A GENETIC ALGORITHM FOR THE LOCATION-ROUTING PROBLEM WITH TIME WINDOWS

HANDE ÖZGÖNENÇ

JULY 2006

A GENETIC ALGORITHM FOR THE LOCATION-ROUTING PROBLEM WITH TIME WINDOWS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HANDE ÖZGÖNENÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JULY 2006

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Çağlar Güven
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Sedef Meral
Supervisor

Examining Committee Members

Prof. Dr. Murat Köksalan	(METU,IE)	
Asst. Prof. Dr. Sedef Meral	(METU,IE)	
Assoc. Prof. Dr. Canan Sepil	(METU,IE)	
Asst. Prof. Dr. Ayten Türkcan	(METU,IE)	
Mehmet Kayhan (M.S.)	(ASELSAN)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Hande ÖZGÖNENÇ

Signature :

ABSTRACT

A GENETIC ALGORITHM FOR THE LOCATION-ROUTING PROBLEM WITH TIME WINDOWS

Özgönenç, Hande
M.Sc., Department of Industrial Engineering
Supervisor: Asst. Prof. Dr. Sedef Meral

July 2006, 156 pages

The emphasis on minimizing the logistics costs to decrease overall system costs has led the researchers to consider the interdependencies between the decisions of locating facilities and planning the routes from those facilities. The location-routing problems considering this issue are the subject of this thesis study. A two-level hierarchical distribution system is considered in which goods are delivered from the sources (plants) to the facilities (depots) and then from the facilities to the customers. The facilities are uncapacitated and operate within the shift times defined. The goods are to be delivered to the customers within their time windows by the vehicles that are capacitated.

Both a mathematical model and a genetic algorithm based heuristic solution approach are proposed for this problem. We discuss the problem specific issues integrated with the general framework of the genetic algorithm applications. The computational studies are realized on a number of test problems. The results indicate that the genetic algorithm based heuristic gives satisfactory results compared with a sequential solution methodology.

Keywords: Location-Routing, Location-Allocation, Vehicle Routing Problem with Time Windows, Metaheuristics, Genetic Algorithms

ÖZ

ZAMAN KISITLI YERLEŞİM-ROTALAMA PROBLEMİ İÇİN BİR GENETİK ALGORİTMA

Özgönenç, Hande
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Y.Doç. Dr. Sedef Meral

Temmuz 2006, 156 sayfa

Toplam sistem maliyetlerini düşürmek amacıyla lojistik maliyetlerini en aza indirgeme üzerindeki vurgu; araştırmacıları depoları konumlandırma ve bu depolardan yapılacak dağıtımların rotalarını belirleme kararları arasındaki ilişkiyi dikkate almaya yöneltmiştir. Bu konuyu ele alan yerleşim-rotalama problemleri, bu tez çalışmasının konusudur. Ürünlerin tedarik kaynaklarından depolara, oradan da müşterilere dağıtıldığı 2 seviyeli hiyerarşik bir dağıtım sistemi düşünülmüştür. Kapasite sınırı olmayan depolar önceden belirlenmiş çalışma saatleri dahilinde çalışmakta; mallar kapasite sınırı olan araçlar tarafından müşterilere belirlenmiş olan zaman aralıkları içinde dağıtılmaktadır.

Üzerinde çalışılan problem ile ilgili olarak bir matematiksel model ve genetik algoritmaya dayalı bir sezgisel yöntem sunulmuştur. Genel genetik algoritma çerçevesi, probleme özgü bir takım bilgiler ile bütünleştirilmiş ve tartışılmıştır. Bir grup test problemi üzerinde önerilen yöntem ile sonuçlar elde edilmiştir. Bu sonuçlar, önerilen yöntemin karşılaştırmada kullanılan ardışık çözüm yöntemine göre yeterince iyi sonuç verdiğini göstermiştir.

Anahtar Kelimeler: Yerleşim-Rotalama, Yerleşim-Atama, Zaman Kısıtlı Araç Rotalama Problemi, Modern Sezgisel Yöntem, Genetik Algoritma

To my parents…

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Asst. Prof. Dr. Sedef Meral for her guidance and all the time she dedicated to this study. Her supporting and friendly attitude motivated me throughout the study, always refreshing my belief in completing it and coming out with good results even during the most difficult phases.

I would also like to thank Oğuz Solyalı and Tarık Dinç for sharing their suggestions and experiences that helped me a lot for the computational part of this work.

I would present my special thanks to Ertuğrul Akgül who went through all the stages of the study together with me with great patience doing the best that could be done from way overseas. I deem myself lucky for feeling his everlasting belief in me and my success all this time and for all he brought into my life; including my little, white, furry friend, who tried to cheer me up in the most devastating hours of study.

And finally, very special thanks to my parents Hale and Ahmet Özgönenç, for their endurance against my unbearable mood while trying to handle various problems at difficult stages of the study, for making me feel that they will always be next to me when I need and for their endless support and encouragement from the beginning of this study.

TABLE OF CONTENTS

PLAGIARISM..iii
ABSTRACT...iv
ÖZ...vi
ACKNOWLEDGEMENTS...ix
TABLE OF CONTENTS...x
LIST OF TABLES..xiii
LIST OF FIGURES..xv
CHAPTERS
1. INTRODUCTION..1
2. LITERATURE REVIEW...5
2.1 LRP Related Literature...5
2.1.1 Single Stage LRPs..10
2.1.1.1. Single Stage LRPs and Heuristic Based Solution Methodologies.................................10
2.1.1.2 Single Stage LRPs and Exact Algorithms.......18
2.1.2 Multiple Stage LRPs..19
2.2 Genetic Algorithms on Location and Routing Problems...........25
3. MATHEMATICAL MODELING OF THE LOCATION-ROUTING PROBLEM...33
3.1 LRP and the Problem Environment in the Study..........................33
3.2 Verbal Description of the LRP Studied...35
3.3. Mathematical Models of the LRP Studied.. 36
3.3.1 Non-linear Models...37
3.3.1.1 LRP with hard time-windows 37
3.3.1.2 LRP with soft time-windows 41

3.3.2 Linear Models……………………………………..43
3.3.2.1 LRP with hard time-windows 43
3.3.2.2 LRP with soft time-windows..............44
3.4 Validation of the Mathematical Model…………………………44
4. THE SOLUTION METHODOLOGY ……………...…………………...46
4.1 Fundamentals of the Genetic Algorithms……………………..…47
4.2 The Proposed Solution Methodology and GA Components…....48
4.2.1 Chromosome Representation…………………………..49
4.2.2 Initial Population Generation………………………….52
4.2.3 Fitness Function Evaluation Including VRPTW………56
4.2.4 Genetic Operators and Techniques……………………60
4.2.4.1 Parent Selection……………………………...60
4.2.4.2 Crossover……………………………………62
4.2.4.3 Mutation……………………………………...68
4.2.4.4 Replacement…………………………………69
4.2.5 Parameter Determination………………………………70
4.2.6 The Overall Algorithm………………………………...72
5. MODEL TESTING AND COMPUTATIONAL STUDY……………….76
5.1 Test Problem Structure and Data………………………………..76
5.1.1 Locations and Capacities………………………………78
5.1.2 Cost Parameters………………………………………..80
5.2 The Method Used in the Evaluation…………………………….82
5.3 The Evaluation of the Proposed Approach……………………..87
5.3.1 Preliminary Work on the Population Size and the Number of Maximum Generations……………………88
5.3.2 The Effect of Problem Specific Operators versus Classical Ones………………...……………………93
5.3.3 Trials on Obtaining the Optimal and Comparison of the Algorithm with the Optimal for an Example Case…....97
5.3.4 Comparison of the Algorithm with the Sequential Approach………...……………………………...…100

5.3.5 More Research on Genetic Algorithms………………106
5.3.5.1 The behavior of genetic algorithms within a run……….……………………………........106
5.3.5.2 A further study on initial population generation…...………………………………107
6. CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH..110
REFERENCES……………………………………………………………….114
APPENDICES
A. GAMS MODELS ………………………..……………….………….124
B. ILOG DISPATCHER…………...…………...…………...………......132
C. THE PSEUDOCODES FOR THE GA BASED HEURISTIC…………136
D. TEST PROBLEM INSTANCES…………...…………...………….......146

LIST OF TABLES

TABLES

5.1 The Coordinates for the Potential Facility Sites………………………………...79

5.2 The Capacities for Three Sizes of Facilities for Different Problem Types...…....81

5.3 Results for instance mr101_25 for Population Size=100 & Number of Generation=250………………………………………………………………...89

5.4 Results for instance mr101_25 for Population Size=300 & Number of Generation=500……………………………………………………………….…..89

5.5 Results for instance mr101_25 for Population Size=500 & Number of Generation=850………………………………………………………………...90

5.6 Results for instance mr101_25 for Population Size=600 & Number of Generation=1000………………………………………………………………90

5.7 Results for instance mrc207_25 for Population Size=500 & Number of Generation=850………………………………………………………………...91

5.8 Results for instance mrc207_25 for Population Size=600 & Number of Generation=1000………………………………………………………………91

5.9 Results for instance mr101_50 for Population Size=1000 & Number of Generation=1750………………………………………………………………93

5.10 Results for instance mr101_50 for Population Size=1000 & Number of Generation=2000………………………………………………………………93

5.11 Results Obtained with the General Crossover and Mutation Operators for Instance mr101_25……………………………………………………………..95

5.12 The Variation of the Computational Time with regard to the Problem Size…..98

5.13 The Comparison of the Results Obtained by GA-based Heuristic with CPLEX Results and with Sequential Approach Results……………………………………..99

5.14 Results for Problems with 25 Customers……………………………………..101

5.15 Results for Problems with 50 Customers……………………………………..101

5.16 Results for Problems with 100 Customers……………………………………101

5.17 Analysis of the Total Cost Values for Both Methods ..……………………....105
5.18 Results Obtained with both the Modified and the Non-Modified Initial Population Generation…………………………….. …………………………...108

LIST OF FIGURES

FIGURES

4.1 The Representation Scheme of a Solution in the Proposed GA………………...51
4.2 A Two-Point Crossover Operator……………………………………………63
4.3 The Flowchart of the Algorithm……………………………………………..74

5.1 The Evolution of the Population through Run4 with General Operators..……..96
5.2 The Evolution of the Population through Run1 with Problem Specific Operators…………………………………………………………………………97
5.3 The Comparison of Results for Problems with 25 Customers…………………102
5.4 The Comparison of Results for Problems with 50 Customers…………………102
5.5 The Comparison of Results for Problems with 100 Customers………………..103
5.6 The Evolution of the Population with regard to Minimum, Maximum and Average Objective Values…………………………………………………………106

CHAPTER 1

INTRODUCTION

Among the cost components in production systems, logistics activities account for a significant percentage. Thus, in order to be competitive and to minimize the losses related to the imperfections in the distribution network, the decision of locating facilities in a distribution system has always gained attention. It is a critical decision as it contains a tradeoff between the fixed facility (depot, warehouse or distribution center) costs and the transportation costs. For full-truckload demand quantities, the distribution strategies are straightforward, however, for less than truckloads, the system becomes more complicated and vehicle routing issues must also be considered along with the location of facilities. This series of relations bring out a different type of problem from the classical location-allocation problem, that's the Location-Routing Problem (LRP).

This study is concerned with the LRP, which considers a number of decisions related with the logistics system as well. They are the primary decisions for logistics environments; the location of facilities, the allocation of customers to the open facilities and the routing of customers which are assigned to the same facility. The interrelations among these basic decisions generate the need to address the location and routing problems in an integrated manner.

In a very general form, we can define LRP as:
in a single/multiple layer environment, given a number of potential sites with specific locations and customers with location and demand data; to determine where to locate a number of capacitated / uncapacitated facilities (the number to locate either

predetermined or not), to decide on the allocation of customers to the open facilities and to determine the routing of the customers with capacitated / uncapacitated vehicles within the time windows for the customers and within the shift times for the facilities if any time constraints exist.

Although the basic decisions to be considered are the ones above, the varieties existing in the environments for location-allocation problems and vehicle routing problems cause to diversify the LRP environment. The studies in the literature are examined in a classification with regard to the problem characteristics in the later parts. But it is possible to list the followings among the critical parameters; the number of hierarchy levels in the problem, the existence of capacity constraints for the facilities and vehicles, the existence of time windows for the deliveries to the customers.

The environment we consider throughout this study is a two-stage LRP where the customer and facility site data (e.g. demands, the time it takes to go from one site to another) are deterministic. Multiple facilities and vehicles are allowed in the system; the facilities are uncapacitated and the vehicles are capacitated. Additional time constraints are also considered; the time intervals for deliveries to the customers and time intervals within which the facilities and vehicles operate are defined. For example, in the case of a retail chain, the characteristics are similar to this environment: the decisions of locating central warehouses which are intermediary facilities between the supply sources and retail points, the allocation of retail points to the central warehouses and the routes for deliveries to retail points are important for the construction of the system. The capacity constraints and shift times on vehicles are usually observed in retail chains. Some traffic regulations or the need for deliveries to the retail points before / after the service hours to customers may be some simple reasons for operating with time windows.

The difficulty in dealing with this type of problem arises from the increase in the number of decision variables in the problem even for reasonable problem sizes which can be encountered easily in practice. That's why, classical mathematical formulation based approaches are usually not satisfactory.

The most common approach in handling LRPs has been to decompose the problem into subproblems that are easier to solve and to obtain either optimal or near-optimal results for those subproblems. Finally by combining the results for separate subproblems and evaluating them trying to consider system-wide trade-offs, a final solution is constructed for the problem.

Our aim in this study is to propose an integrated approach for the LRP. Firstly with the aim of reaching the optimum solution for the problem, a mathematical model is developed. However, the incapability of the solvers in solving even small or medium sized problems has led us to search the use of metaheuristics in different problem environments. For this purpose, we investigate the principles of some metaheuristics and apply an appropriate method, a genetic algorithm (GA) based solution approach for our LRP environment. Some problem specific issues in different phases within the general GA framework are also employed to improve the performance of the algorithm. The algorithm is tested on a number of test problems and the results are compared with the results of a sequential approach. It is observed that in almost all of these instances, the GA-based heuristic has outperformed the sequential approach.

In the following chapters, the details of our study are presented regarding the below outline.

In Chapter 2, starting with the classification of LRP environments with regard to different criteria, a review of the LRP literature is given. The diversity in the literature does not easily allow for the grouping of the studies; but similar problem types and the solution methodologies are introduced together where it is possible. Besides the review of studies concerning LRPs, the second part of Chapter 2 summarizes GA applications in different types of problems. Genetic Algorithms are emphasized as well since our solution approach is based on it.

In Chapter 3, the problem environment that is studied in this thesis work is explained in detail. Both soft and hard time-window cases are considered and the definitions for both environments are given. Based on the verbal problem definitions, the related mathematical models are presented and stated explicitly.

In Chapter 4, we explain the solution methodology we have applied on the location routing problem with time windows. The solution technique we propose is based on Genetic Algorithms; so while describing the proposed algorithm, the basics for the Genetic Algorithms, some different methods for the phases of GA other than the ones we have applied and the reasons for our choices are also given in detail.

In Chapter 5, the test problems generated for computational experiments are defined. Since there is no test problem instances that conform to our problem environment, we have generated the test problem instances. Then the sequential approach used in comparison with the results of the GA-based heuristic is explained. The performance of our approach is evaluated against the sequential approach.

Chapter 6 concludes the study by briefly highlighting the significant parts of our study and giving some suggestions for future study.

CHAPTER 2

LITERATURE REVIEW

This chapter consists of two main parts. In the first part, the literature review on the problem type LRP is presented. After examining different solution procedures, our studies for proposing a solution technique for the LRP have focused on genetic algorithms. So, the second part of the literature review describes some applications of genetic algorithms on some location and routing problems.

2.1 LRP Related Literature

The classical plant location problem and the vehicle routing problem have been studied extensively for years and a wide literature exists on those problems. However, the globalization in logistics environments has supported the view that pure location problems are not sufficient to cope with real situations, and hence, location, allocation and routing issues must be addressed simultaneously. As discussed in Salhi and Rand (1989), the effects of ignoring routes when locating depots are usually significant. Existence of routes including more than one customer may affect the optimal location-allocation solutions, which may not be still optimal after the routing phase. In the study, to reach this conclusion, problem sets are formed according to several scenarios with a single depot or no restrictions on the number of depots. In the solutions of those problems; the Random Destination Algorithm (RDA), the Alternate Location Allocation (ALA), the Drop Method (DROP) for location and a routing procedure developed by Salhi and Rand are used. The ranking of the locations after only location stage and after location-routing stages are then given to show that ignoring the interdependence between location and routing phases would lead us to solutions which are not the lowest cost solutions from a system-wide point of view. With such analysis, the importance of LRPs has gained attention and studies on LRPs have started to appear especially in recent years.

Two comprehensive surveys of LRPs, which may be useful in the review of the literature, are found in Laporte (1988) and in Min et al. (1998).

