Solving Multi-Depot Location Routing Problem Using Lagrangian Relaxation

by

Zeynep Özyurt

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

Master of Science
in
Industrial Engineering

Koç University

January 2007

Koç University

Graduate School of Sciences and Engineering

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

Zeynep Özyurt

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:

_____

Deniz Aksen, Ph. D. (Advisor)

_____

Necati Aras, Ph. D.

_____

Sibel Salman, Ph. D.

Date: _____

ABSTRACT

The design of a distribution logistics system requires quite a number of decisions of different planning levels. The most important strategic decision is the locations of distribution centers, which are also referred to as depots. The allocation of customers to the depots is a decision of tactical level, while determining vehicle routes to visit those customers belongs to the operational level. Multi-depot Location-Routing Problem (MDLRP) involves the decisions of different levels simultaneously. In the problem, the optimal number and locations of depots are decided while allocating customers to depots and determining vehicle routes to visit all customers.

In this thesis, we propose a nested Lagrangian relaxation-based method named [LR-TS] for the 2-layer discrete uncapacitated MDLRP. An outer Lagrangian relaxation embedded in subgradient optimization decomposes the parent problem into two subproblems. The first subproblem is a facility location-like problem. It is solved to optimality with Cplex 10.0.

The second one resembles a capacitated and degree constrained minimum spanning forest problem, which is tackled with an augmented Lagrangian relaxation. The lower bound to the true optimal solution of the comprehensive problem is obtained by summing the objective function value of the Cplex solution of **SubP1** and the lower bound found for **SubP2**.

The solution of the first subproblem reveals a depot location plan. As soon as a new distinct location plan is found in the course of the subgradient iterations, a tabu search algorithm is triggered to solve the multi-depot vehicle routing problem associated with that plan, and a feasible solution to the parent problem is obtained. Its objective value is checked against the current upper bound on the parent problem's true optimal objective value.

The performance of the proposed method is evaluated based on the gap between the best upper bound and the best lower bound achieved. [LR-TS] has been tested on a number of randomly generated test problems as well benchmarking problems from LRP literature, and the results have been tabulated.

ÖZET

Dağıtım sistemlerinin dizayn edilmesi için değişik planlama düzeylerinde bir çok karar alınması gerekir. Kaç adet dağıtım merkezinde faaliyet gösterileceğinin ve bu merkezlerin yerlerinin belirlenmesi stratejik seviyede alınması gereken kararlardır. Hangi deponun hangi müşteriye hizmet vereceği taktik seviyede ele alınırken, teslimat rotaları operasyonel seviyede belirlenir. Çoğul Depolu Tesis Yeri Belirleme - Rotalama Problemi (ÇDTYRP) değişik düzeydeki bu kararları birlikte değerlendirir. Bu problemde, toplam maliyeti enküçülten depo sayısı ve yerleri, her müşteriye hangi deponun hizmet verdiği ve teslimat rotaları eşzamanlı olarak belirlenir.

Bu tez, 2 seviyeli ayrık ve kapasite kısıtsız ÇDTYRP için [LR-TS] adını verdiğimiz iç içe geçmiş iki Lagrange gevşetmeye dayanan bir çözüm yöntemi önermektedir. Subgradient eniyileme yöntemi içerisine oturtulmuş olan dıştaki Lagrange gevşetme ana problemi iki alt probleme ayırmaktadır. İlk alt problem **SubP1**, tesis yeri belirleme problemine benzemektedir ve Cplex 10.0 ile en iyi çözümü elde edilmektedir. Kapasite kısıtlı en küçük kapsayan orman problemine benzeyen ve **SubP2** olarak adlandırılan diğer alt problem ise ilk Lagrange gevşetmenin içine yerleştirilmiş bir çoğalan Lagrange gevşetme yöntemiyle çözülmektedir. **SubP1**'in Cplex çözümünden elde edilen amaç fonksiyon değeri ve çoğalan Lagrange gevşetme yöntemiyle **SubP2** için bulunan alt sınır değeri toplanarak, tüm problemin amaç fonksiyon değeri için bir alt sınır elde edilir.

İlk problemin çözümü sonucunda bir depo yerleşim planı elde edilir. Dıştaki subgradient yönteminin yinelemeleri sırasında, ilk alt problemin sonucu olarak her farklı depo yerleşim planı elde edildiğinde, bir tabu araması algoritması çalışmaya başlar. Tabu araması algoritması ilk alt problem sonucunu temel alarak bir Çok Depolu Araç Rotalama Problemi çözer. Böylece ana problem için olurlu bir sonuç elde edilmiş

olur. Bulunan en iyi olurlu çözümün toplam maliyeti ana problemin en iyi sonucu için bir üst sınır teşkil etmektedir.

Önerilen çözüm metodunun performansı bulunan en iyi üst sınır ile en iyi alt sınır arasındaki aralık temel alınarak değerlendirilmektedir. [LR-TS] bir kısmı rasgele üretilmiş ve bir kısmı da literatürdeki kıyas problemlerinden alınmış test problemleri üzerinde denenmiş ve sonuçlar sunulmuştur.

ACKNOWLEDGEMENTS

First and foremost I would like to thank my thesis advisor Deniz Aksen for his guidance and support. He has shared his experience with me not only on my thesis topic but also on all aspects of research and my personal development. This thesis would not be possible without his assistance and his tolerance especially during the last period of my study.

I want to thank members of my thesis committee Necati Aras, Sibel Salman and my professor Ceyda Oğuz for their valuable comments and suggestions. I am very grateful to Yaman Arkun and Süleyman Özekici for their contributions on the enhancement of my study. I also want to thank Arzu Aras for her support on the completion of my thesis as well as my colleague, Hakan Aydın for his kind welcome and for the fun he adds to every occasion.

My friends Nesrin, Eda, Canan, Suat, Hazal, Eren and Cem, made me enjoy my life in Koç University and provided me with the encouragement when I needed. I am very lucky to meet such people who hold success, kindness and fun together and to be part of this friendship. I also want to thank my roommates Selen, Besray, Deniz and Zeynep for the times that we have together full of good memories.

I am very grateful to my family for being with me every time and for their love. Without them, any success I achieved would not be possible. My special appreciation goes to my brother Seyfettin whom I have founded beside me whenever I got into difficulties.

Last but not least, I want to thank my best friend, my love, my fiancé Ahmet, for his valuable support, everlasting encouragement and permanent belief in my success. The most precious present I have ever received is "you" making me feel the happiness deep inside every moment and making my life wonderful in every way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

PC	Personal computer
CPU	Central processing unit
LRP	Location routing problem
MDLRP	Multi-depot location routing problem
LP	Linear programming
ILP	Integer linear programming
TSP	Traveling salesman problem
FLAP	Facility location allocation problem
VRP	Vehicle routing problem
VRP-TW	Vehicle routing problem with time windows
MDVRP	Multi-depot vehicle routing problem
LR	Lagrangian relaxation
MST	Minimal spanning tree
MSF	Minimal spanning forest
CMST	Capacitated minimal spanning tree
CMSF	Capacitated minimal spanning forest
DCMST	Degree constrained capacitated minimal spanning tree
TS	Tabu search
VNS	Variable neighborhood search
MTZ	Miller, Tucker, Zemlin
[LR-TS]	The name of the complete algorithm designed to solve P
[MSF-ALR]	The algorithm that solves the problem obtained after Lagrangian relaxation of **SubP2**
[PFIH-NN]	Push Forward Insertion-Nearest Neighborhood heuristic

[CW]	Modified Clarke-Wright heuristic for MDVRP
LPO	Local Post Optimization
P	The comprehensive multi-depot location routing problem
SubP1	FLAP-like subproblem of P
SubP2	MSF-like subproblem of P
IC	Set of customers
ID	Set of depots
ID_{pres}	Set of present depots
ID_{cand}	Set of candidate depots
I	Set of all nodes ($IC \cup ID$)
x_{ijk}	Binary decision variable taking the value 1 if node j is visited after node i on a route originating from depot k, 0 otherwise.
y_k	Binary decision variable taking the value 1 if depot k is in service, 0 otherwise.
δ_{ik}	Binary decision variable taking the value 1 if customer i is assigned to depot k, 0 otherwise.
FC_k	The opening or closing cost of depot k
OC_k	The operating cost of depot k
VC_k	Vehicle acquisition cost for depot k
c_{ij}	Traveling cost of one vehicle from node i to node j
Q	Uniform vehicle capacity
$L(S)$	The optimal solution to the one-dimensional bin packing problem where the bin length is equal to the vehicle capacity Q, and demand values d_i ($i \in IC$) are the sizes of the items to be packed into the bins
Z_P^*	The optimal objective function value of the problem P
Z_{lb}	Lower bound on the true optimal objective value of the problem P
Z_{ub}	Upper bound on the true optimal objective value of problem P

Z_{ub}^{q}	The upper bound found for the optimal objective value of P in iteration q of the subgradient optimization
Z_{LR}^{q}	The Lagrangian objective value found in iteration q of the subgradient optimization in the main Lagrangian relaxation
$\boldsymbol{\lambda}$	Lagrange multiplier set in the main Lagrangian relaxation
$\boldsymbol{\mu}$	Lagrange multiplier set in the Lagrangian relaxed problem
$\boldsymbol{\alpha}$	Lagrange multiplier set in the Lagrangian relaxed **SubP2**
LR($\boldsymbol{\lambda}$, $\boldsymbol{\mu}$)	The Lagrangian relaxed problem of P
$Z_{LR}(\boldsymbol{\lambda}, \boldsymbol{\mu})$	The objective function of Lagrangian relaxed LR(λ, μ)
ALR^{SubP2}	The Lagrangian relaxed problem of **SubP2**
Z_{SubP2}^{*}	The true optimal objective function value of **SubP2**
$Z_{ub(SubP2)}^{q}$	The upper bound found for the optimal objective value of **SubP2** in iteration q of the subgradient optimization
$Z_{ALR(SubP2)}^{q}$	The Lagrangian objective value found in iteration q of the subgradient optimization in the inner Lagrangian relaxation
C_{new}	The cost matrix of the objective function of ALR^{SubP2}
$\boldsymbol{SG^{q}}$	Subgradient vector of the problem LR(λ, μ) at iteration q of the subgradient optimization procedure
s^{q}	Step size at iteration q of the subgradient optimization procedure of the problem LR(λ, μ)
$\mathbf{Y}$	Subgradient vector of the problem ALR^{SubP2}
s_{ALR}^{q}	Step size at iteration q of the subgradient optimization procedure of the problem ALR^{SubP2}
p_c	Penalty coefficient used in the evaluation function of tabu search
$V_c(r)$	Penalty term used in the evaluation function of tabu search
R1	Randomly generated problem set 1
R2	Randomly generated problem set 2

TB	LRP instances solved by Tüzün and Burke [23]
$dist_{PD}$	Probability distribution of the present depots locations on the problem space
$dist_{CD}$	Probability distribution of the candidate depots locations on the problem space
$dist_{C}$	Probability distribution of the customers locations on the problem space
U	Uniform distribution
RU	Rectalinear uniform distribution on a specified number of equidistant longitudes and altitudes
C	Clustered around the depot locations
k	Coefficient used for calculating the vehicle capacities during random problem generation
N_C	Number of customers
N_{PD}	Number of present depots
N_{CD}	Number of candidate depots
N_D	Total number of depots

Chapter 1

INTRODUCTION

The design of a distribution logistics system requires quite a number of decisions of different planning levels. The most important strategic decision to be taken is the locations of distribution centers, which are also referred to as depots. The allocation of customers to the depots is a decision of tactical level, while determining vehicle routes to visit those customers belongs to the operational level. Since logistics management plays an important role in customer satisfaction and efficiency of companies, practitioners take considerable notice of location and routing problems. Besides their importance in the logistics business, both location and routing problems have been studied widely by researchers. They are very attractive from research point of view due to the fact that they give rise to a wide variety of challenging mathematical models, and there exist several heuristics and metaheuristics applicable on them.

The construction of a distribution logistics system starts with the depot locations. The potential locations of depots are found based on a series of requirements such as physical and economical accessibility, and particular conditions inherent to the specific application (governmental restrictions, international regulations, company policies etc.). Once the set of candidate locations is determined, the managers select a subset of them which is the best with respect to the objectives. In earlier times, the objective was inadequately settled by only considering the demand of the customers and the distance between the candidate depots and the customers. However, recent research showed that the vehicle acquisition and delivery costs have an impact on the optimum depot location plan. It was observed that the allocation of the customers to depots and the selection of the routes to visit customers, which are decisions of tactical and operational level

respectively, should be considered simultaneously with the selection of the depots to be established among candidate locations.

The need for integrating these three steps of the distribution system design process has been fulfilled by the Location-Routing Problem (LRP). LRP involves finding the optimal number and locations of depots while allocating customers to depots and determining vehicle routes to visit all customers. LRP focuses on three main decisions of different levels simultaneously. The interdependence between these decisions has been noticed by researchers long ago. The effect of ignoring routes when locating depots has also been stressed by Salhi and Rand [1]. However, location and routing parts of the problem have traditionally been solved separately which is justified by the inevitable difficulty of the combined problem due to its components. Extremely high computing power of mainstream processors available today and the success already achieved in solving location as well as routing problems provide us the motivation to deal with the combined location-routing problem without splitting it into its subproblems. In recent years, the literature on the LRP has been increasingly addressing some variants of the problem, which generally correspond to the real world needs.

In this thesis, we solve a two-tier or two-layer (single echelon) multi-depot location-routing problem (MDLRP) where transportation is made directly from depots to customers. There exist two kinds of depots: present and candidate depots. Present depots are already operating facilities that can be preserved or closed. If a present depot is closed, a fixed cost is incurred. This cost may turn out to be a gain since the closure of a depot usually brings about savings in overhead costs. Candidate depot locations are potential sites in which new depots can be opened. For each new depot to be opened, a location dependent fixed cost is incurred. In addition, there exists fixed operating cost which is charged for each preserved or newly opened depot. Customers are visited by a homogeneous fleet of capacitated vehicles. For each of them, a vehicle acquisition cost is charged. Each customer has a deterministic and static demand which should be satisfied by the single visit of a vehicle. There is no capacity constraint on depots. The sum of depot opening-closing, depot operating, vehicle acquisition, and vehicle

traveling costs is minimized subject to the standard vehicle capacity specified in the problem.

To solve this problem, a nested Lagrangian relaxation-based method which is named as [LR-TS] is proposed. An outer Lagrangian relaxation embedded in subgradient optimization decomposes the parent problem into two subproblems. The first subproblem which is a facility location-like problem is solved to optimality with Cplex 10.0. The second subproblem resembles a capacitated and degree constrained minimum spanning forest problem. It is tackled with an augmented Lagrangian relaxation. The sum of objective value of the Cplex solution of **SubP1** and the lower bound found for **SubP2** by the subgradient optimization scheme in the augmented Lagrangian relaxation constitutes the lower bound to the true optimal solution of the comprehensive problem.

The solution of the first subproblem reveals a depot location plan. As soon as a new distinct location plan is found in the course of the subgradient iterations, a tabu search algorithm is triggered to solve the multi-depot vehicle routing problem associated with that plan, and a feasible solution to the parent problem is obtained. Its objective value is checked against the current upper bound on the parent problem's true optimal objective value. The performance of the proposed method has been tested on a number of test problems, and the results have been tabulated.

The remainder of the thesis is organized as follows. In Chapter 2, an overview of the previous studies on the LRP is given. Chapter 3 comprises the problem definition and mathematical formulation. The detailed explanations of the Lagrangian relaxation scheme and the solutions for the subproblems are provided in Chapter 4. In Chapter 5, the tabu search heuristic that is used to obtain upper bounds on the true optimal solution is explained. The computational experiments and results are given in Chapter 7. Finally, Chapter 8 presents a summary with concluding remarks.

Chapter 2

LITERATURE REVIEW

The LRP has been studied since 1970s although researchers have been using different names for the problems they are dealing with. Formerly, the LRP used to be solved by decomposing it into location and routing parts. In recent years, with the ever increasing performance and affordability of PC platforms, algorithms have been proposed to solve the combined problem in reasonable computing time. Most of the solution methods are heuristics while several exact solution approaches have been offered for problems with limited number of customers.

The newest literature survey of the state-of-the-art in location routing is accomplished by Nagy and Salhi [2]. They provide the review of previous studies on the LRP in groups based on the solution techniques and the particular problem investigated. First, they suggest four criteria to classify the studies which they regard as the key aspects of the problems. These are *hierarchical structure*, *type of input data*, *planning period* and *solution method*. Hierarchical structure is the configuration of the distribution system which states the serving and receiving nodes. In most of the problems, there exist facilities servicing a number of customers. Delivery or pick-up between facilities is not common. The authors classify the problems first in terms of the structure of the flow of the goods. The second criterion is the type of input data which can be *deterministic* or *stochastic*. Then, the papers are grouped based on whether the problem is *single period* or *multi period*. The last of the key classification aspects is the proposed solution method, which can be either *exact* or *heuristic*. Although they do not follow a complete taxonomy, they give five further attributes of the problems together with this classification. They clarify the characteristics of an LRP instance using these

five criteria: structure of the objective function, solution space (discrete, continuous, network), number of depots, number and types of vehicles and route structure (arc routing, multiple trips, pick-up and delivery etc.). Finally, they give their suggestions for future research on the LRP after summarizing the suggestions of Balakrishnan et al.[3], Laporte [4] and Min et al. [5].

Another annotated literature review of the LRP and its extensions is due to Ahipaşaoğlu et al. [6]. They provide an overview of the computational aspects and the research prospects. The study has the attribute to be the first survey in the literature that includes a complete analysis of all problem variants. They review the studies in three groups as Deterministic LRP, Stochastic LRP and LRP with obnoxious facilities. LRP with obnoxious facilities refers to the group of problems dealing with locations of anticentral facilities the examples of which are hazardous material storage facilities and waste disposal sites.

A perfect synthesis and survey of the LRP is earlier carried out by Min et al. [5]. They review and compare the literature with respect to algorithmic developments and also include a hierarchical taxonomy. Min et al. define the location-routing model as solving the joint problem of determining the optimal number, capacity and location of facilities serving more than one customer/supplier (demand node), and finding the optimal set of vehicle schedules and routes. Its main difference from the classical location/allocation problem is that, once the facility is located, LRP requires a visitation of demand nodes through tours, whereas the latter assumes a straight-line or radial trip from the facility to the demand nodes. The authors recommend solving the whole LRP concurrently in order to be able to analyze tradeoffs between location and routing decisions at the same level of hierarchy.

The survey by Laporte [7] can be accounted as one of the milestones in LRP literature. This survey is the first comprehensive study of the earlier work on the deterministic LRP. Because it is written in a tutorial style, this survey is particularly important for researchers that are new to the area. It also includes a classification scheme of LRPs based on the number of layers and the type of routes between these

layers. For example, 3/R/T refers to a three layer LRP where routes between the first and second layer is of type Replenishment (direct shipment) while the delivery between the second and third layer is realized via tours (vehicle routes).

Among exact solution methods proposed for the LRP is the method of Laporte and Nobert [8]. They formulate the uncapacitated single facility LRP as an integer linear program (ILP), and solve it using constraint relaxation and a branch-and-bound method. They initially relax all subtour elimination constraints and add them one at a time as needed. Laporte et al. [9] studied the LRP with multiple uncapacitated facilities and uncapacitated vehicles. They locate several facilities among n sites and find the optimal routes to serve the remaining ones. They formulate the problem as an ILP, and solve it using a constraint relaxation procedure similar to Laporte and Nobert [8]. Laporte et al. [10] use a graph transformation approach to model the LRP as an ILP and tackle it with a branch and bound method. They solve instances with 80 nodes and 1 to 3 potential sites to optimality which is, to our knowledge, the largest LRP solved optimally.

Laporte and Dejax [11] studied the dynamic version of the case of multiple facilities with multiple vehicles. Their model represents the system as a network consisting of three-layers where the layers correspond to production sites, distribution centers, and customers. In their paper two solution approaches have been addressed. In the first one, which is proposed to solve small size problems, the LRP is formulated as an ILP using a network representation and solved to optimality. In the second approach, the solution is obtained by an approximation algorithm based on approximating TSP tours with spanning trees. This work of Laporte and Dejax [11] is outstanding in that it is the first analytical study concerning the dynamic LRP.

In his thesis Gezdur [12] formulates the LRP as a set partitioning problem and solves it using a column generation algorithm. The bounds are tightened by 2-path cuts, and a separation algorithm is used to eliminate the subtours. The performance of the proposed method is evaluated by comparing with the solutions of Perl and Daskin [13], Wu et al. [14], and Hansen et al. [15]. It is found to yield better solutions than the others.

Another study on the exact solution of the LRP is due to Boğ [16]. He also tackles the problem with a column generation method. He uses the set-partition based formulation, and solves an elementary shortest path problem for the pricing subproblem. The method is tested on the problems that are compiled by Barreto [17] and tight gaps are obtained for most of them.

Jacobsen and Madsen [18] solve a newspaper distribution system problem with approximately 4500 retailers. The delivery is first made from the printing office to a number of transfer points, and then from the transfer points to the retailers where the number and locations of transfer points are among the decisions to be made. They present three heuristics methods. The first heuristic is a tour construction method with implicit transfer point locations. The second is an alternative location-allocation procedure for locating transfer points followed by savings procedures for routing the trips from the printing office to the transfer points, and subsequently from transfer points to the retailers. In the third heuristic an initial savings procedure is used for the routing of the tours from the transfer points to the retailers. This is followed by a *Drop* procedure determining the locations of the transfer points, and a savings procedure constructing the tours from the printing office to the transfer points. Madsen [19] provides an almost complete and comprehensive survey of previous applications, and comparison of methods for solving combined location-routing problems.

Perl and Daskin [13] analyze the case where both facilities and vehicles are capacitated. They give an ILP formulation, but solve the problem by a heuristic method where the LRP is decomposed into three stages, and the location and routing phases are solved iteratively. They obtain the initial feasible solution by a route-first, location-allocation second heuristic method. The method is applied to a large scale problem of a distribution system of an international manufacturer. Hansen et al. [15] modify the LRP formulation of Perl and Daskin [13] and suggest a more effective heuristic which yields better solutions.

Srivastava [20] designs three heuristic for LRP in his dissertation work which form the basis of an extensive computational study published in Srivastava and Benton [21].

In the computational study of Srivastava and Benton, the authors compare the three heuristics of Srivastava [20], which has been also published in Srivastava [22], with the sequential locate-first, route-second method. The authors use a set of 150 problems, and test the performance of the heuristics under varying conditions defined by the cost structure, spatial distribution of customers, and number of depot sites. They conclude that the three heuristics yielded smaller (better) objective function values than the sequential method, except for the low cost case. They have also commented on which heuristic is more appropriate to use under what settings of the experimental parameters.

Tüzün and Burke [23] propose a two-phase tabu search architecture for the solution of the multi-depot LRP where depots have unlimited throughput capacity. The first stage of the tabu search is the routing phase where routes are improved by swap and insert-delete moves. In the second stage, the location-allocation configuration is improved by moving a facility from one location to another, and by simultaneous adding and dropping of facilities. They compare their results with the results of SAV1 heuristic addressed in Srivastava [22], and conclude that their heuristic performs better than SAV1 on the average. The computation times are also relatively short. Wu et al. [14] decompose the standard LRP with capacitated depots into a discrete facility location-allocation problem (FLAP) and a vehicle routing problem (VRP), and solve each subproblem using simulated annealing in a sequential and iterative manner. They obtain better solutions on the same instances solved by Perl and Daskin [13] and Hansen et al. [15]. For the same class of the LRP, Albareda-Sambola et al. [24] apply a method which generates first a lower bound either from the linear relaxation of the given problem or from the solutions of a pair of ad hoc knapsack and asymmetric traveling salesman problems. This lower bound is then used as a starting point of a tabu search heuristic. They test their method on a set of randomly generated problems. Melechovský et al. [25] address an LRP with non-linear depot opening costs that grow with the total demand satisfied by the depots. They present a hybrid metaheuristic consisting of tabu search (TS) and variable neighborhood search (VNS) heuristics.

We are aware of one study by Aksen and Altinkemer [26] on Lagrangian relaxation for the LRP. They propose a 3-layer distribution logistics model for the conversion from brick-and-mortar to click-and-mortar retailing. They consider a traditional brick-and-mortar retailing model in which such a retailer operates two types of facilities, and serves only one type of customers. Its facilities comprise warehouses (WH's) and physical stores, where goods are transferred from the former to the latter in direct shipments. Goods are then sold to walk-in type customers who are assumed to go to the nearest physical store for shopping. When the retailer opens a website for the online shopping convenience, it needs the capability of receiving, processing and then delivering orders placed by online customers at that website. Some of the present brick-and-mortar stores (BM's) of the retailer might have to be reconfigured, or several new stores might be opened with that capability. A physical store is designated as a click-and-mortar store (CM) if it is Internet-enabled, equipped with the necessary hardware, software and personnel such that it can effectively handle online orders. A physical store can serve online customers only then if it is opened as a CM, or if it is a BM reconfigured as (converted to) a CM. They determine the number and locations of CM and BM type facilities, and the vehicle routes to visit online customers. A static one-period optimization model is built and solved using Lagrangian relaxation. They assess the performance of the Lagrangian-based solution method on a number of randomly generated test problems.

The problem in our study is a 2-layer multi-depot location routing problem with uncapacitated depots. In our study, we deal with a 2-layer multi-depot LRP with unlimited throughput capacities. Tüzün and Burke [23] explain that this problem belongs to the class of $\mathcal{NP}$-hard problems. Once the locations of the facilities are determined, this is, once the uncapacitated FLAP is solved, the LRP reduces to a multiple depot VRP. Tüzün and Burke cite the work of Cornuejols et al. [27] who have shown that the FLAP belongs to the class of $\mathcal{NP}$-hard optimization problems. Similarly, they refer to Karp [28] and Lenstra and Rinnooy Kan [29] who have proven that even the single depot VRP is $\mathcal{NP}$-hard. Supported by these facts, Tüzün and Burke conclude

that the LRP is also an $\mathcal{NP}$-hard problem. We propose a nested Lagrangian relaxation scheme to provide lower bounds to the problem while an upper bound is obtained for the problem using the tabu search heuristic. The tabu search heuristic takes the location plan that comes from the Lagrangian relaxation as an input and solves a multi-depot vehicle routing problem to constitute a feasible solution the complete problem. The gap between the best upper bound and the best lower bound is used to assess the quality of the solution scheme.

