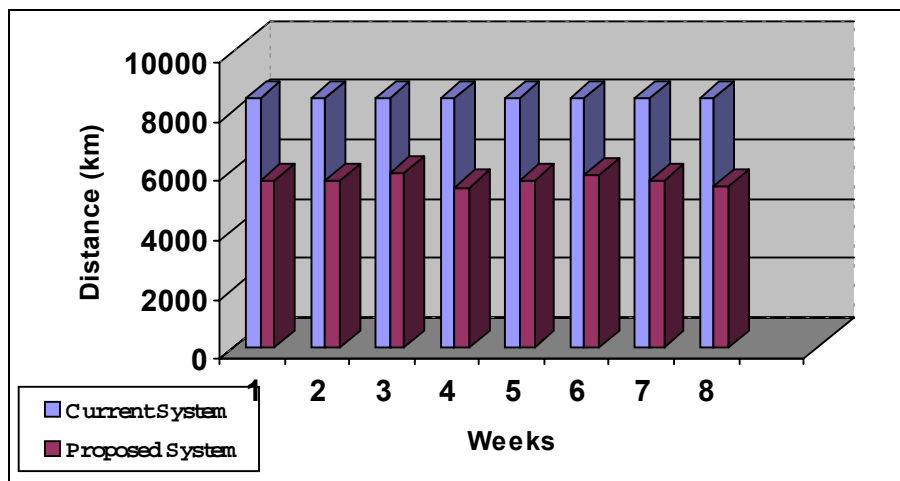


Figure 3.1. Comparison of The Current System with The Proposed System in Distance for weeks in Case 1.

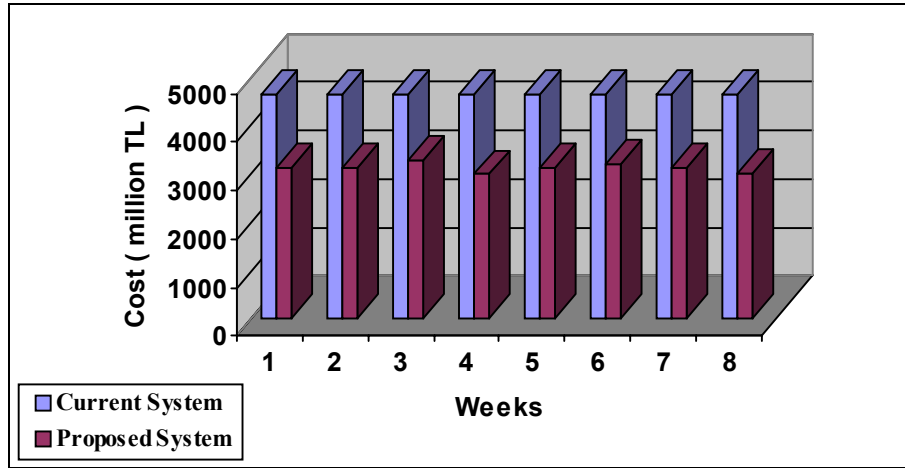


Figure 3.2. Comparison of The Current System with The Proposed System in Cost for weeks in Case 1.

3.5.2 Case 2. Fleets of Four vehicles are employed.

Following table shows the total distance and cost of optimal routes, which we have found and the savings, which we have made for each week when fleets of four vehicles are employed.

	Weeks							
	1	2	3	4	5	6	7	8
Current System's Distance	8346	8346	8346	8346	8346	8346	8346	8346
Optimal Distance	5201	5235	5912	4931	5200	5115	5200	5034
Current System's Cost (Million TL)	4602	4602	4602	4602	4602	4602	4602	4602
Week Cost (Million TL)	2869	2888	3261	2720	2869	2822	2869	2777
Saving (Million TL)	1733	1714	1341	1882	1733	1780	1733	1825

Table 3.6. Total distances, costs and savings for weeks in Case 2.

It can be seen more clearly on the following figures that there is great improvements in distance cost. The distances are shortened for %29-40. So, the same ratio of cost can be saved by applying the model.

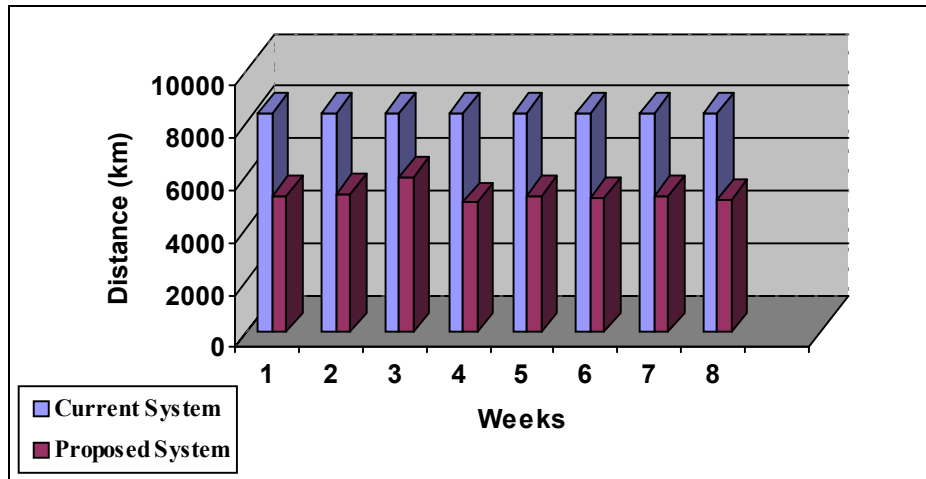


Figure 3.3. Comparison of The Current System with The Proposed System in Distance for weeks in Case 2.