Laporte (1988) classifies all the LRP literature till 1988, in addition to giving a general framework and defining some basic concepts. The systems up to four layers where both primary and secondary facilities are included are presented. The general ideas related with the heuristic solutions are discussed. As the main problem consists of three subproblems as location, allocation and routing, the heuristics can be examined in two groups; in the sequences of location-allocation-routing and allocation-routing-location. It is also noticed that some heuristic algorithms exist which do not fit exactly the given classification. Then as the LRPs are strictly in relation with vehicle routing problems (VRPs), exact algorithms for VRPs are discussed under three titles; (i) the direct tree search algorithms which are mainly based on branch and bound techniques, (ii) dynamic programming algorithms or (iii) ILP algorithms that can be grouped as set partitioning and flow formulations (both commodity and vehicle flows). Although not practical, the vehicle flow formulations of LRPs with three indices are more successful in handling different aspects of the problem (i.e. heterogeneous vehicle fleet) whereas the two index vehicle flow formulations for both symmetrical and asymmetrical problems in single or multi depot cases can also result in successful implementations for some environments. For symmetrical problems, constraint relaxations and branch and bound techniques are proposed for the solution of single depot problems, and for the multi-depot capacitated LRP, a procedure for generating cuts known as chain barring constraints is described. For asymmetrical two index vehicle flow problems, the relaxed problem which is converted to an assignment problem is solved by efficient algorithms and then branch and bound is used for a check on the violated constraints.

Min, Jayaraman and Srivastava (1998) present a detailed classification of the studies on the combined location routing problems till 1998. There exists several classification schemes in the literature (Madsen, 1983; Laporte, 1988) but Min, Jayaraman and Srivastava (1998) involve a more comprehensive scheme regarding to the problem characteristics and the solution methodologies. Generally, the LRP can be classified in two large classes that are single stage LRPs and multistage LRPs. In single stage LRPs, the focus is on primary facilities while multistage LRPs consider both inbound/outbound flows and they are interested in secondary facilities. Another critical point in the classification is whether the location routing parameters are deterministic or stochastic. Also subcategories are developed according to the nature of the vehicle fleet (single/multiple; capacitated/uncapacitated) and the facilities (capacitated/uncapacitated) in addition to the characteristic of the objective function (single objective/multi objective). A different point of view is to classify the LRP studies in the literature regarding the solution procedures proposed. Although location-allocation problems and routing problems are both NP-Hard, some exact procedures are proposed for the solution of LRPs, but these methods are only applicable for limited sized problems. Besides these, a large number of heuristic procedures, some based on metaheuristics such as Tabu Search (TS), Genetic Algorithm (GA), simulated annealing (SA), are proposed and they are usually proved to be acceptable methods. Departing from the survey carried out, the future research areas such as problem types including stochasticity, time windows, multiple periods and multiple objectives are suggested. A summary table showing the classification scheme explained above is given below as in Min, Jayaraman and Srivastava (1998):

Classification of LRP with regard to the problem characteristics

I. Hierarchical Level
   - A. Single stage
   - B. Two stages

II. Nature of demand/supply
   - A. Deterministic
   - B. Stochastic

III. Number of facilities

A. Single facility

B. Multiple facilities

IV. Size of vehicle fleets

A. Single vehicle

B. Multiple vehicles

V. Vehicle capacity

A. Uncapacitated

B. Capacitated

VI. Facility capacity

A. Uncapacitated

B. Capacitated

VII. Facility layer

A. Primary

B. Secondary/intermediate

VIII. Planning horizon

A. Single period (static)

B. Multiple periods (dynamic)

IX. Time windows

A. Unspecified time with no deadline (No time windows)

B. Soft time windows with loose deadlines

C. Hard time windows with strict deadlines

X. Objective function

A. Single objective (e.g. minimize the total system cost)

B. Multiple objectives (e.g. in a health service sector problem, minimize the total system cost and maximize the coverage/total benefit)

XI. Types of model data

A. Hypothetical

B. Real-world data

Classification of LRP with regard to the solution methodology

I. Exact algorithms
   A. Direct tree search/Branch and bound
   B. Dynamic programming
   C. Integer programming
   D. Nonlinear programming

II. Heuristics
   A. Location-allocation first, route second
   B. Route first, location-allocation second
   C. Savings/insertion
   D. Improvement/exchange

As location-routing problems are a developing research area, a number of studies can be reached belonging to the above classes. However, it is preferred to narrow our literature survey by the problems related with the characteristics of the environment that we have planned to study on. As it will be defined at the beginning of Chapter 3 in detail, the problem environment in this study is a multistage LRP. It is supposed to reflect an environment with deterministic demand and supply quantities, with multiple facilities and vehicles that are capacitated, but there is no restriction on the facility capacities. Time windows for the customers and shift times for the depots are considered as well.

The following studies related to our problem environment are chosen and examined. All studies reviewed are deterministic in terms of the demand nature. Another common aspect in the examined studies is that almost all studies do not cover time windows for the customers. However, the difficulty of matching the studies in the literature exactly to our criteria led us to discuss slightly different environments and the solution methodologies related with them.

The rest of the study explains the characteristics of problems by grouping them according to the environment considered. The second criterion in the presentation of the studies is the solution methodology of the problem, whether it is based on heuristic methods or exact algorithms. The studies are then presented in chronological order.

2.1.1 Single Stage LRPs

These problems are concerned with the location of facilities and the outbound transportation from the facilities to the customers. The inbound part of the distribution system is ignored in single stage LRPs. The facilities to locate may be the primary facilities such as manufacturing plants which are the exact origin of the problem environment as well as the secondary facilities like distribution centers or warehouses.

2.1.1.1. Single Stage LRPs and Heuristic Based Solution Methodologies

The first group of studies examined under this title deal with the *uncapacitated facilities and capacitated vehicles*. Srivastava and Benton (1990) consider this environment. Their focus is on the evaluation of the performance of three heuristic location-routing procedures with respect to some external environmental characteristics. Those environmental factors are cost structure (the ratio of warehousing costs to distribution costs), spatial distribution of customers (uniform or clustered) and the maximum number of depot sites allowed in the system. The capability of 0-1 mixed integer programming is limited in handling the models with too many variables and constraints in the case of location-routing problems; so three alternative heuristics are studied and compared with the sequential method, which is a two-stage model. Those heuristics are savings-drop heuristic, savings-add heuristic and cluster-route procedure. As a benchmark, a sequential method is used, where the first stage is composed of a location-allocation problem solved using an efficient branch and bound by Khumawala (1972) and the second stage is composed of the routing of allocated customers using a modified savings based procedure. The performance criterion used in the benchmark is the % difference in total system costs between the alternative heuristic and the sequential model. It is observed that all of the environmental factors have effect on the performance of the heuristics and there are specific structures that favor the use of each heuristic, besides this, each heuristic outperforms the sequential method solutions.

Chien (1993) is another study which considers the uncapacitated facilities and capacitated vehicles in its problem environment. The sequential procedure proposed consists of mainly two steps; solution construction and solution improvement. The significant point in the solution construction is the way the transportation costs are calculated. Several heuristics using actual costs or certain estimators are developed. Twelve different procedures starting from randomly generated feasible solutions to a number of improvements are applied to a set of 560 problem instances. Within those procedures, the customer-vehicle assignment rules, a modified closest depot rule and certain cost estimators are used to obtain feasible customer-depot allocations and to improve them. Each of the instances is solved with each procedure and then the quality of each solution is evaluated by criteria as the mean square error and the percentage error from the estimated lowest total cost, which is generated based on the costs of a group of feasible solutions. The effectiveness of different construction procedures and improvement procedures for different problem characteristics are grouped, detailed ANOVA test results are discussed and given as conclusions. This study based on a great number of computational experiments, stresses that further investigation on the heuristics is a future research area.

The two previous studies are based on heuristics developed by authors; but it is possible to view some studies (Su, 1998 and Tuzun & Burke, 1999) which propose solutions based on metaheuristics for the same environment (uncapacitated facilities, capacitated vehicles). Su (1998) deals with the design of a physical distribution system implementing the Genetic Algorithms (GA) to the problem. The problem involves both the location of distribution centers (DC), the allocation of customers to DCs and the routing for each open facility, but in an environment with no time windows for customers or shift times for facilities. The characteristics and properties of GAs are adapted to the problem in a simple approach and an efficient method is produced. The solution string is composed of 0-1 variables representing whether a facility is opened at that candidate facility. In the calculation of the fitness value for each string, the constraints are allowed to be included as penalty terms. The steps in the proposed approach can be summarized as:

- Generate a random population where each string represents a solution for the location problem.
- Assign each customer to an open facility (in fact to the nearest open facility as the facilities are uncapacitated), considering each customer will be assigned to one and only one warehouse.
- After determining the number of the needed vehicles for each warehouse, solve the routing subproblem for each depot by a cluster first, route second approach as in Fisher and Jaikumar (1981).
- Calculate the fitness values based on the routing solutions.
- Work out the GA mechanism allowing to generate new populations by reproduction, crossover and mutation.
- Repeat this procedure until the prespecified conditions are reached.

In the study, an experimental simulation is realized using the above procedure to show the efficiency of it and also to help to determine some genetic parameters. The proposed methodology is illustrated on a small problem instance with 5 candidate depot locations and 25 customers. Although the experimental work is not very sufficient to evaluate the performance of the proposed methodology on larger problems, it presents some positive aspects of GA application and it seems promising for future research.

Tuzun and Burke (1999) introduce a two-phase solution procedure which integrates the location and routing phases of LRP and uses tabu search (TS) for the solution of both phases. The two-phase approach coordinates two tabu search mechanisms in a computationally feasible way. In the first phase, which is the location phase, swap or add moves are performed and the solution space is searched for better location configurations. However, in order to determine the best swap or add move; a simplistic approach is used to see how good a location configuration is in terms of total system costs. Although the system design includes routes among nodes, only the direct distances are used to estimate the total goodness of a location configuration in swap or add moves. After the *best* swap/add move, which is determined with regard to the above approximate evaluation, is realized, the routing phase is run and the best routing for the given configuration is obtained. The properties of TS allow exploring different regions of the solution space and prevent to get trapped at the local optima. To evaluate the performance of the TS approach, it is compared with the SAV1 algorithm by Srivastava also mentioned above and the TS algorithm is found to be significantly better than the SAV1 algorithm.

Another environment intensively studied among the single stage LRPs is with *capacitated facilities and capacitated vehicles*. Perl and Daskin (1984, 1985) giving the earliest significant results on this environment have been guides for a number of following studies.

The details and results of Perl and Daskin's studies are conveyed via Perl (1983), Perl and Daskin (1984) and Perl and Daskin (1985). Their research presents a solution methodology that allows handling the interdependence between location and routing decisions explicitly. At first a mathematical programming model which aims to minimize total system costs is presented, however, with a huge number of integer variables and constraints, it is not possible computationally to solve the problem to optimality. Then with some simplifying assumptions, the Warehouse Location Routing Problem is converted to the Modified Warehouse Location Routing Problem and a solution methodology, which is based on decomposing the problem to three subproblems and solving them in a sequential manner, is proposed. The mathematical formulations for each subproblem are given; fixed or variable elements in three phases are classified to reflect the aim of each subproblem. Subproblem-1 (the complete multi-depot vehicle dispatch problem) is solved heuristically without considering the fixed and variable warehousing costs and an initial set of routes is obtained. In subproblem-2 (the warehouse location-allocation problem), the optimal number and locations of DCs, the allocation of routes to DCs is determined by considering the related costs. This subproblem is solved to optimality by Implicit Enumeration algorithm of Lemke and Spielberg (1967). Subproblem-3 (the multi-depot routing location-allocation problem) simultaneously reallocates customers to DCs and designs the delivery routes given the DC locations in subproblem-2, using a heuristic method. The solution of subproblem-3 can then be input to subproblem-2 and this continues iteratively until an insufficiently small amount of reduction in total cost is obtained. In addition, a detailed sensitivity analysis on many parameters is carried out. The methodology is shown to help in large sized practical problems though the subproblems are solved sequentially. This methodology is tested on a base case as well as a real life distribution system. The ability of the proposed methodology for solving large-scale models is shown and results better than the current system are obtained.

Hansen et al. (1994) also study the same problem as Perl and Daskin (1984). The discussions on the integer programming model and on the mixed integer programming model, developed by introducing a set of flow variables and flow constraints, are carried out; however, the number of constraints makes the MILP still difficult to solve. The same decomposition strategy as in Perl and Daskin (1984) with differences in solution techniques is then proposed. For subproblem-1 (the multi-depot vehicle-dispatch problem), an initial feasible solution is obtained and improved by a series of *TSP, Single Displacement* and *Exchange* procedures. Using the initial routing from subproblem-1, subproblem-2 (the warehouse location-allocation problem) is solved by applying procedures as *central depot opening, route cluster displacement, single route displacement, forced depot closing, depot opening sequence*. In subproblem-3 (the multi-depot routing-allocation problem), reallocations and new routings are developed without violating the restrictions on depot capacities. Procedures as *customer orientated initiatives, extended node exchange and two by two node displacements* are used. Compared to the Perl (1983), the heuristics proposed produce better solutions without improving computational effort.

Wu, Low and Bai (2002) also introduce a heuristic solution procedure for the multi-depot location-routing problem but as an extension to the studies of Perl and Daskin (1984) and Hansen (1994), non-homogenous vehicle fleet is handled in their study. In the paper, first a mathematical model of the problem is given. Different from the model developed by Perl and Daskin (1984); another set of sub-tour elimination constraints resulting in a lower number of constraints is introduced, vehicle dispatching costs are included in the objective function and different types of vehicles in terms of capacity are allowed. However, because of the computational difficulty, a heuristic approach is proposed for large-scale problems. In this approach, the LRP is decomposed into two subproblems; the location-allocation problem (LAP) and the vehicle routing problem (VRP). The consolidation is realized by performing LAP module followed by the VRP module until the convergence criterion is met. As both LAP and VRP are difficult to solve, heuristic procedures using simulated annealing are proposed. The solutions for the three problem instances from Perl (1983) are obtained and compared with both Perl's and Hansen's method. The efficiency of the problem is evaluated for new problem instances, as well. The proposed methodology performs better than Perl's solutions; however, it is observed that Hansen's method (Hansen, 1994) is better in some instances.

Lin, Chow and Chen (2002) deal with a real case for bill delivery services, which can be placed in this group of capacitated facilities and vehicles in the classification. The case includes locating the bill delivery offices, vehicle routing for in-house delivery and the loading problem to determine the smallest number of vehicles to be used for the established routes. An additional issue handled in this study is the existence of shift times. The staff working hours are limited and considered in the problem, but there are no time windows as the customers do not need to be present during the delivery. The objective of the problem is to minimize the sum of the facility costs, staff costs, vehicle rental and operating costs. A constraint on the trip time of vehicles imposed by the daily working hour of the staff is added. As the exact methods are not sufficient for real-sized problems, decomposing the problem into reasonable subproblems and then solving those with meta heuristics is the approach proposed in the paper. The approach can be summarized as:

- Determine the minimum number of facilities, n_f
- Pick a set of n_f facilities
- Initial clustering and routing with Clarke & Wright savings algorithm, and then improvement by TSP
- Use of meta heuristics (threshold accepting (TA) and simulated annealing (SA)) for improvement
- Final route improvement by TSP
- Loading (assignment of routes to vehicles)
- Evaluate the system cost
- Iteratively repeat the procedure by changing the set of n_f facilities or increasing n_f till it is worth in terms of total system costs

In the use of metaheuristics, there exist four methods for improvement (TA-SA, SA-TA, TA, SA) and to evaluate the efficiency, the results obtained and the CPU times are compared with branch and bound results or with manual approaches for very large problems. It is observed that the method proposed outperforms the manual approach and can do as well as branch and bound.

Cappenara, Gallo and Maffoli (2004) deal with a discrete combined location-routing model referred to as Obnoxious Facility Location and Routing (OFLR). The candidate sites for facilities are all capacitated; there are no constraints on vehicle capacity, but thresholds of exposures on the arcs are defined. The problem is formulated as a capacitated minimum cost network flow model and the Lagrangean Relaxation proposed (by dualizing the capacity constraints on the facilities) allows decomposing OFLR to a location subproblem and a routing subproblem. Additional constraints to the subproblems are used to change their weakly-correlated nature. The location problem turns out to be a 0-1 Integer programming model and the routing problem turns to linear programming. By solving both, a lower bound for the LRP is obtained. On the other hand, an upper bound is obtained by using two simple heuristics (Location-Routing heuristic or Routing-Location heuristic) based on the information from the Lagrangean relaxation. A branch & bound procedure is then used to close the gap between those bounds. The solution procedure is tested on a set of problems generated by OFLR instances generator given a set of parameters. Although it is difficult to draw general conclusions, efficient techniques according to problem characteristics are determined.

Besides the above classes, studies that do not exactly match with those classes exist.
Srivastava (1993) deals with the environment with multiple uncapacitated facilities and a single uncapacitated facility. The heuristics are SAV1, SAV2 and Clust, which are introduced in Srivastava and Benton (1990). SAV1 and SAV2 use similar savings schemes which are modified versions of the "travel time savings" developed by Clarke and Wright and extended to multi-depot cases by Tillman (1969). SAV1 uses the "drop" approach, but SAV2 uses the "add" approach. Clust heuristic is developed on the assumption that if the customers are grouped in clusters at the beginning, more efficient procedures can be constructed, so the procedure begins by identifying the clusters. Then the locations are determined and vehicle routing phase is realized using polar coordinates based technique. To evaluate the efficiency of the heuristics, as a benchmark, MIP model is used where it is possible to solve the test problems. However, for larger problems, it is infeasible computationally to use MIP, so as in Srivastava and Benton (1990), the sequential approach is used. The results show superiority of the heuristics over the sequential approach.