Chapter 3

PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

The problem that we described in Chapter 2 takes its motivation from the real life problems faced by local logistics firms. They operate a number of depots some of which are their own property while the others have been rented by them. At the beginning of certain periods, or when they sign new contracts with the customers on their portfolio, they revise their operating depots. According to the locations and demands of their customers they may decide to hire out their own depots, to leave some of the depots they have rented before, or to rent new ones. In order to use a new depot, they have to cover a certain opening cost. This opening cost comprises the sum of overhead charges related to renting the depot, hiring new workers, moving or buying new equipment. When they quit using a depot which was in service, they have to compensate the overhead costs of laying off or transferring workers, and moving out of the depot. The sum of these cost items constitutes the closing cost of a depot. They also consider such depot operating costs as wages and social packages offered to the workers and maintenance costs. These may vary from depot to depot depending on the quality of the workforce or on the regulations applying to the region of the depot. They adopt the location plan which minimizes the sum of total opening, closing and operating costs of depots plus the traveling cost between depots and customers. This case of the logistics firms turns out to be a location routing problem since the location and routing decisions are made simultaneously.

In our study, we model and solve the problem where the depots to be in service are selected among a number of operating and candidate depots, and the routes to visit the customer locations are determined with the assumption that there is no limit on the

depot capacities. The problem can be represented as 2/T according to Laporte's [7] classification of LRPs. It means there are two layers; namely, depots and customers where the transportation between these layers is realized via tours (vehicle routes) rather than direct shipments.

In the problem we are dealing with, there exist two kinds of depots: present and candidate. Present depots are the ones that are already operating. We (the logistics firm) have the option to close these depots at a fixed cost which amounts to the overhead costs of closure minus the salvage value of the depot if there exists one. Candidate depot locations are potential sites where new depots can be opened. When we decide to start operations in one of these potential depot locations, we have to pay a fixed opening cost. In addition, for each depot in service a fixed operating cost is incurred as well. There is no capacity constraint on the throughput of depots. We assume that there is an unlimited number of vehicles with the same capacity (homogeneous vehicle fleet) where for each one deployed a fixed acquisition cost is charged. Each customer has a deterministic demand which is to be satisfied by a single visit of the assigned vehicle. This means split deliveries are not permitted. We assume that the network of candidate and present depot sites and customer locations forms a complete graph where each node is directly accessible from another. In case there exist more than one depot in service, any customer can be served on a route originating from any of the depots. However, each route should start and end at the same depot.

In the problem, we determine

i. the number and locations of depots,
ii. which customer will be served by which depot (the assignment of customers to depots),
iii. on which routes the customers will be served

minimizing the sum of

i. opening, closing, and operating costs of the depots,
ii. total vehicle acquisition cost
iii. total traveling cost.

For the multi-depot location routing problem that we have described, we give below a mathematical model preceded by its notation.

Notation:

Sets:

IC : set of customers

ID : set of depots

ID_{pres} : set of present depots

ID_{cand} : set of candidate depots

I : set of all nodes ($IC \cup ID$)

Binary Decision Variables:

x_{ijk} : 1 if node j is visited after node i on a route originating from depot k, 0 otherwise.

y_k : 1 if depot k is in service, 0 otherwise.

δ_{ik} : 1 if customer i is assigned to depot k, 0 otherwise.

Parameters:

FC_k : the opening or closing cost of depot k

OC_k : the operating cost of depot k

VC_k : vehicle acquisition cost for depot k

c_{ij} : traveling cost of one vehicle from node i to node j

Q : uniform vehicle capacity

$L(S)$: the optimal solution to the one-dimensional bin packing problem where the bin length is equal to the vehicle capacity Q, and demand values d_i ($i \in IC$) are the sizes of the items to be packed into the bins.

P :

$$\text{Min} \sum_{k\in ID} OC_k y_k + \sum_{k\in ID_{cand}} FC_k y_k + \sum_{k\in ID_{pres}} FC_k (1-y_k) + \sum_{i\in ID}\sum_{j\in IC} VC_i x_{iji} + \sum_{k\in ID}\sum_{i\in I}\sum_{\substack{j\in I \\ i\neq j}} c_{ij} x_{ijk} \quad (3.1)$$

subject to:

$$\sum_{k\in ID} \delta_{ik} = 1 \qquad \forall i \in IC \quad (3.2)$$

$$\sum_{\substack{j\in IC\cup\{k\} \\ j\neq i}} x_{ijk} = \delta_{ik} \qquad \forall i \in IC,\ \forall k \in ID \quad (3.3)$$

$$\sum_{\substack{j\in IC\cup\{k\} \\ j\neq i}} x_{jik} = \delta_{ik} \qquad \forall i \in IC,\ \forall k \in ID \quad (3.4)$$

$$\sum_{i\in IC} x_{ikk} = \sum_{i\in IC} x_{kik} \qquad \forall i \in ID, \forall k \in ID \quad (3.5)$$

$$\sum_{k\in ID}\sum_{i\in IC} x_{kik} + \sum_{k\in ID}\sum_{i\in IC}\sum_{\substack{j\in IC \\ i\neq j}} x_{ijk} = |IC| \quad (3.6)$$

$$\sum_{k\in ID}\sum_{i\in S}\sum_{\substack{j\in S \\ j\neq i}} x_{ijk} \leq |S| - L(S) \qquad \forall S \subseteq IC,\ |S| \geq 2 \quad (3.7)$$

$$\sum_{i\in IC} \delta_{ik} \leq |IC| y_k \qquad \forall k \in ID \quad (3.8)$$

$$x_{ijk} \in \{0,1\} \qquad \forall i \in I, \forall j \in I, \forall k \in ID \quad (3.9)$$

$$\delta_{ik}, y_k \in \{0,1\} \qquad \forall i \in I, \forall k \in ID \quad (3.10)$$

The model structure can be expressed in plain English as follows:

Minimize Z_P = Σ FLAP Objectives + Σ MDVRP Objectives

Subject to: i) Pure FLAP constraints

ii) Pure MDVRP-TD constraints

iii) FLAP and MDVRP-TD coupling constraints.

The objective function of **P** shown in Equation (3.1) is a combination of objectives of a facility location-allocation problem (FLAP) and a multi-depot vehicle routing problem (MDVRP). The constraints are comprised of pure FLAP constraints, pure

MDVRP constraints and coupling constraints linking routing decisions with location decisions. The set of constraints in Equation (3.2) provides that each customer is assigned to a depot. Equations (3.3)−(3.4) are flow conservation constraints which guarantee all customers be visited exactly once on a route originating from the depot they are assigned to. Equation (3.5) ensures that the numbers of incoming and outgoing arcs at each depot are equal which means number of vehicles departing from a depot is equal to the number of vehicles returning to it. Equation (3.6) is identical to the sum of the constraints in Equation (3.2). In order to obtain the second subproblem as a minimum spanning forest like problem after the Lagrangian relaxation, we add this redundant constraint to the model. Equation (3.7) represents the well-known exponential number of subtour elimination constraints which provide that the first and last nodes in all routes are one of the depots. The assignment of a customer to a closed or unopened depot, and routes originating from such a depot are avoided by Equation (3.8). Finally, Equations (3.9)–(3.10) are integrality constraints.

Chapter 4

THE LAGRANGIAN RELAXATION FOR MDLRP

4.1. Overview of the Lagrangian Relaxation Method and the Subgradient Optimization

Lagrangian relaxation (LR) is a decomposition approach used for a variety of $\mathcal{NP}$-hard optimization problems. In this method, the true optimal objective value of the problem Z_P^* is bracketed between a lower and an upper bound [Z_{lb}, Z_{ub}] and $\left|\frac{Z_{ub} - Z_{lb}}{Z_{lb}}\right|$; i.e., the percentage gap between the final values of the bounds; is used to measure the quality of the solution. In a minimization problem, the objective function value of a good feasible solution found by a heuristic method constitutes the upper bound. The lower bound is obtained from the solution of the Lagrangian relaxed problem. The relaxed problem is formed by removing some of the constraints of the original problem, penalizing them with a set of Lagrange multipliers $\boldsymbol{\lambda}$, and adding them to the objective function. If the constraints set is exclusively comprised of linear functions, then the Lagrangian relaxation can be represented as the following transformation.

$$\begin{aligned} \mathbf{L}: \quad & \min f(\boldsymbol{x}) \\ & Ax \le b \\ & x \in X \end{aligned} \qquad\qquad \begin{aligned} \mathbf{L'}: \quad & \min f(\boldsymbol{x}, \lambda) \\ & A'x \le b' \\ & x \in X \end{aligned}$$

A' and b' are obtained from A and b by deleting the respective rows that are associated with the constraints to be removed for the relaxation.

Geoffrion [30] proved that for any given value of multipliers $\boldsymbol{\lambda}$, the optimal objective function value $f_{min}(\boldsymbol{x}, \boldsymbol{\lambda})$ of **L'**, namely Z_{lb}, is a lower bound for the true optimal solution Z_P^* of **L**. Since Z_P^* is guaranteed to be equal to or higher than Z_{lb}, the final gap between the bounds ($Z_{ub} - Z_{lb}$) cannot be lower than the gap ($Z_{ub} - Z_P^*$).

The selection of constraints that will be dualized with Lagrange multipliers $\boldsymbol{\lambda}$ and embedded in the original objective function $f(\boldsymbol{x})$ forms the resulting Lagrangian relaxed problem **L'**. In order to achieve the tightest lower bound Z_{lb} on Z_P^*, one needs to obtain the best Lagrange multipliers $\boldsymbol{\lambda}^*$ by solving the problem in (4.1) .

$$f_{min}(\mathrm{x}, \lambda^*) = \max_{\boldsymbol{\lambda}} f(\boldsymbol{x}, \boldsymbol{\lambda}) \tag{4.1}$$

The success of any Lagrangian method not only depends on how easy it is to solve the relaxed problem **L'**, but also on the method of obtaining the best multipliers $\boldsymbol{\lambda}^*$. One method of Z_{lb} improvement is the iterative subgradient optimization. The subgradient vector of **L'** is given by the difference between left- and right-hand sides of the relaxed constraints in **L**. This difference is to be calculated using the values of decision variables in the current iteration q. The subgradient vector is used as the step direction together with the multipliers iterate $\boldsymbol{\lambda}_q$ and as a step size to calculate the new iterate $\boldsymbol{\lambda}_{q+1}$ of the next iteration $q+1$.

In vehicle routing and capacitated spanning tree problems solved by LR, a number of other methods have been suggested that compute better values for the Lagrange multipliers λ so as to find tighter lower bounds on the original problem. In their paper on the VRP with time windows, Kohl and Madsen [31] describe a bundle algorithm in which a convex combination of previously obtained subgradients is used instead of a single subgradient only. Gavish [32] suggests a dual ascent procedure followed by a subgradient optimization procedure in his paper on centralized network design modeled as a Capacitated Minimal Spanning Tree (CMST) Problem.

The Lagrangian relaxation approach combined with the subgradient optimization techniques have been shown to be useful in obtaining near-optimal dual solutions that can provide lower bounds which are as good as (in some instances tighter than) the lower bounds obtained by the standard linear programming relaxation.

When the Lagrangian relaxation is applied to the MDLRP model, the coupling constraints in Equations (3.3)–(3.4) which combine the FLAP and MDVRP parts of the problem are relaxed. The left-hand sides of the constraints are subtracted from their right-hand sides. The differences are multiplied by the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ respectively, which are unrestricted in sign. The terms are then augmented into the original objective function, and the new objective function $Z_{LR}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ in Equation (4.2) is obtained.

$$Z_{LR}(\lambda,\mu) = \sum_{k\in ID} OC_k y_k + \sum_{k\in ID_{cand}} FC_k y_k + \sum_{k\in ID_{pres}} FC_k (1-y_k) + \sum_{i\in ID}\sum_{j\in IC} VC_i x_{iji} + \sum_{k\in ID}\sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}} c_{ij} x_{ijk} + \sum_{k\in ID}\sum_{i\in IC}\sum_{\substack{j\in IC\cup\{k\}\\ j\neq i}} (\delta_{ik} - x_{ijk})\lambda_{ik} + \sum_{k\in ID}\sum_{j\in IC}\sum_{\substack{i\in IC\cup\{k\}\\ i\neq j}} (\delta_{jk} - x_{ijk})\mu_{jk} \quad (4.2)$$

When we rearrange the last two terms of the objective function as in Equation (4.3), there remains no terms which connect location and routing variables. Consequently, $Z_{LR}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ becomes separable in FLAP and MDVRP parts.

$$Z_{LR}(\lambda,\mu) = \sum_{k\in ID} OC_k y_k + \sum_{k\in ID_{cand}} FC_k y_k + \sum_{k\in ID_{pres}} FC_k (1-y_k) + \sum_{i\in ID}\sum_{j\in IC} VC_i x_{iji} + \sum_{k\in ID}\sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}} c_{ij} x_{ijk} + \sum_{k\in ID}\sum_{i\in IC}\sum_{\substack{j\in IC\cup\{k\}\\ j\neq i}} \delta_{ik}(\mu_{jk} + \lambda_{ik}) - \sum_{k\in ID}\sum_{j\in IC}\sum_{\substack{i\in IC\cup\{k\}\\ i\neq j}} x_{ijk}(\mu_{jk} + \lambda_{ik}) \quad (4.3)$$

After the rearrangement, the structure of the Lagrangian relaxed problem $LR(\boldsymbol{\lambda}, \boldsymbol{\mu})$ can be written in plain English as shown below.

$LR(\lambda, \mu)$: Minimize $Z_{LR}(\lambda, \mu) = \sum$ *Augmented* FLAP *objectives* (Z_{SubP1})

$+ \sum$ *Augmented* MDVRP *objectives* (Z_{SubP2})

subject to:

i. Pure FLAP constraints (3.2), (3.8) (*constraints of SubP*1)
ii. Pure MDVRP constraints (3.5)–(3.7) (*constraints of SubP*2)
iii. Nonnegativity and integrality constraints (3.9)–(3.10)

The Lagrangian relaxed problem $LR(\boldsymbol{\lambda}, \boldsymbol{\mu})$ can be partitioned into two independent subproblems. The first subproblem resembles an uncapacitated FLAP (**SubP1**). The second one is similar to a degree constrained minimum spanning forest problem (DCMSF) (**SubP2**). We solve **SubP1** with Cplex 10.0 in reasonable time. However, **SubP2** is still an $\mathcal{NP}$-hard problem, which is tackled with an augmented Lagrangian relaxation by relaxing the degree constraints. The relaxed **SubP2** becomes a minimum spanning forest problem with a minimum number of outgoing arcs at root nodes (depots). It is solved with a modified version of Prim's minimum spanning tree algorithm. Figure 4.1 displays the flow chart of the iterative subgradient optimization procedure with the Lagrangian relaxation scheme applied to the parent problem **P**. The flow chart's segment in the box indicates the inner augmented Lagrangian relaxation which is applied to the second subproblem **SubP2**. The upper bound for problem **P** is obtained using a tabu search heuristic embedded in the iterations of the Lagrangian relaxation. Therefore, the complete solution method is named [LR-TS].

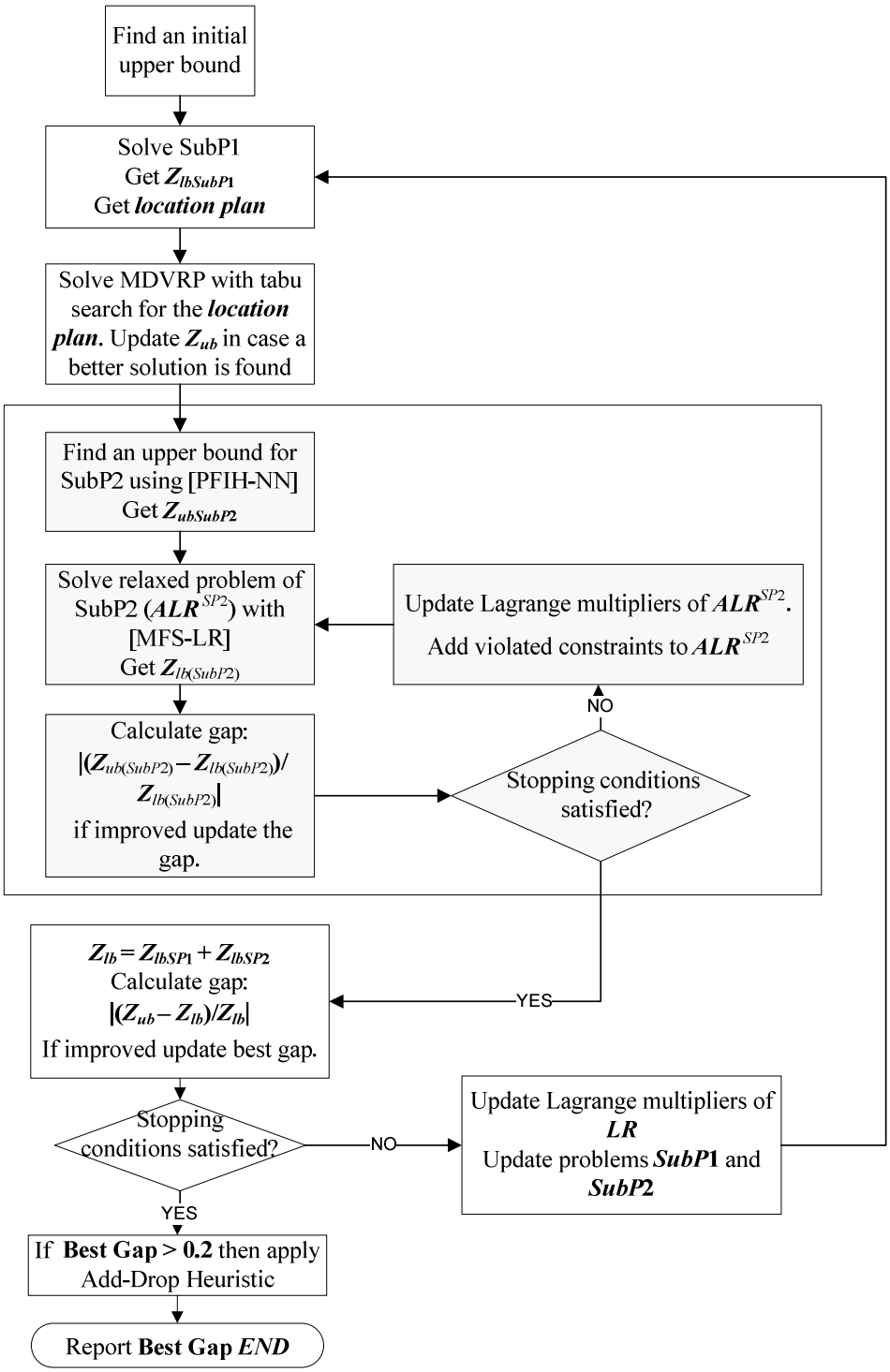

Figure 4.1 Flow chart of the Lagrangian relaxation Scheme for MDLRP

4.2. The Lagrangian Relaxed Problem LR

The objective function of the Lagrangian relaxed problem; i.e., $Z_{LR}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is separable into two components as FLAP and MDVRP objectives. In order to obtain two independent components, $Z_{LR}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ needs to be rearranged by dividing the relaxed constraints that are augmented into the objective function into two parts, as shown in Equation (4.3). Then the part which is stated in Equation (4.4) is added to the FLAP objectives of $Z_{LR}(\boldsymbol{\lambda}, \boldsymbol{\mu})$, while the remainder is included in the MDVRP objectives. The Lagrange multipliers that are added to the FLAP objective component represent pseudo costs of allocating customers to depots. The part which is appended to the MDVRP component, Equation (4.5), augments the traveling costs between nodes.

$$\sum_{i \in IC} \sum_{k \in ID} \delta_{ik} (\mu_{ik} + \lambda_{ik}) \tag{4.4}$$

$$-\sum_{i \in IC} \sum_{k \in ID} x_{ijk} (\mu_{ik} + \lambda_{ik}) \tag{4.5}$$

After reorganizing the terms of the objective function as in Equation (4.6), we derive the 3-dimensional asymmetric and depot dependent traveling cost matrix $C_{new}=[(c_{ijk})^{new}]$. In this cost matrix, the vehicle traveling cost from node i to node j not only depends on the distance between i and j, but also on the depot k which sends off that vehicle.

$$\begin{aligned} Z_{LR}(\lambda, \mu) = & \sum_{k \in ID} OC_k y_k + \sum_{k \in ID_{cand}} FC_k y_k + \sum_{k \in ID_{pres}} FC_k (1 - y_k) + \sum_{i \in ID} \sum_{j \in IC} (c_{ij} + VC_i - \mu_{jk}) x_{iji} \\ & + \sum_{i \in IC} \sum_{j \in ID} (c_{ij} - \lambda_{jk}) x_{ijj} + \sum_{k \in ID} \sum_{i \in IC} \sum_{\substack{j \in IC \\ j \neq i}} (c_{ij} - \mu_{jk} - \lambda_{ik}) x_{ijk} + \sum_{k \in ID} \sum_{i \in IC} \sum_{\substack{j \in IC \cup \{k\} \\ j \neq i}} \delta_{ik} (\mu_{jk} + \lambda_{ik}) \end{aligned} \tag{4.6}$$

Let $G(I,A)$ denote the complete weighted and directed graph of customers and depots, i.e. $A = \{(i,j) \in (I \times I), i \neq j\}$. Let $(c_{ijk})^{new}$ denote the cost of arc (i, j), if it is

traversed with a vehicle dispatched from depot $k \in ID$. Arc costs in G are then defined as follows:

i. $(i,j) \in IC \text{ x } IC, i \neq j, k \in ID$: $(c_{ijk})^{new} = c_{ij} - \lambda_{ik} - \mu_{jk}$

ii. $(i,j) \in IC \text{ x } ID$: $(c_{ijj})^{new} = c_{ij} - \lambda_{ij}$

iii. $(i,j) \in ID \text{ x } IC$: $(c_{iji})^{new} = c_{ij} - \mu_{ji} + VC_i$

iv. $(i,j) \in ID \text{ x } IC, k \in ID, i \neq k$: $(c_{ijk})^{new} = +\infty$

v. $(i,j) \in IC \text{ x } ID, k \in ID, j \neq k$: $(c_{ijk})^{new} = +\infty$

The Lagrange multipliers which are multiplied by the binary decision variable x_{ijk} are embedded into the cost matrix as arc costs. The vehicle acquisition cost is added to the cost of the arcs which emanate from depots and go to customers. The rationale behind doing so is that each time a depot node is directly connected to a customer node a new vehicle has to be acquired at the unit acquisition cost. The last two cost assignments avoid illegal arc definitions. As we mentioned in Chapter 3, the binary decision variable x_{ijk} takes the value 1 when node j is visited immediately after node i on a route originating from depot k. If a vehicle visits node j after depot i, the route cannot be originating from a depot $k \neq i, k \in ID$. Also, if a depot node j is visited after node i, the route cannot be originating from depot $k \neq j, k \in ID$. Therefore, c_{ijk} is assigned to infinity in these two cases in order to prevent x_{ijk} from taking the value 1. After the construction of the new cost matrix C_{new}, the Lagrangian relaxed problem LR(λ, μ) can be stated as follows.

$$\text{Min} \quad Z_{LR}(\lambda,\mu) = \sum_{k\in ID} OC_k y_k + \sum_{k\in ID_{cand}} FC_k y_k + \sum_{k\in ID_{pres}} FC_k (1-y_k) + \sum_{k\in ID}\sum_{i\in IC} \delta_{ik}(\lambda_{ik}+\mu_{ik}) + \sum_{k\in ID}\sum_{i\in I}\sum_{\substack{j\in I \\ j\neq i}} (c_{ijk})^{new} x_{ijk} \tag{4.7}$$

subject to: (3.2), (3.5)–(3.10)

The Lagrangian relaxed problem is solved with a subgradient optimization procedure in order to obtain a lower bound (Z_{lb}) for Z_P^*, the true optimal solution of the main problem **P**. We use a tabu search heuristic to find a good feasible solution whose objective value will be an upper bound (Z_{ub}) on Z_P^*. This upper bound is updated throughout the subgradient iterations of the Lagrangian relaxation. The details of the upper bound generation and updating method are explained in Chapter 5.

4.3. Subgradient Optimization in the Lagrangian Relaxed Problem LR

Iterative subgradient optimization is one of the common methods of obtaining the best Lagrange multipliers that are hoped to produce the tightest bound on the original problem's true optimal objective value. The aim is to update the Lagrange multipliers of the current iteration, and use them in the next iteration of the subgradient optimization. Kohl and Madsen [31] remark on the easy implementation advantage of subgradient optimization. They state that it usually gives rapid improvement in the first iterations. Though, the convergence slows down later, and for some difficult or large problems convergence in a reasonable computing time is not possible.

Let $\boldsymbol{SG^q}$ denote the subgradient vector of the problem LR($\boldsymbol{\lambda}$, $\boldsymbol{\mu}$) at iteration q of the subgradient optimization procedure. Step size s^q is then derived from the norm square of $\boldsymbol{SG^q}$ and the gap between the current best objective Z_{ub}^q (upper bound on Z_P^*) and current Lagrangian objective Z_{LR}^q. It is multiplied also by a scalar Λ^q whose first value Λ^1 is 2.0 by convention (see Fisher [33]). This scalar is halved whenever the objective Z_{LR} does not improve for a specified number of consecutive iterations. At the beginning, we set all Lagrange multipliers to the initial value zero. Formulae of the subgradient optimization routine for the Lagrangian relaxation of problem **P** are given below.

$$(SG_{ik}^{\lambda})^q = (\delta_{ik})^q - \sum_{\substack{j \in IC \cup \{k\} \\ j \neq i}} (x_{ijk})^q \qquad \forall i \in IC, \forall k \in ID \tag{4.8}$$

$$(SG_{ik}^{\mu})^q = (\delta_{ik})^q - \sum_{\substack{j \in IC \cup \{k\} \\ j \neq i}} (x_{jik})^q \qquad \forall i \in IC, \forall k \in ID \tag{4.9}$$

$$\left\| SG \right\|^2 = \left\| SG^{\lambda} \right\|^2 + \left\| SG^{\mu} \right\|^2 \tag{4.10}$$

$$s^q = \Lambda^q \frac{Z_{ub}^q - Z_{LR}^q(\lambda, \mu)}{\left\| SG \right\|^2} \tag{4.11}$$

$$(\lambda_{ik})^{q+1} = (\lambda_{ik})^q + s^q (SG_{ik})^q \qquad \forall i \in IC, \forall k \in ID \tag{4.12}$$

$$(\mu_{ik})^{q+1} = (\mu_{ik})^q + s^q (SG_{ik})^q \qquad \forall i \in IC, \forall k \in ID \tag{4.13}$$

As mentioned in the previous section, Z_{LR}^q has two components. The FLAP part of the objective function comes from the optimal solution of **SubP1**. The second part comes from **SubP2** which is tackled by an augmented Lagrangian relaxation scheme. Then the values are added up to constitute Z_{LR}^q. Before the subgradient optimization procedure starts, we generate a greedy initial solution for the complete problem. The upper bound Z_{ub}^q is assigned to the objective value of this quickly obtained initial solution, for which we assume that all present depots are in service, and solve a MDVRP for the remainder of the problem. The solution is accomplished by a method called [PFIH-NN] proposed by Aksen and Altınkemer [34]. It is a hybrid of Push Forward Insertion and Nearest Neighborhood methods explained in 5.1.1 and Appendix A. In the subsequent iterations, the upper bound comes from a tabu search method which solves a MDVRP for the location part of the solution of **SubP1**. This part of the **SubP1** solution reveals namely the facilities of the problem that will be in service.

