A Method for Pickup and Delivery and Its Application to Truck Route Scheduling

by

Aysu Altun

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Industrial Engineering

Koc University

December 2015

Koc University

Graduate School of Sciences and Engineering

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

Aysu Altun

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

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

____

Prof. Dr. Ceyda Oguz (Advisor)

____

Assoc. Prof. Dr. Sibel Salman

____

Prof. Dr. Necati Aras

Date: ______________________________

ABSTRACT

Vehicle Routing Problem can be briefly explained as distributing goods to a set of customers by using set of vehicles which perform their movements by using appropriate transportation mode such as aviation, land transport such as rail and road, ship transport, pipeline and cable. In particular, the solution of a Vehicle Routing Problem requires determination of a set of routes, each performed by a single vehicle that starts and ends at its own depot, such that all the demand of the customers are fulfilled, all the operational constraints are satisfied, and the overall transportation cost is minimized. The Classical Vehicle Routing Problem (VRP) is one of the most popular problems in combinatorial optimization and it has allowed considerable applications to exact and heuristic solution studies.

In this study, a real world application for Vehicle Routing Problem for Turkish Cargo in Balkan cities is carried out. Delivery routes and pickup routes are modeled and exact solutions are found in General Algebraic Modeling System (GAMS) for a given time period. For the delivery of cargos in Balkan cities a Heterogeneous Fleet of Open Vehicle Routing Problem Model is developed and for the pick up case of cargos a Heterogeneous Fleet of Multi-Depot Open Vehicle Routing Problem Model is developed. A combined backhaul model Heterogeneous Fleet of Vehicle Routing Problem Backhauls with model is written. Comparisons of these models are done. Turkish Cargo requested a GAMS-Excel interface, in order to use the outputs of our models. In GAMS-Excel Interface, a user-friendly application is developed for users of Turkish Cargo. Interface reads and writes the demand values in GAMS file for each month. With the GAMS-Excel interface users in Turkish Cargo can easily see the route for delivery and pickup models for the desired month with a simple notation.

ÖZET

Malların dağıtımı, verilen bir zaman periyodu içinde şoförler tarafından havacılık, karayolu taşımacılığı, demiryolu, gemi taşımacılığı, boru hattı ve kablo ile taşıma çeşitlerinden uygun olan taşıma modu seçilerek müşterilere dağıtılır. Özellikle, Taşıt Rotalama Problemlerinin çözümleri, müşterilerin taleplerini karşılamak, bütün işlevsel kısıtları korumak ve toplam ulaşım maliyetini minimize etmek suretiyle, her biri tek bir araçla o araca ait depoda başlayan ve biten rotalar kümesinin belirlenmesini gerektirir. Tipik Taşıt Rotalama Problemleri kombinasyonal optimizasyonun en popüler problemlerinden biridir ve kesin ve buluşsal çözümlü çalışma uygulamalarının önünü açmıştır.

Bu çalışmada, Taşıt rotalama problemi Turkish Cargo 'nun Balkan şehirlerindeki faaliyeti için gerçek hayata uygulandı. Teslimat rotaları ve ürün toplama rotaları modellendi ve kesin çözümler belirli bir zaman dilimi için GAMS kullanarak bulundu. Balkan şehirlerindeki teslimat kargoları için Heterogeneous Fleet of Open Vehicle Routing Problem, toplama kargoları için Heterogeneous Fleet of Multi-Depot Open Vehicle Routing Problem adlı modelleri yazdık. Ayrica iki modeli toplayan tek bir Heteregeneous Fleet of Vehicle Routing Problem with Backhauls modeli yazdık. Iki farkli modelin karsilastirilmasi yapildi. Turkish Cargo model çözümünün verdiği sonuçları kullanmak istediğini belirtti ve bizden bir GAMS-Excel arayüzü hazırlamamızı talep etti. Bunun için GAMS-Excel arayüzünü yaparken Turkish Cargo memurları için kullanımı kolay bir uygulama yaratmaya çalıştık. Hazırlanan arayüz her ay için talep miktarlarını okuyarak GAMS programının kullanabileceği bir biçime çevirmektedir. Ayrıca GAMS-Excel arayüzü teslimat ve toplama rotalarını gösteren tabloları bütün kullanıcıların anlayabileceği bir notasyona çevirerek Turkish Cargo çalışanlarına kolaylık sağlayacaktır.

ACKNOWLEDGEMENTS

First of all, I would like to thankfully acknowledge the supportive guidance of Prof. Dr. Ceyda Oguz in the all steps of the development of this thesis. I thank her not only for the things I learned from her but also for the respect and understanding I have always felt during my learning phase.

Besides, I would like to thank to Assoc. Prof. Dr. Sibel Salman and Prof. Dr. Necati Aras for taking part in my thesis committee and for their insightful comments and suggestions.

I also want to thank to Koc University for the financial support and the opportunities it has provided.

I thank my admirable parents, Ferda and Aliihsan, and my beloved brother Melih for trusting me at every step I have taken so far. I want to express my thankfulness to my friends in Department of Industrial Engineering, Aybike, Senem, Nazlı and Büşra for standing by me through all good and bad times during the time I spent in Koc University.

I would like to thank my fellow office mates Güneş, Ayşegül, Duygu, Işıl for their valuable and warm friendship; and special thanks to Özgür and Fatih for their willingness to help and their suggestions. I also would like to thank my housemate Serena for her love and patience. I would like to thank specially to my lovely BAL friends Öykü, Bengisu, Hande, Şeyda for providing me motivation during my studies.

I dedicate this thesis to my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

VRP	Vehicle Routing Problem
TSP	Travelling Salesman Problem
LP	Linear Programming
MILP	Mixed Integer Linear Programming
CVRP	Capacitated Vehicle Routing Problem
VRPTW	Capacitated Vehicle Routing Problem with Time Windows
MDVRP	Multi-Depot Vehicle Routing Problem
SDVRP	Split Delivery Vehicle Routing Problem
SVRP	Stochastic Vehicle Routing Problem
PVRP	Periodic Vehicle Routing Problem
DCVRP	Distance Constrained Vehicle Routing Problem
VRPB	Vehicle Routing Problem with Backhauls
HFVRP	Heterogeneous Fleet of Vehicle Routing Problem
i,j	set of cities
k	set of vehicles
b	set of customers
e	set of existing vehicles located at depots from former model
w	set of existing vehicles located at Istanbul
d	set of depots
l	set of cities except Istanbul
x_{ijk}	1 if vehicle k is used between nodes i and j
y_{ik}	1 if city i is the last visited city by vehicle type k
t_{ik}	1 if vehicle type k starts routing from city i

v_{ik}	Auxiliary variable for MTZ subtour elimination constraints
f_k	Fixed cost of using vehicle type k
c_{ij}	Distance between nodes i and j
p_k	Available number of vehicles for each type k
d_i	Demand of city i
Q_k	Capacity of vehicle type k
N	Number of cities
IATA	International Air Transport Association
IST	IATA Code for Istanbul
ATH	IATA Code for Athens
BEG	IATA Code for Belgrade
BUD	IATA Code for Budapest
BUH	IATA Code for Bucharest
KIV	IATA Code for Kishinev
LJU	IATA Code for Ljubljana
PRG	IATA Code for Prague
SJJ	IATA Code for Sarajevo
SKP	IATA Code for Thessaloniki
SKG	IATA Code for Skopje
SOF	IATA Code for Sofia
TGD	IATA Code for Podgorica
TIA	IATA Code for Tirana
VIE	IATA Code for Vienna
WAW	IATA Code for Warsaw
ZAG	IATA Code for Istanbul

Chapter 1

INTRODUCTION

The aim of the Vehicle Routing Problem (VRP) is the distribution of goods, in a given time period, of a set of customers by a set of vehicles, which are located in one or more depots, are operated by a set of drivers, and perform their movements by using an appropriate transportation mode (aviation, land transport which are rail and road, ship transport, pipeline and cable transport). In particular, the solution of a Vehicle Routing Problem calls for the determination of a set of routes, each performed by a single vehicle that starts and ends at its own depot, such that all the demand of the customers are fulfilled, all the operational constraints are satisfied, and the overall transportation cost is minimized. In this section, the typical characteristics of the routing problems by using their main components (road network, customers, depots, vehicles, and drivers), the different operational constraints that can be imposed on the construction of the routes, and the possible objectives to be achieved in the optimization process will be discussed.

The transportation network, used for the distribution of products, is generally described through a graph, whose arcs represent the road sections and whose vertices correspond to the road junctions and to the depot and customer locations. The arcs and graphs can be directed or undirected, depending on whether they can be travelled in only one direction (one-way roads) or in both directions. Each arc has a cost, which generally represents its length, and a travel time, which is possibly dependent on the vehicle type.

The typical information about customers involves their location, amount of goods (demand), possibly of different types, which must be delivered to or collected at the customer, the times required to deliver or collect the goods at the customer location and the subset of the available vehicles that serve the customer (for example, possible access limitations). Sometimes, it is not reasonable to satisfy all of the demands of each customer. The products to be distributed or picked can be reduced, or a subset of customers can be left unserved in these cases. Different priorities, or penalties associated with the partial or lack of service, can be assigned to the customers.

The typical characteristics of the vehicles are the capacity of the vehicles, expressed as the maximum weight, or volume that the vehicles can load and costs associated with utilization of the vehicles (per distance unit, per time unit). The routes must meet several operational constraints, which depend on the nature of the transported goods, on the quality of the service level, and on the characteristics of the customers and the vehicles.

Some typical operational constraints are the following: along each route, the current load of the vehicle cannot exceed the vehicle capacity; the customers served in a route can require only the delivery or the collection of goods, or both possibilities can exist depending on the problem type. Evaluating of the cost of the routes and checking the operational constraints forced on them, require knowledge of the travel cost and the travel time between each pair of customers and between the depots and the customers.