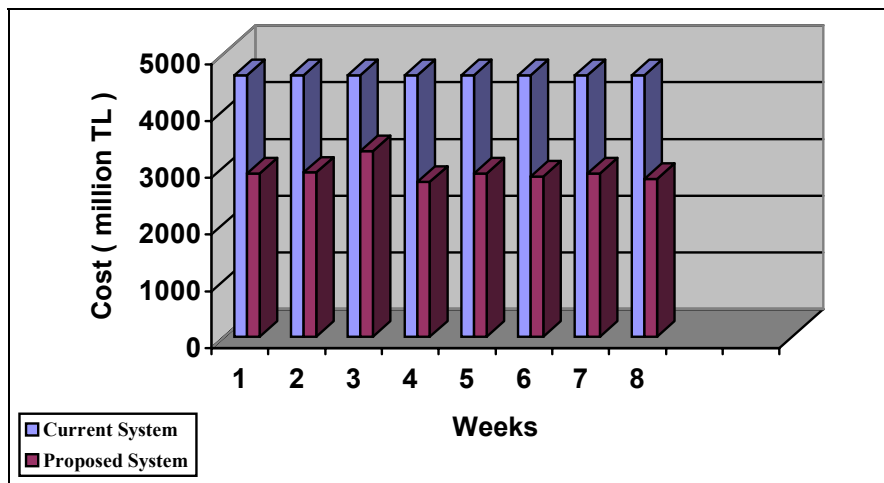


Figure 3.4 Comparison of The Current System with The Proposed System in Cost for weeks in Case 2.

3.5.3 Case 3. Fleets of Six vehicles are employed.

Following table shows the total distance and cost of optimal routes, which we have found and the savings, which we have made for each week when fleets of six vehicles are employed.

	Weeks							
	1	2	3	4	5	6	7	8
Current System's Distance	8346	8346	8346	8346	8346	8346	8346	8346
Optimal Distance	6277	6111	6222	5852	6121	6462	6121	6040
Current System's Cost (Million TL)	4602	4602	4602	4602	4602	4602	4602	4602
Week Cost (Million TL)	3461	3370	3431	3227	3375	3563	3375	3330
Saving (Million TL)	1141	1232	1171	1375	1227	1039	1227	1242

Table 3.7. Total distances, costs and savings for weeks in Case 3.

It can be seen more clearly on the following figures that there is great improvements in distance cost. The distances are shortened for %22-29. So, the same ratio of cost can be saved by applying the model.

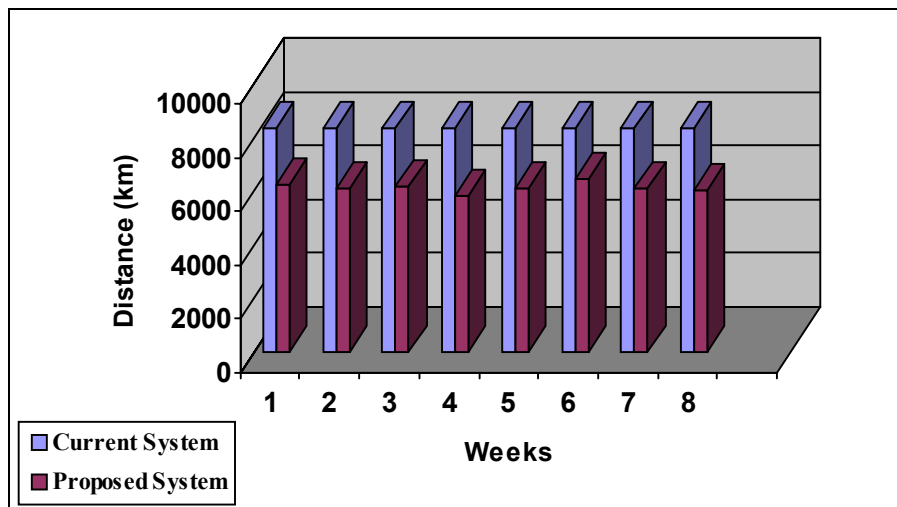


Figure 3.5. Comparison of The Current System with The Proposed System in Distance for weeks in Case 3.

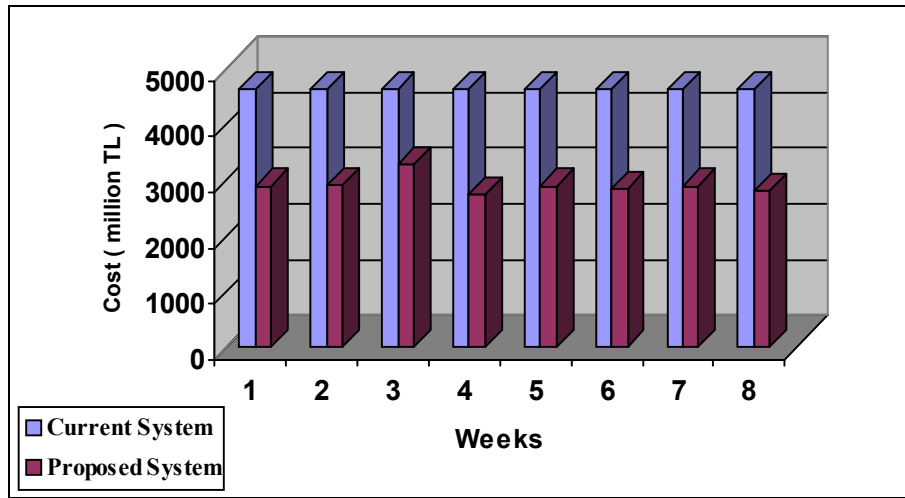


Figure 3.6. Comparison of The Current System with The Proposed System in Cost for weeks in Case 3.