4.4. FLAP-like Subproblem SubP1

The first of the two subproblems of LR($\boldsymbol{\lambda}$, $\boldsymbol{\mu}$) is the FLAP like problem **SubP1**. The formulation of **SubP1** can be stated as follows.

$$\text{Min } Z_{SubP1} = \sum_{k \in ID} OC_k y_k + \sum_{k \in ID_{cand}} FC_k y_k + \sum_{k \in ID_{pres}} FC_k (1 - y_k) + \sum_{k \in ID} \sum_{i \in IC} \delta_{ik} (\lambda_{ik} + \mu_{ik})$$

subject to: (3.2), (3.8), (3.10)

The technological coefficients matrix of **SubP1** is unimodular. For that reason we know that the decision variable δ_{ik} takes integer values even if it is not defined as an integer decision variable. So we can define δ_{ik}'s as nonnegative continuous decision variables between 0 and 1 instead of binary variables. Furthermore, the constraints in Equation (3.8) should be disaggregated as $\delta_{ik} \leq y_k \ \forall i \in IC, \ \forall k \in ID$. This way, we get a formulation with less integer variables, which in turn yields significantly better solution times for **SubP1**. In **SubP1**, the depots to be preserved or opened are determined, and customers are allocated to these. Each depot has its operating (OC_k) and opening cost (FC_k). FC_k is incurred when a candidate depot is opened or when a present depot is closed. FC_k of the present depots is generally negative corresponding to the savings in overhead costs and to the salvage value accrued by closing the depot. The Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ embody allocation costs of customers to depots. These costs change as the Lagrange multipliers get updated during the subgradient optimization iterations on the Lagrangian relaxed problem. At the beginning of each subgradient iteration, allocation costs are plugged in and **SubP1** is solved with Cplex to optimality. The solution times are generally reasonable. A problem instance with 20 depots and 1000 customers takes 2.84 seconds on a present-day desktop PC. Note that the location part of the **SubP1** solution is used by the tabu search procedure as an input telling which depots are going to be used in the MDVRP.

4.5. Minimum Spanning Forest-like Subproblem SubP2

4.5.1. Mathematical formulation and characteristics of SubP2

The second subproblem of LR($\boldsymbol{\lambda}$, $\boldsymbol{\mu}$) is **SubP2** which resembles a degree and capacity constrained minimum spanning forest problem (CMSF). The cost matrix C_{new} comprises the coefficients in the objective function of the subproblem. Since the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are embedded in this matrix and since they change at each subgradient iteration, C_{new} is to be calculated over again at the beginning of each iteration. The mathematical formulation of **SubP2** can be stated as follows:

$$\text{Min } Z_{SubP2} = \sum_{k \in ID} \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} (c_{ijk})^{new} x_{ijk} \tag{4.14}$$

subject to: (3.5)−(3.7), (3.9)

In **SubP2** depot locations $k \in ID$ represent center nodes, while customers $i \in IC$ are terminal nodes. The terminals should be accessible from one of the center nodes via a subtree rooted at that center. Equation (3.5) enforces that the numbers of outgoing and incoming arcs at each center node be equal. This balance-of-in-and-outdegree condition differentiates **SubP2** from the classical MSF. Capacity and subtour elimination constraints are given in Equation (3.7). The capacity constraint requires that the total demand on a subtree rooted at a center node do not exceed Q, the standard vehicle capacity in the main problem **P**. Equation (3.6), which is actually equal to the sum of the constraints stated in Equation (3.2), provides connectivity of the tree. As we mentioned in Chapter 3, (3.6) is added to the problem in order to solve **SubP2** as a MSF–like problem. Otherwise, we would have **SubP2** as a more relaxed problem which in turn would yield low quality lower bounds and high gaps. Equation (3.7) avoids the formation of subtrees which are not linked to any of the center nodes. Since the constraints in Equations (3.3) and (3.4) are relaxed, any node can have more than one offspring nodes.

4.5.2. Augmented Lagrangian relaxation for SubP2

If the balance-of-degree constraints in Equation (3.5) are discarded, and if the number of depots in *ID* is dropped to one, **SubP2** would reduce to the capacitated minimum spanning tree (CMST) problem. Papadimitriou [35] showed that CMST is an $\mathcal{NP}$-hard problem. Consequently, **SubP2** also belongs to the $\mathcal{NP}$-hard class. In order to solve **SubP2** we use the method proposed in Aksen and Altinkemer [26] where the augmented Lagrangian relaxation method of Gavish [32] is adopted and modified to tackle the balance of degree constraints. Gavish's CMST formulation is very similar to our **SubP2**'s formulation in that both have an exponential number of subtour elimination and capacity constraints. Gavish's paper [32] on the CMST problem is a pioneering study that combines an augmented Lagrangian and a subgradient optimization procedure into one solution technique. An early effective use of augmented Lagrangian based procedures was presented in Balas and Christofides [36] for solving asymmetric traveling salesman problems.

The idea behind augmented Lagrangian relaxation is simply to generate and dualize to-be-relaxed constraints dynamically as they are violated by the current iteration's Lagrangian solution instead of relaxing them all right away at the construction stage of the Lagrangian problem. This kind of Lagrangian relaxation has been applied for the classical VRP by Fisher [37], and for VRP-TW by Fisher, Jörnsten and Madsen [38]. A concise overview of Lagrangian lower bounds obtained in an augmented fashion can be found in a study of branch-and-bound algorithms for vehicle routing problems in Toth and Vigo [39]. The authors stress the main difficulty of most Lagrangian relaxation schemes for the VRP, which comes from the exponential cardinality of the set of relaxed constraints. This prevents one from including all of them explicitly into the objective function. Instead, the process of augmenting the Lagrangian problem with dynamically generated constraints as they are violated is iterated until no such constraint is detected or a prefixed number of subgradient iterations have been

executed. Slack constraints could be also purged from the Lagrangian problem during subgradient computations to keep the problem size under control.

The augmented Lagrangian relaxation is applied to **SubP2** as follows. We relax the subtour elimination constraints in **SubP2**, since this relaxation scheme achieves empirically better lower bounds on Z^*_{SubP2}. First, the constraint set in Equation (3.7) is divided into two parts as (3.7.a) and (3.7.b), the second of which is relaxed. Secondly, a trivial constraint which sets the minimum number of vehicles required is added to the original formulation as (3.7.c). This minimum number is calculated by solving the associated bin-packing problem that embraces all demand values d_i, $i \in IC$.

$$\sum_{k\in ID}\sum_{i\in S}\sum_{\substack{j\in S\\ j\neq i}} x_{ijk} \le |S|-1 \qquad \forall S \subseteq IC,\ \exists |S| \ge 2 \quad (3.7.a)$$

$$\sum_{k\in ID}\sum_{i\in S}\sum_{j\in S} x_{ijk} \le |S| - L_S \qquad \forall S \subseteq IC, |S| \ge 2 \quad (3.7.b)$$

$$\sum_{k\in ID}\sum_{i\in IC} x_{kik} \ge L_{IC} \quad (3.7.c)$$

The relaxed constraint set (3.7.b) is multiplied with Lagrange multipliers $\boldsymbol{\alpha}$, $\boldsymbol{\alpha} \le 0$. Left hand side values are subtracted from their right hand sides, and the resulting terms are augmented into the objective function of **SubP2** in Equation (4.14). In order to combine the embedded terms with Z_{SubP2} and to get a compact formulation for the objective function of the problem after the Lagrangian relaxation we separate Equation (4.14) into three parts as follows:

$$\sum_{k\in ID}\sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}} (c_{ijk})^{new} x_{ijk} = \sum_{k\in ID}\sum_{i\in IC} \left(c_{kik}\right)^{new} x_{kik} + \sum_{k\in ID}\sum_{i\in IC} \left(c_{ikk}\right)^{new} x_{ikk} + \sum_{k\in ID}\sum_{i\in IC}\sum_{\substack{j\in IC\\ j\neq i}} \left(c_{ijk}\right)^{new} x_{ijk} \quad (4.15)$$

After appending the dualized constraints and making the necessary rearrangements, the objective function and constraints of ALR^{SubP2} (the Lagrangian relaxed **SubP2**) can be written as follows:

$$\text{Min } Z_{LR}^{SubP2}(\alpha) = \sum_{k\in ID}\sum_{i\in IC}\left[\left(c_{kik}\right)^{new}\right]x_{kik} + \sum_{k\in ID}\sum_{i\in IC}\left[\left(c_{ikk}\right)^{new}\right]x_{ikk} + \sum_{k\in ID}\sum_{i\in IC}\sum_{\substack{j\in IC\\ j\neq i}}\left[\left(c_{ijk}\right)^{new} - \sum_{s\in G_{ij}}\alpha_s\right]x_{ijk} + \sum_{s\in\Psi}\left(|S| - \mathrm{L}_S\right)\alpha_s \qquad (4.16)$$

subject to: (3.5), (3.6), (3.7.a), (3.7.c), (3.9)

The last term in Equation (4.16) is constant for a given set of Lagrange multipliers $\boldsymbol{\alpha}$. At first glance, there seems to be no use to include it in the optimization scheme and one could incorrectly think of omitting the term from Z_{LR}^{SubP2}. However, the solution to ALR^{SubP2} will constitute a lower bound for the optimal solution of LR. Omitting the constant term would overestimate or underestimate the lower bound depending on the negativity of the term. Therefore we keep it in the objective function although it has no effect on the minimization process.

Observe that S in the relaxed constraints represents any unordered subset of IC with a cardinality greater than one, which requires two or more vehicles to deliver orders. The set of such subsets is denoted by $\boldsymbol{\Psi}$. For each $S \in \boldsymbol{\Psi}$, there is an associated Lagrange multiplier $\alpha_s \geq 0$. Let G_{ij} denote the index set of subsets S in $\boldsymbol{\Psi}$ that contain customer nodes i and j. The augmented Lagrangian relaxation feature is used here, because we do not explicitly generate all constraints in Equation (3.7.b). Therefore, we do not compute the entire multiplier vector $\boldsymbol{\alpha}$, either. The augmented Lagrangian relaxed problem ALR^{SubP2} is equivalent to an MSF problem without capacity constraints where the cost matrix C_{new} is dependent on the center node of departure. However, there are two distinct restrictions in this MSF problem:

i. The sum of outgoing degrees of all center nodes has to be equal to or greater than L_{IC} as required by the constraint in (3.7.c).
ii. At each center node, incoming and outgoing degrees should be equal as required by constraints in Equation (3.5).

The solution of the problem ALR^{SubP2} is checked against the violation of constraints in Equation (3.7.b) in **SubP2**. If any violated constraint is detected, it is added together with its associated Lagrange multiplier to the set of active constraints and multipliers. The objective function is augmented with the product of the difference between the violated constraint's right- and left-hand side values and the associated Lagrange multiplier's initial value. We do not remove previously augmented constraints from the set of active constraints in the Lagrangian problem; neither do we generate any such constraint for a second time. Another issue inherent in an augmented Lagrangian relaxation is to determine an initial value for its associated dual multiplier of a newly generated constraint. Typically at the beginning of subgradient iterations, Lagrange multipliers are assigned to zero no matter of their sign restrictions. In an augmented relaxation, however, dualizing a newly generated constraint with a zero multiplier would have no effect on the Lagrangian objective value. The goal in a minimization problem is to obtain a lower bound as tight as possible. This means a higher optimal objective value from the Lagrangian problem is sought for. Gavish [32] explains a sensitivity analysis technique to elevate the value of the Lagrangian objective function by finding an initial multiplier value for every augmented constraint while maintaining the optimality property of the Lagrangian solution before that constraint. With a few modifications fitting the forest rather than tree structure in the problem, this sensitivity analysis technique is adopted also into the augmented Lagrangian relaxation of our problem **SubP2**.

Finally, the degree balance constraints in Equation (3.5) and the minimum sum constraint in Equation (3.7.c) on the center nodes' outgoing degrees should be reckoned with. These constraints could have been relaxed and embedded into the Lagrangian objective as well, leaving us with the multi-center node version of the well-studied

minimum spanning tree (MST) problem. However, such a relaxation would excessively loosen the parent problem **SubP2**, and this would make it even harder to close the gap between the lower and upper bounds on **SubP2**'s true optimal objective value.

4.5.3. Solving DCMST-like problem ALR^{SubP2}

The closest version of ALR^{SubP2} is the degree-constrained minimum spanning tree problem (DCMST). Garey and Johnson [40] prove that the DCMST with arbitrary degree constraints on nodes other than the center is $\mathcal{NP}$-hard. In spite of copious methods and algorithms developed for the DCMST in the literature, we cannot use any of them as is due to the following reasons:

i. ALR^{SubP2} displays a forest structure with asymmetrical and center-node dependent costs
ii. The degree constraints that appear in ALR^{SubP2} relate to the balance of incoming and outgoing degrees at the center nodes only.

There exists also a lower bound on the sum of outgoing degrees at the centers according to Equation (3.7.c). From this perspective, ALR^{SubP2} is conceivably easier to solve than a general DCMST problem. Aksen and Altınkemer [26] develop a polynomial-time procedure called [MSF-ALR] which is largely an adaptation of Prim's MST algorithm. They modify Prims's algorithm to handle the balance-of-degree constraints at center nodes as well as ensuring minimum number of outgoing arcs from these center nodes. We take on their solution method for solving the problem ALR^{SubP2} with some modification in the way the algorithm takes care of the extra constraints. Aksen and Altinkemer's algorithm first constructs a minimum spanning forest using an algorithm like Prim's algorithm without regarding the minimum total number of out degree constraint on center nodes. Then the solution is controlled to discover whether the constraint is violated or not. If the number of arcs emanating from the depots is less then the minimum number of out degrees, new arcs between the depots and customers are added to the forest. In order to select the arc that will yield the minimum additional

cost when it is inserted to the solution, all the arcs between the center nodes and the customers are scanned. The arc costs in ALR^{SubP2} are dependent on the depot node from which the edges of the arc are accessible. Therefore, when an arc between a center and a customer node which is previously accessible from another center node is considered, all of the arc cost in the subtree succeeding the customer is recalculated. The change in the total of arc costs in the subtree is taken into account as well as the cost of inserted arc. After the minimum total outgoing degree at center nodes constraint is satisfied, returning arcs from the customers to the center nodes are inserted to provide the balance of degree which is required by Equation (3.5). Our modification on the solution method is in the way the minimum number of outgoing arcs from center nodes constraint is handled. In order to avoid the prolonged solution time due to the recalculation of arc costs in the subtrees, we propose a more practical way. At the beginning of the algorithm we insert arcs to the forest as much as the minimum number of outgoing arcs, with lowest cost. Once the constraint is satisfied, the rest of the forest is constructed with the method of Aksen and Altinkemer [26]. The pseudo code of the algorithm can be found in Appendix B.

4.6. Subgradient Optimization in the Augmented Lagrangian Relaxation

The subgradient vector $\mathbf{\Upsilon}$ is calculated according to the formulae given below. The cardinality of the subgradient vector increases as the number of violated constraints goes up. In the formulae, G^q denotes the index set of those subtour elimination and capacity constraints in Equation (3.7.b) which have been violated and thus generated either in the current iteration q or in a previous iteration. Each index r in G^q corresponds to some subtree of customer nodes whose indices comprise a particular subset S in $\mathbf{\Psi}$ as explained in 4.5.2. There are as many as $|G^q|$ constraints from Equation (3.7.b) relaxed and augmented into ALR^{SubP2}. In Equation (4.19), s^q_{ALR} denotes the step size of the subgradient optimization, Λ^q_{ALR} is a scalar with the initial value 2.0, $Z_{ub(SubP2)}$ is an upper

bound on the true optimal objective value of **SubP2**, and finally $Z^{q}_{ALR(SubP2)}$ is the current augmented Lagrangian objective value. The scalar Λ^{q}_{ALR} is halved whenever $Z^{q}_{ALR(SubP2)}$ does not increase for a specified number of consecutive iterations. S_r in Equation (4.17) indicates the r^{th} subset of customers in $\boldsymbol{\Psi}$ which are spanned by the same subtree.

$$\left(\Upsilon_r\right)^q = \left(\left|S_r\right| - \mathrm{L}_{S_r}\right) - \sum_{k\in ID}\sum_{i\in S_r}\sum_{\substack{j\in S_r \\ j\neq i}}\left(x_{ijk}\right)^q \qquad \forall r \in G^q \tag{4.17}$$

$$^{q}_{ALR} = \Lambda^{q}_{ALR}\frac{Z_{ub(SubP2)} - Z^{q}_{ALR(SubP2)}(\alpha)}{\left\|\Upsilon\right\|^2} \tag{4.18}$$

$$\left(\alpha_r\right)^{q+1} = \min\left\{0, \left(\alpha_r\right)^q + s^{q}_{ALR}\left(\Upsilon^{\alpha}_{r}\right)^q\right\} \qquad \forall r \in G^q \tag{4.19}$$

The upper bound $Z_{ub(SubP2)}$ for the MSF-like problem **SubP2** is found at the beginning of the augmented Lagrangian relaxation as follows. First, a capacitated MDVRP is solved by a version of [PFIH-NN] where the algorithm is modified to handle the costs structure in C_{new}. All arc costs are calculated depending on the depot to which the customer will be connected. The solution attained is definitely a feasible solution for **SubP2** since the corresponding MDVRP solution satisfies the capacity, connectivity and balance of degree constraints in **SubP2**. Moreover, a returning arc to a depot in **SubP2** need not originate from the last customer on the respective route. Exploiting this fact, we apply a myopic improvement procedure for each depot. We search the arc with the lowest cost that connects the depot to one of the customers on the same route. If that arc is different than the currently defined returning arc, the current one is replaced by the lowest-cost arc found.

Chapter 5

GENERATING UPPER BOUNDS FOR P: THE TABU SEARCH HEURISTIC

We use a tabu search (TS) heuristic integrated into the Lagrangian relaxation scheme in order to generate upper bounds for the optimal solution of the comprehensive location routing problem. TS is an iterative meta-heuristic algorithm first proposed in its present form by Glover [41]. It guides the local search to prevent it from being trapped in premature local optima or in cycling. It starts with an initial solution which is generally obtained by a quick constructive heuristic algorithm. At each iteration of the TS, a neighborhood of solutions is generated for the current solution. The best one from this neighborhood is picked as the current solution depending on a number of criteria. Certain attributes of previous solutions are kept in a tabu list which is updated at the end of each iteration. The selection of the best solution in the neighborhood is done such that it does not attain any of the tabu attributes. Best feasible solution so far (incumbent) is updated if the current solution is both feasible and better than the incumbent. The procedure continues until one or more stopping criteria are fulfilled.

At each subgradient iteration of the outer Lagrangian relaxation of **P**, first the facility location allocation problem **SubP1** is solved. The solution obtained for **SubP1** reveals the depots that are preserved and newly opened, as well as which customers are allocated to which depots. For upper bound generating procedure, we only take facility location plan part of this solution. Once the facilities in service are known, the remainder of the problem becomes a MDVRP any feasible solution of which constitutes an upper bound to **P**. Each time a new depot location plan is obtained by solving **SubP1**, a tabu search (TS) heuristic solving MDVRP is triggered in the hope of

achieving a better upper bound for **P**. When the Lagrangian iterations terminate, a greedy method called Add-Drop heuristic starts in case the final gap is greater than 2%.

In our study, we adopted the tabu search procedure as proposed by Aksen et al. [42] for the open vehicle routing problem with fixed driver nodes. We tailored the procedure for the MDVRP, and enriched it with additional neighborhood generation moves.

5.1. An Initial Solution for P

In order to generate an initial solution for our TS procedure, we tested on two constructive heuristics. One of them is the [PFIH-NN] method proposed by Aksen and Altinkemer [34], which is a hybrid of Push Forward Insertion and Nearest Neighborhood heuristics. The second heuristic is a modified version of Clarke-Wright parallel savings [CW] heuristic, which was first proposed for the single depot capacitated VRP by Clarke and Wright [43]. We have adjusted the algorithm so as to construct a good feasible solution to the capacitated MDVRP.

Since the former algorithm [PFIH-NN] exhibits an empirically better performance, we have chosen [PFIH-NN] to construct an initial solution for the MDVRP when necessary.

5.1.1. The [PFIH-NN] algorithm

In [PFIH-NN] customers are first assigned to the nearest depot. Then they are placed in an array sorted in the non-decreasing order of a special cost coefficient. This coefficient is calculated for each customer based on his distance to the assigned depot. The customer with the lowest cost coefficient is appended to a route. The remaining customers in the array are then chosen one at a time, and inserted into this first route according to the cheapest insertion principle. When the next to-be-inserted customer's demand exceeds the spare capacity on the current route, a new route is initiated.

Before starting the subgradient optimization for **P**, we use [PFIH-NN] to find an initial solution for the comprehensive MDLRP problem. In doing this, all of the present

depots are assumed to be preserved, and the remainder of the problem is solved as a MDVRP by [PFIH-NN]. When there exist no present depots, the candidate depot with the least opening and operating cost is selected to be in service. A pseudo code of the algorithm can be found in Appendix A.

5.1.2. The [CW] parallel savings algorithm for the MDVRP

This savings algorithm was first proposed by Clarke and Wright [43] for the single depot capacitated VRP. It can solve an instance with 1,000 nodes in one second on a fast modern-day desktop PC. Furthermore, it can be implemented without using advanced data structures. Therefore it still remains popular to date. In this algorithm, first a dedicated route is formed for each customer. Savings that will be obtained by merging two routes are calculated for all pairs of dedicated routes. Then, routes are merged starting with the pair that yields the highest savings. The algorithm is repeated until none of the remaining mergers is feasible.

In order to construct a feasible solution for the MDVRP we append an allocation phase at the beginning of the algorithm. In this phase, each customer is allocated to the nearest depot. After determining which customer will be served by which depot, a single depot capacitated VRP is solved using [CW] for each of the depots with more than one customers allocated.

5.2. Evaluation of the Solutions

For a given location plan, the objective of the problem is to minimize the vehicle acquisition and total traveling cost. In our tabu search method, we apply strategic oscillation by admitting infeasible solutions where infeasible solutions are penalized in proportion to the violations of capacity and time constraints. The penalty terms are added to the objective value of an infeasible solution. This penalty is intended to prevent the algorithm from spending too much time with exploring the infeasible regions of the search space. Every 10 iterations, the number of feasible and infeasible

solutions visited are compared. If the number of feasible solutions exceeds that of infeasible solutions, penalty terms are divided by a factor of 1.5; otherwise penalty terms are multiplied by 1.5. The purpose of dynamically adjusting penalty coefficients, referred to as strategic oscillation, is to keep the search procedure around the boundary of feasible and infeasible regions. Since good solutions are expected near this boundary, exploring this region is likely to provide better solutions.

The objective value for a solution is obtained by Equation (5.1).

$$\sum_{k\in ID}\sum_{\substack{j\in I\\ j\neq i}}\sum_{i\in I} c_{ijk}x_{ijk} + \sum_{r\in R} p_c V_c(r) \tag{5.1}$$

The first term is the total traveling cost containing the vehicle acquisition cost. This cost can be appended to the costs of the arcs which connect depots directly to customers. Every time an arc from a depot to a customer is included in the solution, a new vehicle is needed. Thus, its acquisition cost is incurred in the cost of that arc. The second is the penalty term for the routes that hold more demand than the vehicle capacity. R is the set of all routes, $V_c(r)$ denotes the overload (total demand of customers in route r minus vehicle capacity Q), and p_c denotes the penalty coefficient for overload on a route.

The best infeasible objective value is kept in memory during the procedure as well the best feasible objective value. The best infeasible objective value is updated when a feasible or infeasible solution with a better total cost is obtained. We utilize from this value in the aspiration and stopping criteria which are explained in following subsections.

5.3. Neighborhood Structure, Tabu Attributes and Stopping Conditions

In tabu search, at every iteration a neighborhood of the current solution is generated by the use of some move operators. The move operators are defined actions which generate neighboring solutions by altering one or more attributes of the current solution. Some examples to these attributes can be the position of a node on a route, the depot

that a customer is allocated, or the number of routes serving the customers. After the neighborhood generation, the neighboring solutions that do not possess one of the tabu attributes are examined, and the one with the best objective function value is set as the new current solution. Depending on the configuration of the tabu search heuristic, one can generate all possible neighboring solutions that can be obtained by the relevant move, or one can restrict the neighborhood size to a certain number of solutions. The decision depends on the trade-off between the solution time and solution quality. As more neighboring solutions are generated, the possibility to come across a better solution increases.