Sambola, Diaz and Fernandez (2005) define an auxiliary network and give a compact formulation of the combined location-routing problem. The problem handled is one with the following characteristics: multiple capacitated facilities and a vehicle associated with each open facility that has a capacity equal to the facility's capacity. In the auxiliary network, with some simplifying assumptions, the problem is converted to find a set of paths that satisfy the constraints. Solution of the LP relaxation of this compact model is used to obtain an initial solution for the Tabu Search. Starting from a rounded integer solution of the relaxed problem, an integrated approach applying intensification and diversification phases of TS procedure allows the authors to come up with an upper bound for the problem. For the solution, the lower bound is obtained by summing up two types of costs; routing costs from an asymmetric traveling salesman problem and fixed costs from a knapsack problem. To evaluate the quality of bounds, a set of instances in a wide range is developed. A group of them is solved to optimality and the bounds are compared with the optimal solution. For larger instances the gaps between lower and upper bounds are examined. It is observed that tighter relaxations of the compact model with valid inequalities result in better quality bounds.

2.1.1.2 Single Stage LRPs and Exact Algorithms

Laporte, Nobert and Arpin (1986) develop an exact algorithm for solving a single stage capacitated location-routing problem (capacitated vehicles, uncapacitated facilities) with bounds defined on the numbers of facilities and vehicles. Integer linear programming formulation of the problem is provided containing the subtour elimination constraints which ensure the solution does not contain any subtour, and the chain barring constraints which ensure that each route starts and ends at the same depot. A procedure, which is based on relaxing the problem by removing the chain barring constraints, is proposed. The algorithm starts with a heuristic solution giving an objective function value z^*. Then subproblems obtained by eliminating subtour and chains barring constraints are solved and additional constraints are added for infeasible solutions. The objective function values are compared with z^* and the solution proceeds in the search tree till the subproblems are all examined. Three test problem series are generated randomly to evaluate the efficiency of the method proposed; a group with no capacity restrictions, no fixed costs on vehicles or depots, a group including capacity constraints and a group with both capacity constraints and fixed costs. The results of the study show that the algorithm can solve problems up to

20 nodes optimally in reasonable times and this study is known to be the first one to deal with such problems in an exact way.

Laporte, Nobert and Taillefer (1988) address a type of location-routing problem, which is asymmetrical multi-depot location routing problem, reflecting the same environment as Laporte's study above; but they deal with the problem in a manner completely different from the ones discussed up to here. With an appropriate graph representation and a graph extension, LRP is converted to a constrained assignment problem. It is shown that every feasible solution to the original problem corresponds to one or more Hamiltonian circuits and any feasible solution on the transformed graph can be converted to a solution to the original problem on the original graph. Then the constrained assignment problems are solved by a modified version of the branch and bound algorithm for TSP proposed by Carpaneto and Toth (1980). In the algorithm, first the assignment problem relaxation of the problem formulation on the transformed graph is solved, then the violated constraints are checked by branch and bound rules. The algorithm proposed is tested on capacity constrained VRPs (where vehicle capacities are not exceeded), cost constrained VRPs (the cost of a route is predetermined and is not exceeded) and cost constrained LRPs (location routing problems where the VRPs are cost constrained). It is declared that problems containing up to 80 nodes can be solved to optimality without any difficulty.

2.1.2 Multiple Stage LRPs

Multiple stage LRPs consider at least two layers or more in the production-distribution systems. Both the inbound (pickup) and the outbound (delivery) processes are taken into consideration. In those environments, instead of routes originating from a facility to customers, intermediate levels exist and the consolidation issue in those layers is discussed. Cooper (1983) and Hall (1987) are the two studies to get an insight to consolidation.

Consolidation decisions are defined as strategic decisions in a transportation system which can help in taking advantage of lower transportation costs due to larger loads instead of smaller (more frequent or direct) shipments. In Hall (1987) the consolidation strategies are grouped as inventory and vehicle and terminal consolidation; the trade-offs between gains and costs of consolidation strategies are discussed. Inventory consolidation involves storing items that are produced and used at different times, but transporting them together. Vehicle consolidation is as in milk-run systems; collecting items from different suppliers and delivering them to the customer. In terminal consolidation, the terminals are places where different goods from suppliers are collected, grouped according to customer orders and delivered to customers as packages containing different products. Another classification can be as in Cooper (1983) and Min (1996). Freight consolidation can be defined as the aggregation of customer orders for a larger shipment and for a more effective usage of vehicles; and basic freight consolidation strategies are spatial, temporal and product. In multistage LRPs, we are generally interested in the spatial consolidation of orders and the location of intermediate levels (consolidation terminals/distribution centers or warehouses). Especially in cases where customers have time windows, an intermediate level helps to satisfy those constraints and makes it possible to include a greater number of customers in a single route.

A number of multiple stage LRP studies mostly belonging to recent years exist in the literature. As this problem contains a greater number of variables, any exact solution method that handles the whole model is not proposed; instead, heuristics, meta heuristics and the decomposition of the problem are the proposed methods.

Or and Pierskalla (1979) present the first study in the multistage literature. They focus on the location-allocation-transportation aspects of the regionalization of blood centers where the centers are assumed to be uncapacitated but the vehicles are capacitated. The problem is modeled with the objective to minimize the sum of the periodic delivery costs, emergency delivery costs and system costs and with the concern to satisfy the constraints declared by the parameters such as demand quantities or vehicle capacities. The problem is then decomposed into two subproblems. In the first subproblem, the system costs (such as the fixed costs of facilities etc.) are constant and the emergency delivery costs are assumed to be negligible, and the main problem is equivalent to the general transportation problem. If the system costs are constant and the periodic delivery costs are negligible, then the problem turns out to be a location-allocation problem. So the complex problem can be handled as the combination of the two known problems, which are still difficult to solve. Two heuristic solution procedures are developed based on solving the subproblems mentioned above independently and combining them at the end, making tradeoffs between them. These algorithms produce acceptable feasible solutions for the complex blood transportation-allocation problem in reasonable computation times. They are tested on a set of real data from Chicago area and it is shown they can be helpful in real cases.

Apart from the work of Or and Pierskalla (1979), the following studies all model environments where both facilities and vehicles are capacitated.

Bookbinder and Reece (1988) deal with a multi-commodity, capacitated distribution planning model. An exact algorithm, which combines the Geoffrian and Graves (1974) approach to the distribution system design with the Fisher and Jaikumar (1981) approach to the vehicle routing problem, is proposed. The warehouse problem stands for the master problem. After the solution of that master problem, a routing problem is solved for each warehouse that is open. From the solution of the routing problems, outbound cost estimates are calculated and they are used in the transportation subproblems for each commodity. The above procedure is tested on eleven problem instances, some of which can be solved to optimality while it is difficult to find the optimal for some of them. It is noticed that the effect of the starting solution, and hence the lower bound obtained from the master problem on the efficiency to reach the optimality can not be ignored. An alternative solution technique which is an improved version of that procedure is then proposed. This second technique is able to reach optimality for all problem instances.

Min (1996) is another work on the location of consolidation terminals. Consolidation strategies are discussed briefly and solution methodology for a freight consolidation problem, a type of spatial consolidation, is proposed. Similar to the studies in the literature, a multiphase decomposition heuristic procedure where the phases are solved sequentially is proposed. The first phase is the aggregation of customers into capacitated clusters so that the total delivery size of the customers in each cluster does not exceed the vehicle capacities. A p-median type model for capacitated clustering and statistical clustering techniques such as refined Ward's minimum variance method are used. The second phase consists of location of terminals and allocation of sources and customers to terminals. A mathematical model containing both inbound and outbound transportation costs representing a hierarchical structure is given. After each cluster of customer and each source are allocated to an open terminal at the end of the second phase, in the third phase, a detailed route within each customer cluster is realized by solving a well known symmetric TSP. In the first phase, the clusters are formed considering the vehicle constraints, so there is no feasibility problem. The methodology proposed is tested on a set of real data provided by a large logistics firm in USA. Since the problem size that can be solved with exact LRP algorithms is limited, it is not possible to compare the algorithm with those methods. But the efficiency of the methodology is evaluated with the data set with regard to computational time and total savings in travel distances relative to the non-consolidated routing problem.

Wasner and Zäpfel (2004) deal with the network problem of a parcel service provider. The decisions to be made within the problem are the location of depots and hubs, the allocation of postal zones to depots and the transportation routes between customers, depots and hubs. Hence, the structure of the problem is a bit different from the ones mentioned above, as the two interrelated problems are the pickup and delivery design problem and the line haul design problem. The two problems can be handled in a mixed integer optimization model which allows a pure raster system (no hub transports) as well as the location of multiple hubs (depot-hub-depot traffic). The objective function includes distance related variable costs, fixed costs for depots and vehicles and the constraints represent all problem specific issues such as the number of hubs and depots, the assignments of postal zones to depots, the route sequences. However, the computational difficulty of the problem leads the authors to a solution methodology hierarchical in structure but which allows upgrades by feedback loops. The steps of the algorithm can be summarized as follows:

- Determining the number of depots as the initialization phase: even if there is lack of knowledge on the optimal number of depots, some intervals are examined to find a reasonable starting point.
- Determining the depot locations with regard to the demand densities: approximate methods such as Add or Drop algorithms can be used.
- Assignments of postal zones to depots: the assignment procedure can be made on the basis of both time-related and distance-related constraints which differ especially for the border areas. So it becomes useful to make different assignments in different iterations.
- With the depot locations and postal zone assignments, the flows between depots are known. The hub location problem can be formulated mathematically as a part of the general mixed integer problem, but it is not practical to solve that problem. So it is preferred that the hub-location decisions are taken with regard to problem specific knowledge.
- After all the direct and indirect transshipments are known, the depot costs are calculated and a feedback helps to return to the postal zone-depot assignments.
- Then the routing problems for the postal zones assigned to each depot are solved. Again a feedback is sent to the assignment phase.
- The overall system cost is evaluated. Then feedbacks to the initial phases such as the number of the depots and the location of the depots are considered to improve the solution by testing totally different scenarios.

The approach is used for the Austrian parcel delivery service problem and it is observed that significant improvements can be obtained with respect to the current situation.

Another study that deals with the location and routing issues in postal delivery systems is Çetiner (2003). It considers the Turkish postal delivery system and proposes an iterative solution technique combining the decisions of where to locate hubs and how to construct the related routes with each hub. In order to solve the problem, a two–phase approach (location-first, route-second) is studied. Different from a pure sequential technique, a feedback from the second phase (the routing phase) is used to update the distances in the first phase (the hub location and non-hub allocation phase). In the solution of the first phase Campbell's multiple allocation p-hub median formulation is used with some problem related modifications. For the second phase, the solution is obtained by Clarke and Wright savings heuristic followed by a VRP mathematical formulation. The iterative nature of the algorithm helps the two phases to be in interaction, so their solutions converge towards each other by reducing the objective function that represents the overall system costs. The proposed method is tested on a small problem leaving the extension of the method to the counties of Turkey or upgrading it to serve as a DSS with some additional features for further research.

Ambrosino and Scutellá (2005) deal with the integrated distribution network design problem. Their study includes a detailed literature survey. They consider a complex problem with several different kinds of nodes, different layers, both static and dynamic scenarios for a number of periods, and they are concerned with the facility location, transportation and inventory decisions. The general form of the problem consists of 4 layers; supply points (or plants), central depots (CDs), regional facilities (regional depots (RDs) or transit points (TPs)) and demand points. The goal of the problem can be summarized as:

- to locate CDs, RDs, TPs
- to allocate the demand points to the open facilities, allocate the regional facilities to central depots, allocate the central depots to supply points
- to determine the vehicle routes so that the capacity constraints are not violated and the demands are satisfied

- to determine the inventory levels at central and regional facilities.

Under the static scenario, a three-index formulation of the integrated distribution network design problem is given with some special assumptions which make the problem different from the general Warehouse Location Routing Problem (WLRP) with the routing decisions and the presence of a fixed cost for the use of vehicles. Then, in a similar approach with Hansen et al. (1994), the same problem is formulated by the use of flow variables and flow constraints. Also the dynamic scenario where a number of periods are analyzed is considered. Twelve instances are developed and the corresponding mathematical formulations are solved using the commercial code CPLEX 7.0. The cost of the best integer solution, the best lower bound (MIP bound) and the related CPU times along with the cost of the linear programming bound are given. As the gap between the best integer solution and the higher lower bound is significant in almost all cases; it is concluded that the use of some heuristic approaches to close the gap may be helpful.

In our study, we deal with a multistage LRP deterministic in demand nature and route duration among sites. Uncapacitated multiple facilities and capacitated multiple vehicles are elements of the environment. In addition to these characteristics, the problem we study on has constraints defining the time windows for customers and the shift times for facilities / vehicles. These time constraints are also the part of our environment that differ from all the studies mentioned above. As it is a multistage environment, a heuristic solution approach is considered.

2.2 Genetic Algorithms on Location and Routing Problems

As it could be seen from the first part of the literature review about LRPs, many different solution procedures have been proposed. The complexity of the problem environment and the high number of variables in both linear and non-linear mathematical models has led the researchers to concentrate on heuristics and metaheuristics.

Genetic algorithms are one of the metaheuristic search methods based on the mechanism of natural genetics. John Holland, his colleagues and students at the University of Michigan, have developed GA (Goldberg, 1989). It has been applied on a wide range of problems in various fields and has given satisfactory results. Although the applications of GAs on location-routing type problems are rare, applications on both location and location-allocation problems may form a basis for the application of GA on LRP. The application of GA in the following studies are on a wide range of location problems, so the following studies are not grouped according to the problem types, but presented in chronological order only.

Hosage and Goodchild (1986) present one of the first applications of the genetic algorithms on location problems; the problem type studied is p-median. The coding scheme used is binary coding, where '1' corresponds to open facilities and '0' corresponds to closed ones and it is supported by penalty function definition for the infeasible solutions. This representation and feasibility maintenance is the one that is not preferred by later studies in 2000s. Besides single point crossover operator and perturbation mutation, inversion operator is also invoked to replace some existing schema with different ones and to preserve diversity. The effects of some strategies and parameters are then discussed. In a 0-1 representation for the p-median problem, the initial population generation gains importance to maintain feasibility of chromosomes (the individuals more/less than p facilities are undesirable) and the penalty function deteriorates the natural flow of the algorithm. Although the use of problem-specific issues strengthens the best solution found through the runs, it is also declared that too many actions for fine-tuning, considering the problem structure, conflict with the nature of the GA and may constrain the ability of the GA. Without a detailed evaluation of the computational efficiency but with the above aspects emphasized, the results of the trial do not seem to be encouraging for later work.

Preston and Kozan (2001) develop a GA based solution approach for location issues in modeling a seaport system in order to determine the optimum storage locations of containers. A non-binary representation is used for the chromosomes where for each container there exists a gene that stands for the location the container is stored. The crossover operator is a simple single point crossover operator, however, repair operators are included as a different way of handling infeasibilities other than the penalty function. Another concept included in the algorithm is elitism, which results in carrying a certain percentage of the good individuals to the next generations to ensure that the best solutions are no worse than the previous generation. The proposed methodology is then applied on real data set at the Port of Brisbane and the results for different schedules are examined. Based on the results, the authors have concluded that it would be a useful tool for the port authority to determine the port traffic.

Jaramillo et al. (2002) discuss the use of GAs to solve location problems. Different types of problems such as Uncapacitated Fixed Charge Problem, Capacitated Fixed Charge Problem, Maximum Covering Problem and Competitive Location Problems are considered and the major differences in GA strategies like fitness function evaluation are outlined. For all problem types, a binary representation like the one in Su (1998) is used. Different from most studies in the literature, instead of simple one point or two point crossover operators, fitness-based fusion operator is used for crossover. To cope with infeasible individuals penalty function values are also included in the fitness evaluation. The GA heuristic is compared with some other heuristics in terms of both solution quality and time efficiency. For some problem types, the results indicate that GA-based heuristic is robust; however for other types, the computational times are excessively large.

Zhou, Min and Gen (2003) deal with a bi-criteria problem considering allocation of customers to multiple warehouses where both total shipping cost and total transit time between customers and warehouses are important criteria. Each candidate solution is represented by a chromosome of length equal to the number of customers and the genes of a chromosome stand for the warehouse that a customer is allocated to. The bi-criteria nature of the problem is considered in the fitness evaluation phase. The single point crossover and perturbation mutation operators are also supported by a problem-specific repair heuristic operator as in Preston and Kozan (2001) to deal with the capacity constraint of warehouses. The proposed methodology is then tested on real data from a firm that faces the problem in practice and the results are evaluated as successful.

Alp, Erkut and Drezner (2003) also give an example of the GA application on location problems. The uncapacitated p-median problem is discussed and the GA is customized according to the characteristics of the problem. Each solution in the population is represented by a p-length chromosome which gives the indices for facilities to be opened. By this representation, the risk of facing infeasible solutions with more/less than p facilities is diminished. The objective function value is used as the fitness of each chromosome. A significant detail is that the randomness in the initial population generation is somewhat under control and for each gene approximately the same frequency is aimed in the initial population for the gene pool not to be biased. In parent selection, no special rule but just randomness is considered. Another outstanding aspect of this work is the use of a problem-related crossover operator instead of classical one or two point crossover operators and the mutation concept is not in use. Besides containing problem specific issues, a heuristic procedure like "*drop algorithms*" works within the crossover operator. The GA is then tested on problem instances from OR library or from Alberta, Galvao and Koerkel problem sets (Beasley, 1990; Galvão and ReVelle, 1996; Koerkel, 1989) and it proves to be quite efficient by giving very good solutions within small computational times.