Lastly, we compare the cases in the view of distance and cost for each week. Case 2 including fleets of four vehicles has the minimum distance and cost among the other cases. Figures below show that fleets of four vehicles are enough to meet the demands of points in the most economic way.

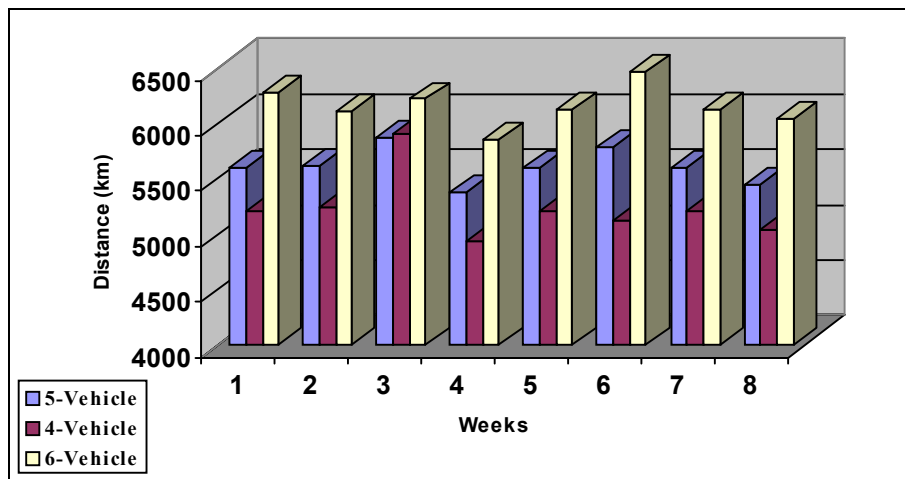


Figure 3.7. Comparison of cases in distance for weeks

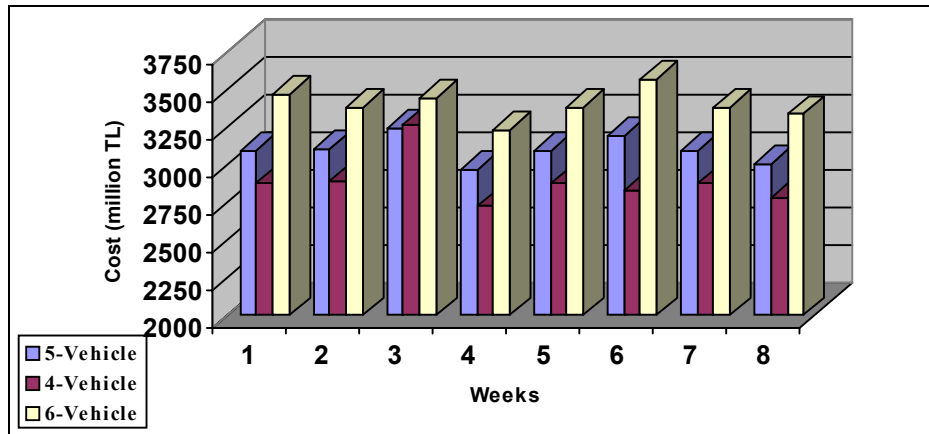


Figure 3.8. Comparison of cases in cost for weeks

CHAPTER 4

CONCLUSION

In this thesis we discussed about the Central Ring Transportation System, which is an important management problem in the field of distribution and logistics in Land Forces. We proposed a mathematical model of the system. The objectives of our study are:

- To find optimal routes which vehicles will follow and to allocate demands to vehicles depending on these optimal routes.
- To decide on vehicle fleet size which satisfies customers' demands in the most economic way.

The model we proposed is in the class of the capacitated vehicle routing problem with backhauls and time windows. Our model has mainly focused on the general assignment based mixed-integer linear programming formulation which utilize subtour elimination constraints (SECs). Results show that our mathematical model can do away with the unproductive mileage. In each case, we can save approximately % 30 of distance (cost) when we compare our model with the current system.

We created our capacitated vehicle routing problem with time windows and backhauls model by using the modeling software GAMS (General Algebraic Modeling System). This process facilitates the coding process of the optimization problems and solves the problems by using another software CPLEX. GAMS provides a high level language for the exact representation of large and complex models and allows change in the model specifications.

We run the model for three cases each of which includes the same eight-week data. For each case, we observed that the proposed system decreases the

transportation costs since non-demanding points are not visited and demanding points are visited only once. For the Case 1, Case 2 and Case3, we can save %30-35, %29-40 and %22-29 of the current transportation cost respectively.

When comparing the cases with each other, we saw that vehicle fleet with four (Case 2) is best to satisfy the demands for eight-week data. We got the shortest and cheapest routes in Case 2. Although all points ((week 5 and week 7) have demands, vehicle fleet with four is enough to satisfy the demands. In our model, the increasing number of vehicles causes to increase the total distance and cost. Because of that it is important to run the model within the different cases to get an optimum vehicle fleet size.