5.3.1. Move operators and neighborhood size

We use four move operators to create a neighborhood for the current solution. The pictorial descriptions of the moves can be found in Figure 5.1 – Figure 5.8. Each move involves two pilot nodes:

1-0 *move* : One of the pilot nodes is taken from its current position and inserted after the other. (Figure 5.1, Figure 5.2)

1-1 *exchange* : Two pilot nodes are swapped.(Figure 5.3, Figure 5.4)

2-*Opt move* : For two pilot nodes in the same route, the arcs emanating from these are removed. Two arcs are added one of which connects the pilot nodes, and the other connects their successor nodes. If the pilot nodes are in different routes, then the route segments following them are swapped preserving the order of nodes succeeding the pilots on each segment. (Figure 5.5, Figure 5.6)

2-2 *exchange* : One of the pilot nodes and its successor are swapped with the other pilot node and its successor. (Figure 5.7, Figure 5.8)

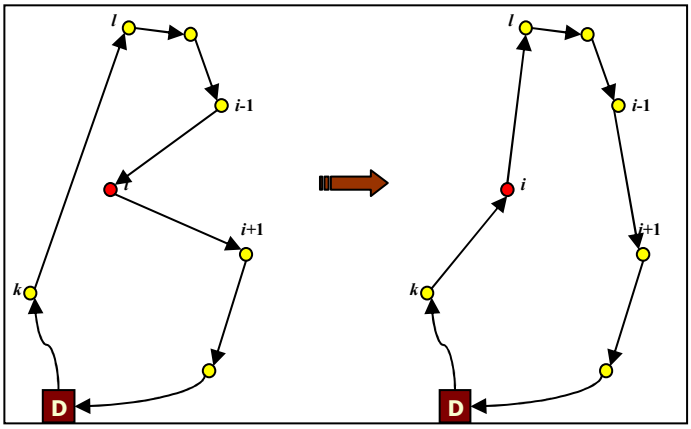

Figure 5.1 1-0 move in same route

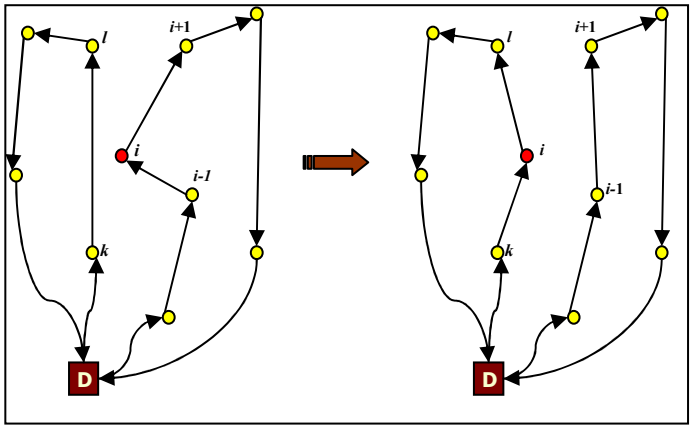

Figure 5.2 1-0 move between two routes

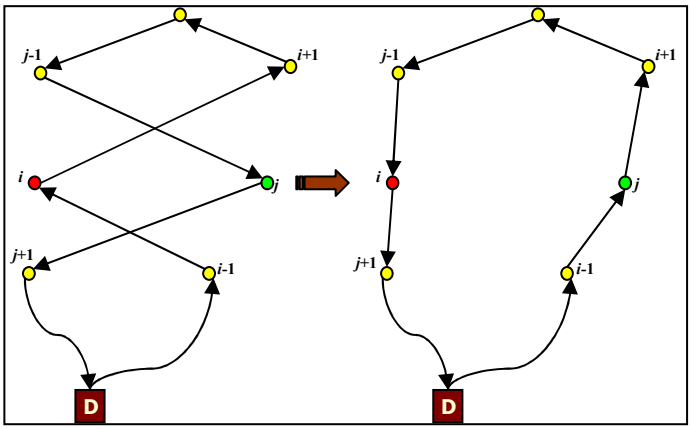

Figure 5.3 1-1 exchange in same route

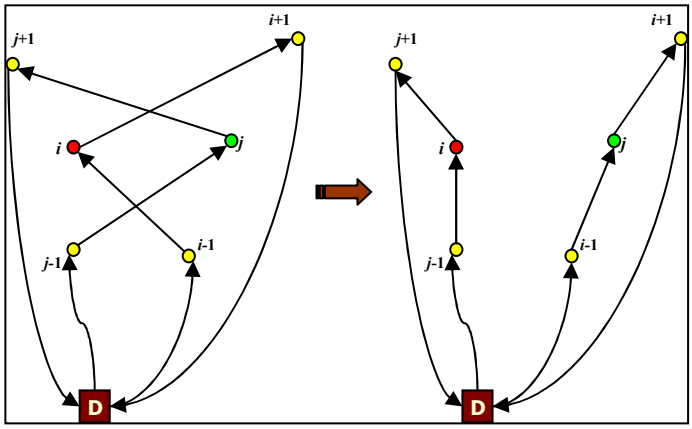

Figure 5.4 1-1 exchange between two routes

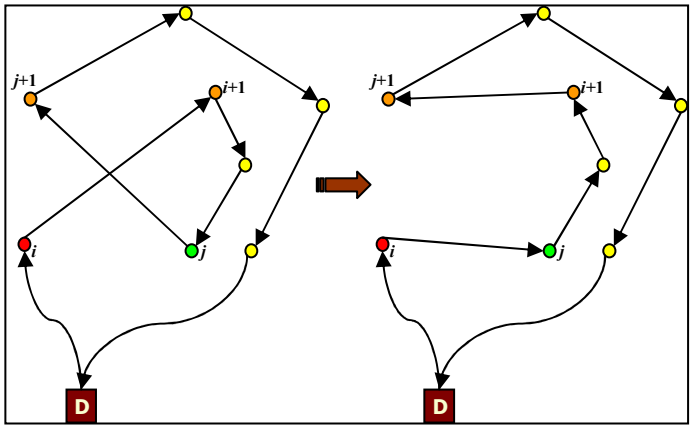

Figure 5.5 2-opt move in same route

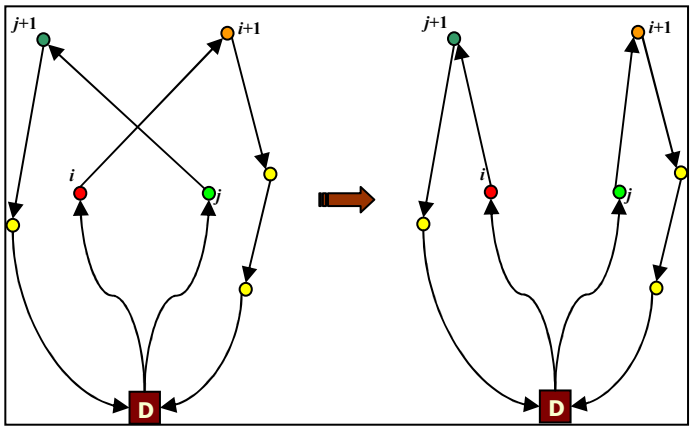

Figure 5.6 2-opt move between two routes

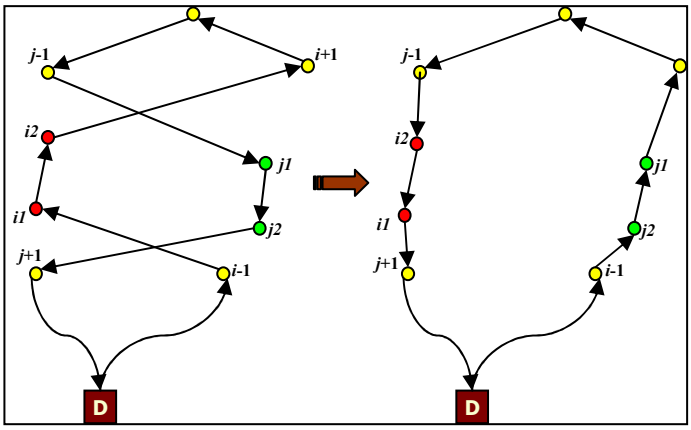

Figure 5.7 2-2 exchange in same route

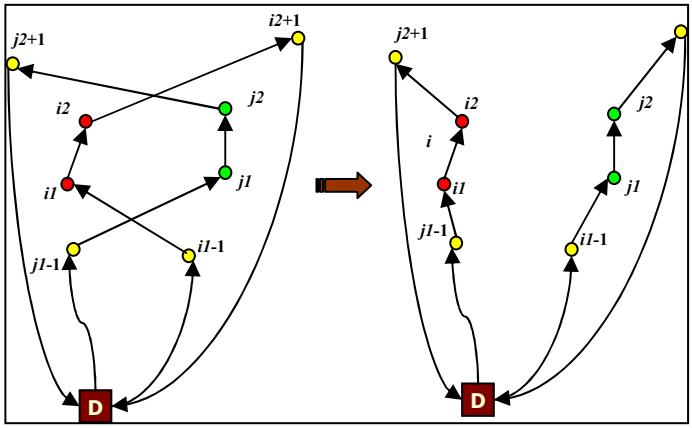

Figure 5.8 2-2 exchange between two routes

In our TS implementation, the size of the neighborhood is restricted because of time consideration. The number of solutions generated in each iteration depends on the number of operating depots and on the number of customer nodes in the problem. The solutions are generated in a probabilistic manner. Before the generation of each neighboring solution we choose one of the move operators with probability equal probability. Then we randomly pick two pilot nodes from the problem space.

5.3.2. Local post optimization

Besides neighborhood generation, we incorporate also a local search with these moves into the tabu search as a tool of local post optimization (LPO). A series of LPO operations are to be applied to the initial solution, to the current solution at the end of every 100 iterations if it is feasible, and also to the incumbent (current best solution) whenever it is updated. This helps the intensification of tabu search on the given MDVRP instance.

In the application of LPO, all customers are set one by one as the first pilot node. For a given pilot node, the second one is chosen such that the related move yields the highest improvement in total distance without causing any infeasibility. Every time we use local post optimization, we apply a sequence of five LPO methods. We have determined the sequence of the LPO operations empirically according to the results of extensive experimentation. We have tested 26 distinct sequences of four LPO methods which are listed in Table 5.1. The tests have been conducted on 33 MDVRP problems compiled by J.-F. Cordeau at the following web address: http://neumann.hec.ca/chairedistributique/data/ .

LPO Type	**1**	**2**	**3**	**4**	**5**
A	2-2 exchange	1-0 move	2-opt move	1-1 exchange	1-0 move
B	2-2 exchange	1-1 exchange	1-0 move	2-opt move	1-0 move
C	2-2 exchange	2-opt move	1-1 exchange	1-0 move	2-opt move
D	2-2 exchange	1-1 exchange	2-opt move	1-0 move	2-opt move
E	2-2 exchange	1-0 move	2-opt move	1-0 move	1-1 exchange
F	1-1 exchange	2-opt move	1-0 move	2-opt move	2-2 exchange
G	1-1 exchange	2-opt move	2-2 exchange	1-0 move	2-opt move
H	1-1 exchange	2-opt move	1-0 move	2-2 exchange	2-opt move
I	1-1 exchange	2-2 exchange	2-opt move	1-0 move	2-opt move
J	1-1 exchange	1-0 move	2-2 exchange	2-opt move	1-0 move
K	1-1 exchange	1-0 move	2-2 exchange	1-0 move	2-opt move
L	1-1 exchange	1-0 move	2-opt move	2-2 exchange	2-opt move
M	1-1 exchange	1-0 move	2-opt move	2-2 exchange	1-0 move
N	2-opt move	2-2 exchange	1-0 move	1-1 exchange	1-0 move
O	2-opt move	2-2 exchange	1-0 move	1-1 exchange	2-opt move
P	2-opt move	1-0 move	2-2 exchange	1-0 move	1-1 exchange
Q	2-opt move	1-1 exchange	1-0 move	2-2 exchange	2-opt move
R	2-opt move	1-0 move	1-1 exchange	1-0 move	2-2 exchange
S	2-opt move	1-1 exchange	2-opt move	2-2 exchange	1-0 move
T	1-0 move	2-2 exchange	2-opt move	1-1 exchange	1-0 move
U	1-0 move	2-2 exchange	1-0 move	1-1 exchange	2-opt move
V	1-0 move	2-opt move	1-1 exchange	2-opt move	2-2 exchange
W	1-0 move	1-1 exchange	2-opt move	2-2 exchange	2-opt move
X	1-0 move	2-opt move	1-1 exchange	2-2 exchange	1-1 exchange
Y	1-0 move	1-1 exchange	2-opt move	1-0 move	2-2 exchange
Z	1-0 move	2-2 exchange	1-0 move	2-opt	1-1 exchange

Table 5.1 LPO sequences tested

5.3.3. Tabu attributes

Tabu search heuristics make use of a list of attributes of the previously visited solutions in order to prevent cycling. Some attributes of the solutions are decided to be tabu during the design of the heuristic. These attributes of the visited solutions are kept

in a tabu list. At each iteration, while choosing the best solution in the neighborhood of the current solution, the neighboring solutions are checked whether they possess at least one of the attributes declared tabu in the list. Tabu attributes of a solution generated by a move can be stated as follows.

1-0 *move* : If node *i* is inserted after node *j*, the position of *i* cannot be changed by the same move while it is tabu-active.

1-1 *exchange* : If nodes *i* and *j* are swapped, they cannot be swapped again while they are tabu-active.

2-*Opt move* : If 2-Opt move is applied to nodes *i* and *j*, the move cannot be applied again to the same nodes while they are tabu-active.

2-2 *Exchange* : If nodes *i* and successor of *i* are swapped with nodes *j* and successor of *j*, these cannot be swapped again while they are tabu active.

The tabu list is updated at the end of each iteration. Some attributes are dropped from the list, while the tabu attributes of the new current solution are added to the list. The number of iterations during which an attribute stays in the tabu list; i.e., the duration that the attribute is tabu-active is called tabu tenure. At each iteration, the tabu tenure is chosen randomly between 5 and 15 iterations.

5.3.4. Aspiration criterion

In some cases, namely if the so-called aspiration criterion is satisfied, a neighboring solution can be accepted although its attributes are tabu-active. In our tabu search implementation, the aspiration criterion is considered to be satisfied if the total cost of the neighboring solution resulting from the move is better than the incumbent's objective value. If the neighboring solution obtained is infeasible, it is compared with the best infeasible solution.

5.3.5. Stopping conditions

The tabu search heuristic terminates when any of two stopping criteria is satisfied. First criterion is the total number of iterations performed. Second criterion is the maximum permissible number of iterations during which the best feasible or the best infeasible solution does not improve. Both values are determined based on the problem size.

5.4. The Probabilistic Nature of the Proposed Tabu Search Algorithm

The tabu search algorithm proposed has probabilistic nature in selection of move operators, determining the pilot nodes in a move and choosing the tabu tenure. The values of these attributes are determined using uniform pseudo-random numbers which are obtained by a linear congruential pseudo-random number generator designed by Park and Miller [44]. The series of the pseudo-random numbers depends on an initial random number seed. At the beginning of every single tabu search run, the initial random number seed is taken from an array of 20 predetermined random number seeds in sequence.

Since the sequence of the random numbers generated is reliant on the initial random number seed, the seed have effect on the moves, pilot nodes and tabu tenure at each iteration of the tabu search heuristic. Starting with the same initial solution but with different seeds, the heuristic can come up with different final solutions. In order to remove the effect of the initial random number seed, in some parts of the Lagrangian relaxation method the tabu search is run several times with different seeds for a specific location plan (6.4).

5.5. Add-Drop Heuristic

When the Lagrangian iterations terminate, an Add-Drop heuristic is applied to the best feasible solution found if the final gap between Z_{ub} and Z_{lb} is greater than 2%. First, closed or unopened depots are added to the solution one by one; then, currently opened

depots are dropped from the solution in a similar decremental fashion. The MDVRP is solved with respect to each of these scenarios using the tabu search heuristic. Because of the probabilistic nature of the tabu search heuristic, the tabu search is run for 10 times with different initial random number seeds. The best of the final solutions of the 10 runs is taken and compared with the best solutions of the other scenarios. If a better feasible solution is realized, the new depot location plan is adopted, and the Z_{ub} value and the final gap are updated as well. A pseudo code of the Add-Drop heuristic is provided in Appendix D.

Chapter 6

COMPUTER EXPERIMENTS AND RESULTS

6.1. More Details on the Proposed Solution Method

Stopping criteria of the Lagrangian relaxation

The mutually exclusive stopping conditions of the subgradient optimization for the outer Lagrangian relaxation are defined as follows. If the number of subgradient iterations performed exceeds 300, or if the number of consecutive subgradient iterations during which the Lagrangian gap does not improve reaches 100, or finally if the amount of absolute change in the Lagrange multipliers is not greater than 1.0e–7, the subgradient optimization procedure for the problem **P** stops. The stopping conditions in case of the augmented Lagrangian relaxation applied to **SubP2** are satisfied if the predefined limit on one of following parameters is reached: 150 subgradient iterations performed in the augmented Lagrangian relaxation, the step size or the gap between $Z_{lb(SubP2)}$ and $Z_{ub(SubP2)}$ dropping below 1.0e–5, and finally 75 consecutive iterations during which the gap does not improve.

The constructive heuristic to generate an initial solution for the tabu search

Tabu search is an improvement heuristic which starts with an initial solution and searches for better ones throughout its iterations. Among the constructive heuristics explained in 5.1, we adopted [PFIH-NN] to generate the initial solution at the beginning of each tabu search run. [PFIH-NN] is also employed in order to generate feasible

solutions for **P** at the beginning of the algorithm [LR-TS] and for **SubP2** each time the inner Lagrangian relaxation is called (4.3, 4.6).

LPO sequence

The local post optimization methods utilized are embedded into the tabu search as an intensification tool. In the tabu search algorithm we designed, the LPO methods are applied both to the initial solution at the beginning of the heuristic and to the best feasible solution whenever it is updated. After extensive experimentation with 26 different sequences specified in Table 5.1, the sequence that is represented with the letter L is selected. The sequence L is as follows: 1-1 exchange, 1-0 move, 2-opt move, 2-2 exchange, 2-opt move.

6.2. The Design of Computer Experiments

All of the codes for the proposed method have been written in ANSI C language, compiled in Visual C++ .NET and executed on a 3.20 GHz Intel Xeon processor with 2 GB RAM. To test the performance of the algorithm we have experimented on a test bed of 96 problems composed of three parts. The first part includes 30 small size randomly generated problems (R1) with 15–35 customers and 2–6 depots. The second part, the larger problems (R2) consist of 30 randomly generated instances with number of customers and depots ranging from 40 to 100 and from 4 to 6, respectively. The third part of the test bed has been designated as TB. It takes in the 36 problems solved in Tüzün and Burke [23] which have 100–200 customers and 10–20 depots. The problems in R1 have been also solved by Cplex 10.0 with a time limit of five hours. The best integer solutions found by Cplex constitute benchmarks for the upper bounds obtained by our method. The upper bounds obtained for the TB problems have been compared with the solutions provided in Tüzün and Burke [23]. We utilized a subset of the test problems for the fine-tuning of the parameters of the tabu search and the Lagrangian relaxation algorithms. In the remainder of this chapter, the details of the random

problem generation procedure are first presented. Then the experiments for the fine-tuning of the algorithmic parameters are explained. Finally, the results and some conclusions are given for the problems of R1, R2 and TB along with the comparisons with other solution methods.

6.3. Random Problem Generation

In order to evaluate the performance of the solution method and to study the configuration of the algorithmic parameters, we randomly generated 60 LRP instances. The number of customers in the problems ranges from 15 to 100, while the number of depots changes between 2 and 6. We divide the test problems into two parts, as R1 and R2, depending on the number of customers in the problem. The instances are created using a generator program coded in ANSI C language. The program works based on several parameters. The values of the parameters are provided by the user. The parameters are explained in two groups as Type 1 and Type 2 parameters. In Table 6.1 the short forms, explanations and values of the Type 1 parameters are given. This group contains the parameters which are set to the same value for all of the random test problems. Table 6.2 shows the information for Type 2 parameters. This group of parameters takes different values for different problems. The details of the random problem generation process and the generated test problems are provided in Tables 6.1 to 6.5.

Among the parameters, $dist_C$ denotes the distribution of the customer locations on the problem space. The parameter *U* indicates that the customers locations are generated using a uniform distribution. *RU* means the customers locations come from a distribution that is rectilinear uniform on a specified number of equidistant longitudes and altitudes. For the entire problems in which *RU* is used the number of equidistant longitudes and altitudes are set as 2 and 3 respectively. Finally, *C* stands for “Clustered Uniform around depots”. The vehicle capacity is assigned based on the sum of the demands of the customers and the coefficient *k,* with the formula given in Table 6.1.

The problems have been generated using 10 different values for the number of customers (N_C). For each specific value of number of customers, two different number of present-candidate depots (N_{PD}-N_{CD}) was used. For each of the 20 $N_C - N_{PD} - N_{CD}$ pairs, three problems with different spatial distribution of customer locations ($dist_C$) have been generated (Table 6.3). The values of Type 2 parameters for each problem in R1 and R2 are presented in Table 6.4 and Table 6.5, respectively.

In the assignment of the fixed depot opening-closing (FC) and operating costs (OC) we assumed that all the costs are projected to the daily time scale as the vehicle traveling costs. We observed that it is reasonable to set the FC and OC values equal to approximately 10% of the total traveling cost in the problem. Also the vehicle acquisition costs (VC) are projected to same time scale. In the average, the VC values correspond to 10% of the fixed depot opening-closing costs. Appendix E details the calculation of FC and OC values.

Parameter	Explanation	Value
$W \times H$	Dimensions of the rectangular problem space (width and height).	$W = 300,\ H = 250$
(x_0,y_0)	Coordinates of the depot at the center of the problem space.	$(x_0,y_0) = (0,0)$
$dist_{PD}$	Distribution of the locations of present depots	U : uniformly distributed in the problem space
$dist_{CD}$	Distribution of the locations of candidate depots	RU : rectilinear uniform on specified number of equidistant longitudes and altitudes
d_i	Demand of customers i	Comes from a Continuous Uniform Distribution between [5,10]
Q	Vehicle capacity	$Q=\left\lceil \frac{\sum_i d_i}{k} \right\rceil$

Table 6.1 Type 1 parameters

Parameter	Explanation	Values used
N_C	Number of customers	R1 : 15,20, 25, 30, 35 R2 : 40, 50, 60, 80, 100
N_{PD}	Number of present depots	R1 : 0, 1 R2 : 0, 1, 2
N_{CD}	Number of candidate depots	R1 : 2, 3, 4, 6 R2 : 4, 5, 6
FC	Fixed opening-closing cost of depots	[20, 45]
OC	Fixed operating cost of depots	[25, 50]
VC	Vehicle acquisition cost	3, 4
k	Coefficient for vehicle capacity	4.5, 5, 5.5, 6, 7
$dist_C$	Distribution of the locations of customers	U : uniformly distributed in the problem space RU : rectilinear uniform on specified number of equidistant longitudes and altitudes C : clustered around the locations of depots
SEED	Initial random number seed.	27, 82, 951

Table 6.2 Type 2 parameters

Parameter	# values
N_C	10
$dist_C$	3
N_{CD}-N_{PD} *pair*	2
TOTAL	60

Table 6.3 The problem generation pattern

ID	N_{CD}	N_{PD}	N_{CD}	$dist_C$	SEED	k
R1-1	15	1	2	U	82	4.5
R1-2	15	1	2	RU	27	4.5
R1-3	15	1	2	C	951	4.5
R1-4	15	-	2	U	82	4.5
R1-5	15	-	2	RU	27	4.5
R1-6	15	-	2	C	951	4.5
R1-7	20	1	3	U	82	5
R1-8	20	1	3	RU	27	5
R1-9	20	1	3	C	951	5
R1-10	20	-	3	U	82	5
R1-11	20	-	3	RU	27	5
R1-12	20	-	3	C	951	5
R1-13	25	1	2	U	82	5
R1-14	25	1	2	RU	27	5
R1-15	25	1	2	C	951	5
R1-16	25	1	4	U	82	5
R1-17	25	1	4	RU	27	5
R1-18	25	1	4	C	951	5
R1-19	30	1	4	U	82	5.5
R1-20	30	1	4	RU	27	5.5
R1-21	30	1	4	C	951	5.5
R1-22	30	-	4	U	82	5.5
R1-23	30	-	4	RU	27	5.5
R1-24	30	-	4	C	951	5.5
R1-25	35	1	4	U	82	6
R1-26	35	1	4	RU	27	6
R1-27	35	1	4	C	951	6
R1-28	35	-	6	U	82	6
R1-29	35	-	6	RU	27	6
R1-30	35	-	6	C	951	6

Table 6.4 The Type 2 parameter values for R1

ID	N_{CD}	N_{PD}	N_{CD}	$dist_C$	SEED	k
R2-1	40	1	4	C	951	5
R2-2	40	1	4	RU	27	5
R2-3	40	1	4	U	82	5
R2-4	40	-	4	C	951	5
R2-5	40	-	4	RU	27	5
R2-6	40	-	4	U	82	5
R2-7	50	1	4	C	951	6
R2-8	50	1	4	RU	27	6
R2-9	50	1	4	U	82	6
R2-10	50	-	5	C	951	6
R2-11	50	-	5	RU	27	6
R2-12	50	-	5	U	82	6
R2-13	60	1	4	C	951	6
R2-14	60	1	4	RU	27	6
R2-15	60	1	4	U	82	6
R2-16	60	-	5	C	951	6
R2-17	60	-	5	RU	27	6
R2-18	60	-	5	U	82	6
R2-19	80	2	4	C	951	7
R2-20	80	2	4	RU	27	7
R2-21	80	2	4	U	82	7
R2-22	80	-	6	C	951	7
R2-23	80	-	6	RU	27	7
R2-24	80	-	6	U	82	7
R2-25	100	2	4	C	951	8
R2-26	100	2	4	RU	27	8
R2-27	100	2	4	U	82	8
R2-28	100	-	6	C	951	8
R2-29	100	-	6	RU	27	8
R2-30	100	-	6	U	82	8

Table 6.5 The Type 2 parameter values for R2

6.4. Experimenting with Algorithmic Parameters

The maximum number of iterations in outer and inner Lagrangian relaxations, the pattern of the tabu search heuristic implementation and the LPO sequence applied in the tabu search have been finalized in accordance with the empirical findings of the tests conducted. In this section, the details of the experiments and the results are explained.

Maximum number of iterations of outer and inner Lagrangian relaxations

The stopping conditions of the Lagrangian relaxation have been fine-tuned by testing on 16 test problems. Since the solution times of the larger problems are not practical for such experimentation, 10 of these problems have been selected among the randomly generated test problems with the number of customers between 15 and 35. The remaining 6 problems have been chosen from among the TB instances.

num_ out	*num_ in*
400	100
350	150
350	200
300	150
300	200
250	150

Table 6.6 Maximum number of iteration of inner and outer LR values tested

Six different combinations of the number of iterations in the outer Lagrangian relaxation (*num_out*) and the number of iterations in the inner Lagrangian relaxation (*num_in*) have tested on these 16 LRP instances (Table 6.6). The "*num_out – num_in"* combination has been evaluated based on three criteria: the gap between the best feasible objective value found and the objective of the benchmark solution, the final gap

between the best lower bound and the best upper bound found and the CPU time elapsed. For 10 small test problems, benchmark results are the Cplex solutions while the ones for the TB problems are the solutions found by Tüzün and Burke [23]. Consequently, the maximum number of iterations in outer and inner Lagrangian relaxation has been determined as 300 and 150, respectively, for a favorable performance and reasonable solution time.

Two different patterns of tabu search implementation

In the comprehensive solution method proposed for the LRP, we make use of the tabu search in order to find a good feasible solution. An MDVRP is solved by the tabu search throughout the iterations of the outer Lagrangian relaxation and in the Add-Drop heuristic (5.5). Because of the probabilistic nature of the tabu search algorithm (5.4), in some cases the tabu search is run more than once starting with different initial random number seeds. The best objective of these runs is taken as a remedy against the probabilistic character of the heuristic. The decision of where and how many times to apply tabu search depends on the balance between the solution time and the solution quality.

In order to determine how to utilize tabu search during [LR-TS], we experimented on two patterns. In the first one, the tabu search is run for once during the outer Lagrangian relaxation whenever the solution of **SubP1** reveals a new location plan. If the Add-Drop heuristic is decided to be applied after the Lagrangian iterations terminate, the tabu search is run once for every distinct location plan generated by the add and the drop moves. After the Add-Drop heuristic the tabu search solves a MDVRP 10 times for the final location plan with different initial random number seeds. In pattern 2, the tabu search is run for once during the Lagrangian relaxation iterations, as well. However, when the Lagrangian iterations terminate, the tabu search is run 10 times for the particular location plan of the best feasible solution found. Then, if the Add-Drop Heuristic is triggered, for each distinct location plan generated, the tabu search works 10 times and the best solution of the 10 runs is adopted. This solution is

compared with the ones of other location plans and if it is better than the best feasible solution, the best upper bound is updated.

The two patterns are tried on the test problems which we utilize for fine tuning of the maximum number of iterations in Lagrangian relaxation. We adopted Pattern 2, for it provides better solutions with allowable increase in solution time.