Different objectives can be used for the vehicle routing problems. Typical objectives are minimization of the global transportation cost, dependent on the global distance traveled and on the fixed costs associated with the used vehicles, minimization of the number of vehicles required to serve all the customers, minimization of the penalties associated with partial service of the customers or any weighted combination of these objectives. (Toth and Vigo, 2002a)

In this study, a real world application of Vehicle Routing Problem, which originates from Turkish Cargo, is studied. In the study three mathematical models are presented to determine delivery routes, pickup routes and backhaul routes and the models are solved by using GAMS (General Algebraic Modeling System). In three mathematical models Heterogeneous Fleet of Open Vehicle Routing Problem, Heterogeneous Fleet of Multi-Depot Open Vehicle Routing Problem and Heterogeneous Fleet of Vehicle Routing Problem with Backhaul models are used for delivery, pickup and backhaul problems, respectively. Computational experiments are done with the data, which is given by Turkish Cargo for all models. For backhaul model Toth and Vigo (1996) backhaul instances are also used in order to test the performance of the backhaul model. Also, the GAMS-Excel Interface is prepared for the models. With this interface Turkish Cargo can fill the demand data in Excel and it will be converted into GAMS format. After solution is found, the interface will convert the GAMS output to a simple route notation.

The content of the thesis is as follows: Chapter 2 gives detailed information about the features and types of Vehicle Routing Problem. Background and literature review on Vehicle Routing Problem is provided, and solution methodologies for classes of Vehicle Routing Problem are detailed.

Chapter 3 presents the problem definition of Turkish Cargo's Balkan region problem. Division of the problem into two problems for delivery and pickup problems and combined backhaul mathematical modeling of these problems are explained.

GAMS-Excel interface details and interface specifications are given in Chapter 4. In addition to application of the interface, the procedures of interface are explained in detail. The computational results of the data are also presented in the chapter.

In Chapter 5, the thesis is concluded with a summary of the performed study and future research work.

Chapter 2

LITERATURE REVIEW

In Vehicle Routing Problem (VRP), we have a distribution of goods problem in which vehicles based at a central facility (depot) are expected to visit geographically scattered customers in order to fulfill known customer requirements such as demands. The main goal of the VRP is to find the minimum total traveling costs for individual vehicles, which is the problem of determining optimal delivery from a depot to a geographically distributed customer, subject to constraints.

“The classical VRP lies at the heart of the distribution management. It is faced every day by thousands of companies and organizations engaged in the delivery and collection of goods or people”(Cordeau et. al., 2004). Desired outputs may change from one problem to another, as a result objectives and constraints are tremendously varying.

The Classical VRP is one of the most popular problems in combinatorial optimization and enables a wide range of applications in exact and heuristic solution studies. The aim of the VRP is to find a path with the minimum cost for the vehicles starting from depot and to satisfy the demand of customers located at different locations. VRP is an NP-Hard problem since it is a generalization of Travelling Salesman Problem (TSP) (Baldacci, 2010). VRP is very easy to describe but finding the globally minimum solution of the problem is computationally complex.

Many developments have been achieved after Danztig and Ramser wrote the first publication for the truck-dispatching problem in 1959 (Toth and Vigo, 2002a). In that paper, authors described a real-world application for the delivery of gasoline to gas stations and proposed the first mathematical programming formulation and algorithmic approach for the solution of the problem. A few years later, Clarke and Wright (1964) proposed an effective greedy heuristic that improved on Dantzig-Ramser approach. Many approaches have been developed to solve single-depot VRP. Exact algorithms can be divided into two. Branch and bound was proposed by Fisher (1994) and branch and cut algorithm was proposed by Ladanyi, Ralph and Trotter (2001).

The most effective exact methods were mainly branch and bound algorithms based on straightforward combinatorial relaxations until the late 1980s. Lately, more sophisticated bounds have been proposed, namely those based on Lagrangian relaxations, which have practically increased the size of problems that can be solved to optimality (Cordeau et.al., 2007).

The use of branch and cut algorithm for VRP was first built in exact algorithm of Laporte et. al. (1985). This algorithm uses Linear Programming (LP) relaxation of the model Capacitated Vehicle Routing Problem (CVRP) without capacity constraints as a basis of solution of the VRP with capacity and maximum distance restrictions. This initial relaxation is iteratively strengthened by adding violated capacity constraints, which are heuristically separated by considering the connected components induced by the set of nonzero variables in the current LP solution. The algorithm was capable of solving randomly generated loosely constrained Euclidean and non-Euclidean instances with two or three vehicles and up to 60 customers (Cordeau et.al., 2007).

There exists several types of VRP studied widely in literature. We can briefly present the leading types as Capacitated VRP, Capacitated VRP with Time Windows, Multiple Depot VRP, Split Delivery VRP, Stochastic VRP, Periodic VRP, Distance Constraint VRP, and Heterogeneous Fleet VRP (Toth and Vigo, 2002a).

Capacitated VRP (CVRP)- Each vehicle's capacity starting at depot is limited.

Capacitated VRP with time windows (VRPTW) - All customers service should be done within the pre-specified time limitations.

Multiple Depot VRP (MDVRP) - There may be more than one depot to service the customers.

Split Delivery VRP (SDVRP) - The customers may be served by different vehicles.

Periodic VRP (PVRP) - In a fixed number of days the deliveries need to be done.

Distance Constraint VRP (DCVRP) – The vehicles cannot exceed a pre-specified route length.

Heterogeneous Fleet VRP (HFVRP) – There are different vehicle types with distinct capacities and costs.

The general VRP problem is with *G= (V, A),* being a directed graph where *V* is set of vertices representing *N* customers and *A* is the set of arcs. Vehicles have the capacity of *Q*. In this problem *K* represents the number of vehicles. Each node except depot has a non-negative demand d_i . Likewise, c_{ij} can be defined as travel time, or travel distance for any of identical vehicles from node *i* to *j*. Purpose of VRP is to route the vehicles, each starting and finishing at depot, so that all customers demands are fulfilled and total travel cost is minimized.

The generic Vehicle Routing Problem model is as follows:

The decision variable x_{ij} is 1 if arc *(i,j)* is used, 0 otherwise.

Generic Vehicle Routing Problem (VRP) Model (Toth and Vigo, 2002a)

$$Min\ z = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{1}$$

Subject to

$$\sum_{i \in V} x_{ij} = 1 \qquad \forall j \setminus \{0\} \tag{2}$$

$$\sum_{j \in V} x_{ij} = 1 \qquad \forall i \setminus \{0\} \tag{3}$$

$$\sum_{i \in V \setminus \{0\}} x_{i0} = K \tag{4}$$

$$\sum_{j \in V \setminus \{0\}} x_{0j} = K \tag{5}$$

$$v_i - v_j + Qx_{ij} \leq Q - d_j \qquad \forall i, \forall j \setminus \{0\}, i \neq j \tag{6}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, \forall j \tag{7}$$

The constraints (2) and (3) ensure that exactly one arc enters and leaves each vertex associated with a customer. Similarly, constraints (4) and (5) force degree requirements for depot. In constraint (6) Miller Tucker Zemlin (MTZ) subtour elimination constraints are used. MTZ subtour elimination constraints make sure that every city visited belongs to a tour starting from the depot city. They also utilize undirected arcs in a route.

Two index vehicle flow models solve the basic versions of symmetric and asymmetric CVRP. However, they are generally insufficient for more complex versions of VRP.

To overcome difficulties about showing which vehicle travels on an arc, index k is added to variables. By adding k index, the problem changes to three-index vehicle flow formulation of Symmetric Capacitated Vehicle Routing Problem (SCVRP) and Asymmetric Capacitated Vehicle Routing Problem (ACVRP).

The decision variables x_{ijk} , w_{ik} and v_{ik} are:

$$x_{ijk} = \begin{cases} 1 \text{ if vehicle } k \text{ travels from node } i \text{ to } j, \text{where } (i,j) \in V \text{ and } k \in K; \\ 0, \text{otherwise} \end{cases}$$

$$r_{ik} = \begin{cases} 1 \; if \text{ vehicle k serves customer } i, i \in V \text{ and } k \in K; \\ 0, \text{otherwise} \end{cases}$$

v_{ik} : Auxiliary variable for MTZ subtour elimination constraints

Capacitated Vehicle Routing Model (Toth and Vigo, 2002a)

$$Min\ z = \sum_k \sum_i \sum_j c_{ij} x_{ijk} \tag{8}$$

Subject to

$$\sum_{k=1}^{K} r_{ik} = 1 \qquad \forall i \setminus \{0\} \tag{9}$$

$$\sum_{k=1}^{K} r_{0k} = K \tag{10}$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = r_{ik} \qquad \forall i, \forall k \tag{11}$$

$$\sum_{i \in V \setminus \{0\}} d_i\, r_{ik} \leq Q \qquad \forall k \tag{12}$$

$$v_{ik} - v_{jk} + Q x_{ijk} \leq Q - d_j \qquad \forall i, \forall j \setminus \{0\}, i \neq j, \forall k \tag{13}$$

$$r_{ik} \in \{0,1\} \qquad \forall i, \forall k \tag{14 }$$

$$x_{ijk} \in \{0,1\} \qquad \forall i, \forall j, \forall k \tag{15}$$

The constraints represented by (9), (10) and (11) force that each customer is visited exactly once and vehicles leave the depot and also the same vehicle enters and leaves a given customer. Route continuity is enforced by (11), once a vehicle arrived at a node, it must also leave that node. No vehicle can service customer demands that exceed the vehicle capacity in (12). Constraint (13) is the three index version of Miller- Tucker-Zemlin (MTZ) subtour elimination constraints.

The two explained models are for Capacitated Vehicle Routing Problems. For the Time windows case, the VRPTW is defined on the network *G = (V, A)*, the depot is represented at two cities 0 and *n+1*. All routes should start with node 0 and end with node *n+1*. Earliest possible departure and latest possible arrival times should be defined for the nodes, for example $[a_i, b_i]$=[E,L], where E and L represent earliest possible departure from the depot and latest possible arrival at the depot. For VRPTW we need two different variable groups x_{ijk} and w_{ik} (Toth and Vigo, 2002c), where

w_{ik} : time variable specifies the start of service at city *i* when it is served by vehicle *k*

Objective function, assignment of vehicles to cities, flow and vehicle capacity constraints are the same with the CVRP model but additionally we add 3 different constraints for VRPTW.

$$w_{ik} + s_i + t_{ij} - w_{jk} \leq \left(1 - x_{ijk}\right)M_{ij} \qquad \forall i, \forall j, \forall k \quad \in A \qquad (16)$$

where M_{ij} is a large constant which can be found by $max\left\{b_i + s_i + t_{ij} - a_j, 0\right\}$,where s_i is the service time.

$$a_i \sum_j x_{ijk} \leq w_{ik} \leq b_i \sum_j x_{ijk} \qquad \forall\, i, \forall k \qquad (17)$$

$$E \leq w_{ik} \leq L \qquad \forall\, i, \forall k \qquad (18)$$

Constraints (16), (17) and (18) ensures feasibility for time schedule.