4.1.Future Research Topics

The points selected by the Logistics Command may change with time and Logistics Command may add new points to CRTS. Because of the fact that solution times of the model will increase as the number of demanding points increase, a heuristic solution method which will solve the model in a reasonable amount of time can be employed in such situations.

A mathematical model giving the optimal vehicle fleet size can be developed. So, we can decide on the optimal vehicle fleet size at once without making any comparison among the alternative fleet sizes.

CRTS and RRTS can be united and managed from one center. This requires a vast and detailed communication network with an enormous vehicle fleet. But transportation costs saved may amortize the system in a short time period.

BIBLIOGRAPHY

- [1] ACHUTHAN, N.R., CACCETTA, L., 1991, 'Integer Linear Programming Formulation for a Vehicle Routing Problem', *European Journal of Operational Research* (52), 86-89.
- [2] APPLGATE, D., BIXBY, R., CHAVATAL, V., COOK, W., 1998, 'On the Solution of Traveling Salesman Problems', *Documenta Mathematica*, Extra Volume ICM,III, 645-656
- [3] BALINSKI, M. L., QUANDT, E., 1964, 'On an Integer Program for a Delivery Problem', *Operations Research* (12), 300-304.
- [4] BELLMORE, M., HONG, S., 1974, 'Transformation of the Multisalesman Problem to the Standard Traveling Salesman Problem', *Journal of Association for Computing Machinery*, (21-3), 500-504.
- [5] BELLMORE, M., MALONE, J.C., 1971, 'Pathology of Traveling-Salesman Subtour-Elimination Algorithms', *Operations Research* (19), 278-307.
- [6] BODIN, L.D., GOLDEN, B.L., ASSAD, A., BALL, M., 1983, 'Routing and Scheduling of Vehicles and Crews. The State of Art', *Computers and Operations Research* (10), 69-211.
- [7] BOTT, K., BALLOU, R., 1986, 'Research Perspectives in Vehicle Routing and Scheduling', *Transportation Research* (20A), 239-243.
- [8] BRODIE, G.R., WATERS, C.D.J., 1988, 'Integer Linear Programming Formulation for Vehicle Routing Problems', *European Journal of Operational Research* (34), 403-404.
- [9] BURKARD, R.E., 1979, 'Traveling Salesman Problem and Assignment Problem: A Survey', in: P.L. Hammer, E.L. Johnson, B.H.

- Korte, editors, Discrete Optimization¹, *Annals of Discrete Mathematics* (4), North-Holland, Amsterdam.
- [10] CHRISTOFIDES, N., 1976, ‘Worst-case Analysis of a New Heuristic for the Traveling Salesman Problem’, Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA.
- [11] CHRISTOFIDES, N., 1985, ‘Vehicle Routing’, in: E. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, editors, The Traveling Salesman Problem, Wiley, Chichester, 431-448.
- [12] CHRISTOFIDES,N., MINGOZZI,A., TOTH,P., 1981, ‘Exact Algorithms for the Vehicle Routing Problem Based on The Spanning Tree and Shortest Path Relaxations’, *Mathematical Programming*(20), 255282, 1981.
- [13] CLARKE, G., WRIGHT, J.W., 1964, ‘Scheduling of Vehicles from a Central Depot to a Number of Delivery Points’ *Operations Research* (12), 568-581.
- [14] DANTZIG, G.B., RAMSER, J.H., 1959, ‘The Truck Dispatching Problem’, *Management Science* (6), 80-91.
- [15] DANZIG, G.B., FULKERSON, R. JOHNSON, S.M., 1954, ‘Solution of A Large-Scale Traveling Salesman Problem’, *Management Science* (6), 80-91.
- [16] DESROCHERS, M., LENSTRA, J.K., SAVELSBERGH, M.W.P., SOUMIS, F., 1988, ‘Vehicle Routing with TimeWindows:Optimization and Approximation’, in: B.L Golden, A.A. Assad, editors, Vehicle Routing: Methods and Studies, North-Holland, Amsterdam, 65-84.
- [17] DESROSIERS, J., SOUMIS, F., DESROCHERS, M., 1984, ‘Routing with Time Windows by Column Generation’, *Networks* (14), 545-565.

- [18] EILON, S., WATSON-GANDY, C.D.T., CHRISTOFIDES, N., 1969, 'Distribution Management: Mathematical Modeling and Practical Analysis' *Operational Research Quarterly* (20), 37-53.
- [19] ENCARTA WORLD ENGLISH DICTIONARY, 1999, St.Martins Press, NY, 1060.
- [20] FIECHTER, C.N., 1990, ' A Parallel Tabu Search Algorithm for Large Scale Traveling Salesman Problems', Working Paper 90/1, De'partement de Math'ematiques, E'cole Polytechnique Fe'de'rale de Lausanne.
- [21] FISHER, M.L., 1994, 'Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees', *Operations Research*(42), 626-642.
- [22] FISHER, M.L., JAIKUMAR, R., 1981, 'A Generalized Assignment Heuristic for Solving the VRP', *Networks* (11), 109-124.
- [23] FLOOD, M.M., 1956, 'The Traveling-Salesman Problem', *Operations Research* (4), 61-75.
- [24] GAMBARELLA, L.M., TAILLARD, E., AGAZZI, G., 1999, ' MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows',in: D. Corne, M. Dorigo, F. Glover, editors, New Ideas in Optimization, McGraw-Hill, 63-76.
- [25] GAMS-A User Guide, <http://www.gams.com/docs/document.htm>
- [26] GAVISH, B., SRIKANTH, K., 1986, 'An Optimal Solution Method for the Large-Scale Multiple Traveling Salesman Problem', *Operations Research* (34-5), 698-717.
- [27] GENDREAU, M., HERTZ, A., LAPORTE, G., 1994, A Tabu Search Heuristic for the Vehicle Routing Problem, *Management Science* (40), 1276-1290.