	***Number of TS runs in Pattern* 1**	***Number of TS runs in Pattern* 2**
Through LR	1	1
End of LR	-	10
Add-Drop	1	10
End of Add-Drop	10	-

Table 6.7 Two patterns of tabu search implementation through [LR-TS]

The selection of the constructive heuristic to generate initial solution and the LPO sequence in tabu search

The constructive heuristic that generates an initial solution for the tabu search (TS), and the LPO sequence applied to the initial and the incumbent solution have been determined after experimenting on 33 MDVRP problems compiled by J.-F. Cordeau at the following web address: http://neumann.hec.ca/chairedistributique/data/ .

First, [PFIH-NN] has been used as the initial solution generating method and the 26 LPO sequences listed in Table 5.1 have been tested. All MDVRP instances have been solved 10 times by TS with different initial random number seeds utilizing each LPO sequence (33 *problems* x 10 *seeds* x 26 *sequences* = 8580 *runs*). For each test problem and LPO sequence, the average, maximum and the minimum objective values of 10 tabu search runs have been identified. Then, these values have been compared with the

ones that were obtained with TS using other LPO sequences. The performance of a given LPO sequence is evaluated based on the following three parameters:

n_min: The number of problems where the minimum objective value found by the tabu search using that LPO sequence is lower than the ones obtained with the tabu search using other LPO sequences.

n_max: The number of problems where the maximum objective value found by the tabu search using that LPO sequence is higher than the ones obtained with the tabu search using other LPO sequences.

n_avg: The number of problems where the average of the objective values found by the tabu search using that LPO sequence is lower than the ones obtained with the tabu search using other LPO sequences.

The three parameter values have been compared where the ones with higher *n_min*, *n_avg* and with lower *n_max* are considered performing better. The priority was given to the parameter *n_min*, while *n_avg* is checked second. In case these two parameter values were equal for two or more LPO sequences, *n_max* was used for determining the best. In accordance with the analysis of the test results which are depicted in Table 6.8, the LPO sequences C and L are found to perform better than the others.

After determining the LPO sequences with better performance in the experiments where the initial solutions of TS have been constructed with [PFIH-NN] heuristic, we tested these LPO sequences with an initial solution provided by [CW] heuristic. The very same method explained above has been used to evaluate the performances of different LPO sequences. As shown in Table 6.9, the tabu search with an [PFIH-NN] initial solution has performed better. Therefore, [PFIH-NN] is chosen as the initial solution construction method while the LPO sequence L is utilized in the tabu search heuristic.

LPO Type	*n_min*	*n_max*	*n_avg*
A	8	2	5
B	8	1	3
C	10	3	9
D	9	1	4
E	6	1	3
F	8	4	4
G	8	3	4
H	7	4	4
I	7	2	4
J	9	1	3
K	8	3	5
L	10	2	9
M	9	1	6
N	6	2	3
O	10	1	6
P	8	2	5
Q	10	2	5
R	10	2	3
S	9	2	5
T	7	3	4
U	8	4	3
V	8	1	5
W	9	2	5
X	7	1	5
Y	6	3	5
Z	9	2	4

Table 6.8 The values of the performance parameters for 26 LPO sequences where the initial solutions of TS have been found by [PFIH-NN] heuristic

LPO Type	*n_min*	*n_max*	*n_avg*
C after [PFIH-NN]	10	3	9
L after [PFIH-NN]	10	2	9
C after [CW]	6	7	4
L after [CW]	7	4	2

Table 6.9 Comparison of performance parameters for LPO sequences C and L

6.5. Results of Randomly Generated Test Problems

The proposed solution method [LR-TS] was first tested on the randomly generated test problems in R1 and R2. All codes have been written in ANSI C, compiled and executed in Microsoft Visual Studio .NET on a 3.20 GHz Intel Xeon® server with 2 GB RAM. For each test problems in R1, we have also constructed a GAMS model, and solved it on the same platform to optimality where possible using the general-purpose MIP solver Cplex 10.0. Each test problem's optimal or best feasible objective value found by Cplex makes a benchmark for the best feasible objective value obtained by [LR-TS]. In GAMS models, the accuracy obtained by the employed solver is controlled with a number of options. A relative optimality criterion (`OPTCR`) can be set for the MIP master problem determining when the solver should terminate its branch-and-bound (or branch-and-cut) procedure. `OPTCR` is defined as the ratio (|`BP`−`BF`|) / (1.0e−10 + |`BF`|) where `BF` is the objective function value of the current best integer solution while `BP` is the best known (current) lower bound in case of minimization. The solver stops trying to improve upon the integer solution `BF` when this ratio drops below the specified value. The options used in our GAMS models are explained in Table 6.10. The Cplex solutions are assessed using the gap between `BF` and `BP` calculated with the ratio given above and referred to as %GAP2 in Table 6.11.

Option	Explanation	Value
`ITERLIM`	Simplex algorithm iteration limit applied per node of the search tree.	5,000,000
`NODELIM`	Maximum number of nodes solved before the algorithm terminates, without reaching optimality.	5,000,000
`OPTCR`	Relative optimality criterion for a MIP problem.	0.01
`RESLIM`	Solution time limit for the MIP solver.	5 hours

Table 6.10 GAMS/Cplex options in the mathematical models

The Cplex solutions that serve as benchmark for the best upper bound found by [LR-TS] to problems in R1 belong to a different LRP model than **P**. We first constructed GAMS models for the ILP formulation **P** which is given in Chapter 3 and tried to solve with Cplex. However, with the available memory of 2 GB, Cplex was not able to even execute the model for problems with 20 and more customers. We supposed that the model could not be executed because of the exponential number of subtour elimination constraints given in Equation (3.7). We replaced those constraints with the ones shown in Equation (6.1). These are the alternative subtour elimination constraints which are referred to as Lifted Miller-Tucker-Zemlin (MTZ) constraints for the VRP. MTZ inequalities were first proposed by Miller, Tucker and Zemlin [45] for the traveling salesman problem, and then extended by Kulkarni and Bhave [46] to the VRP. Kara et al. [47] present in their recent note a correction for the lifted version of the Miller-Tucker-Zemlin equations for the VRP.

$$U_i - U_j + Qx_{ij} + (Q - d_i - d_j)x_{ji} \leq Q - d_j \qquad \forall i, j \in IC \qquad (6.1)$$

Cplex 10.0 was able to find integer solutions to the test problems of R1 solving the GAMS models of the LRP formulations with MTZ equations. With the hope of finding better solutions with Cplex, we also constructed and solved GAMS models with the 2-index LRP formulation proposed by Boğ [16]. The best integer solutions found by Cplex solving the 2-index LRP formulation were better than the ones obtained by solving the 3-index model. So we adopted the solutions of the 2-index model as benchmark for [LR-TS]. The 2-index formulation of Boğ [16] adapted to our comprehensive problem **P** is provided in Appendix F.

In the tables, Z_{Cplex} denotes the best feasible objective found by Cplex (`BF`), while that of [LR-TS] is shown by Z_{ub}. The best lower bound obtained by [LR-TS] for the comprehensive location routing problem **P** is represented by Z_{lb}. %GAP1 and %GAP3 are computed as shown in Equations (6.2) and (6.3) while %GAP2 is taken from the

GAMS output file. We make use of the ratios to assess the performance of the proposed solution method in terms of both the upper bound and the lower bound found. In the formulas the differences are used without taking the absolute values. Thus, %GAP3 taking a negative value means that, [LR-TS] has outperformed the Cplex solution in terms of solution quality.

$$\%GAP1 = 100 \times \frac{Z_{ub} - Z_{lb}}{Z_{lb}} \tag{6.2}$$

$$\%GAP3 = 100 \times \frac{Z_{ub} - Z_{Cplex}}{Z_{Cplex}} \tag{6.3}$$

Except for the problems with 15 customers, the Cplex solver terminated due to solution time limit without obtaining a solution that satisfies `OPTCR`. The randomly generated test problems in R2 are not solved with Cplex, for with more number of customers and depots Cplex is not expected to find comparable solutions in reasonable time. For test problems in R1, the quality of the [LR-TS] solutions is measured by %GAP1. The results obtained for the test problems in R1 and R2 are monitored through Table 6.11 to Table 6.17. The solutions of [LR-TS] and Cplex as well the gaps mentioned above are displayed in Table 6.11 and Table 6.12 for every single problem in R1 and R2, if available. In Table 6.13 to Table 6.17, results are aggregated with respect to the number of customers (N_C), the spatial distribution of customer locations ($dist_C$) and the total number of depots (N_D) in the problem. The average results for each of those parameter triplets are examined in order to find out their effect on the solution quality and the solution time. Observing the results in the tables, we derive some conclusions about our solution method [LR-TS] and Cplex.

Cplex accomplished to find a feasible solution that satisfies `OPTCR` for three of the problems in R1. For nine out of 30 problems Cplex comes up with solutions better than [LR-TS], while the upper bound (best feasible solution) found by [LR-TS] outperforms

the Cplex solution for 15 problems. If we examine the aggregated results in Table 6.13, we observe that as the number of customers in the problem goes up, the number of problems where [LR-TS] does better increases. When the number of customers reaches 35, Cplex cannot match [LR-TS] in any of the instances. On the average, the solutions of Cplex are improved by 1.8% with the proposed method.

The value of %GAP2 monitors the gap between the best possible and the best feasible solutions of Cplex, while %GAP1 serves the same purpose for [LR-TS]. The average %GAP1 is 4.27% for the problems in R1. In four of the problems it is lower than 1% and in two of them higher than 10%. On the other hand, the average %GAP2 is 14.91% with 19 of the problems having a %GAP2 value above 10%. Table 6.14 shows that in 24 out of 30 problems [LR-TS] has found better gaps in comparison with Cplex. From the table one can also detect that the number of problems where %GAP1 is lower than %GAP2 (i.e. $N_{[LR\text{-}TS]}$) increases as the problem size gets bigger.

When the results for problems in R2 are examined, a significant deterioration of the gap between the best feasible objective value and the best lower bound of [LR-TS] (%GAP1) is observed. The average %GAP1 is 16.27% for the problems in R2, while the value remains under 5% for the problems in R1. The deterioration can also be seen in detail in Table 6.15 and Figure 6.1, where the available results are aggregated for problems with the same number of customers. The average gap grows with the problem size. In Table 6.15 we can also observe that solution times get worse for bigger problems which is depicted in Figure 6.2.

In order to evaluate its quality, the final lower bound Z_{lb} is also compared against the optimal objective value of the linear programming (LP) relaxation of the modified 2-index formulation found by Cplex. In other words, the gap between the solution of LP relaxed problem and the upper bound provided by [LR-TS] is compared against %GAP1. As the result of LP relaxation we obtain average gaps of 31.79% and 50.05% for the problems in R1 and R2, respectively. It is observed that [LR-TS] provides significantly better lower bounds in comparison with the LP relaxation where average

%GAP1 is 4.71% and 16.27% for R1 and R2, respectively. The detailed results of the LP relaxation of the LRP model are provided in Appendix G.

To see the effect of the spatial distribution of the customer locations, average results have been computed for each $dist_C$ values U, RU and C for the problems in R1 and R2, both separately and together. We have not concluded any evidence of a correlation between $dist_C$ and the solution quality or the solution time. Finally, in Table 6.17, it is shown that the solution quality as well the solution time deteriorate with the increasing number of depots. However, the result is also related to the fact that the problems with more customers also have more depots. The effect of the number of depots is examined on TB test problems in the next section.

ID	Z_{lb}	Z_{ub}	%GAP1	CPU(s)	Z_{Cplex}	%GAP2	%GAP3
R1-1	1075.58	1127.84	4.86	54.7	1127.84	0.10	0.00
R1-2	994.92	994.92	0.00	27.6	994.92	0.10	0.00
R1-3	805.16	909.72	12.99	45.0	909.72	0.10	0.00
R1-4	975.28	1027.14	5.32	53.5	1024.19	1.01	0.29
R1-5	1031.51	1074.68	4.19	48.5	1032.08	1.01	4.13
R1-6	1274.75	1291.96	1.35	29.2	1280.85	1.01	0.87
R1-7	1128.51	1148.72	1.79	38.2	1136.52	0.10	1.07
R1-8	1262.14	1317.96	4.42	96.7	1285.05	10.75	2.56
R1-9	1114.10	1138.72	2.21	158.1	1138.72	9.83	0.00
R1-10	1435.11	1442.48	0.51	155.1	1442.47	5.58	0.00
R1-11	953.04	1022.49	7.29	74.4	1442.47	5.57	-29.12
R1-12	1220.38	1220.48	0.00	52.5	1220.48	5.26	0.00
R1-13	1321.13	1402.44	6.15	261.6	1407.29	12.72	-0.34
R1-14	1244.53	1286.85	3.40	161.8	1271.85	15.98	1.18
R1-15	1204.01	1204.41	0.03	95.5	1210.09	11.52	-0.47
R1-16	1370.51	1418.18	3.48	241.8	1424.57	14.73	-0.45
R1-17	1367.03	1370.47	0.25	241.3	1368.62	18.05	0.14
R1-18	975.77	1029.39	5.50	223.9	1050.80	19.40	-2.04
R1-19	1471.51	1525.03	3.64	356.8	1629.90	28.58	-6.43
R1-20	1348.73	1441.94	6.91	640.8	1432.56	30.06	0.65
R1-21	1093.31	1198.73	9.64	514.1	1175.44	19.58	1.98
R1-22	1511.43	1543.86	2.15	232.6	1599.46	23.98	-3.48
R1-23	1555.81	1611.87	3.60	482.0	1619.42	26.79	-0.47
R1-24	1386.08	1442.53	4.07	392.9	1472.70	14.57	-2.05
R1-25	1735.65	1812.81	4.45	945.1	1909.93	31.26	-5.08
R1-26	1362.82	1386.02	1.70	285.2	1408.74	31.75	-1.61
R1-27	1140.06	1255.75	10.15	978.4	1289.18	26.58	-2.59
R1-28	1658.61	1801.37	8.61	682.3	1844.70	25.45	-2.35
R1-29	1556.13	1634.60	5.04	582.4	1730.64	31.43	-5.55
R1-30	1081.08	1185.35	9.64	420.4	1244.89	24.54	-4.78
averages	**1255.16**	**1308.96**	**4.27**	**285.7**	**1337.54**	**14.91**	**-1.80**

Table 6.11 Performance comparison between Cplex and [LR-TS] on the problems in R1

ID	Z_{lb}	Z_{ub}	%GAP1	CPU(s)
R2-1	1206.44	1428.56	18.41	706.1
R2-2	1549.84	1649.62	6.44	1006.6
R2-3	1903.06	2144.55	12.69	825.3
R2-4	1708.44	1877.78	9.91	1025.9
R2-5	1770.79	1947.74	9.99	938.0
R2-6	1897.92	2221.40	17.04	1052.4
R2-7	1532.22	1700.76	11.00	3031.9
R2-8	1772.49	1997.27	12.68	2797.8
R2-9	2112.81	2357.29	11.57	2114.9
R2-10	1778.51	1973.15	10.94	754.8
R2-11	1635.76	1990.79	21.70	1084.0
R2-12	2114.33	2306.43	9.09	1660.3
R2-13	1617.15	1921.78	18.84	3017.2
R2-14	2084.90	2336.22	12.05	5425.5
R2-15	2130.51	2735.82	28.41	1594.5
R2-16	2103.90	2200.74	4.60	3468.8
R2-17	1890.00	2339.18	23.77	1984.5
R2-18	2115.01	2751.57	30.10	1231.7
R2-19	1781.52	1996.10	12.04	6285.8
R2-20	2286.75	2642.04	15.54	4571.5
R2-21	2585.78	3076.79	18.99	8182.5
R2-22	1945.81	2190.70	12.59	1909.6
R2-23	2308.77	2670.56	15.67	5315.8
R2-24	2488.13	3050.53	22.60	4343.4
R2-25	1952.80	2184.96	11.89	8800.0
R2-26	2369.93	2893.82	22.11	11337.7
R2-27	2691.79	3320.89	23.37	12650.4
R2-28	2032.33	2448.95	20.50	5026.9
R2-29	2324.22	2896.34	24.62	14666.7
R2-30	2741.77	3258.81	18.86	11207.8
averages	**2014.46**	**2350.37**	**16.27**	**4267.3**

Table 6.12 Results obtained for test problems in R2

N_C	# *problems*	N_{Cplex}[1]	$N_{[LR\text{-}TS]}$[2]
15	6	3	
20	6	2	1
25	6	2	4
30	6	2	4
35	6	-	6

Table 6.13 Performance comparison for Cplex and [LR-TS] in terms of best feasible solution

N_C	# *problems*	N_{Cplex}[3]	$N_{[LR\text{-}TS]}$[4]
15	6	5	1
20	6	1	5
25	6	0	6
30	6	0	6
35	6	0	6

Table 6.14 Performance comparison for Cplex and [LR-TS] in terms of the gap between the best feasible and the best possible solution found

N_C	Avg_{Zlb}	Avg_{Zub}	Avg_{GAP1}	Avg_{CPU}	Avg_{ZCplex}	Avg_{GAP3}
15	1026.20	1071.04	4.79	43.06	1061.60	0.88
20	1185.55	1215.14	2.70	95.83	1277.62	-4.25
25	1247.16	1285.29	3.14	204.29	1288.87	-0.33
30	1394.48	1460.66	5.00	436.54	1488.25	-1.63
35	1422.39	1512.65	5.99	648.98	1571.35	-3.66
40	1672.75	1878.28	12.41	925.69	-	-
50	1824.35	2054.28	12.83	1907.28	-	-
60	1990.25	2380.89	19.63	2787.03	-	-
80	2232.79	2604.45	16.24	5101.44	-	-
100	2352.14	2833.96	20.23	10614.92	-	-

Table 6.15 Comparison of aggregated results for each N_C value

[1] *Number of problems in R*1 *that best feasible objective value found by Cplex outperforms that of [LR-TS]*
[2] *Number of problems in R*1*that best feasible objective value found by [LR-TS] outperforms that of Cplex*
[3] *Number of problems in R*1*that %GAP*2 *is lower than %GAP*1
[4] *Number of problems in R*1*that %GAP*1 *is lower than %GAP*2

$dist_C$	Avg_{Zlb}	Avg_{Zub}	Avg_{GAP1}	Avg_{CPU}	Avg_{Zcplex}	Avg_{GAP3}
R1-U	1368.33	1424.99	4.10	302.2	1454.69	-1.68
R1-RU	1267.67	1314.18	3.68	264.1	1358.64	-2.81
R1-C	1129.47	1187.70	5.10	291.0	1199.29	-0.91
R2-U	2278.11	2722.41	19.27	4486.3	-	-
R2-RU	1999.35	2336.36	16.46	4912.8	-	-
R2-C	1765.91	1992.35	13.07	3402.7	-	-
U	1823.22	2073.70	11.68	2394.2	-	-
RU	1633.51	1825.27	10.07	2588.4	-	-
C	1447.69	1590.03	9.09	1846.9	-	-

Table 6.16 Comparison of aggregated results for each $dist_C$ value

N_D	Avg_{Zlb}	Avg_{Zub}	Avg_{GAP2}	Avg_{CPU}
2	1093.85	1131.26	3.62	43.7
3	1139.32	1179.07	3.91	103.1
4	1481.69	1583.40	6.13	490.7
5	1642.18	1844.67	11.58	1463.8
6	2120.36	2483.45	16.60	6398.9

Table 6.17 Comparison of aggregated results for each N_D value

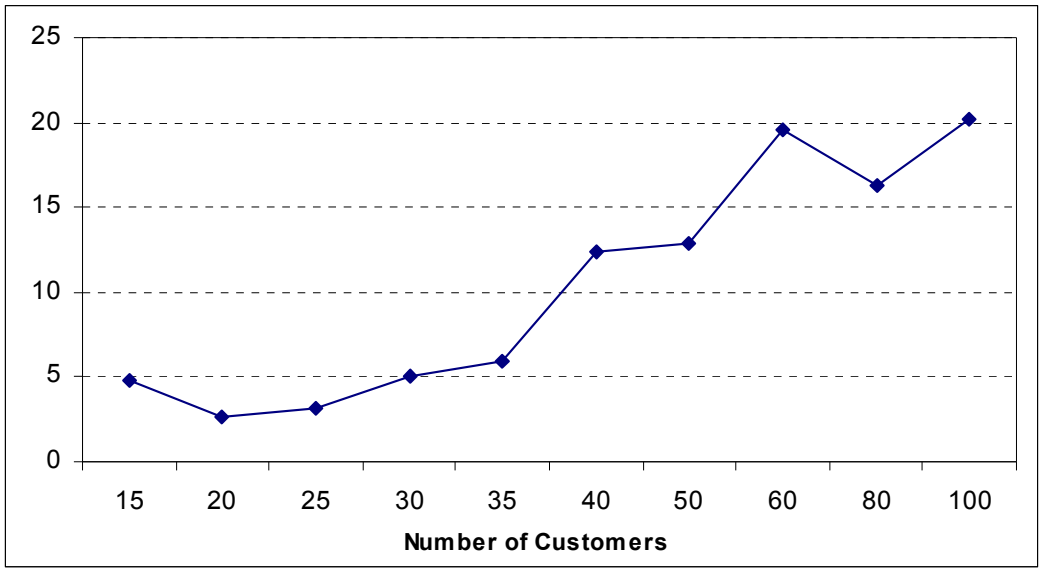

Figure 6.1 %GAP1 versus the number of customers in the problem

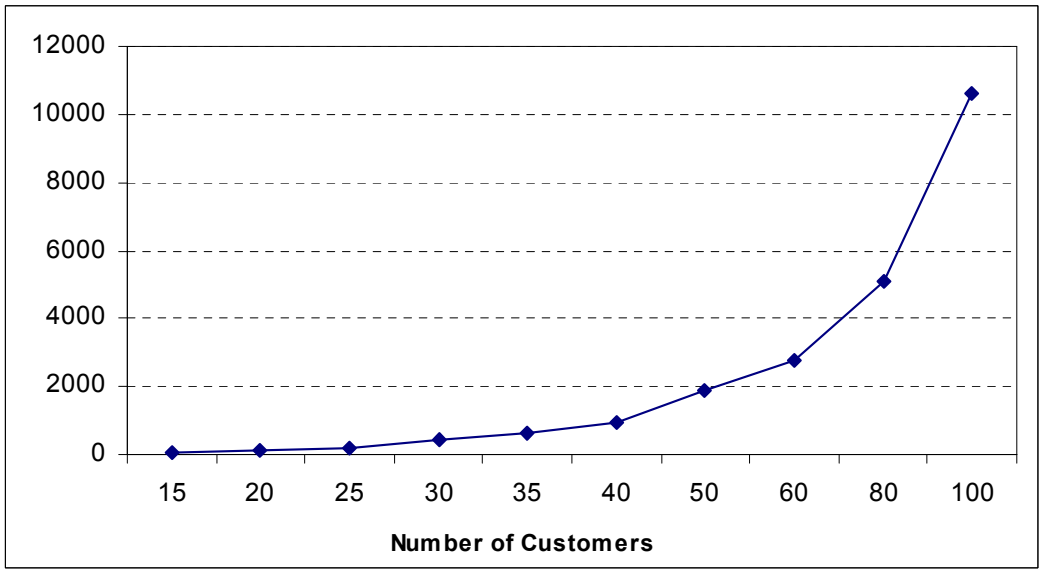

Figure 6.2 CPU time elapsed (sec) versus the number of customers in the problem

6.6. Results for Tüzün-Burke Instances

Tüzün and Burke [23] propose a two-phase tabu search algorithm to solve the 2-layer (single echelon) MDLRP where depots have unlimited throughput capacities. They provide their solutions to 36 benchmark LRP instances and compare their solutions with the SAV1 heuristic of Srivastava [22]. In order to compare the solution quality of [LR-TS] with the benchmark solutions in the recent LRP literature, we solve the 36 LRP instances of Tüzün and Burke [23] with [LR-TS]. Since no present depots exist in those test problems, N_D in this section denotes the total number of depots which comprise only candidate ones.

N_C	N_D	*# problems*
100	10	6
100	20	6
150	10	6
150	20	6
200	10	6
200	20	6
TOTAL		36

Table 6.18 The main characteristics of the 36 LRP instances

The [LR-TS] results for each benchmark problem are provided in Table 6.19 along with the solutions found by Tüzün and Burke. The gap between Z_{ub} and Z_{lb} is denoted by %GAP1 and calculated with the formula given in Equation (6.1). %GAP2 represents the gap between Z_{ub} and Tüzün and Burke's solutions (Z_{TB}). It is obtained as shown in Equation (6.4). If the gap is negative it means [LR-TS] has achieved a better solution than the one provided in Tüzün and Burke [23].

$$\%GAP2 = 100 \times \frac{Z_{ub} - Z_{TB}}{Z_{TB}} \tag{6.4}$$

[LR-TS] updates Tüzün and Burke's solutions for 31 out of 36 problems. The margin between our and their solutions which is represented by %GAP2 ranges between 3.83% and -11.95%. On the average, Tüzün and Burke results are improved by an average margin of 3.75%. However, solutions times of [LR-TS] are significantly higher in comparison with the CPU times of Tüzün and Burke's two phase tabu search method. Although [LR-TS] provides lower bounds along with better feasible solutions for the LRP, the solution times need to be improved appreciably.

%GAP1 which monitors the gap between the best upper bound and the best lower bound found by [LR-TS] increases as the number of customers goes up. On the average, the gap between Z_{ub} and Z_{lb} is 25.04% which is significantly higher than that of the problems in R1 and R2. The minimum %GAP1 is 9.93% which is obtained in an instance with 100 customers and 20 depots. The maximum %GAP1 which is 38.24% is obtained in an instance with 150 customers and 20 depots. In order to evaluate the contribution of the lower bound we provided, an alternative lower bound is obtained by the linear programming relaxation of the LRP model. For this purpose, we have employed the LRP model with 2-index formulation which is shown in Appendix F. The reasoning of using the formulation is provided in 6.6. as well as Appendix F. The percentage gap between the LP bound and Z_{ub} ($\%GAP_{LP}$)is compared with %GAP1. $\%GAP_{LP}$ values ranges between 28.46% and 107.03% with an average of 58.82%.

In order to monitor the impact of the number of customers and the number of depots to the performance of the proposed solution method, the results are aggregated in Table 6.20. The charts in Figure 6.3 to Figure 6.9 illustrate the changes of %GAP1, %GAP2, CPU times elapsed in [LR-TS] in connection with the number of customers and the number of depots. Examining Table 6.20 and the charts closely, the following conclusions can be derived:

1. As the number of customers in the problem increases, %GAP1 deteriorates. Slightly better gaps are obtained for the problems with less number of depots, but the increase in the number of depots does not appear to be as prominent as that of the number of customers. Since the location-allocation problem part of the LRP is solved by Cplex to optimality, the changes in the number of customers seems to be much more liable for the complexity of the overall solution method.
2. Neither the number of customers nor the number of depots shows a consistent impact on %GAP2. We have not observed dependable evidence directing to the impact of N_C or N_D on the gap between the best upper bound generated by [LR-TS] and Tüzün and Burke's solutions.
3. The CPU time of [LR-TS] goes up with the number of customers which is expected due to growing problem complexity. However, the solution times are not affected by the number of depots as predicted. A slight decrease in the CPU time is observed as the number of depots increases.