Salhi and Gamal (2003) introduce a GA solution for the uncapacitated continuous location-allocation problem. Continuous location-allocation problem is also an area where it is difficult to encounter GA solutions. The work of Salhi and Gamal improves the existing studies up to 2003 by adding some problem specific heuristics and some logical and promising aspects into standard procedures such as initial population generation or parent selection in the GA application. A binary representation scheme is not preferred as the problem deals with a continuous search space. So the uses of repair mechanisms and penalty functions which can usually deteriorate the efficiency of the algorithm are also avoided. Another mechanism called injection is employed within the heuristic. It is aimed to mimic the effect of immigration and to maintain diversity. They have tested their algorithm on a 110 instance set for four problem classes and they have seen that their algorithm outperforms the existing GA solutions in most instances.

Correa, Steiner, Freitas and Carnieri (2004) deal with the capacitated p-median problem. The representation of individuals is the same as in Alp, Erkut and Drezner (2003). However, the capacity constraint for the facilities adds another step to the fitness evaluation phase and this step, the assignment of customers to the open facilities represented in each chromosome, is handled with a heuristic developed by Correa (2001). A ranking-based selection method is applied for parent selection. The crossover operator contains some problem specific issues and a new hypermutation operator based on the problem being solved is proposed in this work. The results both with and without the hypermutation operator are compared with a Tabu Search (TS) algorithm and the results show that GA with the hypermutation operator outperforms TS. This emphasizes the effect of problem specific procedures on the solution quality.

Topçuoğlu et al. (2005) present a GA based solution technique for the uncapacitated hub location problem. A two-array representation is used for each chromosome; one being the *HubArray* and the second *AssignArray.* In the first array, each node is a potential hub and binary coding is used, as '1' represents a node that is a hub and '0' represents a non-hub node. The *AssignArray* is used to show which hub each spoke is assigned to. In initial population generation, some rules which seem very logical to obtain good solutions quickly are applied. These rules are applied on the decision of the number of hubs in each chromosome and the nodes that are most likely to be a hub regarding the flow of traffic across them. Fitness proportional selection with roulette wheel sampling for parent selection and one point crossover on both *HubArray* and *AssignArray* is applied. The problem of infeasibility is faced after the crossover operator; then a repair mechanism is developed. The results of the proposed algorithm are compared with the best results in the uncapacitated single allocation p-hub median problem literature, which are obtained by GATS (Genetic algorithm and tabu search) algorithm. The results demonstrate that the proposed algorithm surpasses the existing solution procedures for even larger problems.

Ko and Evans (2005) develop a genetic algorithm based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs. The dynamic supply chain management by 3PLs belongs to the class of multi-period, multi-stage, multi-commodity and capacitated location problems. As this kind of problems belong to the class of NP-hard problems and moreover there exist non-linear terms in the objective function, so a heuristic is developed to obtain good solutions in reasonable times. The representation scheme includes the decision variables in the problem (opening/closing of a warehouse/repair center; expansion decision for open warehouse/repair center and the amount of expansion) and an array stands for each period. After the information of open warehouses/repair centers is read from the chromosomes, the optimal assignment of customers to open facilities is realized by a simplex transshipment algorithm and fitness values are calculated. Besides the representation seems to be appropriate for the problem structure, the drawback of the representation is that it allows to represent infeasible solutions, the infeasibility being related to the capacity limits on warehouse/repair centers and this leads to consider the penalty function. Binary tournament selection for the selection of parents, cloning operator to apply elitism, two-point crossover operator and perturbation mutation operator for reproduction are used. The proposed method is tested on a base-line case which is small when compared to the real cases. For the comparison of the results, the non-linear mathematical model is converted into a linear one with the use of additional variables and the same base-line case is also solved on this linear model. The GA based heuristic gives solutions in small gaps to the optimal solution and moreover, good solutions can be obtained in some instances which can not be solved optimally.

Other than the ones presented above; some studies on the application of GAs to the Vehicle Routing Problem with Time Windows (VRPTW) and the use of different types of operators, both problem specific or not, are also examined to gain further insight.

Starkweather, McDaniel, Mathias and Whitley (1997) discuss six sequencing operators: enhanced edge recombination, two different versions of order crossover, partially mapped crossover, cycle crossover and position based crossover. Each of those operators is then used to solve a 30 city TSP and the results are compared in terms of the value of the best solution and the average of objective values obtained by a number of runs. The use of those operators on a Warehouse/scheduling problem instead of traveling salesman problem gives a different ranking between the operators as the critical information is not adjacency but critical order. They have concluded that the goodness of an operator depends on the information carried by that operator and the importance of that information to determine the solution.

Thangiah et al. (1999) deal with the vehicle routing problem with time deadlines and propose two solution methodologies; one based on Local Algorithms and another based on Genetic Algorithms. In the GA based method, the customers are first clustered using the Genetic Sectoring Heuristic and then they are routed using the cheapest insertion method. The two algorithms are tested on a number of problem sets and the genetic based algorithm is proved to perform well for uniformly distributed customers and tight deadlines.

Blanton and Wainwright (1999) also deal with the vehicle routing problem with time windows and capacity constraints. The outstanding points related with this study are the design of new, problem specific crossover operators and the decoding approach for multiple vehicles. It is a study where the use of problem specific knowledge is emphasized and those operators are shown to outperform other standard operators.

Although the GAs are applied to a variety of problems, satisfactory studies for the LRP or location-routing problem with time windows (LRPTW) are very rare. We have decided to propose a solution methodology for the LRPTW based on GA to evaluate the efficiency of the metaheuristic for this type of problem and to examine how GAs can make any contribution for the solution of such a complex environment.

CHAPTER 3

MATHEMATICAL MODELING OF THE LOCATION-ROUTING PROBLEM

This chapter begins with a general description of the Location-Routing Problem (LRP). The introduction to the model characteristics is then followed by the assumptions in this study, the verbal definition of our model in detail and the related mathematical models.

3.1 LRP and the Problem Environment in the Study

LRP can be defined as follows: The number, locations and demands of customers, the number and locations of all potential sites and the vehicle fleet properties are given. In LRP:

(i) the demand of each customer is to be satisfied,

(ii) the capacities of both vehicles and facilities are not to be exceeded,

(iii) each route is to begin and end at the same depot,

and the total system cost considering both facility and transportation related costs is to be minimized. The decisions to be made are the locations of the facilities to open, the allocation of customers to the open facilities and the routes for each open facility.

The LRP is a complex problem. It is easily observed that with some extensions and / or reductions, it combines various components of different problems such as location, allocation or routing problems.

Besides this general definition; with the help of some assumptions at the beginning, we define the characteristics of the LRP environment in this study as follows:

Characteristics of the LRP Environment

- A multistage LRP is considered and the facilities to be located are secondary facilities, which are intermediaries between the supply sources and many demand points (customers).
- For the inbound transportation, there is no routing consideration. The pricing is on per-item per-unit distance basis, as if all requirements from a supply source to a facility are transported by a single vehicle with a very large capacity.
- The supply sources are already located and they are uncapacitated.
- In outbound transportation, the deliveries are realized via routes consisting of multiple deliveries to multiple customers.
- The customer demands are deterministic.
- Each customer is to be delivered by a single vehicle and a single visit.
- Multiple potential facility sites and multiple vehicles are allowed to be used.
- The facilities to be opened are uncapacitated; however, the fixed cost of a facility differs based on capacity.
- A single facility of any size is allowed at a potential facility site.
- Vehicles are capacitated and of one type; that's all the vehicles have a constant capacity and fixed cost. There is no limit on the number of vehicles in the system.
- Shift time restrictions exist on the facilities to be opened.
- Time windows exist for customers. Both hard and soft time-windows are considered, but the study focuses mainly on hard time-windows. It is possible to handle the soft time-window case with only a slight change in the routing part of the approach.

- Waiting times are allowed at both customers' and facilities' sites; with no penalty cost (however, the models are constructed so that they are able to cope with a unit penalty cost for waiting).

3.2 Verbal Description of the LRP Studied

The verbal description of the problem is given in terms of the objective function, the system constraints, the problem parameters and the decisions to be made.

Objective Function

To minimize the total system costs including the fixed and variable facility costs, fixed vehicle costs, the inbound transportation costs from the supply source to the intermediate facilities and the outbound transportation costs from the intermediate facilities to customers.

System Constraints

- Each customer is served from a single facility (single sourcing) and in one visit only.
- The capacities of vehicles are not exceeded.
- Each vehicle is used once and each tour starts and ends at the same facility.
- The shift times that the facilities are allowed to operate within should be obeyed. The vehicles also operate within the working hours of the facilities.
- The time windows that the customers are to be delivered within should be considered (though in the soft time-window case, they are less restrictive).

Parameters

- The locations and the demand requirements of customers
- The locations of potential facility sites
- The locations of supply sources
- The fixed cost for vehicles

- The breakpoints in the capacity of different types of facilities and the fixed costs for those facility types
- Variable operating unit cost of facilities
- The cost terms related to transportation
- The time windows for customers and shift times for facilities
- The required service times for customers (for unloading etc.)

Decisions

- The potential sites selected for locating the facilities
- The type (size) of facilities to be located at those sites
- The allocation of customers to the open facilities
- The routing of customers that are allocated to the same facility
- The waiting times at customer and facility sites, in order to satisfy the time windows

3.3. Mathematical Models of the LRP Studied

The mathematical model for the LRP as defined above turns out to be non-linear due to the existence of shift time constraints for the potential facility sites and the time windows for the deliveries to the customers. In this section, the non-linear mathematical models are developed first, but later they are linearized.

Mathematical formulations of an LRP with time windows differ, depending on whether the time windows are hard or soft. Both scenarios are considered in this section. Hard time-window where delivery out of time window is not allowed and soft time-window where delivery out of time window is allowed but with penalties for early/late deliveries are considered.

We have extended the basic formulations of Perl and Daskin (1984) and Wu, Low, Bai (2002) with regard to the types of costs included in our environment and to the time window issue.

3.3.1 Non-linear Models

3.3.1.1 LRP with hard time-windows

The nonlinear model for the case where the time windows are hard is developed as follows. First the sets, parameters, decision variables are presented; then the mathematical formulation is given.

Sets

I	set of customers	$i = 1,\ldots,n$
J	set of potential sites	$j = n+1,\ldots,m$
S	set of supply sources	$s = m+1,\ldots,p$
K	set of vehicles	$k = 1,\ldots,r$
L	set of potential facility types (according to the capacity)	$l = 1,\ldots,t$

Parameters

D_i	demand of customer-i (unit)
d_{gh}	distance between sites g and h ; $g \in I \cup J \cup S, h \in I \cup J$
CV	capacity of a vehicle (unit)
C_l	capacity of a potential site of type-l (unit)
FC_l	fixed cost of a potential site of type-l
fc	fixed cost of a vehicle
VC	variable cost of a potential site (per unit)
TC	transportation cost from any facility site to any customer or between customers (per unit distance)
tc	transportation cost from any supply source to any facility site (per unit distance per item)
h	the time it takes for a vehicle to cross a km
e_g	the earliest service starting time of site-g ; $g \in I \cup J$
se_g	the required service time of site-g; $g \in I \cup J$
la_g	the latest service time for site-g ; $g \in I \cup J$

w_g the cost of waiting at site-g per unit time; $g \in I \cup J$

N number of customers

Decision Variables

$$X_{ghk} = \begin{cases} 1 & \text{if g immediately precedes h in route of vehicle - k} \\ 0 & \text{o / w} \end{cases} \qquad g,h \in I \cup J \quad k \in K$$

$$Z_{jl} = \begin{cases} 1 & \text{if a facility of type - l is located at site - j} \\ 0 & \text{o / w} \end{cases} \qquad j \in J \quad l \in L$$

$$ZS_j = \begin{cases} 1 & \text{if a facility (of any type) is located at site - j} \\ 0 & \text{o / w} \end{cases} \qquad j \in J$$

$$Y_{ij} = \begin{cases} 1 & \text{if customer- i is assigned to DC at site - j} \\ 0 & \text{o / w} \end{cases} \qquad i \in I \quad j \in J$$

$$V_k = \begin{cases} 1 & \text{if vehicle- k is used} \\ 0 & \text{o / w} \end{cases} \qquad k \in K$$

q_{sj} = quantity shipped from source-s to facility site-j $s \in S, j \in J$

wt_g = waiting time at site-g $g \in I \cup J$

at_g = arrival time at site -g $g \in I \cup J$

rt_j = returning time to facility site-j $j \in J$

U_{ik} = auxiliary variable (for subtour elimination constraints) $i \in I, k \in K$

Model

Minimize

$$\sum_{l\in L}\sum_{j\in J} FC_l\ Z_{jl} + \sum_{k\in K} fc\ V_k + \sum_{j\in J} VC\left(\sum_{i\in I} D_i\ Y_{ij}\right) + \sum_{s\in S}\sum_{j\in J} tc\ q_{sj}\ d_{sj} + \sum_{k\in K}\sum_{g\in I\cup J}\sum_{h\in I\cup J} TC\ d_{gh}\ X_{ghk} + \sum_{g\in I\cup J} w_g\ wt_g \tag{3.1}$$

subject to

$$\sum_{k}\sum_{\substack{h\in I\cup J\\ h\neq i}} X_{hik} = 1 \qquad \forall i\in I \tag{3.2}$$

$$\sum_{i\in I} D_i \sum_{\substack{h\in I\cup J\\ h\neq i}} X_{hik} \le CV\ V_k \qquad \forall k\in K \tag{3.3}$$

$$U_{lk} - U_{jk} + N\ X_{ljk} \le N-1 \qquad \forall l,j\in I, \forall k\in K \tag{3.4}$$

$$\sum_{\substack{g\in I\cup J\\ g\neq h}} X_{hgk} - \sum_{\substack{g\in I\cup J\\ g\neq h}} X_{ghk} = 0 \qquad \forall h\in I\cup J, \forall k\in K \tag{3.5}$$

$$\sum_{j\in J}\sum_{i\in I} X_{ijk} \le 1 \qquad \forall k\in K \tag{3.6}$$

$$\sum_{s\in S} q_{sj} - \sum_{i\in I} D_i\ Y_{ij} = 0 \qquad \forall j\in J \tag{3.7}$$

$$\sum_{i\in I} D_i\ Y_{ij} - \sum_{l\in L} C_l\ Z_{jl} \le 0 \qquad \forall j\in J \tag{3.8}$$

$$\sum_{l\in L} Z_{jl} = ZS_j \qquad \forall j\in J \tag{3.9}$$

$$-Y_{ij} + \sum_{u\in I\cup J}(X_{iuk} + X_{ujk}) \le 1 \qquad \forall i\in I, \forall j\in J, \forall k\in K \tag{3.10}$$

$$at_i\ X_{ijk} + wt_i\ X_{ijk} + se_i\ X_{ijk} + d_{ij}\ h\ X_{ijk} \le rt_j \qquad i\in I,\ j\in J, k\in K \tag{3.11}$$

$$at_g\ X_{gik} + wt_g\ X_{gjk} + se_g\ X_{gik} + d_{gi}\ h\ X_{gik} \le at_i \quad g \in I \bigcup J, i \in I, k \in K \quad (3.12)$$

$$at_g + wt_g \ge e_g \qquad \forall g \in I \cup J \qquad (3.13)$$

$$at_i + wt_i \le la_i \qquad \forall i \in I \qquad (3.14)$$

$$rt_j \le la_j ZS_j \qquad \forall j \in J \qquad (3.15)$$

$$at_j = 0 \qquad \forall j \in J \qquad (3.16)$$

$$X_{ggk} = 0 \qquad g \in I \bigcup J, k \in K \qquad (3.17)$$

$$X_{ghk}, Z_{jl}, ZS_j, V_k, Y_{ij} \ \ binary \qquad g,h \in I \bigcup J, k \in K, i \in I, j \in J, l \in L \qquad (3.18)$$

$$q_{sj}, at_g, wt_g, U_{ik}, rt_j \ge 0 \qquad s \in S, j \in J, g \in I \bigcup J, i \in I, k \in K \qquad (3.19)$$

The objective function (3.1) minimizes the sum of fixed warehouse costs, fixed vehicle costs, variable warehouse costs, transportation costs from supply sources to facilities, routing costs for deliveries from facilites to customers and a penalty term which stands for the waiting times of vehicles at nodes (it is also possible with this model to handle the case where there is no penalty cost for waiting by $w_g = 0\ \forall g$).

Constraint (3.2) requires that each customer is assigned to a single facility and a single vehicle. Constraint (3.3) assures that a customer can be assigned to a vehicle if that vehicle is being used and the capacity of that vehicle is not exceeded by the customers assigned to that vehicle. Constraint (3.4) is the sub-tour elimination (connectivity) constraints. Constraint (3.5) assures that every customer/facility node entered is left by the same vehicle. Constraint (3.6) assures that a vehicle can not be operated from multiple warehouses. Constraint (3.7) specifies that the amount delivered from a facility to customers is equal to the amount delivered to that facility from suppliers. Constraint (3.8) requires that the prespecified capacity of any opened facility is not exceeded or in other words the size of the facility to be opened is sufficient to satisfy the demands of the assigned customers. Constraint (3.9) assures that at most a single facility of a single type is opened at a potential site and the status of all types of potential sites determine the usage of that site. Constraint (3.10) requires that a customer can be assigned to a facility only if they belong to the same vehicle. Constraint (3.11) is used to evaluate the returning time to a facility to be later used in the shift time constraints. The time interval up to return to the facility includes the service times, the waiting times at the customers and the time spent on the route between nodes. Constraint (3.12) defines the arrival time of the vehicle at each customer site. Constraints (3.13), (3.14) and (3.15) assure that the service starting time for each node and the time for returning back for the facility sites is between the predefined time windows. Constraint (3.16) consists of an equation for each potential site that represents the depots as starting points of each route. Constraint (3.17) prevents any meaningless route from a node to itself. The last groups (3.18) and (3.19) define the variables' types used in the mathematical model. In the model, the two constraints (3.11) and (3.12) cause the nonlinearity.