In the Multi Depot Vehicle Routing Problem, we need two different types of decision variables, which are x_{ijk}, which was defined before, and z_{ij} as defined below:

$$z_{ij} = \begin{cases} 1 \text{ if customer } j \text{ is assigned to depot } i\,, i \in I \text{ and } j \in J\,; \\ \qquad\qquad 0, \text{otherwise} \end{cases}$$

For the MDVRP mathematical model, each customer should be assigned to a single route. The constraints for the MDVRP are the capacity constraints for vehicles, subtour elimination constraints and flow conservation constraints. Different from other models, depots may have a capacity V_i and total activity on depot should not exceed this capacity. Constraint (19) solves this problem.

$$\sum_{j \in J} d_j\, z_{ij} \leq V_i \qquad \forall\, i \qquad (19)$$

A customer can only be assigned to a depot only if there is a route from that depot going through that customer. Constraint (20) helps to assign a customer to depot if there is a route going through that customer. Constraints (19) and (20) are added to CVRP model.

$$-z_{ij} + \sum_{u \in I \cup J} \left(x_{iuk} + x_{ujk} \right) \leq 1 \qquad \forall i, \forall j, \forall k \qquad (20)$$

In Split Delivery Vehicle Routing Problem (SDVRP) case, more than one vehicle can serve a customer.

$$\sum_{i=0}^{n} \sum_{k=1}^{K} x_{ijk} \geq 1 \qquad \forall j \qquad (21)$$

Only by adding constraint (21) we can convert CVRP model into SDVRP.

In Periodic Vehicle Routing Problem, each customer should be satisfied in terms of the pre-specified period information of delivery. The decision variable has another index for time.

$$x_{ijkt} = \begin{cases} 1 \text{if vehicle } k \text{ travels from node } i \text{ to } j \text{ at the beginning of the day } t \text{ , where} \\ (i,j) \in V \text{ and } k \in K, t \in T; \\ 0, \text{otherwise} \end{cases}$$

In Distance Constraint Vehicle Routing Problem (DCVRP), the vehicles can travel within a maximum travel distance. In DCVRP case, maximum travel distance constraint (22) is added to CVRP model, where *D* is the maximum travel distance.

$$\sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ijk} \leq D \qquad \forall k \qquad (22)$$

In Open Vehicle Routing Problem (OVRP), the vehicles are not obligated to return to depot (Schrage, 1981). For converting CVRP model to OVRP, constraints (23) and (24) are added.

$$\sum_{j=1}^{n} x_{0jk} \leq 1 \qquad \forall k \qquad (23)$$

$$\sum_{i=1}^{n} x_{i0k} \leq 1 \qquad \forall k \qquad (24)$$

In Vehicle Routing Problem with Backhauls (VRPB) a set of customers is divided in two subsets consisting of linehaul and backhaul customers. The VRPB is also known as linehaul-backhaul problem. VRPB is an extension of the VRP since it involves both delivery and pickup locations.

Linehaul (delivery) points are the locations that need to receive the merchandises from the distribution centers. Backhaul (pickup) points are the locations where the merchandises should be sent back to distribution centers.

Grocery industry can be considered as an application of VRPB. Every shop and market are linehaul customers and grocery suppliers are backhaul customers. VRPB is NP-Hard in the strong sense and many applications can be found in real life routing problems. Each vehicle in VRPB performs one route. In each route vehicle capacity is not exceeded separately for linehaul and backhaul customers. Linehaul customers should be visited first, then vehicles start backhaul customer routing. The routes starting with backhaul customers are not allowed. Similarly with other versions of the problem, the aim is to minimize total traveling cost.

Another important variant of VRP is Heterogeneous Fleet VRP, which was first introduced by Golden et. al. (1984). In HFVRP, each customer is served by exactly one vehicle, each route must begin and end at the depot and total demand of all customers served on a route must not exceed the capacity of the vehicle. In HFVRP, the model selects the number and the capacities of the vehicles in the fleet from the group of different types of vehicles. Different from VRP there is a fixed cost of using a vehicle.

There is a central depot 0, and the customers are presented as $i = 1, \dots, n$. K is the number of vehicle types. Q_k and f_k are the capacity and the fixed cost of vehicle *k*. Different from Capacitated VRP model, we add a new variable, u_{ij} , which is the flow of commodities from *i* to *j* into the model.

Heterogeneous Fleet of Vehicle Routing Problem Model (Golden et. al. ,1984)

$$Min\ z = \sum_{k=1}^{K} f_k \sum_{j=1}^{n} x_{0jk} + \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ijk} \qquad (25)$$

Subject to

$$\sum_{k=1}^{K} \sum_{i=0}^{n} x_{ijk} = 1 \qquad \forall\ j \backslash \{0\} \qquad (26)$$

$$\sum_{i=0}^{n} x_{ijk} - \sum_{l=0}^{n} x_{jlk} = 0 \qquad \forall\ j\ , \forall\ k \qquad (27)$$

$$\sum_{i=0}^{n} u_{ij} - \sum_{l=0}^{n} u_{jl} = d_j \qquad \forall\ j \backslash \{0\} \qquad (28)$$

$$u_{0j} \leq \sum_{k=1}^{K} Q_k\, x_{0jk} \qquad \forall\ j \backslash \{0\} \qquad (29)$$

$$u_{ij} \leq M \sum_{k=1}^{K} x_{ijk} \qquad \forall\ i\ , \forall\ j, i \neq\ j \qquad (30)$$

$$u_{ij} \geq 0 \qquad \forall\ i\ , \forall\ j, i \neq\ j \qquad (31)$$

$$x_{ijk} \in \{0,1\} \qquad \forall\ i\ , \forall\ j, i \neq\ j\ \ , \forall k \qquad (32)$$

In the objective function (25) we are trying to minimize the total transportation cost. Constraints (26) and (27) check that only one vehicle can enter and leave the customer. Constraint (28) controls the demand satisfaction for the customers. Constraint (29) checks that for a trip of vehicle k capacity of vehicle k should not be exceeded. In Constraint (30) if there is no path from city i to j, y_{ij} is set to 0.

As we mentioned before there exists two exact solution methodologies for VRP, these are branch and bound algorithms and branch and cut algorithms. Since VRP is an NP-hard problem, solving the model with exact methods are taking significant time. In order to overcome this issue, various heuristic methods have been developed for the VRP. Initially these were mainly standard route construction algorithms, but more recently powerful metaheuristic approaches have also been developed. We can divide heuristic methods of VRP into two groups; classical heuristics and metaheuristics (Toth and Vigo, 2002b).

Classical heuristics can be divided into three groups: route construction methods, two-phase methods, and route improvement methods.

The first heuristics for the CVRP were route construction methods and still they form the core of many software implementations for various routing applications. These algorithms generally start from an empty solution and iteratively construct routes by adding one or more customers at every iteration, until all customers are in the route of vehicles. The first and most known heuristic of this CVRP problem was proposed by Clarke and Wright (1964) and is derived on the construction of saving, an estimate of the cost reduction obtained by serving two demand vertices consequentially in the same way, rather than in two separate ones. The resulting algorithm is quite fast but may have a poor performance (Laporte and Semet, 2002). Golden et al. (1977), Paessens (1988), and Nelson et al. (1985) have proposed various enhancements to this heuristic with the strategies of the savings approach focus on upgrading either its effectiveness or its computational efficiency with better data structures (Toth and Vigo, 2002b).

Mole and Jameson (1976) offered another classical route construction heuristic which contained the sequential insertion algorithm. The algorithm uses a selection and insertion criterion for the evaluation of the extra distance resulting from the insertion of an unrouted customer k between two consecutive vertices i and j on the current route. Christofides et al. (1979) proposed a more general and effective two-step insertion heuristic. As a first step, a sequential insertion algorithm is used to determine a set of feasible routes. A parallel insertion approach is applied as the second step. The resulting algorithm is better than Mole and Jameson (1976) algorithm and represents a good compromise between effectiveness and efficiency (Toth and Vigo, 2002b).

In two-phase heuristics, two-phase methods are based on the decomposition of the VRP solution into two separate sub problems:

(1) Clustering: Customer subsets which are corresponding to a route is determined in this phase.

(2) Routing: Sequence of customers on each route is determined.

In a cluster-first-route-second method, customers are first grouped into clusters and the routes are then determined by sequencing the customers within each cluster appropriately. Different techniques have been proposed for the clustering phase, while the routing phase comes up to solving a TSP.

Bramel and Simchi-Levi (1995) worked with fixed number of vehicles with another two-phase method. This algorithm determines route offsprings by solving a capacitated location problem. The customers are selected by minimizing the total distance between each customer and its closest offspring, and by imposing that the total demand associated with each route assigned to a vehicle should be less than capacity Q. After offsprings have been determined and the single-customer routes are created, the remaining customers are added in the current routes by minimizing insertion costs. The location based approach of

Bramel and Simchi-Levi produces better solutions but requires much larger computing times.

Significant amount of metaheuristic methods have been applied to the VRP. With respect to classical heuristics, they perform a more extensive search of the solution space. Metaheuristics can be broadly divided into three classes:

(1) Local search: including deterministic annealing.

(2) Population search: including genetic search and adaptive memory procedures.

(3) Learning mechanisms: including neural networks and ant colony optimization.

Local search algorithms explore the solution space by iteratively moving from a solution x_t at iteration t to a solution x_{t+1} in the neighborhood $N(x_t)$ of x_t until a stopping criterion is satisfied. If $f(x)$ denotes the travelling cost for problem, then $f(x_{t+1})$ is not necessarily smaller than $f(x_t)$. Because of that reason, some mechanisms must be implemented to avoid cycling.

Deterministic annealing was first applied to the VRP by Golden et al. (1998). In this study, Golden et. al. (1998) used deterministic annealing concept in 20 large scale instances of VRP. This study did not yield effective solutions when it is compared to the other heuristic methods with respect to solution quality.

Tabu search depends on various tactics to implement tabu tenures (also known as short term memory), search diversification (also known as long term memory), and search intensification, which is emphasizing the search in a promising area of the search space. Genetic algorithm is performed by repeating the following operation k times: extract two parent solutions from the populations to create two offspring using a crossover operation, and apply a mutation operation to change each offspring with a probability; then remove the *2k* worst elements from the population and replace them with the new *2k* offsprings. Several crossover rules are available for sequencing problems (Bean, 1994; Potvin, 1996; Drezner, 2003; Prins, 2004). The offspring is created by extracting and recombining elements of several parents in adaptive memory procedures. In the initial version proposed by Rochat and Taillard (1995) for the VRP, nonoverlapping routes are extracted from several parents to create a partial solution. This solution is then completed and optimized by tabu search (Toth and Vigo, 2002b).