- [28] GOLDEN, B., ASSAD, A., 1986, 'Perspectives on Vehicle Routing: Exiting New Developments', *Operations Research* (34), 803-810.
- [29] GREFENSTETTE, J., ROPAL, R., ROSMAITA, B.J., GUCHT, D., 1985, 'Genetic Algorithms for the Traveling Salesman Problem', in: J.J., Grefenstette, editor, *Proceedings of International Conferance on Genetic Algorithms and Their Appications*, Lawrence Erlbaum Associates, Hillsdale, NJ, 160-168.
- [30] HONG, S., PADBERG, M.W., 1977, 'A Note on the Symmetric Multiple Traveling Salesman Problem with Fixed Charges', *Operations Research* (25-5), 871-874.
- [31] JOHNSON, D.S., MCGEOCH, L.A., 1997, 'The Traveling Salesman Problem: A Case Study in Local Optimization', in: (E.H.L Aarts, J.K. Lenstra, editors, Local Search in Combinatorial Optimization, John Wiley and Sons, London, 215-310.
- [32] JÜNGER, M., REINELT, G., RINALDI, G., 1994, 'The Traveling Salesman Problem', in: M.O. Ball, T.L. Magnananti, C.L. Monma, G.L. Nemhauser, editors, Handbook on Operations Research and the Management Science, North Holland Press, 225-230.
- [33] KINDERVATER, G.A.P., SAVELSBERGH, M.W.P., 1997, ' Vehicle Routing: Handling Edge Exchanges', in: E.H.L Aarts, J.K. Lenstra, editors, Local Search in Combinatorial Optimization, Wiley, Chichester, 337-360.
- [34] KNOX, J., 1988, 'An Application of TABU Search to the Symmetric Traveling Salesman Problem', Ph.D. Thesis, Center for Applied Artificial Intelligence (CAAI), Graduate School of Business, University of Colorado.

- [35] KOLEN, A.W.J., RINNOOY KAN, A.H.G., TRIENEKENS, H.W.M.J., 1987, 'Vehicle Routing with Time Windows', *Operations Research* (35), 266-273.
- [36] KULKARNI, R.V., BHAVE, P.R., 1985, 'Integer Programming Formulations of Vehicle Routing Problems', *Operations Research* (20), 127-132.
- [37] LAPORTE, G., 1992, 'The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms', *European Journal of Operational Research* (59), 231-247.
- [38] LAPORTE, G., 1992, 'The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms', *European Journal of Operational Research* (59), 345-358.
- [39] LAPORTE, G., NOBERT, Y., 1980, 'A Cutting Plane Algorithm for the m-Salesman Problem', *Operations Research Quarterly* (31), 231-247.
- [40] LAPORTE, G., MERCURE, H., NOBERT, Y., 1991, 'A Branch-and-Bound Algorithm for a Class of Asymmetrical Vehicle Routing Problems', *Journal of Operational Research* (43-5), 469-481.
- [41] LAWLER, E.L., LENSTRA, J.K., RINNOOY KAN, A., SHMOYS, D.B., editors, 1985, 'The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization', Wiley, New York.
- [42] LENSTRA, J. K., RINNOY KAN, A. H. G., 1975, 'Some Simple Applications of the Traveling Salesman Problem', *Operational Research Quarterly*(26), 717-734.
- [43] LENSTRA, J. K., RINNOY KAN, A. H. G., 1981, 'Complexity of Vehicle Routing Problems', *Networks* (11), 221-227.

- [44] LIN, S., 1965, 'Computer Solution of the Traveling Salesman Problem', *Bell System Tech. Journal*, (44), 2245-2269.
- [45] LIN, S., KERNIGHAN, B.W., 1973, 'An Effective Heuristic Algorithm for the Traveling Salesman Problem', *Operations Research* (21), 498-516
- [46] MALEK, M., 1988, 'Search Methods for Traveling Salesman Problems', Working Paper, University of Texas, Austin, TX.
- [47] MILLER, C.E., TUCKER, A.W., ZEMLIN, R.A., 1960, ' Integer Linear Programming Formulations of Traveling Salesman Problem', *Journal of Association for Computing Machinery* (7), 326-329.
- [48] MINGOZZI, A., BALDACCI, R., GIORGI, S., 1999, 'An Exact Method for the Vehicle Routing Problem with Backhauls', *Transportation Science* (33-3), 315-329.
- [49] NAHAR, S., SAHNI, S., SHRAGOWITZ, E., 1989, ' Simulated Annealing and Combinatorial Optimization', *International Journal of Computer Aided VLSI Design* (1), 1-23.
- [50] ORLOFF, C.S., 1974, 'Routing a Fleet of M vehicles to/from a Central Facility', *Networks* (4), 147-162.
- [51] REINELT, G., 1994, 'The Traveling Salesman: Computational Solutions For TSP Applications', Springer-Verlag, Germany.
- [52] ROCHAT, Y., TAILLARD, E., 1995, 'Probabilistic Diversification and Intensification in Local Search for Vehicle Routing,' *Journal of Heuristics* (19), 147-167.
- [53] ROSSIER, Y., TROYON, M., LIEBLING, T.M., 1986, 'Probabilistic Exchange Algorithms and the Euclidean Traveling Salesman Problem', *Operations Research Spectrum* (8), 151-164.