3.3.1.2 LRP with soft time-windows

The non-linear model of the system with soft time-windows is given below. The sets are the same as in the hard time-window case. The additional parameters and decision variables are presented below.

<u>Additional Parameters</u>

pen1 penalty cost for starting the delivery before the earliest service starting time (penalty for earliness)

pen2 penalty cost for starting the delivery after the latest service starting time (penalty for lateness)

<u>Additional Decision Variables</u>

lte_i = the earliness of a delivery (the time interval between the service starting time and the predefined earliest delivery time) $i \in I$

ltt_i = the tardiness of a delivery (the time interval over the predefined latest delivery time) $\quad i \in I$

(it is assured by the constraints that both earliness and tardiness are equal to zero if the delivery is realized within the time window)

<u>Model</u>

Minimize

$$\sum_{l\in L}\sum_{j\in J} FC_l\, Z_{jl} + \sum_{k\in K} fc\, V_k + \sum_{j\in J} VC \left(\sum_{i\in I} D_i\, Y_{ij}\right) + \sum_{s\in S}\sum_{j\in J} tc\, q_{sj}\, d_{sj} + \sum_{k\in K}\sum_{g\in I\cup J}\sum_{h\in I\cup J} TC\, d_{gh}\, X_{ghk} + \sum_{g\in I\cup J} w_g\, wt_g + \sum_{i\in I\cup J} lte_i\, pen1 + \sum_{i\in I\cup J} ltt_i\, pen2 \tag{3.20}$$

subject to

Constraints (3.2) – (3.12)

Constraints (3.15) – (3.18)

$$ltt_i \geq at_i + se_i - la_i \qquad \forall i \in I \tag{3.21}$$

$$lte_i \geq e_i - at_i \qquad \forall i \in I \tag{3.22}$$

$$q_{sj}, at_g, wt_g, U_{ik}, rt_j, lte_i, ltt_i \geq 0 \qquad s \in S, j \in J, g \in I \cup J, i \in I, k \in K \tag{3.23}$$

There are just few differences between the hard and soft time-window models that are described below:

- The model with soft time-windows has two new penalty costs for earliness and tardiness of deliveries.
- The objective function value has two additional terms; the total additional cost for the early deliveries and tardy deliveries to the customers.
- Equation (3.21) sets the lower bounds on ltt_i; the time that passes between the delivery and the latest service time. In case the delivery is realized

before the latest delivery time, value of ltt_i becomes negative, but it is guaranteed to be greater than zero in the model.

- Equation (3.22) stands for the same purpose for lte_i ; it defines the lower bound on the amount of time between the delivery and the earliest service time. If the delivery is realized after the earliest delivery time, this value is again negative, that is zero.

3.3.2 Linear Models

As it can be observed from the above models, the non-linearity arises from the time window and shift time constraints. The definitions of the arrival time to a customer and returning time to a facility depend on the sequence of nodes that follow each other on the route and this situation is expressed with non-linear constraints. The models are linearized below. The variables and constraints that have been changed/added due to the linearization are given in ***bold italic.***

3.3.2.1 LRP with hard time-windows

All the sets, parameters and decision variables are valid for the linear models. In addition to them the following are defined:

M a big constant

$\boldsymbol{T_{ghk}}$ = auxiliary binary variable $g,h \in I \cup J,\ k \in K$

The mathematical model for the hard time-window case is the same as the model in section 3.3.1.1 except constraints (3.11) and (3.12). Constraint (3.11) will be replaced by the set of inequalities defined by ***(3.24a)*** to ***(3.24c):***

$$at_i + se_i + wt_i + d_{ij}\ h - rt_j \le MT_{ijk} \qquad \forall i \in I, \forall j \in J, \forall k \in K \qquad \textbf{\textit{(3.24a)}}$$

$$\text{-}\,at_i - se_i - wt_i - d_{ij}\ h + rt_j \le MT_{ijk} \qquad \forall i \in I, \forall j \in J, \forall k \in K \qquad \textbf{\textit{(3.24b)}}$$

$$X_{ijk} \leq M\ (1 - T_{ijk}) \qquad \forall i \in I, \forall j \in J, \forall k \in K \qquad \textbf{\textit{(3.24c)}}$$

Constraint (3.12) is replaced by the set of equations defined by ***(3.25):***

$$at_g + se_g + wt_g + d_{gi}\ h - at_i \leq M\ T_{gik} \qquad \forall g \in I \cup J, \forall i \in I, \forall k \in K \qquad \textbf{\textit{(3.25a)}}$$

$$-at_g - se_g - wt_g - d_{gi}\ h + at_i \leq M\ T_{gik} \qquad \forall g \in I \cup J, \forall i \in I, \forall k \in K \qquad \textbf{\textit{(3.25b)}}$$

$$X_{gik} \leq M\ (1 - T_{gik}) \qquad \forall g \in I \cup J, \forall i \in I, \forall k \in K \qquad \textbf{\textit{(3.25c)}}$$

3.3.2.2 LRP with soft time-windows

The model with soft time-windows can be linearized in the same way as the model with hard time-windows is linearized. The only point is that the non-linear equivalents of the returning times (rt_j) to depots and the arrival times (at_i) at the customers are defined by three inequalities which maintain the linearity.

The linear soft time-window model is as in section 3.3.1.2 except that constraint (3.11) is replaced by the group of constraints (3.24a) to (3.24c) and (3.12) is replaced by the group of constraints (3.25a) to (3.25c).

3.4 Validation of the Mathematical Models

The formulation of the environment is followed by modeling the environments in the General Algebraic Modeling System (GAMS) which is capable of solving both MIP and NLMIP with the hope that the results of the mathematical model also help in evaluating the results of our genetic algorithm. However, for the non-linear model, as the version of the solver for the NLMIPs is in demo version, GAMS is capable of solving only problems up to 50 variables. In addition, for the rest of the study, it is decided to use CPLEX to solve the linear programs. So the models (both linear and non-linear for hard and soft time windows) translated to the GAMS language, which are given in Appendix A, are only used to solve very small sample problems to validate the mathematical models, that's to see if the objective function and the constraints are formulated so that they define the environment as desired.

CHAPTER 4

THE SOLUTION METHODOLOGY

LRPs are classified within the general class of network optimization problems (Golden et al., 1981) and within the family of arc-node problems (Schrage, 1981) since their solutions are constituted from selection of a number of nodes and arcs on a network. These problems are proved to be NP-hard with a great number of variables and constraints. Practical-size problems can not be solved to optimality based on their mathematical formulations. Especially in our case where time windows/shift times exist for nodes, it becomes much more difficult to solve the problem through a mathematical programming approach.

After reviewing the solution approaches in the literature for LRPs, we have decided to base our solution methodology on GA.

In this chapter, this methodology is explained with all its details. At the beginning, some basic concepts and descriptions related to GA are given. While explaining the methodology that we propose for the solution of the problem, discussions on different phases and parameters of GA applications are included as well. In addition to those, while developing our algorithm, the solution of the VRPTW problem has gained great importance, and hence we have paid a significant effort for the solution of this subproblem in our GA approach. A brief explanation on the solutions of VRP / VRPTW in the literature and how we handle this issue in our algorithm is also covered in this chapter.

4.1 Fundamentals of Genetic Algorithms

Genetic Algorithms are search methods based on the genetic processes in the nature. They were first developed by Holland and his colleagues at Michigan University (Goldberg, 1989) and the basic principles were first published in Holland (1975). The fundamentals of GA which are in a direct analogy with the nature can be summarized as follows:

- In nature, the individuals in a population compete with each other to survive and to reproduce. It is known that according to the evolution theory, better individuals have higher chance to stay alive and pass their genetic material to next generations. The GA applies to populations which are composed of individuals in the form of chromosomes. The individuals (or chromosomes) are in fact solutions for the problem represented in various ways. The chromosomes are made up of genes, each standing for a decision variable of the solution.
- The probability of survival and reproduction is generally determined by how good an individual is in nature. The value which represents the goodness of a solution is the objective function value for the problem with the parameters represented in the solution. This is the fitness function value for a chromosome. As it is explained later, the fitness function may not be purely the objective function, but more than that.
- The initial genetic material is included in a number of individuals generated at the beginning, which constitutes the initial population. This population is then updated with processes similar to the processes of reproduction in the nature. During the reproduction processes, the parent selection phase determines the genetic data that is passed to the next generation among the whole data present in the population. Operators like crossover and mutation treat those genetic materials and generate a new offspring. Each population updated with one/more offspring is called a generation.

Like in nature, the genetic process evolves without knowing anything about the problem type that is being solved. However, along with applying the general fundamentals, it is shown that the use of problem specific knowledge to guide the operators towards better strings helps to improve the solutions obtained. Those operators designed with the help of the problem specific data were first used in problem types like TSP where the general operators were not sufficient.

Although there exists a general framework to apply genetic algorithms as a solution procedure to different problems, there are some key components to be decided on. The components which are the fundamentals of genetic algorithms at the same time are as follows:

- A genetic representation for the possible solutions in the search space
- A way to generate the initial population
- Fitness function evaluation for the chromosomes
- Genetic operators and techniques such as the parent selection technique, the crossover and mutation operators and the replacement strategy that is followed while updating the existing generation with the offspring
- Parameters for the genetic algorithm procedure such as the population size, crossover and mutation rates, maximum number of offspring to be generated

4.2 The Proposed Solution Methodology and GA components

The GA theory and different aspects related to each of the above components are explained under separate headings. How those components are adapted to the solution procedure for LRP with time window is also discussed in detail.

4.2.1 Chromosome Representation

Chromosome representation is a very critical issue in the success of the GA. An appropriate representation must be capable of representing any possible solution for the problem and at the same the representation scheme must not support to include the infeasible solutions in the population if it is possible to do so. The evolution of the representation scheme for the p-median problem is a good example of the aspect mentioned above. In early studies the representation which shows the open and closed facilities but which does not restrict the number of facilities is used and this brings out the solutions with more/less than p facilities into the population. However, in later studies, representation schemes which consider only solutions with p facilities are preferred.

The main decision is whether to use a binary or integer scheme for the representation. The binary coding consists of 0s and 1s. These values may be sufficient to represent the solution or any integer value may be converted to base 2 from base 10 and bit strings of 0/1s are used as chromosomes. As the fundamentals of GA and the genetic theory are founded on binary representation and it is easy to apply most operators on this kind of representation, it is used for most types of problems.

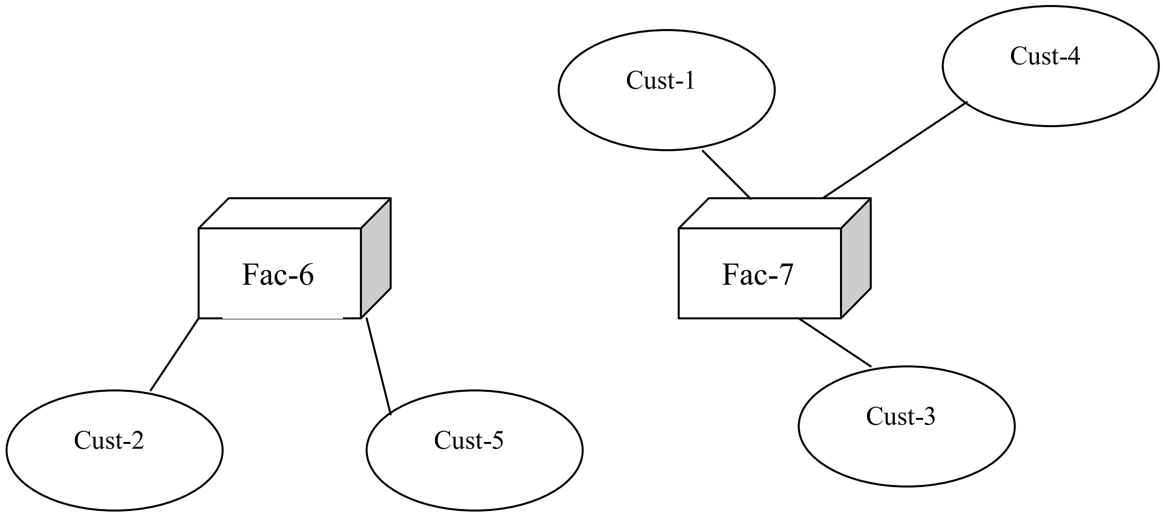
However, for certain problem environments, a binary representation would not be sufficient and the work of converting the solution to base 2 may be inefficient. Then the use of integers for each gene becomes more appropriate. It is also an obligatory task to design the genetic operators according to the non-binary representation of chromosomes.

Depending on the problem environment and its constraints, sometimes use of non-binary representation can be a solution to the problem of infeasible individuals. However, for cases where it is still possible to generate infeasible individuals in initial population or by the genetic operators with an integer representation, there exist a number of ways to handle the infeasible individuals in the GA literature:

- Adding a penalty function to the fitness of an infeasible chromosome so that the infeasibility incurs a noticeable cost in the fitness value and thus the possibility of being chosen for mating is decreased
- Design of heuristic operators which transform infeasible solutions into feasible solutions and applying them before an offspring is included in the population
- Exploring a two-population genetic algorithm such that while the aim in one population is maximizing/minimizing the objective value, the aim in the other population is minimizing the infeasibility

The third method above is proposed recently and therefore not applied extensively in the literature. The first two methods are common wherever there is the probability of obtaining infeasible individuals. If there are infeasible individuals through the execution of the genetic algorithm, to apply one of these methods is inevitable. However, there is a general consensus as to the negative effect of such approaches to the efficiency of the overall algorithm.

For the LRP that we study, the non-binary chromosome representation is chosen. For an *n*-customer, (*m*-*n*) potential facility site problem, each chromosome will consist of n genes and each gene takes a value ranging from n+1 to m (as the facility sites are numbered as nodes after the customers) which shows the potential facility site this customer is assigned to in that solution. The figure below illustrates the representation scheme for a problem of 5 customers and 2 potential facility sites.



Customer no	*1*	*2*	*3*	*4*	*5*
Assigned Facility no	*7*	*6*	*7*	*7*	*6*
Chromosome	7	6	7	7	6

Figure 4.1 The Representation Scheme of a Solution in the Proposed GA

The LRP with time windows could be treated as consisting of 3 sub-problems; the location of facilities, the assignment of customers to the open facilities (this problem is actually two-stage, but the assignment of facilities to suppliers is straightforward as the suppliers are assumed to be uncapacitated in our problem environment and each facility could be assigned to the nearest supplier) and the routing of the customers assigned to a facility considering all time-related constraints.

With our representation the first two decisions (location-allocation) of a solution could be read from the chromosome itself. Su (1998) that deals with the same kind of problem has chosen to represent each solution by chromosomes only informing us about the potential facility sites that will be opened and then assigns each customer to its nearest facility. The routing phase is also solved sequentially. However the routing considerations led us to think that there may exist solutions good enough to consider and to improve even if customers are not assigned to the nearest facility and the information of "*which customer will be assigned to which facility*" is critical than the information of "*which facility will be opened*" in a location routing problem.

A further step might be to represent the location, allocation, and moreover the routing decisions in the chromosome. But when the huge number of possible solutions containing all of these information is considered, it is obviously seen that any population size could hardly be capable of covering the solution space and there may be very poor solutions included in the population. It is not possible to eliminate those chromosomes in advance, because the violation of time windows is only visible when the whole route is constructed. The above representation is decided to be the most suitable one in our study, considering all those aspects and paying effort to find a representation which may be possible to handle in terms of length, and at the same time which may carry the greatest amount of information related to a solution.

4.2.2 Initial Population Generation

The initial population for GA is the first group of solutions among which the search begins. As declared in Reeves and Rowe (2003), the point in generating the initial population is that "*every point in the search space or in other words any solution to the original problem could be reached from the solutions in the initial population by crossover only*" and this could only be satisfied by the existence of each possible value for each gene in the initial population. This emphasizes the importance on the way the initial population is generated.

The most common way of generating the initial population is doing this randomly without any control on the existence of alleles for genes. While this approach is in accordance with the stochastic nature of the GAs, individuals generated in this way do not necessarily cover the solution space and the above principle can not be satisfied with this random approach. So the use of some control routines on the chromosomes and more sophisticated statistical methods will provide advantages especially with non-binary representations. An example method could be the generalization of the Latin hypercube (Reeves and Rowe, 2003).

There are some other strategies to begin with a better initial population. One strategy could be to generate a number that is twice or three times the population size and then use the best "population size" group of individuals as the initial population. Another could be to insert the known good solutions directly to the population and generate the remaining solutions randomly. But there is always the possibility of getting trapped in a certain region of the solution space and premature convergence in these attempts.

Above all, the second important aspect to satisfy the idea by Reeves and Rowe (2003) mentioned in the first paragraph is working with an appropriate population size.

Considering all the discussions in the literature related with the issues above and the characteristics of the problem, our "Initial Population Generation" phase is composed of a number of small sub-algorithms covering the topics below:

<u>Determine the feasible assignments regarding the shift times for facilities and the time windows for customers</u>

GA is designed to find good solutions for problems without knowing the problem characteristics, but including problem specific knowledge in the solution method via some operators is an important tool to improve the efficiency of the GA application. In problem types such as the one we are concerned with, when the population size needed to cover the whole search space is very large, it is more critical to narrow the solution space according to our constraints.