Neural networks are models formed of richly interconnected units of weighted links like neurons in the brain. Neural networks slowly construct a solution through a feedback mechanism to find a better match for an observed output to a described output. In VRP neural network models named as the elastic net and the self-organizing map are deformable templates to generate a feasible VRP solution. Ghaziri (1993) provided a neural network solution (Toth and Vigo, 2002b).

Chapter 3

PROBLEM DEFINITION AND MATHEMATICAL MODELING

3.1 Introduction

Turkish Airlines is the national carrier airline of Turkey, with its headquarter located at Turkish Airlines General Management Building in Yesilkoy, Istanbul. Turkish Airlines serves 41 domestic and 210 international airports around the Europe, Asia, Africa and America. Turkish Airlines is the forth-largest carrier in the world, with 251 flight destinations. Istanbul Ataturk Airport is the main hub of Turkish Airlines. Besides Istanbul Ataturk Airport they have their secondary hubs at Esenboga International Airport, Sabiha Gokcen International Airport and Adnan Menderes International Airport. Turkish Airlines has a fleet of 257 airplanes, of which, 248 of them are passenger airplanes.

Turkish Airlines has a sub-company for cargo delivery operations, known by the name Turkish Cargo. Turkish Cargo has 9 cargo aircrafts, and serves 47 destinations around worldwide. Around Turkey, Turkish Cargo delivers cargos with road transportation rather than air transportation since airfreight has some limitations on handling, cost and safety. Turkish Cargo department wants to use the same concept of road transportation in Balkan deliveries since the distances are very similar compared to Turkey.

In this chapter, we formulate appropriate Vehicle Routing Problem for the implementation of real life cargo deliveries of Turkish Cargo for Balkan region. We separated problem into two different solution methodologies. The first solution methodology is for delivery and pickup case while the second one is the combined version which is the backhaul model. We start by explaining the delivery pickup model and then we continue with the backhaul model that we had written for this real life problem.

3.2 Problem Definition

Turkish Cargo is a subsidiary company of Turkish Airlines. The firms have a fleet of 248 passenger and 9 cargo airplanes. Turkish Airlines passenger planes transport cargos to all of their 251 destinations. Because of air transportation restrictions and high of cost of transporting cargos with planes, Turkish Cargo also wants to use road transportation as an alternative option. Turkish Cargo has 9 cargo aircrafts, 3 of them are Airbus A310 type and they can carry 38610 kg and the other 6 aircrafts are Airbus A330-200F type and they can carry 64000 kg per flight.

A transportation hub (interchange) is a place where we can change transportation modes for cargos and passengers. In our problem Istanbul is the main hub. Istanbul is at a very strategic point since it is located between Europe and Asia and also nearly all of the Turkish Airlines scheduled foreign flights fly from Istanbul. Therefore, gathering all cargo in Istanbul is not difficult for Turkish Cargo.

Besides the advantages of having a strategic hub, Turkish Cargo can deliver their cargos to Europe with road transportation. Road transportation is cheaper than airfreight because they do not have to pay handling and jet fuel costs. Because of airfreight regulations Turkish Cargo cannot carry cargos heavier than 250 kilograms and dangerous materials due to possible accident risks. Some cargos need special delivery conditions such as the ones that require refrigeration. In cargo and passenger planes, Turkish Cargo does not have refrigeration conditions; hence, they are carried in storage docks. Other reasons for road transportation can be listed as: not all cities have airports and some cities have scheduled flights only for some days of the week. They can overcome these problems by using road transportation. As of 2014, Turkish Cargo uses road transportation for 20 destinations.

These cities are the ones with high demand values in Turkey. For delivering cargo in Turkey they are using simultaneous deliver and pick model. Turkish Cargo wanted us to optimize the truck routes for Balkan cities since they want to use a similar approach for Balkans. The distances that they will travel using truck routes for Balkans is almost same as the distances they are using in Turkey. However, Turkish Cargo cannot simultaneously deliver and pick cargos from Balkan cities because of European Union laws. They have to first deliver the cargos, and then they have to pick the cargos from the Balkan cities. Because of that we have to divide our model into two parts. First we have to deliver the cargos, and then we have to pick the cargos from the cities. This real world implementation will increase their profit since they are forming the routes based on their experiences. The optimal truck routes for Balkan cities will increase the carried cargo demands and also increase the customer satisfaction. The objective of this thesis is to find the optimal truck routes for Turkish Cargo. In this chapter we are going to explain how we model this real world application of cargo distribution for Turkish Cargo.

3.3 Delivery Model

In our delivery model, all trucks should start from Istanbul since it is the main hub for Turkish Airlines. All the cargo from all over the world comes to Istanbul then they are delivered to their final destinations. For the Balkan delivery problem, we have 16 cities. These are; Athens (ATH), Belgrade (BEG), Budapest (BUD), Bucharest (BUH), Kishinev (KIV), Ljubljana (LJU), Prague (PRG), Sarajevo (SJJ), Thessaloniki (SKP), Skopje (SKG) , Sofia (SOF), Podgorica (TGD), Tirana (TIA), Vienna (VIE), Warsaw (WAW) and Zagreb (ZAG). The abbreviations in parenthesis are the IATA airport codes. For delivering the cargo starting from Istanbul (IST) is Heterogeneous Fleet of Open Vehicle Routing Problem. In our problem we have two different types of vehicles; trucks and lorries. Since we are trying to determine how many trucks and lorries that we need, we set the default vehicle number to 10. In the delivery model we have 5 trucks and 5 lorries waiting in the depot Istanbul. Each truck has a capacity of 13000 kilograms and each lorry has a capacity of 20000 kilograms. Fixed cost of using a truck and lorry are 5790 and 8340 Euros. Cost of using a vehicle per kilometer is 0.3 Euros. All the distances between cities are taken from Google Maps and according to distances a symmetric distance matrix is formed. The aim of this delivery model is to find optimal routes for delivering the cargos from Istanbul to Balkan cities with cargo demands.

3.3.1 Mathematical Modeling for Delivery Model

Indices

i, j : Cities

k : Vehicles

Sets

$I = \{\text{IST}, \text{ATH}, \text{BEG}, \text{BUD}, \text{BUH}, \text{KIV}, \text{LJU}, \text{PRG}, \text{SJJ}, \text{SKP}, \text{SKG}, \text{SOF}, \text{TGD}, \text{TIA}, \text{VIE}, \text{WAW}, \text{ZAG}\}$ Set of cities

$J = \{\text{IST}, \text{ATH}, \text{BEG}, \text{BUD}, \text{BUH}, \text{KIV}, \text{LJU}, \text{PRG}, \text{SJJ}, \text{SKP}, \text{SKG}, \text{SOF}, \text{TGD}, \text{TIA}, \text{VIE}, \text{WAW}, \text{ZAG}\}$ Set of cities

$K = \{\text{truck}, \text{lorry}\}$ Set of vehicle types

Parameters

f_k: Fixed cost of using vehicle type k

c_{ij}: Distance between nodes i and j

d_i : Demand of city i

Q_k: Capacity of vehicle type k

p_k: Available number of vehicles for each type k

Decision Variables

z: Objective function value

x_{ijk}: 1 if vehicle type k travels between city i to city j

y_{ik}: 1 if city i is the last visited city by vehicle type k

v_{ik} : Auxiliary variable for MTZ subtour elimination constraints

Model

$$Min\ z = \sum_{j=0}^{n}\sum_{k=1}^{K} f_k x_{0jk} + \sum_{k=1}^{K}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ijk} \qquad (31)$$

Subject to:

$$\sum_{j=1}^{n} x_{0jk} \leq p_k \qquad \forall k \qquad (32)$$

$$\sum_{j=1}^{n}\sum_{k=1}^{K} x_{ijk} = 1 - \sum_{k=1}^{K} y_{ik} \qquad \forall i\backslash\{0\}, i \neq j \qquad (33)$$

$$\sum_{j=0}^{n}\sum_{k=1}^{K} x_{jik} = 1 \qquad \forall i\backslash\{0\}, i \neq j \qquad (34)$$

$$\sum_{j=0}^{n} x_{ijk} + y_{ik} = \sum_{j=0}^{n} x_{jik} \qquad \forall i\backslash\{0\}, i \neq j, \forall k \qquad (35)$$

$$\sum_{i=1}^{n} y_{ik} = \sum_{j=1}^{n} x_{0jk} \qquad \forall k \qquad (36)$$

$$v_{ik} - v_{jk} + Q_k x_{ijk} + (Q_k - d_i - d_j)x_{jik} \leq Q_k - d_j \qquad \forall i\backslash\{0\}, \forall j\backslash\{0\}, i \neq j, \forall k \qquad (37)$$

$$d_i \leq v_{ik} \leq Q_k \qquad \forall i\backslash\{0\}, \forall k \qquad (38)$$

$$x_{i0k} = 0 \qquad \forall i\backslash\{0\}, \forall k \qquad (39)$$

$$v_{ik} \geq 0 \qquad \forall i, \forall k \qquad (40)$$

$$x_{ijk} \in \{0,1\} \qquad \forall i, \forall j, \forall k, \quad i \neq j \qquad (41)$$

$$y_{ik} \in \{0,1\} \qquad \forall i, \forall k \qquad (42)$$

Equation (31) is the objective function that; finds the minimum travelling costs, which are fixed costs of using vehicles and total travelled distance cost. In constraint (32), the number of vehicles that start routing from depot should be less than the number of vehicles that we have for each type. In constraint (33), if city i is the last city that the vehicle type k travels, we are forcing that x_{ijk} variable to 0. In constraint (34), every city i should be visited once. Route continuation constraints are expressed in equation (35). The number of vehicles for each type k should be equal to sum of last cities variable y_{ik}. Equations (37) and (38) are the three-index version of Lifted - Miller Tucker Zemlin subtour elimination constraints. Equation (39) eliminates the returns to depot. Last three equations are the non-negativity and binary variable definitions.