- [54] SÖRENSEN, B., 1986, 'Interactive Distribution Planning', Ph.D. Thesis, Technical University of Denmark, Lyngby, DENMARK.
- [55] SURAL, H., BOOKBINDER, J., 2000, 'The Single Vehicle Routing Problem with Unrestricted Backhauls, Working Paper UW.
- [56] SVESTKA, J.A., HUCKFELT, V.E., 1973, 'Computational Experience with an m-Salesman Traveling Problem Algorithm', *Management Science* (19-7), 790-799.
- [57] THOMPSON, P.M., PSARAFTIS, H.N., 1993, 'Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems', *Operations Research* (41-5), 935-946.
- [58] TOTH, P., VIGO, D., 1997, ' An Exact Algorithm for the Vehicle Routing Problem with Backhaul', *Transportation Science* (31), 372-385.
- [59] VOIGT, B.F., 1831, ' Der Handlungsreisende, wie er sein soll und was er zu Thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiss zu sein. Von einem alten Kommiss-Voyageur', Ilmenau, (Republished 1981), Verlag Bernd Schramm, Kiel)
- [60] YANO, C. A., CHAN, T.J., RICHTER, L., CULTER, T., MURTY, K.G., MCGETTIGAN, D., 1987, 'Vehicle Routing at Quality Stores', *Interfaces* (17-2), 52-63.

Appendix A : Data File

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m³)	Weight (1000 kg)	Volume (m³)
DEPOT	0	0	0	0
2	3	4	2	4
3				
4	5	6	4	6
5				
6	3	4	2	1
7	4	4	3	4
8	5	6	4	6
9	6	3	3	3
10	7	7	2	7
11				
12				
13	1	2	2	1
14	4	6	3	6
15	3	8	2	8
16				
17	4	5	1	5
18				
19				
20	6	7	8	7
21	7	7	7	7
22	2	5	7	5
GARAGE	0	0	0	0

Table A.1. Demand (pick-up/delivery) quantities of points for Week 1

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m ³)	Weight (1000 kg)	Volume (m ³)
DEPOT	0	0	0	0
2	7	8	2	2
3	6	5	3	2
4	3	2	0	0
5	2	3	4	5
6				
7	3	3	4	4
8	4	4	6	6
9	7	6	7	6
10				
11	2	3	2	2
12	4	5	5	6
13				
14	6	4	5	4
15				
16	2	3	4	3
17				
18	3	2	2	2
19	5	5	6	7
20	8	9	5	5
21	7	7	7	7
22	8	7	4	6
GARAGE	0	0	0	0

Table A.2. Demand (pick-up/delivery) quantities of points for Week 2.

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m ³)	Weight (1000 kg)	Volume (m ³)
DEPOT	0	0	0	0
2				
3	6	4	4	4
4				
5	5	6	7	7
6	7	7	5	5
7				
8	7	8	8	9
9	6	6	5	5
10				
11	8	9	6	6
12	6	5	6	5
13				
14	8	7	8	7
15				
16				
17	8	9	7	7
18	4	5	6	5
19	6	7	8	7
20	10	6	6	6
21				
22	10	11	7	7
GARAGE	0	0	0	0

Table A.3. Demand (pick-up/delivery) quantities of points for Week 3.

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m ³)	Weight (1000 kg)	Volume (m ³)
DEPOT	0	0	0	0
2				
3	3	3	2	2
4	4	3	3	3
5	2	2	1	1
6	5	5	4	3
7	4	5	4	4
8				
9	3	3	3	3
10	4	3	3	4
11				
12	4	4	2	1
13				
14	3	2	2	2
15	4	6	4	3
16	3	5	2	2
17				
18	5	4	5	5
19				
20	9	10	8	8
21				
22	9	9	10	10
GARAGE	0	0	0	0

Table A.4. Demand (pick-up/delivery) quantities of points for Week 4.

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m ³)	Weight (1000 kg)	Volume (m ³)
DEPOT	0	0	0	0
2	2	2	1	1
3	1	1	1	1
4	0	0	2	2
5	1	1	0	0
6	1	1	0	0
7	2	2	1	1
8	0	0	3	3
9	2	3	3	2
10	1	1	1	1
11	2	2	2	2
12	4	3	1	1
13	0	0	3	2
14	1	2	2	2
15	1	1	0	0
16	0	0	1	1
17	2	3	0	0
18	3	2	1	1
19	1	1	0	0
20	3	3	1	1
21	2	2	2	2
22	5	5	5	5
GARAGE	0	0	0	0

Table A.5. Demand (pick-up/delivery) quantities of points for Week 5.

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m ³)	Weight (1000 kg)	Volume (m ³)
DEPOT	0	0	0	0
2	4	4	3	3
3				
4	3	2	1	1
5				
6				
7				
8	2	2	1	2
9				
10	1	2	3	3
11	2	2	2	2
12	1	1	2	2
13	2	2	1	1
14	2	2	3	3
15				
16	2	1	1	1
17	4	4	3	3
18	2	2	1	1
19				
20	9	10	8	8
21	1	1	1	1
22	9	9	10	10
GARAGE	0	0	0	0

Table A.6. Demand (pick-up/delivery) quantities of points for Week 6.