Related with the vehicle routing problem that is solved for every open facility, the time windows and shift time information are very restrictive in order to obtain good solutions including all the decision variables. Although the final arrival and departure times of customers can be revealed after the VRPTW is solved, there are some assignments which can easily be shown to be infeasible, considering the cases below:

(i) A customer can not be reached from a facility and served before its latest service time, even if a vehicle directly goes from that facility to that customer, in this case it is obvious that any routing scenario including other customers can not be within the time window of that customer.

(ii)The sum of the direct shipment duration from a facility to a customer, the service time at the customer and the duration for the return of the vehicle from the customer back to the facility exceeds the desired shift time value for the facility.

(iii)Even if a vehicle starts service of a customer at the earliest service starting time, serving him and returning back to the facility may exceed the shift time. (it is supplementary to the second condition; (arrival + service + return time) may seem feasible, but if arrival is before the earliest service time of a customer, then the vehicle must wait before starting service and (earliest starting time + service + return time) may not be feasible.)

So it is necessary to eliminate solutions including these infeasible assignments. In order to provide this, for each customer, the set of facilities this customer can be assigned to is determined before the chromosomes are generated randomly.

Randomly generate individuals as many as the Population Size

Using the elimination criteria above ((i),(ii),(iii)), “the set of facilities each customer can be assigned to” or in GA terms “the values each gene can take” are determined. By randomly choosing a feasible facility from that set for each customer, a chromosome can be generated. This process is repeated as many times as the population size.

Check if there exist duplicates of individuals in the population and create a population of non-identical individuals

As the population size is limited and the idea is to search as much of the region of the solution space as possible to have good solutions, it is preferred to include non-identical individuals to maintain the diversity in the population. So the identical individuals are determined, the copies other than the first one are deleted from the population, and new, non-identical individuals are inserted into the population till the population size is reached.

It must be noted that for problems, which have a very large solution space, to obtain duplicates in the population is very unlikely. In such cases, it is worth considering whether to include this sub-routine or not because of the computational effort it requires.

Check whether every feasible value for each customer exists in the population, and if not, satisfy this requirement

After the population composed of non-identical individuals is generated, it should be checked whether the population can satisfy the idea at the beginning of this section, that is, the idea of existence of each possible value for each gene. Then, for each customer, it must be checked whether all the facilities in the feasible set for that customer are included in the initial population. If there are feasible potential facilities that the customer is not assigned to in any chromosome, then any other facility that the customer is assigned to more than once is chosen randomly and it is exchanged with the unused facility. Doing this check for every customer provides us to begin with an initial generation that is able to produce any solution possible (the infeasible ones being eliminated) for the problem.

4.2.3 Fitness Function Evaluation Including VRPTW

The fitness function value for a chromosome is generally the objective function value for the solution the chromosome represents. The objective function value can be easily calculated with the help of the data directly read from the chromosome by decoding the chromosome or some intermediate calculations may be needed using the data read from the chromosome to obtain a complete solution and to reach the final objective function value; i.e. in a capacitated p-median as in Correa et al. (2004), the chromosome represents the p open facilities and the allocation to the capacitated facilities is realized by an algorithm developed by Correa et al (2004).

In cases where the chromosomes can represent infeasible solutions as well as the feasible ones and the way to handle the infeasibilities is to use the penalty function, a penalty function value is added to the objective function value to obtain the fitness value. The penalty for feasible chromosomes is equal to 0. The values for the infeasible chromosomes are calculated according to how much the constraints are violated.

In our solution, the objective function consists of the following terms:

- the fixed costs of facilities
- the variable costs of facilities
- the transportation costs from supply sources
- the fixed costs of vehicles
- the costs incurred in routes from facilities to customers

By decoding a chromosome, the facility each customer is assigned to is directly read. The facilities to which at least a customer is assigned are assumed to be open and the total flow of a facility is the sum of the demands of the customers assigned to that facility. So the fixed and variable costs related to the facilities are easily calculated. As the supply sources are uncapacitated, the supply sources and the facilities can be easily linked by assigning each open facility to the nearest supply source and this gives the transportation costs from the supply sources to the facilities. The only cavities in the objective function are the routing related costs: the fixed costs of the vehicles calculated based on the number of vehicles and the routing costs.

It is observed that in many studies proposing even integrated solution approaches for LRP using heuristics or metaheuristics in the literature, it is assumed that there is direct shipment between the facilities and the customers while exploring neighborhoods. It is assumed that a good solution regarding the direct shipment costs is also a good solution when the customer demands are not directly shipped but routing is considered. But in our solution method, in order to be able to evaluate the real cost of a solution and to maintain integrity in the solution, we prefer to handle the VRPTW instead of replacing the routing related costs with approximate costs.

However, the difficulty faced during the study is that, although VRPTW is not the main concern, and we focus on a general solution framework for the location-routing problem with time windows (LRPTW), the efficiency of the VRPTW solution has a great impact on the quality of the LRPTW solution. Related with our goal, to develop a heuristic which is guaranteed to give good solutions for VRPTW is not among our purposes. But the VRPTW being a very difficult problem to solve and has a significant effect on the LRPTW solution, it has to be handled with great emphasis. As a great number of VRPTWs is solved within the above approach (for each open facility on each gene in the initial population and for each open facility on each gene generated during the run), the solution method for the VRPTW has to be efficient in terms of computational time.

With these requirements, the two alternatives below are evaluated to handle the VRPTW issue.

- To use some simple heuristics which may provide some upper bounds on the VRPTW solution

- To use a detailed, well-defined approach which is proved to be successful to find the best solution in a short computational time

In order to find a satisfactory solution, first the solution approaches in the literature for VRP and VRPTW are reviewed and briefly discussed below.

For the VRP and TSP, which are NP-hard problems, any exact solution method does not give satisfactory results due to the great computational effort required. Instead, a number of heuristic approaches have been proposed in the literature. Nelson et al. (1985) and Laporte (1992) are studies which provide an overview of both exact and approximate VRP algorithms. The Clarke and Wright heuristic for the capacitated VRP (CVRP) (Clarke and Wright, 1964) which has been studied later and implemented to a number of environments, the Sweep Algorithm by Gillett and Miller (Gillett and Miller, 1974) and the two- phase approach by Christofides, Mingozzi and Toth (Christofides, Mingozzi and Toth, 1979) can be listed among the earliest approximate approaches. Later on, a tabu search algorithm developed by Gendreau, Hertz and Laporte (1994), a location based heuristic for the routing problems by Bramel and Levi (1995) and a heuristic based on genetic clustering for the multi- depot VRP by Thangiah and Salhi (2001) are improvements on this problem. Among the exact methods are two/three indexed formulations, direct tree search methods and dynamic programming. A different approach is developed by Ghiani and Improta (2000) by the transformation of the generalized VRP to the capacitated arc routing problem.

For the VRPTW, there exist a number of heuristics or bound calculation techniques based on relaxations of the problem. In Solomon (1987), a number of tour building heuristics for the VRPTW are discussed. Among those algorithms, “Savings Heuristic” which is based on the savings heuristic originally proposed by Clarke and Wright (1964) is an example. While the general savings equality is still valid, the necessary and sufficient conditions for time feasibility when inserting a customer are also adapted. “The time-oriented, nearest neighbor heuristic” starts every route by the unrouted customer closest to the depot and assigns the customer closest to the last customer at every subsequent iteration. Here, the *closest* measure is calculated by considering both the geographical and temporal closeness of customers so the time windows are taken into consideration in the solution. “Insertion heuristic” inserts customers into routes regarding several criteria which include the time feasibility of serving each customer and the minimization of total distances traveled. “Time-oriented sweep heuristic” is a two-phase heuristic with clustering first and scheduling second. The original sweep heuristic of Gillett and Miller (Gillett and Miller, 1984) is improved to cover the time window constraints. Those algorithms are then compared according to time efficiency and the quality of results.

Some other heuristics developed in other thesis studies or published in related journals are also examined but the main problem related to them is the long computational times. Lower and upper bounding heuristics are also provided in Kontoravdis and Bard (1995) and Kontoravdis, Bard and Yu (2002).

Coding one of the Solomon’s algorithms (Solomon, 1987) or the bounding heuristics could be a solution for the VRPTW part of our algorithm, unless a better solution technique is found. However, the main drawback of this choice is that for different solutions the gap between the upper/lower bound and the best solution possible differs and a bounding technique finds near-to-optimal solutions for some instances while finding not-so-good solutions for the others. So the heuristic stands for some instances and against others in an unfair way.

Considering this, our search continued for any improved technique.

In addition to heuristics, another method encountered in later studies and in improved solution techniques is the use of constraint programming and constraint-based operators. Hybrid approaches which combine heuristics, metaheuristics and constraint programming are proven to be successful in logistics problems. As referred in Rousseau, Gendreau and Pesant (2002); Pesant and Gendreau (1996;1999), Baptiste, LePape and Nuijten (1995), Caseau and Laburthe (1998), the works of the ILOG team and some other commercial VRPTW components have given examples of applications in which OR techniques and constraint programming are combined.

Finally, for solving VRPTW, the prerequisites in our case on the solution quality and the computation time for a good solution led us to use an algorithm with some predefined classes and functions within ILOG Dispatcher 4.2 for which the basics are defined in Backer et al.(2000). The solution method of the VRPTW is based on constraint programming, some local search techniques and metaheuristics. In our applications we followed the general framework given by ILOG documents. We used savings heuristic as the first solution heuristic and defined five neighborhoods possible in Dispatcher (2-opt, or-opt, relocate, cross, exchange) to construct and search for new solutions. We progressed with a first accept strategy and did not use the support of metaheuristics, as our routing problems and search space are not so huge. The role of each technique and how the algorithm is designed is discussed in detail in Appendix B. In the flow of our algorithm for LRP, after reading the customer assignment for a depot from a chromosome, the VRPTW is solved for each open facility and the missing cost terms of the objective function are then added. The fitness function calculation is thus terminated.

4.2.4 Genetic Operators and Techniques

Genetic operators contain a number of milestones which are playing a great role in the flow of the genetic algorithm.

4.2.4.1 Parent selection

This is the task of choosing parents from the population to generate an offspring that inherits properties from the parents as in the nature. From the very beginning of the GA literature, many methods have been used to select the parents, varying from random selection of two parents to complex methods. For a detailed explanation of a variety of techniques, Haupt and Haupt (1998), Reeves and Rowe (2003), Beasley, Bull and Martin (1993) and Goldberg and Deb (1991) could be referred to. In a general way, the techniques could be summarized as follows:

- *Proportionate selection techniques such as roulette wheel selection, stochastic remainder selection and stochastic universal selection.* After the fitness values for the chromosomes are calculated, selection probabilities related to each chromosome is calculated regarding the total fitness of the population. Parents are selected generating random numbers in [0,1] and using the cumulative probability distributions. These are comparatively easy techniques to apply, but they have drawbacks: (1) while working directly with the fitness values, the most fit individuals dominate easily in the early generations and (2) in the later generations where the fitness values of individuals become close to each other; the cumulative probabilities are not so sensitive to choose better individuals which leads to low growth ratio. In addition to many other references, Whitley (1989) emphasizes the three disadvantages related to the selection pressure: (i) Stagnation because the search lacks selective pressure, (ii) premature convergence because selective pressure has caused the search to narrow and (iii) the difficulty of maintaining adequate selective pressure when the population tends to be homogenous.

- *Ranking and scaling based techniques*. Instead of using the raw fitness values of chromosomes, they are mapped onto a new scale and the selection is realized according to this scale or the chromosomes are sorted according to their fitness values and the selection probabilities are assigned to the ranks independent of the fitness values. As explained in Gen and Cheng (1997), the main intention in doing so is to maintain a reasonable difference between fitness ratings of chromosomes and to prevent a too-rapid takeover by some super chromosomes. Linear scaling, dynamic linear scaling, power law scaling and logarithmic scaling are examples of scaling techniques. Although they require computationally more effort than the proportionate selection methods, ranking and scaling based techniques can recover the deficiencies of the proportionate selection techniques.

- *Tournament selection*. This technique depends on choosing a group of individuals of size equal to the tournament size, then picking the best fit one among them and assigning it as a parent. Although the most common tournament size is 2, larger tournament sizes could also be used. The best fit individual could win the tournament with probability '1' in the deterministic version, whereas it could win the tournament with a probability between '0.5' and '1' in the probabilistic version. This method presents a good balance among computational effort/difficulty, probabilistic nature of GAs and the emphasis on the effects of the raw fitness values of chromosomes. It is computationally practical, because it only needs a preference ordering in a small set.

When the literature is examined, the last two groups of techniques are more commonly preferred in later studies. The pros and cons of the methods are as explained above, the computational effectiveness, not causing to premature convergence and conserving the fitness value differences independent of ranks have led us to use deterministic binary tournament selection with tournament size 2 in our algorithm. We repeat the reproduction phase a significant number of times to give chance to many different individuals in the population to be parents so choosing the better of the two candidate parents in a deterministic fashion does not cause us to skip the probabilistic nature.

4.2.4.2 Crossover

The crossover operator recombines the genetic material from two parents to make one or two offspring. The type of the operator defines how the offspring is generated from the parents and which information is inherited from which parent.

It is one of the most suitable phases where the problem specific knowledge can be inserted into the algorithm. However, it is still very common to apply a single point crossover or two-point crossover with no regard of the problem characteristics in GA applications. As explained and illustrated in Davis (1991), the basic idea behind the one/two-point crossover operators is to determine the crossover points and exchange parts between parents to pass to the offspring.

For the chromosomes being represented in a non-binary format, an example of a two-point crossover operator can be viewed in Figure 4.2.

Parent 1	2	1	2	|	2	1	|	2	1
Parent 2	**1**	**2**	**2**	**|**	**1**	**2**	**|**	**1**	**1**
Child 1	**2**	**1**	**2**		1	2		**2**	**1**
Child 2	1	2	2		**2**	**1**		1	1

Figure 4.2 A Two-Point Crossover Operator

In later studies, alternatives for the crossover operator are also developed. Uniform crossover is one of them where a mask is determined a priori to give information on the crossover points and the same mask is used for each crossover, whereas the crossover points are generated randomly in one/two point crossovers. Also the usage of mask eases the application of operator with even more than two crossover points.

Other advancement is the generalized fitness-based crossover operator called the fusion operator proposed in Beasley and Chu (1996). There the rationale is described below: as the overall fitness is determined by the values of genes, the genes of a more fit individual are more likely to produce a more fit offspring, so the chance of getting a gene from a parent would be proportional to the fitness of that parent in a probabilistic fashion in order not to create new individuals identical to their parents.

In permutation problems for which the most famous examples are traveling salesman or scheduling problems, any of the above crossover techniques may not be sufficient. For instance, when we think of the problem of finding the optimal order for a 6-city tour, the result of applying a one-point crossover on chromosomes representing routes {2,3,4,1,6,5} and {4,6,2,3,1,5} produces two infeasible offspring; {2,3,4,3,1,5} and {4,6,2,1,6,5} where some cities are visited twice and some are not visited at all. For such kind of problems, Goldberg (1989) discusses in "Genetic Algorithms in Search, Optimization and Machine Learning" some crossover operators appropriate for permutation problems to overcome the above difficulties such as the Partially Matched Crossover, Order Crossover and Cycle Crossover.

In our solution methodology, we prefer to design our own crossover operator for the LRP instead of using the standard operators. If it were a pure location-allocation problem, the concern would be to minimize the distances between customers and open facilities via direct shipment. But the routing problem makes us think the group of customers assigned to the facility as a whole and leads to a check whether the customers closer to each other and therefore more probable to create efficient routes are assigned to the same facility.

Moving from this idea, the crossover operator consists of two phases. The first phase is evaluated only once at the beginning of the heuristic after the distance data between customers are read and for each customer, it helps to create the order of customers starting from the nearest one to the farthest. The second phase uses this order and realizes the following steps:

Step 1. For each customer, the facilities it is assigned to in each parent are read from the chromosomes and assigned to an array (they are the possible values to appear in the offspring).

Step 2. If there are customers assigned to the same facility, then assign them to that facility in the offspring, else go to step 4.

Step 3. Check the possible facilities for the nearest customers to the ones that satisfy the condition in step 2. If one of the possible facilities is the one the customer in step 2 is assigned to, then assign that nearest customer to that facility.

Step 4. For each customer that is unassigned to a facility; starting from the nearest neighbor, check the possible facilities they are assigned in parents. When a common facility is met, assign the customer to that facility and update its possible facilities both being the one it is assigned to. If a common facility is not met, then assign this customer to the facility from the parent with better offspring.

In order to clarify how this crossover operator works, a representative example is given. We consider a small environment of 6 customers (1-6) and 4 facilities (7-10). The distances among sites are already calculated. The two arrays that will be used are:

pos [i] : to store the facilities customer-i is assigned to in the parents, in other words the possible facilities customer-i can be assigned to in the offspring

near [i] : to store the neighbors of customer-i starting from the nearest to the farthest. (this array is generated for each customer at the beginning of the algorithm just after the distance data are calculated).