3.4 Pick Up Model

In our pick up model, we are going to pick the deliveries up that need to be sent to worldwide destinations from Balkan cities to Istanbul. The entire pick up cargos should be first collected in Istanbul since nearly all flights originate from Istanbul. After being collected, the cargos are sent to final destinations. Different from the delivery model, in this model we are going to use the vehicles that we established in delivery model. The last cities that the vehicles reached will be new depots for our new model. The vehicles will start collecting cargos from these depot cities. The model should save the depots and the corresponding vehicles accordingly. In pick up model we need to use the vehicles that are originating from the depots first. If the capacity of these vehicles is not enough, then we should send new vehicles from Istanbul in order not to lose customers. The pick up model is Heterogeneous Fleet of Multi-Depot Open Vehicle Routing Problem. In this model, the depots should not have demands, so we should remove the depots demand from the vehicle that is assigned to that depot from the delivery model. The vehicles start routing with their full capacities. All vehicles should return to Istanbul. In this model, we have to divide the cities into 3 groups; customers, depots and Istanbul. For the vehicles we need two sets; one for the vehicles starting from Istanbul and another one for the vehicle starting from their own depots. The capacities and fixed costs of using vehicles are the same with the delivery model. In Pick Up Model our aim is to find the optimal routes for picking up cargo from Balkan cities.

3.4.1 Mathematical Modeling for Pick Up Model

Indices

i, j : Cities

k : Vehicles

Sets

$I = \{\text{IST}, \text{ATH}, \text{BEG}, \text{BUD}, \text{BUH}, \text{KIV}, \text{LJU}, \text{PRG}, \text{SJJ}, \text{SKP}, \text{SKG}, \text{SOF}, \text{TGD}, \text{TIA}, \text{VIE}, \text{WAW},$ ZAG } Set of cities

$J = \{\text{IST}, \text{ATH}, \text{BEG}, \text{BUD}, \text{BUH}, \text{KIV}, \text{LJU}, \text{PRG}, \text{SJJ}, \text{SKP}, \text{SKG}, \text{SOF}, \text{TGD}, \text{TIA}, \text{VIE}, \text{WAW},$ ZAG } Set of cities

$K = \{\text{truck}, \text{lorry}\}$ Set of vehicle types

Parameters:

c_{ij}: Distance between nodes i and j

d_i : Demand of city i

Q_k: Capacity of truck type k

N: Number of cities

p_k: Available number of vehicles for each type k

Decision Variables:

x_{ijk}:1 if truck k is used between nodes i and j

t_{ik} : 1 if vehicle type *k* starts routing from city i

v_{ik} : Auxiliary variable for MTZ subtour elimination constraints

Model

$$Min\ z = \sum_{k=1}^{K}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ijk} + \sum_{j=0}^{n}\sum_{k=1}^{K} f_k x_{0jk} \tag{43}$$

$Subject\ to$:

$$\sum_{j=1}^{n} x_{0jk} \le p_k \qquad \forall k \tag{44}$$

$$\sum_{i=1}^{n} x_{i0k} = \sum_{j=1}^{n} x_{0jk} + \sum_{i=1}^{n} t_{ik} \qquad \forall k \tag{45}$$

$$\sum_{j=0}^{n}\sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall i\backslash\{0\}\ , i \neq j \tag{46}$$

$$\sum_{j=1}^{n} x_{i0k} = \sum_{j=1}^{n} x_{0jk} + t_{ik} \qquad \forall i\backslash\{0\}\ , i \neq j\ , \forall k \tag{47}$$

$$v_{ik} - v_{jk} + Q_k x_{ijk} + \left(Q_k - d_i - d_j\right)x_{jik} \le Q_k - d_j \qquad \forall i\backslash\{0\}, \forall j\backslash\{0\}\ , i \neq j\ , \forall k \tag{48}$$

$$d_i \le v_{ik} \le Q_k \qquad \forall i\backslash\{0\}, \forall k \tag{49}$$

$$x_{ijk} \in \{0,1\} \tag{50}$$

$$t_{ik} \in \{0,1\} \tag{51}$$

$$v_{ik} \ge 0 \tag{52}$$

Objective function (43) calculates the total travel costs and works towards a minimum. The number of vehicles start routing from depot should be less than the number of vehicles that we have for each type is forced by constraint (44). In constraint (45) the sum of number of vehicles returning to depot should be equal to sum of new vehicles starting from depot and existing vehicles for each type *k*. In constraint (46), every city *i* should be visited once. Route continuation constraints are expressed in equation (47). Constraints (48) and (49) are the three-index version of the lifted Miller Tucker Zemlin subtour elimination constraint. Last three equations are the non-negativity and binary variable definitions.

3.5 Backhaul Model

In our combined model, we are going to deliver the entire delivery cargos that need to be sent to Balkan cities from worldwide destinations originating from Istanbul. Similar to our delivery model, we have 16 Balkan cities and each delivery cargo should be sent to these cities. The fleet starts routing from Istanbul. In the first phase of the model each truck and lorry delivers the delivery cargos from worldwide destinations and then in the second phase they start picking up the Balkan cities demand to worldwide cities and bring them to main hub Istanbul. The combined model is Heterogeneous Fleet of Vehicle Routing Problem with Backhauls. In this model, all vehicles start from main hub Istanbul and they deliver the demands without exceeding user specified lorry and trucks. With their own capacities the vehicles start routing. All vehicles should stop by and deliver the cargos to Balkan cities if their demand is greater than zero. Similarly same vehicles start picking up the demands in Balkan cities and bring them back to Istanbul. In this model, we have to divide the cities into 3 groups; linehaul customers, backhaul customers and Istanbul. Linehaul customers and backhaul customers are the same Balkan cities. In the backhaul model, the separated models are combined in a new model. The combined model solves the problem in one step. In order to separate the delivery and pickup demands we created new backhaul cities and their demand values are the pickup demand values that are given from the firm. Because of backhaul model specifications, we doubled the city size. They are combined as linehaul and backhaul customers. In the combined backhaul model, linehaul customers must be served before backhaul customers. Distance matrix is adapted to new model. The capacities and fixed costs of using vehicles are the same with the delivery and pickup model. In Combined Model our aim is to find the optimal routes for Balkan cities without separating the model into two parts.

3.5.1 Mathematical Modeling for Backhaul Model

Indices

i, j : Cities

k : Vehicles

Sets

$I = \{\mathrm{IST, ATH, BEG, BUD, BUH, KIV, LJU, PRG, SJJ, SKP, SKG, SOF, TGD, TIA, VIE, WAW,}$ ZAG } Set of all linehaul and backhaul cities$-\{0, \dots, 2\mathrm{n}\}$

$J = \{\mathrm{IST, ATH, BEG, BUD, BUH, KIV, LJU, PRG, SJJ, SKP, SKG, SOF, TGD, TIA, VIE, WAW,}$ ZAG } Set of all linehaul and backhaul cities $-\{0, \dots, 2\mathrm{n}\}$

$K = \{\mathrm{truck, lorry}\}$ Set of vehicle types

E: Set of linehaul customers except Istanbul $\{1, \dots, \mathrm{n}\}$

H: Set of backhaul customers except Istanbul $\{\mathrm{n}+1, \dots, 2\mathrm{n}\}$

L: Set of cities except Istanbul $\{1, \dots, 2\mathrm{n}\}$

Parameters:

c_{ij}: Distance between nodes i and j

d_i : Demand of city i

Q_k: Capacity of truck type k

N: Number of cities

Decision Variables:

x_{ijk}:1 if truck k is used between nodes i and j

v_{ik} : Auxiliary variable for MTZ subtour elimination constraints

$$Min\ z = \sum_{j=0}^{2n}\sum_{k=1}^{K} f_k x_{0jk} + \sum_{k=1}^{K}\sum_{i=0}^{2n}\sum_{j=0}^{2n} c_{ij} x_{ijk} \tag{53}$$

$Subject\ to$:

$$\sum_{j=1}^{2n} x_{0jk} \leq\ p_k \qquad \forall k \tag{54}$$

$$\sum_{i=1}^{2n} x_{i0k} = \sum_{j=1}^{2n} x_{0jk} \qquad \forall k \tag{55}$$

$$\sum_{j=0}^{2n}\sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall i\backslash\{0\} \tag{56}$$

$$\sum_{j=0}^{2n}\sum_{k=1}^{K} x_{jik} = 1 \qquad \forall i\backslash\{0\} \tag{57}$$

$$\sum_{j=0}^{2n} x_{ijk} = \sum_{j=0}^{2n} x_{jik} \qquad \forall i\backslash\{0\},\ \forall k \tag{58}$$

$$v_{ik} - v_{jk} + Q_k x_{ijk} + \left(Q_k - d_i - d_j\right) x_{jik} \leq Q_k - d_j \qquad \forall i \in E, \forall j \in E\ , i \neq j\ , \forall k \tag{59}$$

$$v_{ik} - v_{jk} + Q_k x_{ijk} + \left(Q_k - d_i - d_j\right) x_{jik} \leq Q_k - d_j \qquad \forall i \in H, \forall j \in H\ , i \neq j\ , \forall k \tag{60}$$

$$d_i \leq v_{ik} \leq Q_k \qquad \forall i\backslash\{0\}, \forall k \tag{61}$$

$$x_{ijk} = 0 \qquad \forall i \in H, \forall j \in E\ , \forall k \tag{62}$$

$$x_{ijk} \in \{0,1\} \qquad \forall i, \forall j, \forall k \tag{63}$$

$$v_{ik} \geq 0 \qquad \forall i, \forall k \tag{64}$$

Equation (53) is the objective function of the combined model, which aims to minimize the total cost. The first part of this equation relates to the fixed cost of using different types of vehicles trucks, lorries and the second part sums the total travelling cost. The number of vehicles start routing from depot should be less than the number of vehicles that we have for each type is forced by constraint (54). In constraint (55) the number of vehicles start routing from depot should be equal to the number of vehicles returning to depot. Constraints (56) and (57) ensure that each city is visited once. Constraint (58) provides the route continuity. Equations (59), (60) and (61) are three-index version of generalized lifted Miller Tucker Zemlin subtour elimination constraints (Kara. et. al., 2004). Since linehaul routes and backhaul routes have different capacity restrictions, the lifted Miller Tucker Zemlin constraints are separated for linehaul customers and backhaul customers. Constraint (62) sets backhaul to linehaul routes to 0. Last three equations are the non-negativity and binary variable definitions.