ID	N_D	N_C	Z_{lb}	Z_{ub}	%GAP1	CPU(s)	Z_{TB}	%GAP2	CPU(s)
P111112	100	10	1283.09	1417.30	10.46	19875.3	1556.64	-8.95	5
P111122	100	20	1178.19	1410.04	19.68	10554.9	1531.88	-7.95	3
P111212	100	10	1140.54	1406.33	23.30	9562.8	1443.43	-2.57	3
P111222	100	20	1186.54	1464.84	23.45	16420.2	1511.39	-3.08	4
P112112	100	10	1079.16	1209.88	12.11	14444.0	1231.11	-1.72	4
P112122	100	20	925.16	1019.44	10.19	18333.1	1132.02	-9.95	2
P112212	100	10	627.05	726.48	15.86	7158.2	825.12	-11.95	3
P112222	100	20	541.66	738.34	36.31	15391.9	740.64	-0.31	3
P113112	100	10	1069.98	1296.04	21.13	16432.6	1316.98	-1.59	3
P113122	100	20	1055.33	1160.09	9.93	12327.2	1274.50	-8.98	4
P113212	100	10	753.37	908.79	20.63	6190.9	920.75	-1.30	4
P113222	100	20	780.93	929.22	18.99	11696.9	1042.21	-10.84	3
P131112	150	10	1561.25	1869.43	19.74	52546.7	2000.97	-6.57	12
P131122	150	20	1465.80	1899.42	29.58	54043.2	1892.84	0.35	12
P131212	150	10	1589.11	2099.50	32.12	43472.2	2022.11	3.83	14
P131222	150	20	1438.10	1807.63	25.70	55900.3	1854.97	-2.55	13
P132112	150	10	1151.67	1488.29	29.23	42149.1	1555.82	-4.34	9
P132122	150	20	1144.07	1502.16	31.30	59226.1	1478.80	1.58	12
P132212	150	10	959.29	1234.50	28.69	26122.6	1231.34	0.26	9
P132222	150	20	742.16	938.22	26.42	69757.7	948.28	-1.06	9
P133112	150	10	1232.78	1667.65	35.28	10469.4	1762.45	-5.38	9
P133122	150	20	1051.04	1452.97	38.24	32540.3	1488.34	-2.38	9
P133212	150	10	930.82	1173.29	26.05	55394.5	1264.63	-7.22	10
P133222	150	20	973.35	1189.44	22.20	26393.2	1182.28	0.61	9
P121112	200	10	1747.10	2337.60	33.80	107893.1	2379.47	-1.76	22
P121122	200	20	1639.88	2176.88	32.75	75101.7	2211.74	-1.58	22
P121212	200	10	1800.51	2144.31	19.09	144487.6	2288.17	-6.29	23
P121222	200	20	1683.70	2303.29	36.80	122279.4	2355.81	-2.23	26
P122112	200	10	1591.88	2011.02	26.33	188714.8	2158.60	-6.84	20
P122122	200	20	1320.11	1757.52	33.13	206415.4	1787.02	-1.65	18
P122212	200	10	1079.33	1484.87	37.57	74098.2	1549.79	-4.19	18
P122222	200	20	1001.98	1094.71	9.25	76432.0	1112.96	-1.64	18
P123112	200	10	1576.96	2009.21	27.41	72359.7	2056.11	-2.28	23
P123122	200	20	1433.07	1885.89	31.60	130101.3	2002.42	-5.82	20
P123212	200	10	1498.26	1783.77	19.06	159535.9	1877.30	-4.98	20
P123222	200	20	1064.47	1362.84	28.03	61384.5	1414.83	-3.67	17
averages			**1202.71**	**1510.03**	**25.04**	**58478.1**	**1566.77**	**-3.75**	**12**

Table 6.19 Performance comparison of [LR-TS] and TS of Tüzün and Burke [23]

N_D	N_C	Z_{lb}	Z_{ub}	%GAP1	CPU(s)	Z_{TB}	%GAP2	CPU(s)
100	10	992.20	1160.80	17.25	12277.3	1215.67	-4.68	3.7
100	20	944.64	1120.33	19.76	14120.7	1205.44	-6.85	3.2
150	10	1237.49	1588.78	28.52	38359.1	1639.55	-3.24	10.5
150	20	1135.75	1464.97	28.91	496435	1474.25	-0.58	10.7
200	10	1549.01	1961.80	27.21	124514.9	2051.57	-4.39	21.0
200	20	1357.20	1763.52	28.59	111952.4	1814.13	-2.77	20.2
averages		**1202.71**	**1510.03**	**25.04**	**58478.0**	**1566.77**	**-3.75**	**11.5**

Table 6.20 Aggregated results for each N_C and N_D combination

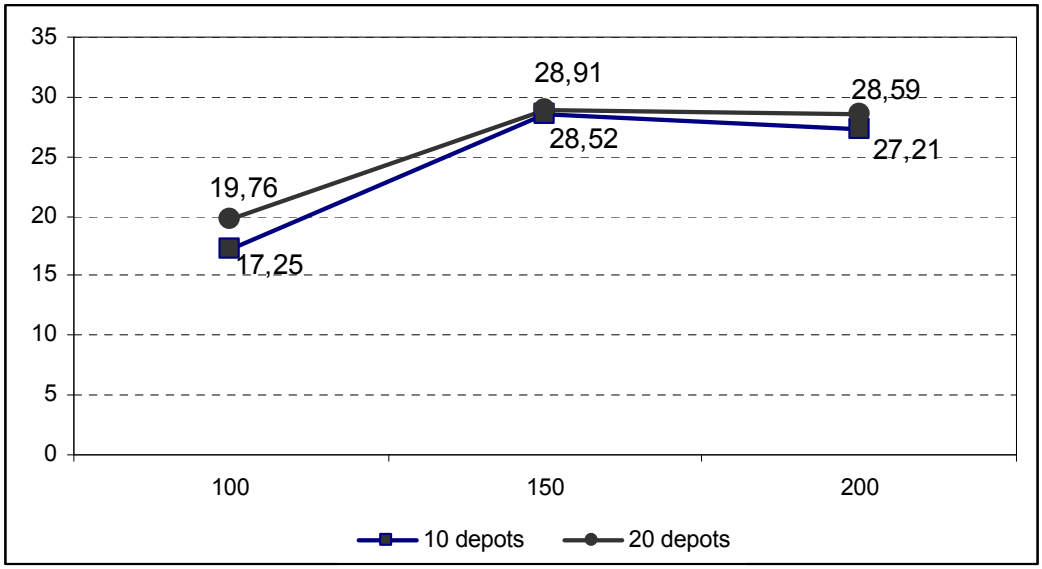

Figure 6.3 %GAP1 vs N_C for two different N_D values

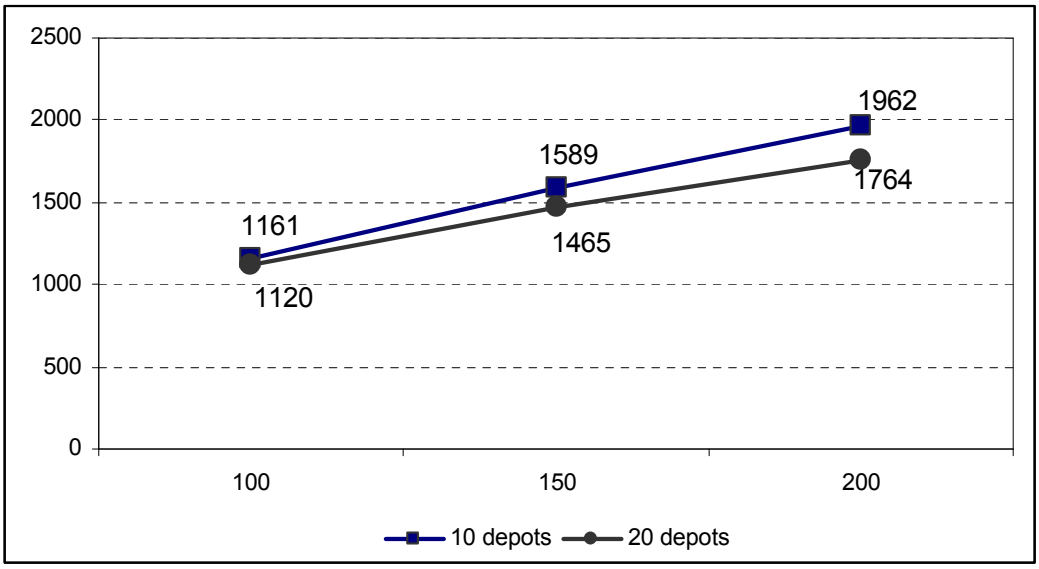

Figure 6.4 CPU time elapsed by [LR-TS](sec) vs N_C for two different N_D values

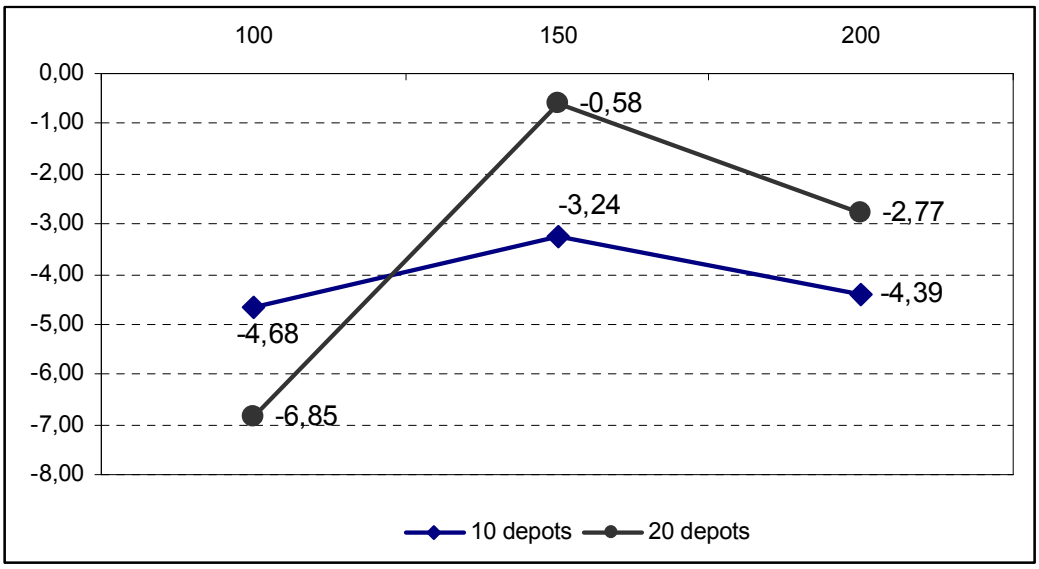

Figure 6.5 %GAP2 vs N_C for two different N_D values

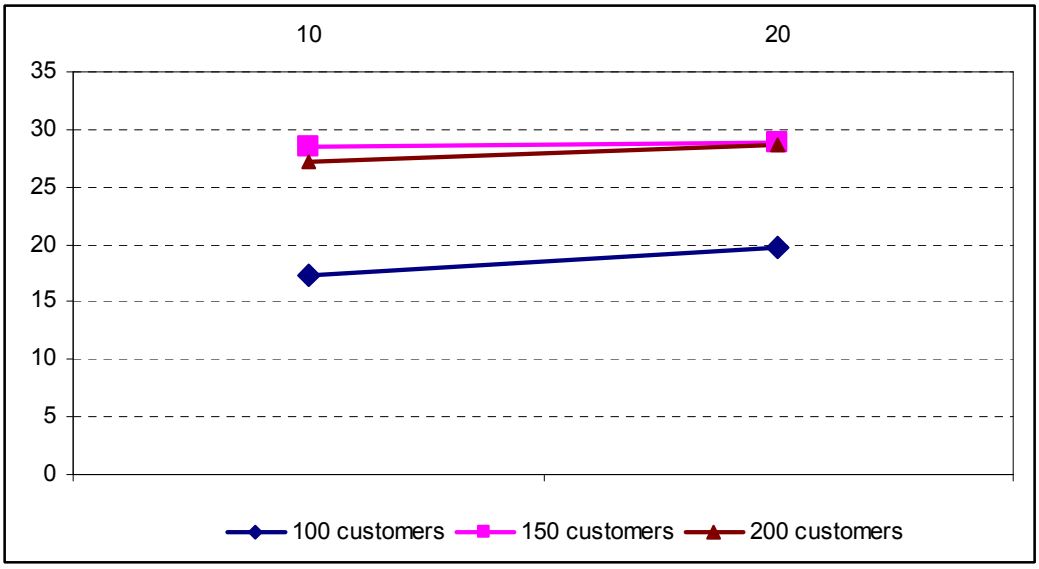

Figure 6.6 %GAP1 vs N_D for three different N_C values

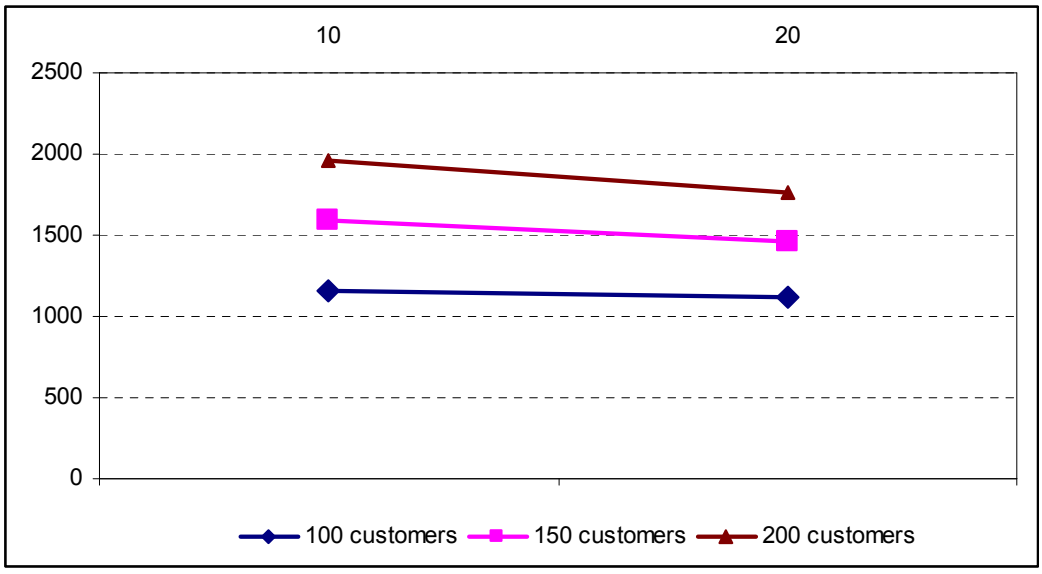

Figure 6.7 CPU time elapsed by [LR-TS](sec) vs N_D for three different N_C values

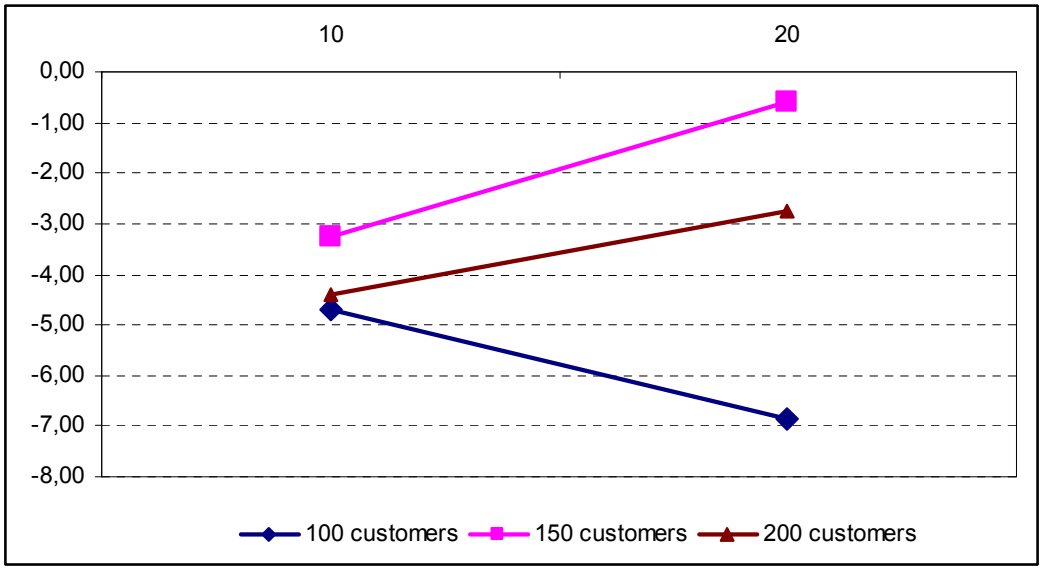

Figure 6.8 %GAP2 vs N_D for three different N_C values

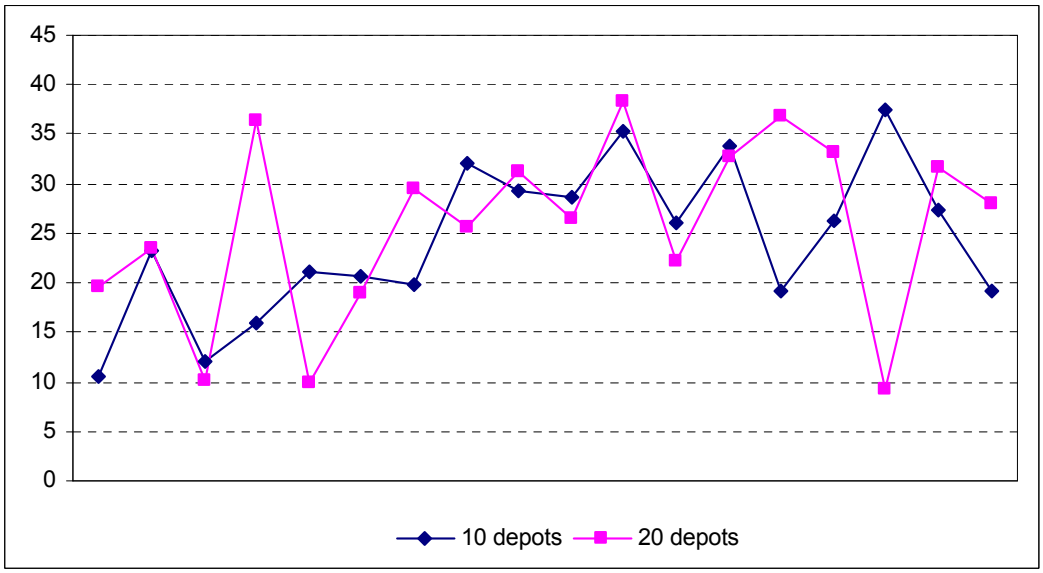

Figure 6.9 %GAP2 for all of the TB problems grouped by N_D

6.7. The Computational Time Elapsed by TS and LR Parts of [LR-TS]

The experimentation on 96 test problems revealed that the solution time of the proposed algorithm needs to be improved. In order to figure out possibilities of shortening the computational time, we have examined the computational time in detail. In Tables 6.21–6.23, $CPU_{[LR\text{-}TS]}$ denotes the solution time of the proposed method [LR-TS]. The total times consumed by the tabu search and by the Lagrangian relaxation part of the algorithm are denoted by CPU_{TS} and CPU_{LR}, respectively. The former includes also the solution time of the Add-Drop heuristic. The latter, namely CPU_{LR} comprises the time for inner and outer Lagrangian relaxations as well as the time for the solution of **SubP1** with [PFIH-NN] heuristic whenever the inner Lagrangian relaxation starts. The proportion of CPU time due to tabu search is given in column $\%CPU_{TS}$. Finally, CPU_{TB} in Table 6.23 represents the solution times of Tüzün and Burke [23].

For the R1 problems, the tabu search comprises 47.4% of the solution time in the average. As the problem size increases the time elapsed by the LR part increases significantly, and the average $\%CPU_{TS}$ drops to14.1% in R2 problems. In the case of Tüzün and Burke's [23] problems, only 8.1% of the solution time is used for tabu search. Examining the values in the tables, it is observed that the significant increase in $CPU_{[LR\text{-}TS]}$ is due to the Lagrangian relaxation part of the solution method. In order to find out the potential improvement points in terms of the solution time, the complete Lagrangian relaxation procedure is reviewed. We see that at each iteration of the inner Lagrangian relaxation a DCMSF-like problem is solved with the algorithm [MSF-ALR]. We perform at most 300 outer subgradient iterations where at each of them a maximum of 150 inner subgradient iterations are executed. As a result, the algorithm [MSF-ALR] is executed at most 45,000 times throughout the complete solution method. Note that our [MSF-ALR] implementation has an order of time complexity equal to $O(N_C^2(N_C + N_D))$ which is significantly high.

We believe that a better implementation of the [MSF-ALR] algorithm may achieve less order of complexity, and this could possibly improve the CPU times of the

proposed solution method. Furthermore, CPU_{TS} is not comparable with CPU_{TB}, which points out the need for improvement also in the solution times of our tabu search for the MDVRP.

ID	$CPU_{[LR-TS]}$	CPU_{LR}	CPU_{TS}	$\%CPU_{TS}$
R1-1	54.7	23.0	31.7	57.9
R1-2	27.6	2.2	25.3	91.9
R1-3	45.0	2.0	43.0	95.6
R1-4	53.5	15.0	38.5	71.9
R1-5	48.5	17.8	30.6	63.2
R1-6	29.2	3.9	25.4	86.8
R1-7	38.2	7.4	30.7	80.6
R1-8	96.7	20.4	76.2	78.9
R1-9	158.1	67.5	90.6	57.3
R1-10	155.1	125.9	29.2	18.8
R1-11	74.4	25.1	49.4	66.3
R1-12	52.5	14.2	38.3	72.9
R1-13	261.6	192.9	68.7	26.3
R1-14	161.8	114.1	47.6	29.4
R1-15	95.5	42.5	53.0	55.5
R1-16	241.8	138.3	103.6	42.8
R1-17	241.3	183.5	57.8	23.9
R1-18	223.9	80.2	143.6	64.2
R1-19	356.8	237.6	119.2	33.4
R1-20	640.8	576.4	64.3	10.0
R1-21	514.1	340.7	173.4	33.7
R1-22	232.6	121.8	110.8	47.6
R1-23	482.0	394.3	87.7	18.2
R1-24	392.9	282.3	110.7	28.2
R1-25	945.1	799.1	146.0	15.4
R1-26	285.2	220.9	64.3	22.6
R1-27	978.4	783.9	194.6	19.9
R1-28	682.3	454.3	228.0	33.4
R1-29	582.4	452.6	129.8	22.3
R1-30	420.4	193.2	227.2	54.0
averages	**285.7**	**197.8**	**88.0**	**47.4**

Table 6.21 Computational times for R1 problems

ID	$CPU_{[LR-TS]}$	CPU_{LR}	CPU_{TS}	$\%CPU_{TS}$
R2-1	706.1	477.7	228.3	32.3
R2-2	1006.6	864.8	141.8	14.1
R2-3	825.3	595.5	229.8	27.8
R2-4	1025.9	895.7	130.1	12.7
R2-5	938.0	799.8	138.2	14.7
R2-6	1052.4	903.2	149.1	14.2
R2-7	3031.9	2627.6	404.3	13.3
R2-8	2797.8	2609.7	188.1	6.7
R2-9	2114.9	1806.1	308.8	14.6
R2-10	754.8	540.7	214.2	28.4
R2-11	1084.0	897.1	186.9	17.2
R2-12	1660.3	1291.4	368.9	22.2
R2-13	3017.2	2717.8	299.4	9.9
R2-14	5425.5	5097.7	327.8	6.0
R2-15	1594.5	1315.8	278.7	17.5
R2-16	3468.8	3223.6	245.1	7.1
R2-17	1984.5	1735.5	249.0	12.5
R2-18	1231.7	931.2	300.5	24.4
R2-19	6285.8	5602.8	683.1	10.9
R2-20	4571.5	4194.3	377.2	8.3
R2-21	8182.5	7671.5	511.0	6.2
R2-22	1909.6	1319.3	590.3	30.9
R2-23	5315.7	4788.1	527.6	9.9
R2-24	4343.4	3719.0	624.4	14.4
R2-25	8800.0	8116.9	683.1	7.8
R2-26	11337.7	10648.2	689.4	6.1
R2-27	12650.4	11599.7	1050.7	8.3
R2-28	5026.9	4466.6	560.2	11.1
R2-29	14666.7	14025.5	641.2	4.4
R2-30	11207.9	10208.7	999.2	8.9
averages	**4267.3**	**3856.4**	**410.9**	**14.1**

Table 6.22 Computational times for R2 problems

ID	$CPU_{[LR-TS]}$	CPU_{LR}	CPU_{TS}	$\%CPU_{TS}$	CPU_{TB}
P111112	19875.3	18331.6	1543.7	7.8	5
P111122	10554.9	8313.1	2241.8	21.2	3
P111212	9562.8	8419.8	1143.0	12.0	3
P111222	16420.2	14839.3	1580.9	9.6	4
P112112	14444.0	13585.3	858.6	5.9	4
P112122	18333.1	15737.3	2595.8	14.2	2
P112212	7158.2	6114.5	1043.7	14.6	3
P112222	15391.9	13899.7	1492.3	9.7	3
P113112	16432.6	15490.0	942.6	5.7	3
P113122	12327.2	9707.0	2620.2	21.3	4
P113212	6190.9	5260.2	930.7	15.0	4
P113222	11696.9	9725.7	1971.2	16.9	3
P131112	52546.7	49267.2	3279.5	6.2	12
P131122	54043.2	49999.1	4044.2	7.5	12
P131212	43472.2	41597.7	1874.4	4.3	14
P131222	55900.3	51380.0	4520.3	8.1	13
P132112	42149.1	40270.5	1878.7	4.5	9
P132122	59226.1	56162.1	3064.0	5.2	12
P132212	26122.6	24682.9	1439.7	5.5	9
P132222	69757.7	67392.3	2365.4	3.4	9
P133112	10469.4	8791.9	1677.5	16.0	9
P133122	32540.3	29404.3	3136.0	9.6	9
P133212	55394.5	53388.9	2005.6	3.6	10
P133222	26393.2	24038.6	2354.6	8.9	9
P121112	107893.1	103408.8	4484.4	4.2	22
P121122	75101.7	65697.0	9404.7	12.5	22
P121212	144487.6	141462.7	3024.9	2.1	23
P121222	122279.4	115925.1	6354.3	5.2	26
P122112	188714.8	185395.7	3319.1	1.8	20
P122122	206415.4	201249.3	5166.1	2.5	18
P122212	74098.2	71898.2	2200.1	3.0	18
P122222	76432.0	72377.5	4054.5	5.3	18
P123112	72359.7	69992.9	2366.8	3.3	23
P123122	130101.3	122222.7	7878.6	6.1	20
P123212	159535.9	157170.3	2365.6	1.5	20
P123222	61384.5	55929.3	5455.3	8.9	17
averages	**58478.1**	**55514.67**	**2963.29**	**8.1**	**12**