In our example, assume that:

near [1] = 3,6,4,5,2

near [2] = 6,4,5,1,3

near [3] = 1,4,6,2,5

near [4] = 2,3,1,6,5

near [5] = 6,2,1,4,3

near [6] = 5,2,1,4,3

parent-1:

7	8	9	7	8	9

fitness of parent-1 = 42

parent-2 :

8	8	9	10	7	8

fitness of parent-2 = 30

Step 1. Reading the data from parents

pos [1] = 7,8

pos [2] = 8,8

pos [3] = 9,9

pos [4] = 7,10

pos [5] = 8,7

pos [6] = 9,8

Step 2.

offspring :

	8	9			

Step 3.

nearest customer to customer-2 : assign customer-6 to facility-8 if possible

nearest customer to customer-3 : assign customer-1 to facility-9 if possible

offspring :

	8	9			8

Step 4.

customer-1 :

check if it can be assigned to the facility customer-3 is assigned to : NOT POSSIBLE

check if it can be assigned to the facility customer-3 is assigned to : POSSIBLE

offspring :

8	8	9			8

update pos [1] = 8,8

customer-2 : already assigned

customer-3 : already assigned

customer-4 :

check if it can be assigned to the facility customer-2 is assigned to : NOT POSSIBLE

check if it can be assigned to the facility customer-3 is assigned to : NOT POSSIBLE

check if it can be assigned to the facility customer-1 is assigned to : NOT POSSIBLE

check if it can be assigned to the facility customer-6 is assigned to : NOT POSSIBLE

check if it can be assigned to the facility customer-5 is assigned to : NOT POSSIBLE

assign customer-4 to the facility coming from the better parent (with lower fitness)

offspring :

8	8	9	10		8

update pos [1] = 10,10

customer-5 :

check if it can be assigned to the facility customer-6 is assigned to : POSSIBLE

offspring :

8	8	9	10	8	8

customer-6 : already assigned

This approach also resembles one of the most common ideas from the beginning of the LRP literature which is the “cluster first, route second” heuristics. As the LRPs contain so many variables, it is inevitable to treat the problem as the sum of sub-problems. However, what we are trying to do in our approach is to evaluate the goodness of a solution based on the overall objective value of the LRP, instead of assessing the overall solution by the goodness of the sub-problems separately.

4.2.4.3 Mutation

The mutation is another way to create new individuals and it is used to maintain diversity in the population. In an ongoing GA, there is always the probability that the algorithm can converge too quickly or can be trapped in a local optimum. While it is called as premature convergence for early generations, it has a higher risk in later generations as the chromosomes are getting alike. To avoid this problem, randomly changing some existing chromosomes and maintaining diversity in the population is needed and mutation is used for this purpose. The mutation operators used in the literature are mostly in totally probabilistic nature; the most common one being the perturbation mutation. The way the perturbation mutation works is to choose randomly a chromosome and then one or more genes to mutate. In a binary chromosome changing those genes from a 0 to a 1 or vice versa; and in a non-binary chromosome changing the selected genes to any possible value would work.

In our solution, instead of using a standard operator, we again prefer to include some problem specific issues in the operator to increase the quality of the new individuals that will be inserted into the population. But care is to be taken about the use of problem specific issue in order not to destroy the randomness principle in the mutation. To apply the mutation operator in the algorithm, a sub-routine at the beginning of the algorithm must be run first to determine the nearest facility to each customer. Later on, each time the realization of the mutation operator is decided in a probabilistic way after the crossover operator, a customer is selected at random and the facility it is assigned to is changed with the nearest facility to that customer. In this way while conserving the randomness, the quality of the solution is likely to be augmented.

With this mutation operator, it is also guaranteed that any infeasible individual will not be generated because the nearest facility to a customer should always be in the feasible set of facilities for that customer. If not, this means the set of feasible facilities for that customer is empty as any farther facility should not be able to satisfy the three conditions defined while determining the feasible assignment set. In this case, the algorithm just stops at the beginning explaining that a customer has no facility to be assigned to in a feasible way in the hard time-window environment so there is no feasible solution for this problem instance.

4.2.4.4 Replacement

As the rational behind GA is to produce new individuals from the existing ones in order to improve the best solution found till that time by inserting them into the population and considering them as parents for the genetic operators, the strategy to update the population members (inserting the offspring and deleting some existing individuals in the population) is an important issue to decide. In the literature, there exist two main strategies depending on the generation gap for this phase of GA. The generation gap is the proportion of individuals in the population, which are replaced in each generation. The first strategy is the use of a generation gap equal to '1', which is the case of replacing the whole population in each generation. This is called generational or non-overlapping replacement. To avoid the loss of good individuals while generating a totally new population, an elitist strategy which directly passes a number of most fit individuals to the next generation is incorporated with generational replacement and applied with operators like cloning. The second strategy is to replace only a few individuals in each generation which is called steady-state or overlapping replacement. While inserting a few individuals into the population, it is also a point of concern to decide on the individuals to replace in the existing population with the new ones. For this choice, a number of methods are applied, like replacing randomly chosen ones, replacing the worst ones, replacing the oldest ones, replacing the parents or choosing by Kill Tournament. A comparative study on the alternatives (Smith and Vavak, 1999) declares the benefits, deficiencies and the effects on the computational effort for these methods. Exact Markov models for some and approximate models for some of the replacement strategies are evaluated in this study and with the help of simulations, general conclusions for different strategies are drawn.

Although there is no evidence that one strategy is strictly better than the other; the experimental work in Syswerda (1991) has resulted on the following advantages of the steady-state replacement:

- Letting new fit individuals immediately join the population and be ready for reproduction, while preserving the best individuals from the previous generation.
- Obtaining better results in less time.
- Imposing the condition that identical individuals are not inserted so that the copies of individuals are not present in the population

Evaluating all the theoretical and experimental work in the literature and with the above advantages, in our work we choose to apply steady state replacement without duplicates. In a problem with large solution space as LRPTW, it is thought to be advantageous to insert a good solution immediately in the population. The new individual generated at the end of the reproduction phase is inserted in the population if

- its copy is not included in the population

and

- it is better than the worst individual in the population.

To insert the new individual, the worst individual in the population is deleted.

4.2.5 Parameter Determination

Besides the above-mentioned issues important for the GA application, there are two other important parameters: population size, and crossover & mutation rates. The reason to mention these under the same topic is that they are dependent on each other. The studies, Goldberg (1989), Schaffer et al. (1989), Fogarty (1989) and Davis (1989), all comment on those parameters and the relations between them.

Population size gives the number of individuals in a population, which is constant throughout all generations, while crossover and mutation rates determine the frequency for applying those operators.

The first factor to decide on the population size is the length of the chromosomes. Some linear or exponential relations are developed between the chromosome length and the population size. However, applying those relations for long chromosomes can result in impractically large population sizes. The mostly supported idea is the use of a population size in the range [n, 2n] for a chromosome of length n.

In general, while small populations can converge too quickly without sufficient exploration of the search space, very large populations lead to long waiting times for significant improvements. To balance those effects, higher crossover and mutation rates are recommended for small populations while lower rates can be used with larger populations.

Also when the "steady-state replacement" and "replace worst" strategies are chosen, large populations are preferred to avoid premature convergence.

Another point is to use varying mutation rates in accordance with the needs of the generations. In earlier generations, as the diversity is high and the individuals represent solutions scattered over different areas in the search space, what is needed is to find good solutions to continue with and make use of the knowledge stored in them. However, in later generations, the search narrows as the convergence begins and what is needed is to explore different regions not to get trapped to local optimum. To satisfy these differing needs, use of low mutation rate in earlier generations and higher mutation rates in later generations seem to give more efficient results.

Based on the above-mentioned ideas in the literature, we determined the parameter values in our study. As it can be seen in some of the studies, a crossover rate equal to

‘1’ is used. This means that any two individuals chosen as parents mate. As we repeat the reproduction for a great number of times, it is possible to let any pair of parents to continue with mating. In addition a varying mutation rate is decided to be applied. The mutation ratio varies from 0.05 to 0.2. These ratios may seem a bit high for mutation, however, they are chosen so due to the fact that unlike a classical two-point crossover operator, our operator generates the same offspring each time the same individuals are chosen as parents and diversity is more emphasized by mutation.

The population size in our experiments varies with regard to the size of the problem (the number of customers and potential facility sites). However, the large solution space does not allow us to work with small populations, although it is computationally expensive. The values used as population size and why it is preferred to work with such populations are also discussed in Chapter 5.

4.2.6 The Overall Algorithm

The key concepts are designed as explained above in detail. The following four steps summarize the overall genetic algorithm-based heuristic procedure:

Step 1. Read the required data and generate the initial population

Step 2. Decode each chromosome in the initial population and evaluate the fitness function for each

Step 3. Calculate the population statistics

Step 4. Until the predetermined stopping criterion is satisfied:

- **4-a** Select parents from the population,
- **4-b** Apply the genetic operators,
- **4-c** Evaluate necessary checks for the offspring, calculate its fitness value and insert into the population if appropriate,
- **4-d** Update population statistics.

The only sub-routines not explained in detail are the calculation of population statistics. Being standard procedures, they are finding the best and worst fitness values and individuals, and the average fitness value in the population. The information about which individual is the worst is used to determine the individual to delete during replacement and the information about which individual is the best helps to find the globally best solution through the runs.

The flowchart at the end of the chapter also summarizes our algorithm.

The algorithm is coded in C++. The whole code is not presented; but the pseudocodes for the main body and for each procedure in the main body are given in Appendix C in detail.

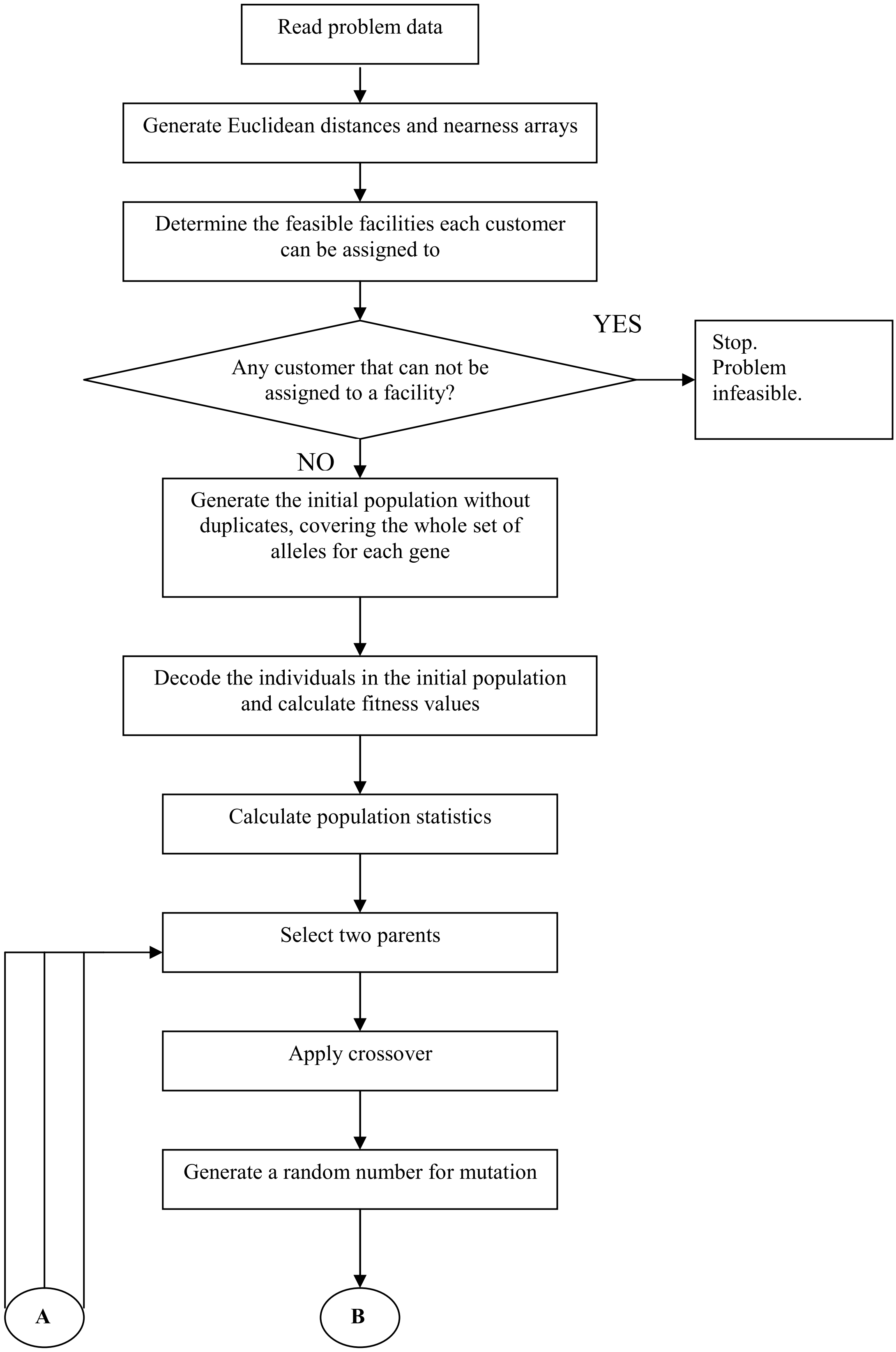


Figure 4.3 The Flowchart of the Algorithm

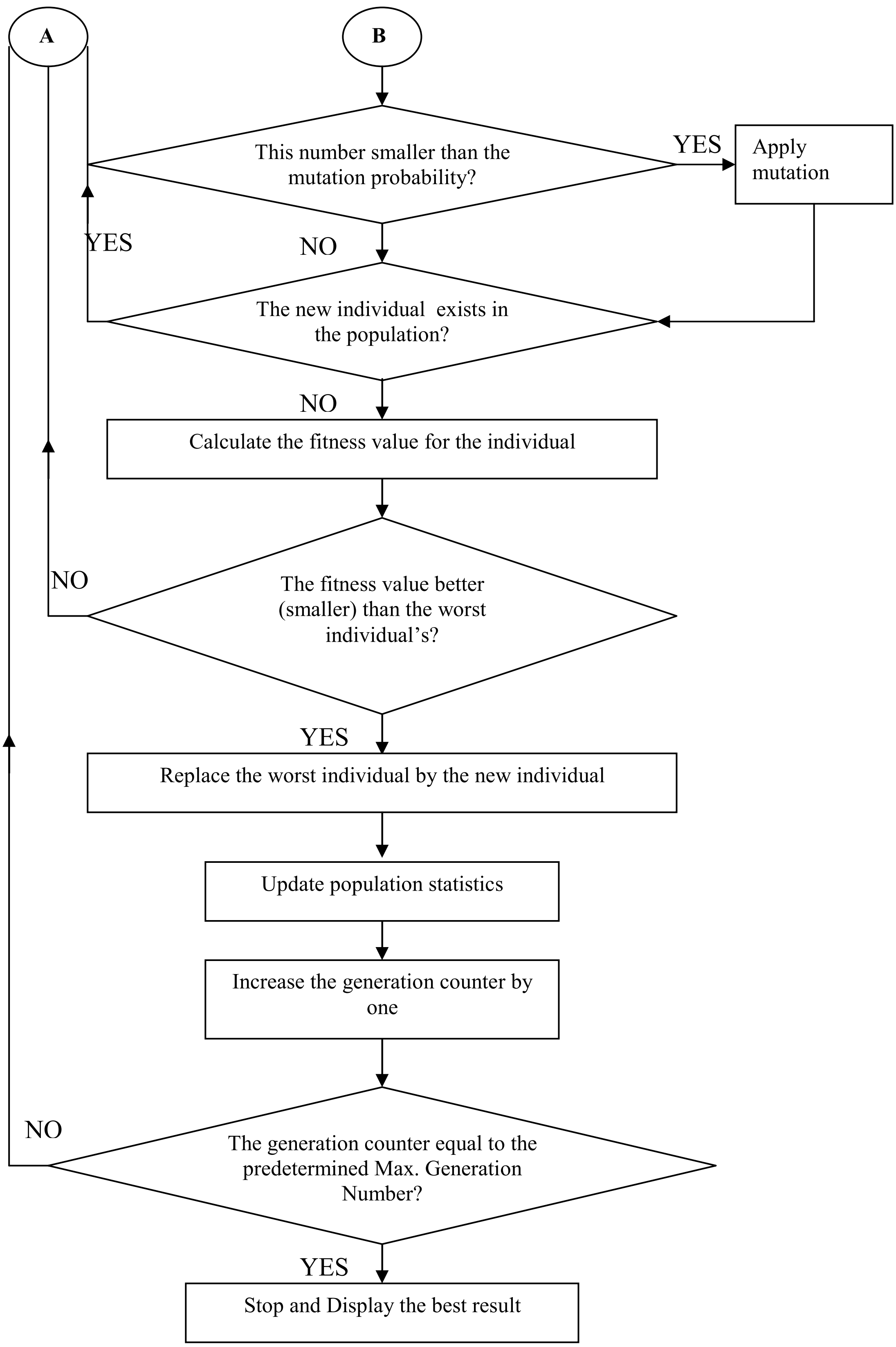


Figure 4.3 (cont'd)

CHAPTER 5

MODEL TESTING AND COMPUTATIONAL STUDY

In this chapter, the goal is to evaluate the accuracy and validity of the proposed methodology for the LRP with time windows for the customers and shift times for the potential sites. This evaluation consists of three phases:

(i) Finding or generating the appropriate set of problem instances for the problem environment

(ii) Determining the appropriate solution method

(iii) Realizing the computational studies on the test problems with both our approach and the comparison method.

5.1 Test Problem Structure and Data

As described in Chapter 2, the elements of the problem environment that we deal with can be summarized as follows:

- Supply source locations
- Potential sites for uncapacitated facilities
- Capacity breakpoints which determine the type of the facility being opened
- Shift time intervals for the potential facilities
- Vehicles with capacities
- Customers with location coordinates, demand quantities, earliest service times (ready times), latest service times (due times) and service times related with the deliveries

- Cost parameters including the fixed and variable costs for potential facility sites and vehicles
- Cost parameters for transportation

Search on the test problems that exactly represent our environment unfortunately gave no results. This caused us to work on generating an appropriate set of data for testing the methodology that we propose.