Chapter 4

COMPUTATIONAL EXPERIMENTS AND GAMS-EXCEL INTERFACE

The VRP is an NP-Hard problem, since it is a generalization of Travelling Salesman Problem. In this thesis, all of the problems studied are NP-Hard problems since they are generalizations of VRP. In this chapter we are going to explain the data that we get from Turkish Cargo and how we adjusted them for our models. Also, we will explain the computational experiments for the data that we have. After computational experiments, we will explain the GAMS-Excel interface developed for Turkish Cargo. With this interface the users can input their data easily and also can read the output in a format that is familiar to them.

4.1 Data Analysis

We are using the data of Turkish Cargo's monthly Balkan delivery and pickup amounts. We have the data for the years 2011 and 2012 for months from January to October. In these data, we can see the demand for cargos for Balkan cities coming from all Turkish Airlines worldwide destinations. In our models we are only using the data for Balkan cities. For calculating one month data we had to sum all the demand for delivery cargos from worldwide destinations to a Balkan city and similarly sum all demand for pickup cargo. In order to gather our data for all models we made this operation for all 16 Balkan cities. All the pickup and delivery data for a city, which are used to be in different files, are combined. In interface it is easier to read one excel file rather than two files. After all operations we saved the demand for pickup and delivery cargo in a neat format in an excel file. Users in Turkish Cargo can easily modify this demand values.

4.2 Computational Experiments

We code the VRP models for delivery, pickup and backhaul in GAMS (General Algebraic Modeling System) software and Gurobi software. The computational experiments are performed on a machine with an Intel(R) Core(TM) i5-2520M CPU at 2.50 GHz processor with 4.00 GB RAM. We have 20 months data for delivery and 20 months data for pickup. In computational experiments we divide our experiments in two sections. Primarily, we start with delivery pick up model, and then conclude with the backhaul model. For each model type, experiments are done with Miller Tucker Zemlin subtour elimination constraints and Lifted Miller Tucker Zemlin subtour elimination constraints. Also, we tested our backhaul model with Toth and Vigo (1996) VRPB test instances using GAMS Cplex solver and Gurobi solver.

4.2.1 Computational Experiments for Delivery Pickup Model

Tables 4.1 and 4.2 show the values of the objective function and CPU time and optimality status of delivery model. In the second column, objective function values for the delivery models are listed. Third column is the CPU time; in all cases we have a stopping criterion of 50000 seconds. In the fourth column, the optimality gaps are shown. In the fifth column, the objective function values for the models that lifted Miller Tucker Zemlin constraints used is shown. In the last two columns, CPU times and optimality gaps of the delivery model are shown for the lifted versions.

Table 4.1 Objective Function Value, CPU Times and Optimality Values of Delivery Model for 2011 Data

Month	MTZ Value	CPU Time	Optimality/ Gap	Lifted MTZ Value	CPU Time	Optimality/ Gap
11-Jan	48698	15264	23.08%	48406	39748	22.32%
11-Feb	51321	50000	26.69%	51202	50000	27.62%
11-Mar	57130	50000	28.89%	54861	22622	22.84%
11-Apr	55617	50000	23.14%	55935	50000	23.13%
11-May	51048	48960	20.68%	50903	50000	20.77%
11-Jun	55078	39957	23.27%	55087	38970	25.01%
11-Jul	48927	50000	20.78%	49383	50000	21.78%
11-Aug	47487	50000	27.93%	48487	45434	33.83%
11-Sep	53492	46527	20.23%	53492	50000	20.31%
11-Oct	58241	50000	23.15%	58183	50000	22.50%

Table 4.2 Objective Function Value, CPU Times and Optimality Values of Delivery Model for 2012 Data

Month	MTZ Value	CPU Time	Optimality/ Gap	Lifted MTZ Value	CPU Time	Optimality/ Gap
12-Jan	68505	38750	22.79%	49254	50000	10.00%
12-Feb	57421	50000	22.16%	57421	50000	20.80%
12-Mar	54861	50000	26.25%	54861	50000	22.84%
12-Apr	51065	50000	23.24%	55077	50000	25.80%
12-May	42110	50000	23.23%	51065	50000	23.32%
12-Jun	41390	50000	31.42%	42110	42359	31.42%
12-Jul	48886	50000	23.11%	49383	50000	24.91%
12-Aug	39318	50000	28.47%	39318	50000	29.51%
12-Sep	48824	50000	33.83%	48895	50000	33.93%
12-Oct	33346	50000	22.50%	33346	50000	22.08%

In Tables 4.3 and 4.4, objective function values, CPU times and optimality statuses are shown for the pick up models. In the second column, objective function values for the pickup models are listed. Third column is the CPU time; similarly in all cases we have a stopping criterion of 50000 seconds. In the fourth column, the optimality gaps are shown for pickup models with Miller Tucker Zemlin subtour elimination constraints. In the fifth column, the objective function values for the models that lifted Miller Tucker Zemlin subtour elimination constraints used is shown. In the last two columns, CPU times and optimality gaps are shown for the lifted versions of pickup model.

Table 4.3 Objective Function Value, CPU Times and Optimality Values of Pickup Model for 2011 Data

Month	MTZ Value	CPU Time	Optimality/Gap	Lifted MTZ Value	CPU Time	Optimality/Gap
11-Jan	10628	0.647	optimal	9153	1.242	optimal
11-Feb	10378	0.425	optimal	9835	0.174	optimal
11-Mar	20999	0.449	optimal	20999	0.998	optimal
11-Apr	10868	0.234	optimal	10013	0.282	optimal
11-May	8751	1.79	optimal	8402	0.919	optimal
11-Jun	9324	0.101	optimal	9324	8.356	optimal
11-Jul	9242	0.466	optimal	9856	0.227	optimal
11-Aug	8413	0.293	optimal	8491	0.321	optimal
11-Sep	9996	0.284	optimal	9996	0.473	optimal
11-Oct	10539	0.431	optimal	9958	0.304	optimal

Table 4.4 Objective Function Value, CPU Times and Optimality Values of Pickup Model for 2012 Data

Month	**MTZ Value**	**CPU Time**	**Optimality/Gap**	**Lifted MTZ Value**	**CPU Time**	**Optimality/Gap**
12-Jan	19867	0.287	optimal	19394	0.515	optimal
12-Feb	12507	0.651	optimal	12507	0.880	optimal
12-Mar	19757	18.124	optimal	19757	53.42	optimal
12-Apr	20320	0.593	optimal	20320	0.947	optimal
12-May	8632	1.337	optimal	8632	0.932	optimal
12-Jun	8730	0.647	optimal	8730	0.827	optimal
12-Jul	21191	1.07	optimal	20114	0.301	optimal
12-Aug	20019	1.05	optimal	20019	0.78	optimal
12-Sep	8365	0.959	optimal	8486	0.837	optimal
12-Oct	7459	0.845	optimal	7459	1.467	optimal

As it can be seen from Tables 4.1 to 4.4 solving the model takes significant times. Since VRP is an NP-Hard problem, in some cases due to out of memory errors the model could not find the optimal value with our computer. In our computational experiments, all of the experiments are done with the Turkish Cargo’s monthly cargo delivery and pickup data. Although Turkish Cargo gave the monthly data, it is required to schedule the vehicle routings on a weekly basis. They stated that solving time would not be a problem for them as long as they got the optimal solutions. As the demand of a city exceeds vehicle capacity, the interface adds a new city as additional variable into the model and divides the demand value of that city appropriately. Because of that operation, the number of decision variables increases and CPU time needed to solve the problem also increases. Turkish Cargo enters the weekly demand and they can also enter the maximum time from the interface for solving the models. When demand values in the model are converted into weekly data, it makes the problem easier to solve and the CPU time will decrease. All cities may not have delivery and pickup demands every week because the city sizes in the problems are reduced. Similarly, the vehicle sizes to serve these customers are also reduced. Reduction in customer and vehicle sizes decreases the number of decision variables and constraints. As a result of this reduction, the problems become easier to solve. With the help of the interface, Turkish Cargo can also change the CPU time limits for the models. When they keep CPU time shorter, the solution quality for the models decreases, and on the contrary when they keep CPU times longer the models will find better solutions. As a result, they will decide the tradeoff between solution time and solution quality.

4.2.2 Computational Experiments for Backhaul Model

We start computational experiments for backhaul models by solving the firms' monthly data and after that we make our computational experiments on Toth and Vigo (1996) VRPB instances. Tables 4.5 and 4.6 show the values of the objective function; CPU time and optimality status for the backhaul model solutions for the datasets of Turkish Cargo in years 2011 and 2012. In the backhaul model, the separated models are combined in a new model to solve the problem in one step. In the first column, problem months are given. In the second column objective function values of the backhaul models are listed. Third column is the CPU time; similarly in all cases we have a stopping criterion of 50000 seconds. In the fourth column, the optimality gaps are shown for backhaul models with Miller Tucker Zemlin subtour elimination constraints. In the fifth column, the objective function values for the models that lifted Miller Tucker Zemlin subtour elimination constraints used is shown. In the last two columns CPU times and optimality gaps are shown for the lifted versions of backhaul model. In order to solve linehaul backhaul problem in one step for the firms data, the cities are separated into two groups, linehaul customers and backhaul customers.