Table 6.23 Computational times for TB problems

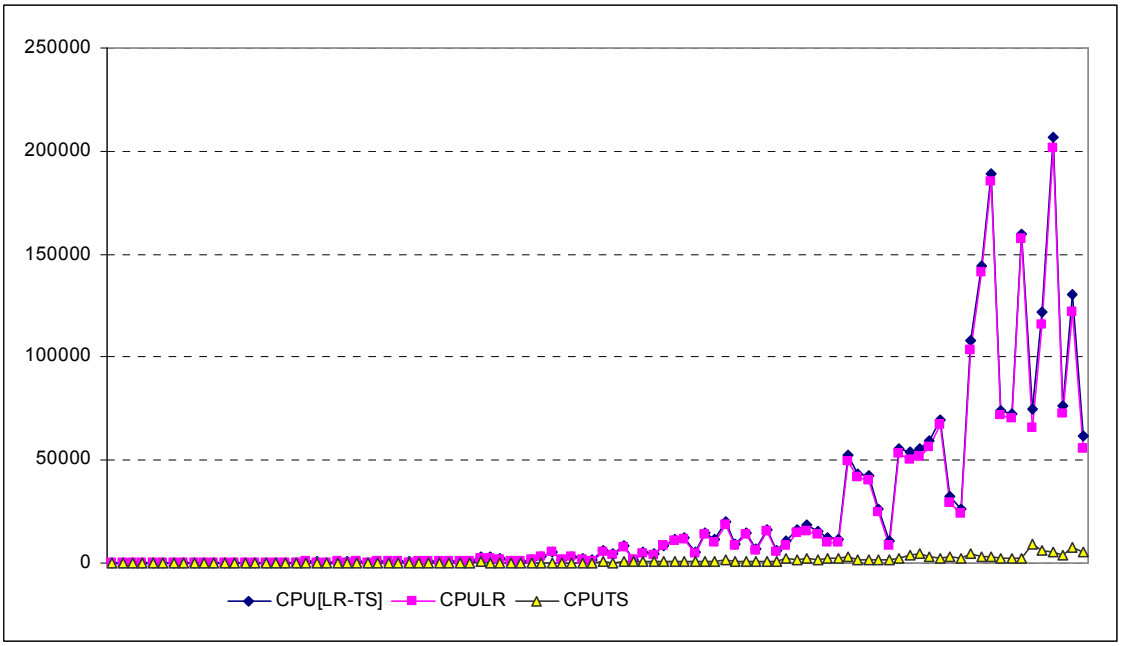

Figure 6.10 Change in $CPU_{[LR-TS]}$, $CPU_{[LR]}$ and $CPU_{[TS]}$ for 96 test problems*

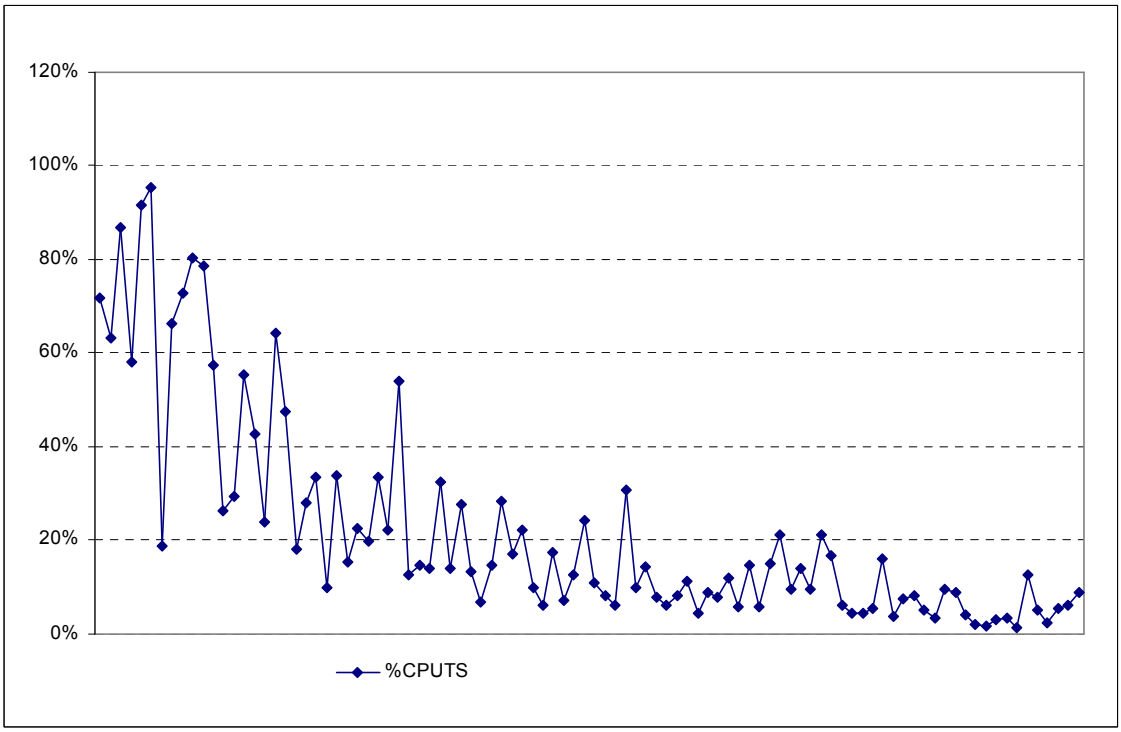

Figure 6.11 Change in % $CPU_{[TS]}$ for 96 test problems[1]

[1] *The problems are sorted first in increasing number of customers, secondly in increasing number of depots*

Chapter 7

CONCLUSION

In this thesis, we have studied a 2-layer (single echelon) discrete uncapacitated multi depot location routing problem (MDLRP). The problem that we are dealing with takes its motivation from real life problems faced by local logistics companies. They operate a number of depots some of which are their own property while the others have been rented by them. At the beginning of certain periods, or when they sign new contracts with the customers on their portfolio, they revise their operating depots. According to the locations and demands of their customers they may decide to hire out their own depots, to leave some of the depots they have rented before, or to rent new ones. Therefore, there exist two kinds of depots in the problem: present and candidate. Present depots are already operating facilities that can be preserved or closed. Candidate depot locations are potential sites in which a new depot can be opened. The problem involves determining which present depots to preserve and which candidate depots to open while allocating customers to depots and designing the vehicle routes to visit all customers.

We propose a nested Lagrangian relaxation based method named [LR-TS] to solve the MDLRP. An outer Lagrangian relaxation embedded in subgradient optimization decomposes the parent problem into two subproblems. The first subproblem **SubP1**, which is a facility location-like problem, is solved to optimality with Cplex 10.0. The solution times are generally reasonable. A problem instance with 20 depots and 1000 customers takes 2.84 seconds on a present-day desktop PC. The second subproblem, namely **SubP2**, resembles a capacitated and degree constrained minimum spanning forest problem. It is tackled with an augmented Lagrangian relaxation. In the

augmented Lagrangian relaxation that we apply to **SubP2,** the subtour elimination constraints are generated and dualized only as they are violated by the current iteration's Lagrangian solution. This procedure is preferred to relaxing all of them right away at the construction stage of the Lagrangian problem. $\boldsymbol{ALR^{SubP2}}$, namely the Lagrangian relaxed problem of **SubP2** resembles a degree-constrained minimum spanning tree problem. It is solved by a modified version of Prim's minimum spanning tree algorithm. This algorithm, called [MSF-ALR], was first proposed by Aksen and Altinkemer [26]. Some modifications are made to the algorithm in the way they tackle the additional constraints of the ALR^{SubP2} which do not exist in the classical degree-constrained minimum spanning forest problem. The sum of the objective value of the Cplex solution of **SubP1** and the lower bound found for **SubP2** by the subgradient optimization scheme in the augmented Lagrangian relaxation constitutes a lower bound to the true optimal solution of the comprehensive problem **P**.

Besides finding a lower bound, [LR-TS] generates a good feasible heuristic solution the objective value of which makes up an upper bound for **P**. The feasible solution is built using a tabu search heuristic implanted in the Lagrangian relaxation procedure. The tabu search procedure for the MDVRP is designed by tailoring the OTS (Open Tabu Serach) proposed by Aksen et al. [42] for the open vehicle routing problem with fixed driver nodes. The procedure is also enriched with additional neighborhood generation moves. The feasible heuristic solution to **P** is generated by solving an MDVRP with tabu search as soon as a new location plan is revealed by the solution of **SubP1**.

The proposed solution method [LR-TS] is first tested on 60 randomly generated instances (30 small size instances designated as R1 and 30 large size instances designated as R2). Subsequently, it has been tested on the 36 LRP instances solved in Tüzün and Burke [23]. We denote the set of these instances as TB. The parameters of [LR-TS] are fine tuned in accordance with the results of experiments conducted on a subset of the test problems. The performance of [LR-TS] is evaluated based on two outcomes. First is %GAP1 which is the gap between the best lower bound and the best

upper bound (Z_{lb}, Z_{ub}) obtained by [LR-TS]. The average %GAP1 is 4.27% for the smaller test problems. For 28 of the problems %GAP1 is below 10%. The gaps of the problems in R2 are higher where the average %GAP1 reaches 16.27% for this part of the test problems. For the problems in R1, the final gap is also compared with the gap between the best feasible and the best possible solution found by Cplex. We have observed that for 24 out of 30 problems [LR-TS] has come up with better gaps than Cplex. The quality of %GAP1 deteriorates for the TB due to growing problem size where the average gap is 25%. The behavior of the final gap changing with problem parameters is examined, and some conclusions are derived. As the number of customers in the problem goes up, the quality of the gap found by [LR-TS] gets worse while the solution times get longer. On the other hand, the performance of the [LR-TS] against Cplex improves with increasing number of customers. No significant effect of the spatial distribution of customer locations has been observed on the tested problems. Alternative lower bounds for the test problems are found by solving the linear relaxation of the complete LRP model to optimality with Cplex. The gap between the LP bound and the best feasible solution of [LR-TS] is compared with %GAP1 values. The LP bounds do not match the lower bounds found by [LR-TS] in any of the test problems. The average gaps are 31.79%, 50.05% and 58.82%, for R1, R2 and TB, respectively.

The second outcome that is utilized to assess the performance of [LR-TS] is the objective function value of the best feasible solution achieved, namely Z_{ub}. For randomly generated test problems in R1, Z_{ub} is compared with the respective Cplex solution while Tüzün and Burke's solutions constitute benchmarks for the instances in TB. On the average, Cplex solutions have been improved by 1.8%. For 15 test problems out of 30, [LR-TS] comes up with a better feasible objective value than Cplex. In case of TB instances [LR-TS] updates the best feasible objective value in 31 test problems out of 36. On the average, their solutions are improved by a margin of 3.75%. The effect of increasing number of customers on the gap between Z_{ub} and the benchmark solution changes with the benchmarked method. The number of problems where [LR-

TS] finds better solutions than Cplex increases with the number of customers in the problem. However, we have not been able to observe a significant effect of N_C on the gap between our solutions and the solutions provided by Tüzün and Burke. The outcomes for TB problems are also examined in order to evaluate the impact of the number of depots in the problem. %GAP1 decreases when the number of depots increases and the number of customers is kept constant. N_D does not seem to have a consistent effect on the gap between Z_{ub} and the respective solution found by Tüzün and Burke.

The proposed method [LR-TS] provides better feasible objective values for 70% of the test problems in comparison with the benchmarked solution method. The main disadvantage of [LR-TS] is its extremely high solution times. Although [LR-TS] accomplishes to find a lower bound on the true optimal objective value of the problem besides a heuristic solution, its CPU solution time needs to be improved in order to compete with other methods. For problems with less number of customers, favorable gaps are obtained while the gaps deteriorate considerably with growing problem size.

The results of the computer experiments not only assess the performance of [LR-TS], but also point to new research directions. The next step would be solving the MDLRP with time windows. This type of time restrictions is a crucial quality of service (QoS) guarantee promised more and more often to customers in distribution logistics. Finally, long solution times especially for problems with a customer number above 100 are a severe disadvantage of the proposed method. This might be overcome by a new implementation of the modified Prim's algorithm which is used for the Lagrangian relaxed subproblem ALR^{SubP2}.

BIBLIOGRAPHY

[1] S. Salhi and G.K. Rand, The Effect of Ignoring Routes When Locating Depots, European Journal of Operational Research, 39 (1989), 150–156.

[2] G. Nagy and S. Salhi, Location-Routing: Issues, Models and Methods, European Journal of Operational Research, 177 (2007), 649–672.

[3] A. Balakrishnan, J.E. Ward and R.T. Wong, Integrated Facility Location and Vehicle Routing Models: Recent Work and Future Prospects, Management Sciences, 7 (1987), 35–61.

[4] G. Laporte, A Survey of Algorithms for Location-Routing Problems, Investigación Operativa, 1 (1989), 93–123.

[5] H. Min, V. Jarayaman and R. Srivastava, Combined Location-Routing Problems: A Synthesis and Future Research Directions, 108 (1998), 1–15.

[6] S.D. Ahipaşaoğlu, G. Erdoğan and B. Tansel (2003), Location-Routing Problems: a Review and Assessment of Research Directions, Working Paper IEOR 2003-07, Department of Industrial Engineering, Bilkent University: Ankara, Türkiye.

[7] G. Laporte, Location-routing problems, in B. L. Golden and A. A. Assad , editor, Vehicle Routing: Methods and Studies, 163–198. North-Holland Publishing, Amsterdam, Holland, (1988).

[8] G. Laporte and Y. Nobert, An Exact Algorithm for Minimizing Routing and Operating Costs in Depot Location, European Journal of Operational Research, 6 (1981), 224–226.

[9] G. Laporte, Y. Nobert and P. Pelletier, Hamiltonian Location Problems, European Journal of Operational Research, 12 (1983), 82–99.

[10] G. Laporte, Y. Nobert and S. Taillefer, Solving a Family of Multi-Depot Vehicle Routing and Location Routing Problems, Transportation Science, 22 (1988), 161–172.

[11] G. Laporte and P.J. Dejax, Dynamic Location-Routing Problems, Journal of Operational Research Society, 40 (1989) 471–482.

[12] Gezdur, Integrated Vehicle Routing and Warehouse Location Problem, M. Sc. Thesis, Graduate School of Sciences and Engineering, Industrial Engineering, Koç University, İstanbul, Türkiye (2003).

[13] J. Perl and M.S. Daskin, A Warehouse Location-Routing Problem, Transportation Research, 19B (1985), 381–396.

[14] T-H. Wu, C. Low and J-W. Bai, Heuristic Solutions to Multi-Depot Location-Routing Problems, Computers & Operations Research, 29 (2002), 1393–1415.

[15] P.H. Hansen, B. Hegedahl, S. Hjortkajær and B. Obel, A Heuristic Solution to the Warehouse Location-Routing Problem, European Journal of Operational Research, 25 (1986), 281-291.

[16] S. Boğ, Algorithms for the Vehicle Routing Problem with Time Windows and the Location-Routing Problem, Koç University, İstanbul, Türkiye (2006).

[17] S. S. Barreto, Analysis and Modeling of Location-Routing Problems. PhD. Thesis, Aveiro University, 2004

[18] S.K. Jacobsen and O.B.G. Madsen, A Comparative Study of Heuristics for a Two-Level Location-Routing Problem, European Journal of Operational Research, 5 (1980), 378–387.

[19] O.B.G. Madsen, Methods for Solving Combined Two Level Location-Routing Problems of Realistic Dimensions, European Journal of Operational Research, 12 (1983), 295–301.

[20] R. Srivastava, Algorithms for Solving the Location-Routing Problem, The Ohio State University, Ohio(1986).

[21] R. Srivastava and W.C. Benton, The Location-Routing Problem: Considerations in Physical Distribution System Design, Computers & Operations Research, 6 (1990), 427–435.

[22] R. Srivastava, Alternate Solution Procedures for the Location-Routing Problem, OMEGA, 21 (1993), 497-506.

[23] D. Tüzün and L. I. Burke, A Two-Phase Tabu Search Algorithm to the Location-Routing Problem, European Journal of Operational Research, 116 (1999), 87–99.

[24] M. Albareda-Sambola, J. A. Díaz and E. Fernández, A Compact Model and Tight Bounds for a Combined Location-Routing Problem, Computers & Operations Research, 32 (2005), 407–428.

[25] J. Melechovský, C. Prins and R. W. Calvo, A Metaheuristic to Solve a Location-Routing Problem with Non-linear Costs, Journal of Heuristics, 11 (2005), 375–391.

[26] D. Aksen and K. Altinkemer, A Location-Routing Problem for the Conversion to the 'Click-and-Mortar' Retailing: The Static Case, Working Paper, College of Administrative Science and Economics, Koç University, İstanbul, Türkiye (2005).

[27] G. Cornuejols, M. L. Fischer and G.L. Nemheuser, Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms, Management Science, 23 (1977), 789–810.

[28] R. Karp, Reducibility among Combinatorial Problems, in R. Miller and J. Thatcher, editors, Complexity of Computer Computations, Plenum Press, New York, USA, (1972), 85–104.

[29] J.K. Lenstra and A. H. G. Rinnooy Kan, Complexity of Vehicle Routing and Scheduling Problems, Networks, 11 (1981), 221–227.

[30] M. Geoffrion, Lagrangian Relaxation and Its Uses in Integer Programming, Mathematical Programming Study, 2 (1974), 82–114.

[31] N. Kohl and O. B. G. Madsen, An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation, Operations Research, 45 (1997), 395–406.

[32] B. Gavish, Augmented Lagrangian Based Algorithms for Centralized Network Design, IEEE Transactions on Communications, COM-33 (1985)., 1247–1257.

[33] M. Fisher, The Lagrangian Relaxation Method for Solving Integer Programming Problems, Management Science, 27 (1981), 1–18.

[34] D. Aksen and K. Altınkemer, Efficient Frontier Analysis and Heuristics for E-tailing Logistics, Working Paper, Purdue University, Krannert Graduate School of Management: West Lafayette, Indiana, USA, (2003).

[35] C. H. Papadimitriou, The Complexity of the Capacitated Tree Problem. Networks, 8 (1978), 217–230.

[36] E. Balas and N. Christofides, A Restricted Lagrangian Approach to the Traveling Salesman Problem, Mathematical Programming, 21 (1981), 19–46.

[37] M. Fisher, Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees, Operations Research, 42 (1994), 626–642.

[38] M. Fisher, K. O. Jörnsten and O. B. G. Madsen, Vehicle Routing with Time Windows: Two Optimization Algorithms, Operations Research, 45 (1997), 488–492.

[39] P. Toth, and D. Vigo, VRP with Time Windows, in: P. Toth and D. Vigo, editors, The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications: Philadelphia, USA, (2002), 29–51.

[40] G. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York, USA, (1979).

[41] F. Glover, Future Paths for Integer Programming and Links to Artificial Intelligence, Computers & Operations Research, 13 (1986), 533–549.

[42] D. Aksen, Z. Özyurt and N. Aras, Open Vehicle Routing Problem with Driver Nodes and Time Windows, available online in Journal of Operational Research Society, August 2006, (doi:10.1057/palgrave.jors.2602249).

[43] G. Clarke and J. W. Wright, Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, 12 (1964), 568–581

[44] S. K. Park and K. W. Miller, Random Number Generators: Good Ones are Hard to Find, Communications of the ACM, 31 (1988), 1192–1201.

[45] C.E. Miller, A.W. Tucker and R.A. Zemlin, Integer Programming Formulations and Traveling Salesman Problems, Journal of the Association for Computing Machinery, **7** (1960), 326–329.

[46] R.V. Kulkarni and P.R. Bhave, Integer Programming Formulations ForVehicle Routing Problems, European Journal of Operational Research, 20 (1985), 58–67.

[47] I. Kara, G. Laporte and T. Bektas, A note on the Lifted Miller-Tucker-Zemlin Subtour Elimination Constraints for the Capacitated Vehicle Routing Problem, European Journal of Operational Research, 158 (2004), 793-795.

Appendix A. [PFIH-NN] Heuristic for Generating a Feasible Solution for MDVRP

[PFIH-NN] heuristic is used to construct a feasible solution for Multi-depot Vehicle Routing Problem. In the proposed solution method it is used in three parts:

1. [PFIH-NN] is run for once at the beginning of the main Lagrangian iterations in order to find an initial value of the upper bound for P. All of the present depots are assumed to be in service, a MDVRP is solved for the remainder of the problem with [PFIH-NN]. In case there does not exist any present depots, the candidate depot minimizing the sum of Euclidian distances from customers to the depot is selected to be opened.
2. Each time the tabu search is triggered the initial solution is constructed using the [PFIH-NN] heuristic.
3. In order to find an upper bound for **SubP2**, a modified version of the [PFIH-NN] heuristic is utilized. **SubP2** has a three dimensional asymmetric cost matrix denoted as C_{new}. The third dimension of the cost matrix implies that the arc cost between two nodes is not only dependent on their locations in the solution space but also on the depot which they are connected to. So all the distance calculations in [PFIH-NN] heuristic for **SubP2** are modified to take the third dimension into account.

J	Set of depots
I	Set of customers
N_j	Subset of customers initially assigned to depot *j*
n_j	Cardinality of subset N_j
(x_k, y_k)	Coordinates of (customer or depot) *k*
α	Weight for the distance criterion
γ	Weight for the polar coordinate angle
θ_i	Polar coordinate angle of customer *i* with respect to the current depot
Q	Uniform vehicle capacity
VCj	Unit vehicle acquisition cost at depot *j*
$[dist_{ki}]$	Matrix of distances between the nodes of the problem
$[C_{jk}]$	Asymmetric matrix of traveling costs with vehicle acquisition costs embedded
ac	Cost coefficient used to convert miles traveled into dollars and cents
R_j	Set of routes that emanate from depot *j*
last(*r*)	Index of the customer node last visited on route *r* (the depot node for an empty route)
spare(*r*)	Spare capacity of the vehicle assigned to route *r*
$i_{[k]}$	Customer node *i* with k^{th} lowest cost value
d_i	Demand of customer *i*
TD_i	Time deadline for the completion of service at customer *i*
$Cost_i$	Cost value for customer node *i*
succ(*i*)	Successor of customer node *i* on the pertinent route
pred(*i*)	Predecessor of customer node *i* on the pertinent route

Table A.1 Notations and symbols used in the pseudo code of [PFIH-NN]

Algorithm [PFIH-NN]

Step 0. Establish the cost matrix $[C_{jk}]$ and embed vehicle acquisition costs into it, i.e. set: $C_{ji} := dist_{ji} + VC_j \quad \forall j \in \mathbf{J},\ \forall i \in \mathbf{I},\ \ C_{ik} :=\cdot dist_{ik} \ \forall i \in \mathbf{I},\ \forall k \in \mathbf{I} \cup \mathbf{J}$.
From now on, proximity judgments will be made on the basis of this cost matrix instead of the distance matrix.

Step 1. Assign each customer temporarily to the nearest depot and form the subsets $\mathcal{N}_j$.

Step 2. For $\forall j \in \mathbf{J} \ni \mathcal{N}_j \neq \{\}$ do:

Step 3. For $\forall i \in \mathcal{N}_j$ do:

Compute the polar coordinate angle of customer node i with respect to its depot j, i.e. set:

$$\theta_i = \arctan \frac{y_i - y_j}{x_i - x_j}.$$

Compute then the following cost expression:

$$Cost_i := -\alpha \cdot C_{ji} + \gamma \cdot \frac{\theta_i}{2\pi} \cdot C_{ji} \quad \text{with } \alpha = 0.7 \text{ and } \gamma = 0.3$$

Sort the $Cost_i$ values in ascending order to obtain the sorted set $\mathcal{N}_j = \{i_{[1]}, i_{[2]}, \ldots, i_{[nj]}\}$ for depot j. Set route index $r := 1$, $spare(r) := Q$, and $R_j = \{1\}$.

Step 4. Repeat until Nj = {}:

Step 4a. Select the first customer $i_{[1]} \in \mathcal{N}_j$ that is capacity feasible. Append $i_{[1]}$ to the current route r, set $\mathcal{N}_j := \mathcal{N}_j \setminus \{i_{[1]}\}$ and $i := i_{[1]}$ and update $spare(r) := spare(r) - d_i$.

Step 4b. For all unrouted customers u ($u \in \mathcal{N}_j$) and all edges $(k, l) \in r$ do:

Compute the cost of inserting unrouted customer u between nodes k and l as follows:

$$\text{INSERTION\_COST}(u, k, l) := C_{ku} + C_{ul} - C_{kl}$$

Step 4c. Match the "best" candidate node $u^* \in \mathcal{N}_j$ with the "best" candidate edge $(k^*, l^*) \in r$ such that the lowest INSERTION_COST is incurred

If there does not exist such node u^*, then set BEST_INSERTION_COST := ∞ and go to Step 4d.

O/w let BEST_INSERTION_COST := INSERTION_COST(u^*, k^*, l^*).

Step 4d. Consider another "best" candidate node $v \in \mathcal{N}_j$ according to the following metric: $v^* := \arg\min_{v \in \mathcal{N}_j} \{C_{last(r),\, v}\}$ ∋ $spare(r) \geq d_{v^*}$.

If there does not exist such node v^*, then set BEST_INSERTION_COST := ∞ and go to Step 4e.

O/w let BEST_NEAREST_NEIGHBOUR_COST := $C_{last(r),\, v^*}$

Step 4e. If there does not exist u^* AND v^*, then go to Step 4f.

O/w if BEST_NEAREST_NEIGHBOUR_COST < BEST_INSERTION_COST, then insert the edge ($last(r)$, v^*) to the end of route r. Update r, $spare(r) := spare(r) - d_{v^*}$, $last(r) := v^*$ and $\mathcal{N}_j := \mathcal{N}_j \setminus \{v^*\}$.

else insert the node u^* into the edge (k^*, l^*) between nodes k^* and l^*. Update r, $spare(r) := spare(r) - d_u$ and $\mathcal{N}_j := \mathcal{N}_j \setminus \{u^*\}$. Also update the service completion times for the nodes l^* and its successors on r.

Go to Step 4c to continue working on the same route r with the remaining nodes in $\mathcal{N}_j$.

Step 4f. If $\mathcal{N}_j = \{\}$ then set $j := j+1$ and go to Step 2.

O/w enumerate the leftover nodes in $\mathcal{N}_j$ such that the first node has index 1, and begin a new route from depot j. Set $r := r + 1$, $spare(r) := Q$ and $R_j := R_j \cap \{r\}$. Go to Step 4.