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m ³)	Weight (1000 kg)	Volume (m ³)
DEPOT	0	0	0	0
2	2	3	4	4
3	1	1	2	1
4	2	2	1	2
5	1	1	0	0
6	1	1	1	1
7	2	2	2	1
8	1	1	1	2
9	2	3	3	3
10	1	1	1	1
11	1	1	0	0
12	1	1	0	0
13	0	0	3	2
14	3	2	2	2
15	1	1	0	0
16	1	1	1	1
17	0	0	1	2
18	0	0	1	1
19	1	1	1	1
20	2	2	4	4
21	2	2	2	1
22	7	7	0	0
GARAGE	0	0	0	0

Table A.7. Demand (pick-up/delivery) quantities of points for Week 7.

	DELIVERY		PICK-UP	
	Weight (1000 kg)	Volume (m ³)	Weight (1000 kg)	Volume (m ³)
DEPOT	0	0	0	0
2	2	2	3	3
3				
4	1	1	2	1
5	2	3	0	0
6	1	1	2	2
7				
8	3	2	2	2
9	1	1	1	1
10				
11				
12	1	1	0	0
13	2	3	2	2
14	1	2	2	1
15	0	0	1	1
16	2	2	0	0
17	1	1	2	3
18				
19				
20	5	5	4	3
21	0	0	1	1
22	8	8	6	6
GARAGE	0	0	0	0

Table A.8. Demand (pick-up/delivery) quantities of points for Week 8.

	G	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
GARAGE	10000	460	404	340	191	233	382	533	580	369	257	443	689	877	258	363	483	424	319	660	761	912
2		10000	56	120	269	337	250	394	572	508	463	890	1044	1232	675	819	939	1079	779	1120	1221	1372
3			10000	64	213	281	294	345	516	452	407	844	988	1176	619	763	883	1023	723	1064	1165	1316
4				10000	149	182	155	301	465	341	303	780	882	924	555	699	829	959	659	1000	1101	1252
5					10000	296	272	423	594	467	422	631	7753	963	445	550	670	810	510	851	952	1103
6						10000	149	300	412	219	146	676	922	1110	338	581	681	821	542	883	984	1135
7							10000	151	322	310	275	825	1047	1235	487	770	830	979	691	1032	1133	1284
8								10000	173	224	323	976	1198	1386	546	812	889	1029	842	1183	1284	1423
9									10000	211	323	1023	1269	1457	546	812	889	1029	867	1208	1309	1423
10										10000	112	812	1058	1246	335	601	678	818	656	997	1098	1212
11											10000	700	946	1134	223	489	566	706	544	885	966	1099
12												10000	246	434	521	339	497	462	194	247	329	480
13													10000	188	767	585	743	685	440	364	263	406
14														10000	955	773	882	742	628	421	320	324
15															10000	218	349	489	327	668	769	883
16																10000	181	316	142	470	587	738
17																	100000	202	326	461	562	596
18																		100000	320	348	446	506
19																			10000	341	442	593
20																				10000	101	252
21																					10000	151
22																						10000
DEPOT	10000	462	406	342	193	235	384	535	582	371	259	445	691	879	260	365	485	426	321	662	763	914

Table A.9. Distance between points

Appendix B: GAMS CODE

SETS

```
i /
GARAGE, POINT2, POINT3, POINT4, POINT5, POINT6, POINT7, POINT8, POINT9, POI
NT10, POINT11, POINT12, POINT13, POINT14, POINT15, POINT16, POINT17, POINT
18, POINT19, POINT20, POINT21, POINT22, DEPOT/
a(i)
/POINT2, POINT3, POINT4, POINT5, POINT6, POINT7, POINT8, POINT9, POINT10, P
OINT11, POINT12, POINT13, POINT14, POINT15, POINT16, POINT17, POINT18, POI
NT19, POINT20, POINT21, POINT22/
b(i)
/GARAGE, POINT2, POINT3, POINT4, POINT5, POINT6, POINT7, POINT8, POINT9, PO
INT10, POINT11, POINT12, POINT13, POINT14, POINT15, POINT16, POINT17, POIN
T18, POINT19, POINT20, POINT21, POINT22/
c(i)
/POINT2, POINT3, POINT4, POINT5, POINT6, POINT7, POINT8, POINT9, POINT10, P
OINT11, POINT12, POINT13, POINT14, POINT15, POINT16, POINT17, POINT18, POI
NT19, POINT20, POINT21, POINT22, DEPOT/

veh vehicle /TYPE1, TYPE2, TYPE3, TYPE4, TYPE5/
```

```
ALIAS(i, j, p)
ALIAS(veh, k)
```

PARAMETERS

```
EARLYT
LATET
D
U
V
TRAVELT
LOADT
DEL1
DEL2
PICK1
PICK2
```

```
EARLYT(i) /GARAGE 1, POINT2 1, POINT3 1, POINT4 1, POINT5 1,
POINT6 1, POINT7 1, POINT8 1, POINT9 1, POINT10 1, POINT11 1,
POINT12 1, POINT13 1, POINT14 1, POINT15 1, POINT16 1, POINT17 1,
POINT18 1, POINT19 1, POINT20 1, POINT21 1, POINT22 1, DEPOT 0/
```

```
LATET(i) /GARAGE 72, POINT2 72, POINT3 72, POINT4 72, POINT5 72,
POINT6 72, POINT7 72, POINT8 72, POINT9 72, POINT10 72, POINT11
72, POINT12 72, POINT13 72, POINT14 72, POINT15 72, POINT16 72,
POINT17 72, POINT18 72, POINT19 72, POINT20 72, POINT21 72, POINT22
72, DEPOT 96/
```