Table 4.5 Objective Function Value CPU Times and Optimality Values of Backhaul Model for 2011 Data

Month	MTZ Value	CPU Time	Optimality/Gap	Lifted MTZ Value	CPU Time	Optimality/Gap
11-Jan	57333	50000	17.36%	57333	50000	17.06%
11-Feb	60634	50000	25.18%	60634	50000	24.35%
11-Mar	72047	50000	26.91%	67577	50000	26.50%
11-Apr	66089	50000	23.12%	66109	50000	23.19%
11-May	59462	50000	17.83%	59305	50000	18%
11-Jun	64636	50000	19.54%	64126	50000	14.63%
11-Jul	57500	50000	26.87%	57500	50000	26.05%
11-Aug	57436	50000	31.09%	57436	50000	31.09%
11-Sep	66061	50000	24.12%	65054	50000	20.47%
11-Oct	67936	50000	23.18%	68139	50000	22.95%

Table 4.6 Objective Function Value CPU Times and Optimality Values of Backhaul Model for 2012 Data

Month	MTZ Value	CPU Time	Optimality/Gap	Lifted MTZ Value	CPU Time	Optimality/Gap
12-Jan	58393	50000	23.78%	59270	50000	27.03%
12-Feb	67855	50000	20.83%	67885	50000	21.07%
12-Mar	68399	50000	34.05%	68018	50000	34.81%
12-Apr	64879	50000	20.90%	64622	50000	14.90%
12-May	59697	50000	23.96%	60557	50000	20.01%
12-Jun	50321	50000	24.57%	50321	50000	24.29%
12-Jul	60944	50000	24.37%	60902	50000	24.80%
12-Aug	56533	50000	21.70%	48023	50000	12.08%
12-Sep	57180	50000	31.20%	57180	50000	30.15%
12-Oct	40202	50000	25.59%	40202	50000	27.85%

Table 4.7 Backhaul Objective Function Value, CPU Times and Optimality Values for TV96 Instances with Gurobi Application and Cplex

Problem Features					Gurobi Optimizer Solutions			Cplex Solver Solutions		
Problem	**Size**	**LH**	**BH**	**FS**	**Objective**	**CPU**	**Gap**	**Objective**	**CPU**	**Gap**
TV-1	22	11	10	5	371	0.8	optimal	371	0.01	optimal
TV-2	22	14	7	4	366	1.29	optimal	366	1.07	optimal
TV-3	22	16	5	5	372	4.17	optimal	372	2.72	optimal
TV-4	22	17	4	4	375	30.04	optimal	364	0.09	optimal
TV-5	23	11	11	3	677	0.14	optimal	677	0.02	optimal
TV-6	23	15	7	3	640	2.48	optimal	640	0.05	optimal
TV-7	23	17	5	3	191	2.73	optimal	191	0.06	optimal
TV-8	23	18	4	3	623	0.71	optimal	623	0.03	optimal
TV-9	30	15	14	4	501	49.15	optimal	501	0.22	optimal
TV-10	30	20	9	4	537	229.45	optimal	541	5.24	optimal
TV-11	30	22	7	4	544	93.92	optimal	544	110	optimal
TV-12	30	24	5	4	514	535.26	optimal	514	91	optimal
TV-13	33	16	16	5	738	600	0.09%	738	600	0.13%
TV-14	33	22	10	4	775	600	0.14%	750	600	0.11%
TV-15	33	24	8	4	768	600	0.18%	783	600	19.49%
TV-16	33	26	6	4	792	600	0.21%	793	600	0.21%
TV-17	51	25	25	6	559	9.26	optimal	559	3.24	optimal
TV-18	51	34	16	6	548	600	0.01%	548	311.7	optimal
TV-19	51	38	12	6	572	600	0.02%	578	600	0.03%
TV-20	51	40	10	5	576	600	0.06%	577	600	0.06%
TV-21	76	38	37	11	776	600	0.11%	772	600	0.10%
TV-22	76	50	25	11	802	600	0.12%	770	600	0.11%
TV-23	76	57	18	11	873	600	0.22%	853	600	0.20%
TV-24	76	60	15	10	829	600	0.21%	841	600	0.22%

Table 4.7 (continued) Backhaul Objective Function Value, CPU Times and Optimality Values for TV96 Instances with Gurobi Application and Cplex

Problem Features					**Gurobi Optimizer Solutions**			**Cplex Solver Solutions**		
Problem	**Size**	**LH**	**BH**	**FS**	**Objective**	**CPU**	**Gap**	**Objective**	**CPU**	**Gap**
TV-25	101	50	50	8	848	600	0.06%	845	600	0.05%
TV-26	101	67	33	9	862	600	0.06%	874	600	0.07%
TV-27	101	75	25	8	890	600	0.12%	902	600	0.13%
TV-28	101	80	20	8	977	600	0.19%	913	600	0.13%
TV-29	76	38	37	15	522	600	0.15%	680	600	0.19%
TV-30	76	50	25	15	940	600	0.25%	930	600	0.14%
TV-31	76	57	18	14	859	600	0.20%	982	600	0.20%
TV-32	76	60	15	15	1002	600	0.24%	966	600	0.21%
TV-33	101	50	50	14	976	600	0.18%	982	600	0.19%
TV-34	101	67	33	14	1057	600	0.23%	1083	600	0.24%
TV-35	101	75	25	14	1112	600	0.19%	1213	600	0.20%
TV-36	101	80	20	14	1168	600	0.22%	1154	600	0.21%
TV-37	76	38	37	9	755	600	0.09%	721	600	0.04%
TV-38	76	50	25	9	757	600	0.07%	759	600	0.07%
TV-39	76	57	18	9	793	600	0.13%	795	600	0.14%
TV-40	76	60	15	9	761	600	0.14%	778	600	0.16%
TV-41	76	38	37	7	712	600	0.03%	704	600	0.02%
TV-42	76	50	25	7	720	600	0.02%	724	600	0.02%
TV-43	76	57	18	7	726	600	0.06%	727	600	0.06%
TV-44	76	60	15	8	711	600	0.08%	717	600	0.09%

LH=Number of Linehaul Customers, BH= Number of Backhaul Customers, FS= Fleet Size

In order to test our models we solve 44 instances with GAMS Cplex optimizer and Gurobi optimizer with Toth and Vigo (1996) VRPB instances. Also, for the sake of making performance comparison of GAMS Cplex Optimizer, we also coded Gurobi optimizer for our backhaul model. In Table 4.7 problem sizes, linehaul and backhaul city sizes, fleet sizes, the objective function value, CPU time and optimality status are shown for TV96 instances separately for Gurobi Optimization Application and GAMS Cplex Solver. In TV96 instances, the city sizes are changing between 22 and 101 and in each case the number of vehicles are changing between 3 and 15. TV96 instances have separate linehaul and backhaul vehicles. In TV96 problem, Toth and Vigo (1996) used linehaul vehicles for linehaul customers and backhaul vehicles for backhaul customers. Since we made our modeling according to company specifications, we summed the number of vehicles assigned for linehaul and backhaul customers. The solutions for Gurobi Optimization Application and GAMS Cplex Solver are listed in Table 4.7. In the first five columns problem characteristics are given. In the sixth column objective function values for TV96 instances and in the seventh and eighth columns CPU times and optimality gaps are given when the model is solved with Gurobi Application. In the ninth column objective function values are given for GAMS Cplex Solver. Tenth and eleventh columns are the CPU times and optimality gaps with respect to GAMS Cplex Solver.

All the instances are solved within a time limit of 600 seconds. When we compare the results of Gurobi optimizer and Cplex optimizer, Cplex optimizer outperforms. Nearly in all cases Cplex solver found better results than Gurobi optimizer.

4.3 Comparison of Delivery Pickup Model and Backhaul Model

We solved our model with two different perspectives. In the first model we first deliver the worldwide cargo demands for Turkish Cargo to Balkan cities with using Heterogeneous Fleet of Open Vehicle Routing Problem model and then we note the stopping city as a dummy depot and pick the pickup cargos from Balkan cities and bring them back to Istanbul with using Heterogeneous Fleet of Multi Depot Open Vehicle Routing Problem. In the second model we used Heterogeneous Fleet of Vehicle Routing Problem with Backhauls and combined the first two models in one model. Furthermore, we compared the objective function values with using and without using lifted version of Miller Tucker Zemlin subtour elimination constraints. In Table 4.8, total costs found by four models are given. In the second and third columns, the total costs found from delivery pickup models and integrated backhaul models without using lifted Miller Tucker Zemlin subtour elimination constraints are shown. In the fourth and fifth columns, total costs found from delivery pickup models and backhaul models with using lifted Miller Tucker Zemlin subtour elimination constraints are shown.

In most cases lifted version of backhaul model found better results than delivery pickup model. In 3 months we found the best objective function values in delivery pickup model. Because of the optimality gaps, backhaul models could not outperform in these months when we compare it with the delivery pickup models. In 17 months, backhaul models outperform the combined delivery pickup models. In these 17 months, in 12 months with using Lifted Miller Tucker Zemlin subtour elimination constraints we found better objective function values. In the first solution perspective, we think of the problem as two consecutive problems. In the first phase, we try to find the optimal routes for delivering the Balkan region cargos, and after that we select the last cities as new depots for the pick up model. In the second phase, we start routing from these depots and try to collect the pickup cargo demands from Balkan region. In some cases the total capacity for the vehicles is not sufficient. In order to overcome this infeasibility we send extra vehicles from Istanbul for collecting pickup cargo demands. Sending extra vehicles from Istanbul increases the objective function value. On the other hand, in the combined backhaul model we doubled the city number size and with the new constraints, model became harder to solve. Since the city numbers are doubled, decision variables increased. Also, in the backhaul model we added new constraints such as all linehaul customers must be served before backhaul customers. These changes in the backhaul model made the model harder to solve and in almost all months optimality gaps increased and more time needed to solve the problem with optimality.

Table 4.8 Comparison of Objective Function Value for two Modeling aspects with lifted Miller Tucker Zemlin versions and without lifted Miller Tucker Zemlin

Month	**Delivery Pickup Value**	**Backhaul Model Value**	**Lifted Delivery Pickup Value**	**Lifted Backhaul Model Value**
11-Jan	59326	**57333**	57559	**57333**
11-Feb	61699	**60634**	61037	**60634**
11-Mar	78129	72047	75860	**67577**
11-Apr	66485	66089	**65948**	66109
11-May	59799	59462	**59305**	**59305**
11-Jun	64402	64636	64411	**64126**
11-Jul	58169	**57500**	59239	**57500**
11-Aug	**55900**	57436	56978	57436
11-Sep	63488	66061	**63488**	65054
11-Oct	68780	**67936**	68141	68139
12-Jan	88372	**58393**	68648	59270
12-Feb	69928	67855	69928	67885
12-Mar	74618	68399	74618	**68018**
12-Apr	71385	64879	75397	**64622**
12-May	**50742**	59697	59697	60557
12-Jun	**50120**	50321	50840	50321
12-Jul	70077	60944	69497	**60902**
12-Aug	59337	56533	59337	**48023**
12-Sep	57189	57180	57381	**57180**
12-Oct	40805	**40202**	40805	**40202**

4.4 GAMS-Excel Interface

In this section, the GAMS – Excel interface, which is requested by Turkish Cargo, is explained. The interface will read the monthly excel data that the Turkish Cargo users fill. Then, these data will be input to the models when the user choses the appropriate model. The solution of the model is extracted back to the user in a simple notation.

In the pickup model certain changes such as constraints for the depots and vehicles have to be made each time. We designed the interface to determine and handle these changes according to the data extracted from the excel file. For the vehicles interface should do the operations for the vehicle capacities because the depots should not have any demand. After writing the model with these constraints and pickup demands users need to start the pickup GAMS Model. After the solution is found, users have to enter the solution into the interface and they can easily see the route for delivery and pickup models for the desired month. The interface is coded with C# language. In Figures 4.1 and 4.2 the algorithms related to the interface are presented.