Appendix B. [MSF-ALR] Algorithm to Solve ALR^{SubP2}

S	Current set of connected nodes.
U	Current set of unconnected nodes.
A	The arcs set of the current MSF.
outdeg(*j*)	The outgoing degree of a given store node *j*, i.e. the number of arcs leaving store *j*.
indeg(*j*)	The incoming degree of a given store node *j*, i.e. the number of arcs entering store *j*.
depot(*i*)	The ID of the depot node from which customer node *i* is accessible.
parent(*i*)	The ID of the parent node from which an arc directly enters into node *i*.
Σ*outdeg*(*j*)	The current sum of all depots' outgoing degrees.
Min_out	Minimum number of outgoing arcs from depot nodes

Table A.2 Notations and symbols used in the pseudo code of [MSF-ALR]

Procedure [MSF-ALR]

Step 0. Using to the current vector of Lagrange multipliers α compute the matrix $C_{new} = \left[c_{ijk}^{new} \right]$ as explained in 4.2.

Step 1. Put each depot node $j \in ID$ into **S**. They will be the roots of potential trees in the MSF. Put each customer node $i \in I$ into **U**. Initialize each customer node's *parent*(*i*) and *depot*(*i*) as zero. Set each depot node's *outdeg*(*j*) and *indeg*(*j*) to zero. Set the adjacency list of each node in the problem to the empty set {}. Also initialize **A** to {}.

Step 2. Repeat until $|\mathrm{S}| = min\_out$

Step 2a. Find the node $(k^*, l^*) \in \mathbf{ID} \times \mathbf{U}$ with the lowest arc cost $(c_{k^* l^* k^*}^{new})$

Step 2b. Update $\mathbf{S} := \mathbf{S} \cap \{l^*\}$, $\mathbf{U} := \mathbf{U} \setminus \{l^*\}$, $\mathbf{A} := \mathbf{A} \cap \{(k^*, l^*)\}$, and $Z_{ALR}^{SubP2} := Z_{ALR}^{SubP2} + c_{k^*l^*j}^{new}$. Set $parent(l^*) := k^*$, $depot(l^*) := k^*$.

Step 2c. Add node l^* to the adjacency list of node k^*. Then increase $outdeg(k^*)$ by one. Go to Step 2.

Step 3. Repeat until $\mathbf{U} = \{\}$:

Step 3a. Find the pair of nodes $(k^*, l^*) \in \mathbf{S} \times \mathbf{U}$ with the lowest arc cost $(c_{ijk}^{new})^*$ where j is either k^* itself if $k^* \in ID$, or denotes the depot node of k^* if $k^* \in I$.

Step 3b. Update $\mathbf{S} := \mathbf{S} \cap \{l^*\}$, $\mathbf{U} := \mathbf{U} \setminus \{l^*\}$, $\mathbf{A} := \mathbf{A} \cap \{(k^*, l^*)\}$, and $Z_{ALR}^{SubP2} := Z_{ALR}^{SubP2} + c_{k^*l^*j}^{new}$. Set $parent(l^*) := k^*$, $depot(l^*) := j$.

Step 3c. Add node l^* to the adjacency list of node k^*. Also if $k^* \in ID$ then increase $outdeg(k^*)$ by one. Go to Step 2.

Step 4. Restore the resulting MSF's conformation to the center (depot) nodes' degree balance constraints in Equation (3.5). In doing so, select for each depot node $j \in ID$ that customer node i^* with $depot(i^*) = j$, whose connection to j would yield the lowest-cost arc (i^*, j). Repeat this selection for depot j until $indeg(j)$ matches $outdeg(j)$, which was established in *Steps* 2-3.

Appendix C. Pseudo Code of the Tabu Serach Algorithm for MDVRP

num_iter	Number of iterations performed.
num_neigh	Number of neighbors generated in current iterations.
num_nonimp_iter	Number of iterations through which the best feasible or the best infeasible solution do not improve.
max_iter	Maximum number of iterations.
max_nonimp_iter	Maximum number of iterations through which best feasible or best infeasible solution do not improve.
size_neigh	Number of neighboring solutions to be generated in an iteration.

Table A.3 Notations used in the pseudo code of tabu search

Algorithm Tabu Search

Generate an initial solution using [PFIH- NN] heuristic**.**
$num\_iter := 0$, $num\_nonimp\_iter := 0$, and $num_i := 1$ $(i = 1,2,3)$.

While $(num\_iter \leq max\_iter)$ AND $(num\_nonimp\_iter \leq max\_nonimp\_iter)$ **do**

$num\_neigh := 0$.

While $(num\_neigh \leq size\_neigh)$ **do**

Select one of the move operators with equal probabilities. Call this move *k*.

Select two pilot nodes randomly.

Generate a neighboring solution with pilot nodes and move *k*.

If the newly generated neighboring solution is not tabu and better than the best neighboring solution, update the best neighboring solution.

If the newly generated neighboring solution is feasible and better than the incumbent or it is infeasible and better than the best infeasible solution, update the best neighboring solution and end loop

$num\_neigh := num\_neigh + 1$.

End While

Set the best neighboring solution as the current solution.

If the current solution is feasible and also better than the incumbent,
update the incumbent and set *num_nonimp_iter* = 0.

If the current solution is infeasible, but better than the best infeasible solution,
update the best infeasible solution and set *num_nonimp_iter* = 0.

Decrement the tabu durations of all attributes by one.

Select the tabu tenure between 5 and 15 with randomly.

Make the attributes of the best neighboring solution tabu for as much as tabu tenure iterations.

num_iter := *num_iter* + 1.

num_nonimp_iter := *num_nonimp_iter* + 1.

End While

Appendix D. Pseudo Code of the Add-Drop Heuristic

BESTPLAN	Location plan of the best feasible solution found
ZUB_IMPROVED	Flag variable indicating whether a feasible solution with better objective value has been achieved
TOTCOST	The best objective function value found for P through 10 tabu search runs
P	Denotes a present depot which is decided to be preserved
O	Denotes a candidate depot which is decided to be opened
C	Denotes a present depot which is decided to be closed
X	Denotes a candidate depot which is not opened

Table A.4 Notations and symbols used in the pseudo code of Add-Drop heuristic

Procedure [ADD]

Step 0. Copy the current best depot location plan **BESTPLAN** onto another parameter **CURRPLAN**.

Do

Step 1. Set flag variable ZUB_IMPROVED:= NO.

Step 2. Scan all closed present depots (status "**C**") in **CURRPLAN**, and promote them to "**O**" one by one.
For each resulting new depot location plan (**NEWPLAN**):

Step 2a. **TOTCOST := ∞**

Apply the tabu search heuristic 10 times with different initial random number seeds. At the end of each run calculate the objective function value of the comprehensive problem **P**. If the new objective function is better update **TOTCOST**.

Step 2b. If **TOTCOST** < Z_{ub}, then update Z_{ub}:= **TOTCOST** and **BESTPLAN**:= **NEWPLAN**.
Also update flag variable ZUB_IMPROVED:= YES.

Step 3. Scan all the candidate depot locations in which a depot is not established (status "**X**") in **CURRPLAN**, and promote them to "**O**" one by one. For each resulting new depot location plan (**NEWPLAN**), repeat Steps 2a-2b in the search for a better **BESTPLAN** which can give a tighter Z_{ub}.

Step 6. If ZUB_IMPROVED = YES, then update **CURRPLAN**:= **BESTPLAN**.

While ZUB_IMPROVED = YES;

Procedure [DROP]

Step 0. Copy the current best depot location plan **BESTPLAN** onto another parameter **CURRPLAN**.

Do

Step 1. Set flag variable ZUB_IMPROVED:= NO.

Step 2. Scan all preserved present depots (status "**P**") in **CURRPLAN**, and demote them to "**C**" one by one.

For each resulting new depot location plan (**NEWPLAN**):

Step 2a. **TOTCOST:=** ∞

Apply the tabu search heuristic 10 times with different initial random number seeds. At the end of each run calculate the objective function value of the comprehensive problem **P**. If the new objective function is better update **TOTCOST**.

Step 2b. If **TOTCOST** < Z_{ub}, then update Z_{ub}:= **TOTCOST** and **BESTPLAN**:= **NEWPLAN**.

Also update flag variable ZUB_IMPROVED:= YES.

Step 3. Scan all candidate depot locations in which a depot is established (status "**O**") in **CURRPLAN**, and demote them to "**X**" one by one. For each resulting new depot location plan (**NEWPLAN**), repeat Steps 2a-2b in the search for a better **BESTPLAN** which can give a tighter Z_{ub}.

Step 5. If ZUB_IMPROVED = YES, then update **CURRPLAN**:= **BESTPLAN**.

While ZUB_IMPROVED = YES

Appendix E. The Depot Costs of the Randomly Generated Test Problems

The random problem generator takes on two sets of parameters where each one comprises a lower limit (L), an upper limit (U) and an increment (I) value. One of the parameter sets is used for calculating FC values while the second is used for OC values. FC_k and OC_k values are calculated using the following formula where $k \in ID$ and $rand_k$ denotes a standart uniform random number; i.e., $rand_k \in [0, 1)$.

$$FC_k(OC_k) = ROUND\left(rand_k * \frac{(U - L)}{I} \right) * I + L \qquad \text{(A.1)}$$

We have determined different L, U and I values for calculating FC and OC for each different N_C-N_{PD}-N_{CD} triplet, which are provided in Table A.6. For each N_C-N_{PD}-N_{CD} triplet we have set a vehicle acquisition cost VC in proportion to the fixed depot opening-closing costs. On the average, VC corresponds to 10% of the FC.

N_C	Number of customers in the problem
N_{PD}	Number of present depots in the problem
N_{CD}	Number of candidate depots in the problem
FC	Fixed opening-closing cost of depots
OC	Fixed operating cost of depots
VC	Vehicle acquisition cost

Table A.5 Notation used in Table A.6

			VC	*FC*			*OC*		
N_C	N_{PD}	N_{CD}		*L*	*U*	*I*	*L*	*U*	*I*
15	1	2	3	20	30	5	30	40	5
15	-	1	3	25	35	5	35	45	5
20	1	3	3	25	35	5	35	45	5
20	-	3	3	20	30	5	30	40	5
25	1	2	3	20	30	5	30	40	5
25	1	4	3	25	35	5	35	45	5
30	1	5	3	30	40	5	40	50	5
30	-	4	3	25	35	5	25	35	5
35	1	4	4	25	35	5	35	45	5
35	-	6	4	30	40	5	40	50	5
40	1	4	3	25	35	5	35	45	5
40	-	4	3	30	40	5	40	50	5
50	1	4	3	25	35	5	35	45	5
50	-	5	3	30	40	5	40	50	5
60	1	4	3	25	35	5	35	45	5
60	-	5	3	30	40	5	40	50	5
80	2	4	3	30	40	5	40	50	5
80	-	6	3	35	45	5	40	50	5
100	2	4	3	30	40	5	40	50	5
100	-	6	3	35	45	5	40	50	5

Table A.6 The fixed opening-closing, the operating and the vehicle acquisition costs of the randomly generated test problems

Appendix F. The 2-index LRP Model

The 2-index LRP model proposed by Boğ [16] is adapted to our problem that is described in Chapter 3. Boğ has defined replicates of depot nodes, and names the first set of the depot nodes as departure depots while referring to the others as arrival depots. We have not preferred adopting this way of depot definition as is. In order to represent the present depots and the candidate depot locations in our problem, we have defined ID_{pres} and ID_{cand} which do not exist in the formulation of Boğ. The objective function of the modified model **P'** also includes the terms for the depot closing and operating costs while the variable operating cost term in Boğ's formulation which depends on the amount of demand assigned to a depot is eliminated. In order to tighten the subtour elimination constraints, we have employed the Lifted MTZ equations proposed by Kara et al. [47] instead of the original ones which Boğ has used in his formulation. In our problem, we do not define an upper bound on the number of vehicles acquired in the solution. Therefore, Equation (A.6) only forces a lower bound on the number of vehicles unlike the one defined by Boğ which brackets the number of vehicles between an upper and a lower bound.

Notation:

Sets:

IC : set of customers

ID : set of depots

ID_{pres} : set of present depots

ID_{cand} : set of candidate depots

I : set of all nodes ($IC \cup ID$)

Binary Decision Variables:

x_{ij} : 1 if node j is visited after node i, 0 otherwise.

y_j : 1 if depot j is in service, 0 otherwise.

δ_{ij} : 1 if customer i is assigned to depot j, 0 otherwise.

Parameters:

FC_j : the opening or closing cost of depot j

OC_j : the operating cost of depot j

VC_j : vehicle acquisition cost for depot j

c_{ij} : traveling cost of one vehicle from node i to node j

Q : uniform vehicle capacity

d_i : demand of customer i

$$\mathbf{P'}: \text{Min} \sum_{j\in ID} OC_j y_j + \sum_{j\in ID_{cand}} FC_j y_j + \sum_{j\in ID_{pres}} FC_j(1-y_j) + \sum_{j\in ID}\sum_{i\in IC} VC_j x_{ij} + \sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}} c_{ij}x_{ij} \quad \text{(A.2)}$$

subject to:

$$\sum_{\substack{j\in I\\ j\neq i}} x_{ij} = 1 \qquad \forall i \in IC \qquad \text{(A.3)}$$

$$\sum_{\substack{j\in I\\ j\neq i}} x_{ji} = 1 \qquad \forall i \in IC \qquad \text{(A.4)}$$

$$\sum_{i\in IC} x_{ji} = \sum_{i\in IC} x_{ij} \qquad \forall j \in ID \qquad \text{(A.5)}$$

$$\sum_{j\in ID}\sum_{i\in IC} x_{ji} \geq \left\lceil \frac{\sum_{i\in IC} d_i}{Q} \right\rceil \qquad \text{(A.6)}$$

$$U_i - U_j + Qx_{ij} + (Q - d_i - d_j)x_{ji} \leq Q - d_j \qquad \forall i, j \in IC \qquad \text{(A.7)}$$

$$d_i \leq U_i \leq Q \qquad \forall i \in IC \qquad \text{(A.8)}$$

$$\delta_{ij} \leq y_j \qquad \forall i \in IC, j \in ID \qquad \text{(A.9)}$$

$$\sum_{j \in ID} \delta_{ij} = 1 \qquad \forall i \in IC \qquad \text{(A.10)}$$

$$x_{ij} \leq \delta_{ij} \qquad \forall i \in IC, j \in ID \qquad \text{(A.11)}$$

$$x_{ji} \leq \delta_{ij} \qquad \forall i \in IC, j \in ID \qquad \text{(A.12)}$$

$$x_{ik} + \delta_{ij} - \delta_{kj} \leq 1 \qquad \forall i,k \in IC, j \in ID \qquad \text{(A.13)}$$

$$x_{ik} + \delta_{kj} - \delta_{ij} \leq 1 \qquad \forall i,k \in IC, j \in ID \qquad \text{(A.14)}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \in I \qquad \text{(A.15)}$$

$$\delta_{ij} \in \{0,1\} \qquad \forall i \in IC, j \in ID \qquad \text{(A.16)}$$

$$y_j \in \{0,1\} \qquad \forall j \in ID \qquad \text{(A.17)}$$

$$U_i \geq 0 \qquad \forall i \in IC \qquad \text{(A.18)}$$

Appendix G. The Lower Bounds Found by LP Relaxation

The lower bounds found by solving the LP relaxation of the LRP model given in Appendix F (**P'**) are given in Tables A.8 – A.10. We tried to solve LP relaxed version of the 3-indexed formulation **P**. However, Cplex was able to execute the model only for the test problems with 15 customers. In Table 10, the LP bounds found using two different formulations are compared. The time elapsed by the Lagrangian relaxation part of [LR-TS] is provided in the tables that we consider this value as the solution time of lower bound finding procedure. The notation used in the tables in as follows:

The lower bounds found by solving the LP relaxation of the LRP model given in Appendix F (problem **P'**) are given in Tables A.7–A.9. The CPU time elapsed by the Lagrangian relaxation part of [LR-TS] is also provided in the tables. We consider this value as the CPU time of our lower bound procedure. The notation used in the tables is as follows:

Z_{ub} : The best upper bound found by [LR-TS]

Z_{lb} : The best lower bound found by [LR-TS]

$\%GAP_{LR}$: The gap between Z_{ub} and Z_{lb} calculated with the formula in Equation (6.2) (%GAP1 in Chapter 6)

CPU_{LR} : The solution time of the LR part of [LR-TS]

Z_{LP} : The lower bound obtained as the optimal solution to the LP relaxation of **P'**

$\%GAP_{LP}$: The gap between Z_{ub} and Z_{LP} calculated with the formula $100 \times \left| \frac{Z_{ub} - Z_{LP}}{Z_{LP}} \right|$

CPU_{LP} : The Cplex 10.0 solution time of the LP relaxed problem

ID	Z_{ub}	Z_{lb}	$\%GAP_{LR}$	CPU_{LR}	Z_{LP}	$\%GAP_{LP}$	CPU_{LP}
R1-1	1127.84	1075.58	4.86	23.0	932.93	20.89	0.1
R1-2	994.92	994.92	0.00	2.2	725.23	37.19	0.1
R1-3	909.72	805.16	12.99	2.0	753.72	20.70	0.1
R1-4	1027.14	975.28	5.32	15.0	795.49	29.12	0.1
R1-5	1074.68	1031.51	4.19	17.8	769.19	39.72	0.0
R1-6	1291.96	1274.75	1.35	3.9	1035.65	24.75	0.0
R1-7	1148.72	1128.51	1.79	7.4	933.39	23.07	0.1
R1-8	1317.96	1262.14	4.42	20.4	978.2	34.73	0.1
R1-9	1138.72	1114.10	2.21	67.5	858.66	32.62	0.1
R1-10	1442.48	1435.11	0.51	125.9	1141.86	26.33	0.1
R1-11	1022.49	953.04	7.29	25.1	760.98	34.36	0.1
R1-12	1220.48	1220.38	0.00	14.2	1013.1	20.47	0.1
R1-13	1402.44	1321.13	6.15	192.9	1111.58	26.17	0.1
R1-14	1286.85	1244.53	3.40	114.1	966.04	33.21	0.1
R1-15	1204.41	1204.01	0.03	42.5	961.95	25.21	0.1
R1-16	1418.18	1370.51	3.48	138.3	1104.82	28.36	0.2
R1-17	1370.47	1367.03	0.25	183.5	1017.98	34.63	0.2
R1-18	1029.39	975.77	5.50	80.2	753.62	36.59	0.2
R1-19	1525.03	1471.51	3.64	237.6	1111.85	37.16	0.3
R1-20	1441.94	1348.73	6.91	576.4	989.58	45.71	0.2
R1-21	1198.73	1093.31	9.64	340.7	882.43	35.84	0.3
R1-22	1543.86	1511.43	2.15	121.8	1147.2	34.58	0.2
R1-23	1611.87	1555.81	3.60	394.3	1177.42	36.90	0.2
R1-24	1442.53	1386.08	4.07	282.3	1129.55	27.71	0.2
R1-25	1812.81	1735.65	4.45	799.1	1365.01	32.81	0.4
R1-26	1386.02	1362.82	1.70	220.9	987.37	40.37	0.4
R1-27	1255.75	1140.06	10.15	783.9	935.44	34.24	0.5
R1-28	1801.37	1658.61	8.61	454.3	1371.22	31.37	0.4
R1-29	1634.60	1556.13	5.04	452.6	1157.95	41.16	0.5
R1-30	1185.35	1081.08	9.64	193.2	928.02	27.73	0.5
averages	**1308.96**	**1255.16**	**4.27**	**197.8**	**993.25**	**31.79**	**0.2**

Table A.7 Comparison of Z_{lb} with LP bounds for R1

ID	Z_{ub}	Z_{lb}	%GAP$_{LR}$	CPU$_{LR}$	Z_{LP}	%GAP$_{LP}$	CPU$_{LP}$
R2-1	1428.56	1206.44	18.41	477.7	999.7	42.90	0.6
R2-2	1649.62	1549.84	6.44	864.8	1070.03	54.17	0.5
R2-3	2144.55	1903.06	12.69	595.5	1576.17	36.06	0.6
R2-4	1877.78	1708.44	9.91	895.7	1421.13	32.13	0.4
R2-5	1947.74	1770.79	9.99	799.8	1220.21	59.62	0.4
R2-6	2221.40	1897.92	17.04	903.2	1595.29	39.25	0.5
R2-7	1700.76	1532.22	11.00	2627.6	1206.74	40.94	0.8
R2-8	1997.27	177 2.49	12.68	2609.7	1257.35	58.85	0.7
R2-9	2357.29	2112.81	11.57	1806.1	1627.42	44.85	0.6
R2-10	1973.15	1778.51	10.94	540.7	1458.69	35.27	1.9
R2-11	1990.79	1635.76	21.70	897.1	1160.28	71.58	0.8
R2-12	2306.43	2114.33	9.09	1291.4	1674.59	37.73	0.9
R2-13	1921.78	1617.15	18.84	2717.8	1341.11	43.30	1.4
R2-14	2336.22	2084.90	12.05	5097.7	1453.82	60.70	1.0
R2-15	2735.82	2130.51	28.41	1315.8	1826.79	49.76	1.2
R2-16	2200.74	2103.90	4.60	3223.6	1648.42	33.51	1.4
R2-17	2339.18	1890.00	23.77	1735.5	1368.35	70.95	1.1
R2-18	2751.57	2115.01	30.10	931.2	1879.15	46.43	1.3
R2-19	1996.10	1781.52	12.04	5602.8	1459.54	36.76	3.2
R2-20	2642.04	2286.75	15.54	4194.3	1496.59	76.54	3.4
R2-21	3076.79	2585.78	18.99	7671.5	2113.54	45.58	3.0
R2-22	2190.70	1945.81	12.59	1319.3	1683.00	30.17	6.8
R2-23	2670.56	2308.77	15.67	4788.1	1591.55	67.80	5.2
R2-24	3050.53	2488.13	22.60	3719.0	2295.82	32.87	8.9
R2-25	2184.96	1952.80	11.89	8116.9	1620.48	34.83	6.3
R2-26	2893.82	2369.93	22.11	10648.2	1571.02	84.20	5.2
R2-27	3320.89	2691.79	23.37	11599.7	2217.65	49.75	7.6
R2-28	2448.95	2032.33	20.50	4466.6	1584.96	54.51	3.4
R2-29	2896.34	2324.22	24.62	14025.5	1578.24	83.52	3.3
R2-30	3258.81	2741.77	18.86	10208.7	2217.78	46.94	5.2
averages	**2350.37**	**2014.46**	**16.27**	**3856.4**	**1573.85**	**50.05**	**2.6**

Table A.8 Comparison of Z_{lb} with LP bounds for R2

ID	Z_{ub}	Z_{lb}	$\%GAP_{LR}$	CPU_{LR}	Z_{LP}	$\%GAP_{LP}$	CPU_{LP}
P111112	1417.30	1283.09	10.46	18331.6	1103.29	28.46	22.4
P111122	1410.04	1178.19	19.68	8313.1	1048.64	34.46	50.7
P111212	1406.33	1140.54	23.30	8419.8	958.32	46.75	13.4
P111222	1464.84	1186.54	23.45	14839.3	1050.67	39.42	42.4
P112112	1209.88	1079.16	12.11	13585.3	855.56	41.41	11.2
P112122	1019.44	925.16	10.19	15737.3	757.00	34.67	38.8
P112212	726.48	627.05	15.86	6114.5	433.45	67.60	9.2
P112222	738.34	541.66	36.31	13899.7	440.52	67.61	21.1
P113112	1296.04	1069.98	21.13	15490.0	823.99	57.29	11.0
P113122	1160.09	1055.33	9.93	9707.0	782.73	48.21	31.7
P113212	908.79	753.37	20.63	5260.2	467.44	94.42	8.1
P113222	929.22	780.93	18.99	9725.7	477.83	94.47	15.0
P131112	1869.43	1561.25	19.74	49267.2	1391.5	34.35	96.3
P131122	1899.42	1465.80	29.58	49999.1	1306.28	45.41	213.6
P131212	2099.50	1589.11	32.12	41597.7	1307.45	60.58	46.6
P131222	1807.63	1438.10	25.70	51380.0	1327.85	36.13	202.5
P132112	1488.29	1151.67	29.23	40270.5	916.00	62.48	34.2
P132122	1502.16	1144.07	31.30	56162.1	920.58	63.18	169.5
P132212	1234.50	959.29	28.69	24682.9	596.28	107.03	17.7
P132222	938.22	742.16	26.42	67392.3	597.31	57.07	51.9
P133112	1667.65	1232.78	35.28	8791.9	943.21	76.81	36.4
P133122	1452.97	1051.04	38.24	29404.3	857.19	69.50	99.7
P133212	1173.29	930.82	26.05	53388.9	624.06	88.01	12.9
P133222	1189.44	973.35	22.20	24038.6	708.35	67.92	73.7
P121112	2337.60	1747.10	33.80	103408.8	1591.75	46.86	457.3
P121122	2176.88	1639.88	32.75	65697.0	1542.58	41.12	547.2
P121212	2144.31	1800.51	19.09	141462.7	1614.53	32.81	572.2
P121222	2303.29	1683.70	36.80	115925.1	1613.38	42.76	549.5
P122112	2011.02	1591.88	26.33	185395.7	1299.35	54.77	358.3
P122122	1757.52	1320.11	33.13	201249.3	1154.61	52.22	182.4
P122212	1484.87	1079.33	37.57	71898.2	844.62	75.80	247.6
P122222	1094.71	1001.98	9.25	72377.5	713.85	53.35	158.4
P123112	2009.21	1576.96	27.41	69992.9	1147.47	75.10	390.4
P123122	1885.89	1433.07	31.60	122222.7	1264.49	49.14	353.1
P123212	1783.77	1498.26	19.06	157170.3	946.00	88.56	93.5
P123222	1362.84	1064.47	28.03	55929.3	749.13	81.92	102.2
averages	**1510.03**	**1202.71**	**25.04**	**55514.67**	**977.15**	**58.82**	**148.4**

Table A.9 Comparison of Z_{lb} with LP bounds for TB

	2-*index formulation*			3-*index formulation*		
ID	Z_{LP}	$\%GAP_{LP}$	CPU_{LP}	Z_{LP}	$\%GAP_{LP}$	CPU_{LP}
R1-1	932.93	20.89	0.1	1034.19	9.06	8.1
R1-2	725.23	37.19	0.1	858.69	15.86	8.0
R1-3	753.72	20.70	0.1	800.21	13.69	7.6
R1-4	795.49	29.12	0.1	945.99	8.58	5.1
R1-5	769.19	39.72	0.0	960.7	11.86	6.2
R1-6	1035.65	24.75	0.0	1176.34	9.83	5.6
averages	835.37	28.73	0.1	962.69	11.5	6.8

Table A.10 Comparison of LP bounds found by 3-index and 2-index formulations

VITA

Zeynep Özyurt was born in Tokat, Turkey on October 6, 1982. She graduated from Tokat Anadolu Lisesi in 2000. She received her B.Sc. degree in Boğaziçi University Industrial Engineering Department in July 2004. In September 2004 she joined Koç University as teaching and research assistant. Since September 2006 she has been working at Borusan Lojistik A.Ş. as Quality and Management Systems Development Specialist.