VARIABLES

```
X
Z
T
```

```

M
Y
BINARY VARIABLE X;

EQUATIONS
DIST
SUC
ANT
VEHIC
CAPA
CAPA2
ARR
DEP
TIME1
TIME2
TIME3
LOAD1
LOAD2
LOAD11
LOAD22;

DIST .. Z=E= SUM ((c(i),b(j),veh), D(i,j)*X(veh,i,j));
SUC(a(i)) .. SUM((b(j),veh),X(veh,i,j)) =E=1;
ANT(a(j)) .. SUM((c(i),veh),X(veh,i,j)) =E=1;
VEHIC(veh,a(p)) .. SUM(c(i), X(veh,i,p))- SUM (b(j),
X(veh,p,j))=E=0;
CAPA (veh) .. SUM(a(i), (DEL1(i)*SUM(b(j),X(veh,i,j))))=L=U(veh);
CAPA2 (veh) .. SUM(a(i), (DEL2(i)*SUM(b(j),X(veh,i,j))))=L=V(veh);
ARR(veh) .. SUM((k(veh),a(i)),X(veh,i,"GARAGE"))=E=1;
DEP(veh) .. SUM((k(veh),a(j)),X(veh,"DEPOT",j))=E=1;
TIME1(i) .. T(i)=G= EARLYT(i);
TIME2(i) .. T(i) =L= LATE T(i);
TIME3(c(i),b(j)) .. T(i)+LOADT(i)+TRAVELT(i,j)-T(j)=L=
5000*(1-SUM(veh,X(veh,i,j)));
LOAD1(veh,a(i)) .. Y(i)=L=U(veh);
LOAD2(c(i),b(j)) .. Y(i)+PICK1(i)-DEL1(i)-Y(j)=L=
SUM(veh,X(veh,i,j)));
LOAD11(veh,a(i)) .. M(i)=L=V(veh);
LOAD22(c(i),b(j)) .. M(i)+PICK2(i)-DEL2(i)-M(j)=L=
SUM(veh,X(veh,i,j)));

MODEL TEZ/all/;
*DATA
PARAMETERS

U(veh)
/ TYPE1 35
TYPE2 23
TYPE3 16
TYPE4 20
TYPE5 25 /
V(veh)
/ TYPE1 55
TYPE2 46

```

```
TYPE3      53
TYPE4      40
TYPE5      54 /
```

```
LOADT(i) /GARAGE 1, POINT2 1, POINT3 1, POINT4 1, POINT5 1, POINT6
1, POINT7 1, POINT8 1, POINT9 1, POINT10 1, POINT11 1, POINT12
1, POINT13 1, POINT14 1, POINT15 1, POINT16 1, POINT17 1, POINT18
1, POINT19 1, POINT20 1, POINT21 1, POINT22 1, DEPOT 1/
```

```
PICK1(i) /GARAGE 0, POINT2 4, POINT3 2, POINT4 1, POINT5 0, POINT6
1, POINT7 2, POINT8 1, POINT9 3, POINT10 1, POINT11 0, POINT12 0,
POINT13 3, POINT14 2, POINT15 0, POINT16 1, POINT17 1, POINT18
1, POINT19 1, POINT20 4, POINT21 2, POINT22 0, DEPOT 0/
```

```
PICK2(i) /GARAGE 0, POINT2 4, POINT3 1, POINT4 2, POINT5 0, POINT6
1, POINT7 1, POINT8 2, POINT9 3, POINT10 1, POINT11 0, POINT12
0, POINT13 2, POINT14 2, POINT15 0, POINT16 1, POINT17 2,
POINT18 1, POINT19 1, POINT20 4, POINT21 1, POINT22 0, DEPOT 0/
```

```
DEL1(i) /GARAGE 0, POINT2 2, POINT3 1, POINT4 2, POINT5 1, POINT6
1, POINT7 2, POINT8 1, POINT9 2, POINT10 1, POINT11 1,
POINT12 1, POINT13 0, POINT14 3, POINT15 1, POINT16 1,
POINT17 0, POINT18 0, POINT19 1, POINT20 2, POINT21 2, POINT22
7, DEPOT 0/
```

```
DEL2(i) /GARAGE 0, POINT2 3, POINT3 1, POINT4 2, POINT5 1, POINT6
1, POINT7 2, POINT8 1, POINT9 3, POINT10 1, POINT11 1, POINT12
1, POINT13 0, POINT14 2, POINT15 1, POINT16 1, POINT17 0,
POINT18 0, POINT19 1, POINT20 2, POINT21 2, POINT22 7, DEPOT 0/;
```

```
TABLE D(i,j) (Appendix A, Page 72)
```

```
SCALAR SPEED /70/;
TRAVELT(i,j)=C(i,j)/SPEED;
OPTION RESLIM=60480000;
OPTION ITERLIM = 1000000000;
TEZ.OPTFILE=1;
OPTION OPTCR=0.0;
SOLVE TEZ USING MIP MINIMIZING Z;
```

```
display x.l, T.l;
```