The interface makes the model preparation and solution visualization user friendly. The user does not see the operations behind the GAMS and Excel. The interface combines them and writes the model according to changing demand. The user never sees the GAMS output file, he only uploads the solution files into interface. The users clearly can see the routes trucks and the solution time together in the interface screen. When the new data comes the user only has to change the excel file for the desired time period. For someone who has very little optimization theory and GAMS knowledge, this interface is a lifesaver.

GAMS-Excel Interface Algorithm for Delivery and Pickup Models

1. Set up GUI including buttons and text box.
2. Check vehicle numbers entry when set number of vehicles button is pressed and record number of lorries and truck when they are valid.
3. Upon clicking delivery call pop up window and browse for GAMS delivery model file
4. When file is selected read the file and store it in a string. The string contains separator phrases to mark where imported data goes.
5. Browse for excel files after clicking the excel file button
6. Call Microsoft Office reference library to open selected excel file and import data.
7. If the file format is correct, store city codes, delivery and pickup demands in arrays
8. Insert the imported delivery demand information into the string.
9. Write combined string into a new GAMS file.
10. Browse for GAMS solution file after load delivery solution button is clicked.
11. Find and extract vehicle route data from the loaded file.
12. Form the delivery route information for each vehicle and display the routes on interface.
13. Browse for pick up template model file and load selected file.
14. Store file contents in a string and locate separators.
15. Insert remaining truck capacities, pick up demands, constraints and other information into stored pickup template.
16. Write combined pickup model into a new GAMS file.
17. Browse for GAMS solution file upon clicking load pick up solution button.
18. Load the selected file and extract vehicle route data.
19. Form the pick up route information for each vehicle and display the routes on interface.

Figure 4.1 GAMS-Excel Interface Algorithm for Delivery and Pickup Models

GAMS-Excel Interface Algorithm Backhaul Models

1. Set up GUI including buttons and text box.
2. Check vehicle numbers entry when set number of vehicles button is pressed and record number of lorries and truck when they are valid.
3. When file is selected read the file and store it in a string. The string contains separator phrases to mark where imported data goes.
4. Browse for excel files after clicking the excel file button.
5. Call Microsoft Office reference library to open selected excel file and import data.
6. If the file format is correct, store city codes, delivery and pickup demands in arrays.
7. Insert the imported delivery demand information into the string.
8. Write combined string into a new GAMS file.
9. Browse for GAMS solution file after load solution button is clicked.
10. Find and extract vehicle route data from the loaded file.
11. Form the delivery and pickup route information for each vehicle and display the routes on interface.
12. Search for GAMS log file. If found, extract execution time information and display on interface.

Figure 4.2 GAMS-Excel Interface Algorithm for Backhaul Models

Chapter 5

CONCLUSIONS AND FUTURE RESEARCH

5.1 Conclusions

In this thesis, we study Turkish Cargo's real world application of Balkan region truck delivery and pickup problem. We divide problem into two parts for delivery and pickup routes. We write the models according to Turkish Cargo's specifications.

In Delivery Problem we use Heterogeneous Fleet of Open Vehicle Routing Problem. All trucks and lorries are starting from Istanbul and travelling to 16 Balkan cities. If the demand is higher than lorry capacity, we do pre-processing in order to prevent infeasibility. In each month, we fulfill the demand for these 16 Balkan cities with optimal routes. We try to find the optimal cost of transportation cost based on distances and fixed cost of vehicles.

In Pickup Problem we use Heterogeneous Fleet of Multi-Depot Open Vehicle Routing Problem. The vehicles from the delivery model start from the Balkan cities that they end in the first model and all other vehicles start from Istanbul. The depot cities do not have demand values so we remove their values from the vehicles' capacities. The vehicles start routing with these values. If the supply of these vehicles are not enough we send extra vehicles from Istanbul to collect the remaining demand values from the customer Balkan cities. In this pickup model we try to find the optimal cost of transportation cost based distances.

In Backhaul Problem we use Heterogeneous Fleet of Vehicle Routing Problem with Backhauls (HFVRPB). All vehicles start from main hub Istanbul and they deliver the demands without exceeding user specified lorry and trucks. After finishing delivering the demand cargos, all vehicles should start picking up the demands in Balkans cities and bring them back to Istanbul. In this model, our aim is to find minimum cost for transportation. The backhaul model finds better solutions with respect to transportation costs when we compare it with the delivery pickup models.

In all problem cases, we compared our problem instances with using Miller Tucker Zemlin subtour elimination constraints and with using lifted Miller Tucker Zemlin subtour elimination constraints. The model performed better when using lifted Miller Tucker Zemlin subtour elimination constraints and optimality gaps and CPU times are reduced.

We used VRPB test instances of Toth and Vigo (1996) for testing our backhaul model. Gurobi Optimization Application and GAMS Cplex Solver are compared with these test instances. GAMS Cplex Solver performed better in our problem. Datasets from the firm are solved using GAMS Cplex Solver.

In GAMS-Excel Interface, we attempt to make a user-friendly application for users of Turkish Cargo. Interface reads and writes the demand values in GAMS file for each month. Furthermore, it displays the optimal routes for delivery model by reading at the GAMS output for delivery model. Then, interface separates the customer Balkan cities and depot Balkan cities by looking at the GAMS output and make the subtraction operation for vehicles and write the GAMS file for pickup model by looking at the monthly pickup demands. After solving the model, the interface shows the optimal pickup routes for the month by looking at the GAMS output file.

5.2 Future Research

Vehicle Routing Problem is used by thousands of companies and organizations everyday since it is first proposed by Dantzig in 1959. Since VRP is an NP-Hard problem, it takes a lot of time to solve the problem optimally. In our thesis, we are using specifications of VRP problem which are Heterogeneous Fleet of Open Vehicle Routing Problem, Heterogeneous Fleet of Multi-Depot Vehicle Routing Problem and Heterogeneous Fleet of Vehicle Routing Problem with Backhauls. These models take a significant amount of time to solve. In order to overcome this problem, as a future research Constraint Programming or heuristic methods such as Genetic Algorithms can be used. These methods will decrease CPU times for problems but this can hurt the possibility of finding the global optimal solution. There will be a tradeoff between time to reach a solution and solution quality. Since this thesis is a real world application we want to give the optimal results for Turkish Cargo. If Turkish Cargo accept the decrease in solution quality, heuristic methods and Constraint Programming can be used as a future research.

BIBLIOGRAPHY

R. Baldacci, E. Bartolini and G. Laporte. Some Applications of Generalized Vehicle Routing Problem . *Journal of Operations Research Society*, 61: 1072–1077, 2010.

G. B. Dantzig and J. H. Ramser. The Truck Dispatching Problem. *Management Science*, 6:80-91,1959.

G. Clarke and J. W. Wright. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research,* 12: 568-581, 1964.

L. Ladányi, T. K. Ralphs, and L. E. Trotter Jr. Branch, Cut, and Price: Sequential and Parallel. *Computational Combinatorial Optimization*, 2241: 223-260, 2001.

G. Laporte , Y. Nobert and M. Desrochers. Optimal Routing under Capacity and Distance Restrictions. *Operations Research*, 33: 1050-1073, 1985.

B. L. Golden, T. L. Magnanti and H. Q. Nyugen. Implementing Vehicle Routing Algorithms. *Networks*, 7:113-148, 1977.

B. L. Golden, A. A. Assad and R. Dahl. Analysis of a large-scale vehicle routing problem with an inventory component. *Large Scale Systems*, 7: 181-190, 1985.

G. Laporte and F. Semet. Classical heuristics for the capacitated VRP. *SIAM Monographs on Discrete Mathematics and Applications*. 109-128, 2002.

H. Paessens. The savings algorithm for the vehicle routing problem. *European Journal of Operational Research*, 34: 336–344, 1988.

M.D. Nelson., K.E. Nygard, J. H. Griffin, W.E. Shreve. Implementation techniques for the vehicle routing problem. *Computers & Operations Research*, 12, 273–283, 1985.

R.H. Mole and S.R. Jameson. A sequential route-building algorithm employing a generalized savings criterion. *Operational Research Quarterly*, 27: 503–511, 1976.

N. Christofides, A. Mingozzi and P. Toth. The vehicle routing problem, *Combinatorial Optimization. Wiley,* 315–338, 1979.

J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. *Operations Research,* 43: 649–660, 1995.

B. L. Golden, E. A. Wasil, J. P. Kelly and I-M Chao. Metaheuristics in vehicle routing. *Fleet Management and Logistics. Kluwer Academic.* 33–56, 1998.

F. Li, B.L. Golden and E. A. Wasil. Very large-scale vehicle routing: New test problems, algorithms and results. *Computers & Operations Research.* 32: 1165–1179, 2005.

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

J.C. Bean. Genetic algorithms and random keys for the sequencing and optimization. *ORSA Journal on Computing.* 6*:* 154–160, 1994.

J.-Y. Potvin, Genetic algorithms for the traveling salesman problem. *Annals of Operations Research.* 63: 339–370, 1996.

Z. Drezner. A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing.* 15:320–330, 2003.

C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research.* 31: 1985–2002, 2004.

Y. Rochat and É.D Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1:147–167,1995.

H. Ghaziri. Algorithmes connexionistes pour l’optimisation combinatoire. Thèse de doctorat, *École Polytechnique Fédérale de Lausanne, Switzerland.* 1993.

J.F. Cordeou, G. Laporte, M. W. P. Savelsbergh and D. Vigo. Vehicle Routing. *Handbook in OR &MS,* 14:365-426, 2007.

P. Toth and D. Vigo. An Overview of Vehicle Routing Problems,*The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, 1-53, 2002a.

P. Toth and D. Vigo . Classical Heuristics for the Capacitated VRP. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, 109-126, 2002b.

P. Toth and D. Vigo. VRP with Time Windows. *The Vehicle Routing Problem. SIAM* Monographs on Discrete Mathematics and Applications, 157-158, 2002c.

L. Schrage. Formulation and structure of more complex/realistic routing and scheduling problems. *Networks.* 11: 229–232, 1981.

İ. Kara, G. Laporte and T. Bektaş. A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operation Research*.158: 793–795, 2004.

M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*.10: 27–36, 1991.

P.Toth, and D. Vigo. A heuristic algorithm for the vehicle routing problem with backhauls. *Transportation Analysis*, 585-608, 1996.

P.Toth, and D. Vigo. An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science,* 372-385, 1997