The most commonly used test problems are those of Solomon's (Solomon, 2005) for the VRPTW. The customer related information including coordinates, demand quantities, time windows and service times and vehicle capacities for each instance could be supplied from those problems and we have decided to use those by completing other aspects of the problem environment.

In Solomon's problems, the geographical data are randomly generated in instances of R1 and R2, clustered in instances of C1 and C2, semi-clustered in RC1 and RC2. Problem instances R1, C1 and RC1 have a short planning horizon, while R2, C2 and RC2 have longer horizons. Along with the capacity of the vehicles, the scheduling horizon affects the number of customers a vehicle can serve. Solomon (2005) and Solomon (1987) could be referred to for the generation and properties of the problem sets. The geographical and demand data are identical within a class, but the difference between those groups is the time window structure (i.e. in R1 and R2, the customer locations and demand data are the same, but the time windows are short in R1 and longer in R2). Each group consists of instances ranging from 8 to 12 with differing time windows. The instances include 100 customers' data; but smaller problems are also created from them considering only first 25 or first 50 customers.

However, the data from Solomon's problems lack two important issues. First, each problem instance has only one facility site as they are VRPTW instances, but not LRP instances. Second, as the objective function in a VRPTW may be represented as minimizing the total distance traveled or minimizing the number of vehicles, no cost parameters to cover both location and routing costs are present in those problems. So these issues have needed further study.

The test problem generation techniques in previous studies on LRP (the ones which are listed in the references) are all examined. The number and locations for potential sites and supply sources are determined in accordance with most of the studies in the literature. The cost data are supplied mainly based on Perl (1983), Perl and Daskin (1984), Perl and Daskin (1985) and Wu, Low and Bai (2002).

The details of the problem instances can be found in Appendix D, but below is the proposed test problem structure in our study:

5.1.1 Locations and Capacities

The general issues applied in the problems are defined below. In addition, the data for customers and facilities are given in Appendix D.

The number of customers: 25, 50 and 100. Different sized problems are based on Solomon's problems as explained above. To evaluate the results of the algorithm on different problem sizes, all three sizes are used.

The number of potential sites: 5 potential sites for 25 customers, 10 potential sites for 50 customers and 15 potential sites for 100 customers exist in the test problems.

The capacity of potential sites: The potential sites are uncapacitated.

The number of supply sources: The algorithm has the capability to handle a number of supply sources, but as the supply sources are uncapacitated, this is not a critical point. In order to avoid unnecessary complexity, the number is limited to one.

The locations of the customers: The coordinates of customers are from Solomon's test problems. The distances between nodes are calculated as Euclidean distances and travel times are taken as equal to corresponding distances.

The locations of potential sites: The general approach is to generate the nodes in $[0,100]^2$ according to a uniform distribution. The potential sites are generated accordingly. The generated nodes are given below. The first 5 or 10 sites or all of them can be used according to the problem size.

Table 5.1 The Coordinates for the Potential Facility Sites

	x coordinate	y coordinate
P1	30	94
P2	3	27
P3	31	1
P4	85	21
P5	14	76
P6	85	41
P7	32	28
P8	54	16
P9	73	41
P10	55	5
P11	39	86
P12	99	10
P13	8	98
P14	65	43
P15	24	58

The location of supply source: The supply source is located outside the region $[0,100]^2$. Point (130,130) is where the supply source is located.

The demands of customers: The values given in Solomon's test problems are used.

The time windows of customers: The values given in Solomon's test problems are used.

The shift time of potential sites: In Solomon's test problems, a depot and the ready and due times are present for each test problem environment. While increasing the number of the potential sites, the shift times are preserved and the same planning horizon is valid for each potential depot site.

The number of vehicles: In Solomon's problems, there is an upper bound on the number of vehicles but not necessarily used in our environment.

The capacity of vehicles: As in Solomon's problems (Solomon, 2005), a homogenous vehicle fleet for each instance is used, but the capacities differ among instances. The capacities specific to instances are also given in the Appendix D for each instance.

5.1.2 Cost Parameters

Fixed cost for potential sites: As the warehouses are uncapacitated, to determine a unique fixed cost for all sizes is not suitable. So the fixed cost function is a step function. The fixed costs are

- 700, 1100, 1400 for 25-customer instances
- 1400, 2200, 2800 for 50-customer instances
- 2800, 4400, 5600 for 100-customer instances

In addition, the total demands in classes R, C, RC are different. Regarding the total demands in problem instances, with 25, 50 or 100 customers, the ranges of the function can be defined as in the following table:

Table 5.2 The Capacities for Three Sizes of Facilities for Different Problem Types

		Nature of Customer Locations		
		Random	**Clustered**	**Random-Clustered**
25 customer	Size-I	total flow < ***120***	total flow < ***160***	total flow < ***180***
	Size-II	***120*** < total flow < ***220***	***160*** < total flow < ***300***	***180*** < total flow < ***340***
	Size-III	***220*** < total flow	***300*** < total flow	***340*** < total flow
50 customer	Size-I	total flow < ***250***	total flow < ***300***	total flow < ***320***
	Size-II	***250*** < total flow < ***500***	***300*** < total flow < ***600***	***320*** < total flow < ***640***
	Size-III	***500*** < total flow	***600*** < total flow	***640*** < total flow
100 customer	Size-I	total flow < ***525***	total flow < ***650***	total flow < ***600***
	Size-II	***525*** < total flow < ***1025***	***650*** <total flow < ***1250***	***600*** < total flow < ***1150***
	Size-III	***1025*** < total flow	***1250*** < total flow	***1150*** < total flow

Here total flow defines the sum of demands of customers assigned to the facility.

These fixed costs seem logical as they support the known fact that opening more facilities increases the fixed costs for depots. They also show economies of scale.

Fixed cost for vehicle: In Wu, Low and Bai (2002), a fixed vehicle cost is added to the parameters of Perl's (1983) problem environment. Similar to that paper, we use fixed vehicle cost as $30.

Variable facility cost: Similar to Perl's environment, it is taken as $0.75/unit.

Variable vehicle cost: Considering the variable vehicle cost and the distances in Perl's environments, the variable vehicle cost is determined to be $12/mile in our environment.

Variable transportation cost from supply source to facilities: It is taken as $0.10/mile/unit (by economies of scale).

5. 2 The Method Used in the Evaluation

There are a number of ways to test the goodness of a heuristic; some of which are:

- The model can be tested on a problem for which the optimal solution is known.
- The performance of the model can be compared to that of other heuristic models or the best-known results in the literature developed for the same problem.
- It may be possible to develop an expression for the worst case behavior of the heuristic.
- The results from the heuristic can be compared to a lower bound on the optimal solution by relaxing some constraints of the problem or to an upper bound.
- Some probabilistic and statistical analyses may be carried out. The probabilistic analysis supposes a probability distribution of problem data to establish statistical properties of a heuristic. The statistical analyses are based on estimating the point or interval the optimal value is located with regard to a series of iterative solutions (Zanakis and Evans, 1981).

Unfortunately for the problem we deal with, the alternatives above are not very easy to implement.

The mathematical model of the problem is given in Chapter 3. However, with so many variables and constraints, both the non-linear and the linear models are practically impossible to solve with the mixed integer programming and mixed integer non-linear programming solvers available. First trials on the non-linear model are realized with GAMS IDE 2.0.13.0. However, the non-linear mixed integer problem solvers DICOPT and SBB in GAMS IDE are in demo version at METU and are not capable of handling reasonable problem sizes.

Then the linear model is used to get optimal solutions for the problem. The model is coded with C programming language. However, obtaining optimum was not also possible for our test problems for 25 customers or more. Our trials to obtain the optimal solution are shown in section 5.3.3 along with the other computational studies. It is seen that this is not a valid comparison method as we cannot obtain solutions for the test problems. Consequently, comparing the results of our heuristic with the optimal solutions is not possible.

Another problem faced to evaluate the quality of the proposed heuristic is the rareness of the studies on exactly our problem environment. Although there exist a number of studies in the LRP literature, the time windows for customers are not considered in those studies. So it was not possible to use neither some results obtained by other heuristics or some predetermined test problem instances.

The worst case behavior is easy to determine for simple heuristics but for complex ones, such estimates will not be sufficient to evaluate the efficiency of the heuristic. As reported in Ball and Magazine (1981), some worst case bounds may be too loose for a good justification.

Probabilistic or statistical analysis may not produce good solutions in every problem type. Probabilistic analysis tends to consider the probabilistic space which contains unrealistic situations. Statistical analysis is not preferred in cases where the true optimal is not known a priori.

Considering all of those, in order to evaluate the accuracy of the proposed methodology, we have decided to use a sequential approach for the comparison. This approach consists of handling the problem by solving two subproblems sequentially. Test problems are solved to obtain an upper bound on the objective function in a sequential manner as well as with the methodology proposed in this study. The sequential approach is used also in many studies as a benchmark because it is accepted as a method to yield very accurate solutions (Srivastava and Benton, 1990; Min,1996; Haimovich and Rinnooy, 1985).

The sequential approach is based on the structure of LRP. Also mentioned before, this problem could be decomposed into subproblems in a number of ways. In the sequential approach we first solve a constrained location-allocation model to optimality by ILOG CPLEX by coding the model in C programming language and then determine the routes for each open facility by ILOG Dispatcher via a code in C++. The mathematical formulation of the location-allocation model is presented below.

Sets

I set of customers $i = 1,\ldots,n$

J set of potential sites $j = n+1,\ldots,m$

S set of supply sources $s = m+1,\ldots,p$

L set of potential site types (according to the capacity) $l = 1,\ldots,t$

Parameters

D_i	demand of customer-i (unit)
d_{gh}	distance between sites g and h ; $g \in I \cup J \cup S, h \in I \cup J$
C_l	capacity of a potential site of type-l (unit)
FC_l	fixed cost of a potential site of type-l
VC	variable cost of a potential site
TC	transportation cost from any facility site to any customer or between customers (per unit distance)
tc	transportation cost from any supply source to any facility site (per unit distance per item)
h	the time it takes for vehicle-k to cross a km
e_g	the earliest service starting time of site-g ; $g \in I \cup J$

se_g the required service time of site-g ; $g \in I \cup J$

la_g the latest service time for site-g ; $g \in I \cup J$

Variables

$$Z_{jl} = \begin{cases} 1 & \text{if a facility of type - l is located at site - j} \\ 0 & \text{o / w} \end{cases} \qquad \begin{matrix} j \in J \\ l \in L \end{matrix}$$

$$ZS_j = \begin{cases} 1 & \text{if a facility (of any type) is located at site - j} \\ 0 & \text{o / w} \end{cases} \qquad j \in J$$

$$Y_{ij} = \begin{cases} 1 & \text{if customer- i is assigned to DC at site - j} \\ 0 & \text{o / w} \end{cases} \qquad \begin{matrix} i \in I \\ j \in J \end{matrix}$$

q_{sj} = quantity shipped from source-s to facility site-j $\quad s \in S,\ j \in J$

Model

Minimize

$$\sum_{l \in L} \sum_{j \in J} FC_l \ Z_{jl} + \sum_{j \in J} VC \left(\sum_{i \in I} D_i \ Y_{ij} \right) + \sum_{s \in S} \sum_{j \in J} tc \ q_{sj} \ d_{sj} + 2 \sum_{i \in I} \sum_{j \in J} TC \ d_{ij} \ Y_{ij} \tag{5.1}$$

subject to

$$\sum_{j \in J} Y_{ij} = 1 \qquad \forall i \in I \tag{5.2}$$

$$Y_{ij} \le ZS_j \qquad \forall i \in I, \forall j \in J \tag{5.3}$$

$$\sum_{s \in S} q_{sj} - \sum_{i \in I} D_i \ Y_{ij} = 0 \qquad \forall j \in J \tag{5.4}$$

$$\sum_{i \in I} D_i \ Y_{ij} - \sum_{l \in L} C_l \ Z_{jl} \le 0 \qquad \forall j \in J \tag{5.5}$$

$$\sum_{l \in L} Z_{jl} = ZS_j \qquad \forall j \in J \qquad (5.6)$$

$$\left(h\, d_{ij} + se_i - la_i\right) Y_{ij} \leq 0 \qquad \forall i \in I\ ,\ \forall j \in J \qquad (5.7)$$

$$\left(h\, d_{ij} + se_i + h\, d_{ji} - la_j\right) Y_{ij} \leq 0 \qquad \forall i \in I\ ,\ \forall j \in J \qquad (5.8)$$

$$\left(e_i + se_i + h\, d_{ji} - la_j\right) Y_{ij} \leq 0 \qquad \forall i \in I\ ,\ \forall j \in J \qquad (5.9)$$

$$Z_{jl},\ ZS_j,\ Y_{ij} \quad binary \qquad j \in J, l \in L, i \in I \qquad (5.10)$$

$$q_{sj} \geq 0 \qquad s \in S, j \in J \qquad (5.11)$$

The objective function (5.1) presents the sum of costs to be minimized. The first term stands for fixed facility costs and the second term stands for variable facility costs, which depend on the flow through that facility. The third term is for the outbound transportation costs. As there are no routing considerations in the location-allocation, the last term represents the inbound transportation costs via direct shipment. It is assumed that a single vehicle leaves for each delivery and the cost is calculated on the basis that if a customer is assigned to a facility; the cost of the delivery is equal to the distance between those two points multiplied by the cost of a unit distance. Taking into account the two ways - both going to the customer and returning back to the facility - this cost is also multiplied by 2.

Constraint (5.2) requires that each customer is assigned to a single facility and constraint (5.3) assures that each customer is assigned to an open facility. Constraint (5.4) and (5.5) show the equality between the quantity of goods that enter the facility and that leave the facility. Also the capacity of the type of the facility that is being opened is sufficient for supplying the demands of customers assigned to that facility. Constraint (5.6) is used to guarantee that at most a single type of facility is opened at any potential site.

Constraints (5.7), (5.8) and (5.9) are additional constraints to a classical location-allocation model, because of the shift time and time windows. Without those constraints, time windows can not be considered in the model at all; then assignments for impossible deliveries within time windows may be done and the result obtained by the sequential method turns out to be infeasible. In fact it is the significant drawback of such a sequential method. But in order to obtain a good upper bound for comparison without falling in infeasibility, we have added these three constraints and allowed any feasible assignment even in cases in which a route can not be constructed including more than one delivery, but a vehicle can serve a single customer.

Constraints (5.10) and (5.11) define positive and binary variables.

After this model is solved, the location and allocation variables are taken as input for the second phase, for the VRPTWs, which are solved in the same way as in the genetic algorithms, that is, by using ILOG Dispatcher. The explanations on ILOG Dispatcher 4.2 are in Appendix B. The total cost is obtained by omitting the direct shipment costs from the objective value of the mathematical model and summing up the routing costs instead.

5.3 The Evaluation of the Proposed Approach

In this section, some discussions on different aspects of the proposed solution methodology are presented. First some genetic algorithm parameters and the effectiveness of the problem specific crossover operator are discussed. Later, after presenting some runs to obtain the optimum solution for LRPTW and explaining why it is not possible to use this approach for larger problems; the comparison of the proposed algorithm against the sequential method is examined. A final discussion is on how the genetic algorithms work and how the solution is improved through a run.

Through all this work the mathematical models are solved by CPLEX 8.1.0 on Pentium IV 1.6 GHz processor and 512 MB RAM running on Windows NT. Our algorithm and the routing phase of the sequential approach are run on a computer with Pentium IV 2.4 GHz processor and 512 MB RAM running on Windows XP.

In most of the trials, instance “mr101_25” was the most widely used instance. A problem instance can be defined by the characteristics such as the structure of customer locations, the length of the planning horizon, the tightness of the time windows and the number of customers and potential sites. This instance contains 25 customers and 5 potential facility sites. The locations of customers are random, not clustered. It is the modified version of the Solomon’s instance r101 with the increase in the number of potential sites. The vehicle capacities are not too high and the planning horizons are short. This case has nothing special or is not selected on purpose, however it is just used so frequently when a randomly generated case is needed.

5.3.1 Preliminary Work on the Population Size and the Number of Maximum Generations

As explained in Chapter 4, the population size and number of generations are two important parameters that are effective in determining the quality of the solution obtained from the genetic algorithms. Unnecessarily increasing these parameters is not preferred, as the computational effort needed also increases; however, limiting them to insufficiently small numbers causes not to evaluate the capability of the method properly. So, before testing the proposed approach on a number of test problems, we have spent some effort on determining the appropriate parameters.

Most of this work is done on 25-customer problems. The conclusions obtained from those trials has given a general perspective on determining the parameters for also larger problems, so the whole evaluation realized for small problems is not repeated.

We have started with small population sizes and small number of generations. We have progressed by increasing these parameters as long as it is not computationally very expensive and the objective value is improved. Moreover, for increasing the parameters, the additional computational effort should be worth the improvement in the objective value. Some of our trials on some problem instances and the discussion on the parameters are given below.

Problem instance mr101_25

The characteristics of instance mr101_25 are already described. The results obtained on this problem instance with different parameter sets are given in Table 5.3, 5.4, 5.5 and 5.6.

Table 5.3 Results for instance mr101_25 for Population Size=100 & Number of Generation=250

	Objective value for the best solution	Computational time (sec.)
run 1	15162	332.31
run 2	15627	217.45
run 3	15132	165.36
run 4	16042	3600

Table 5.4 Results for instance mr101_25 for Population Size=300 & Number of Generation=500

	Objective value for the best solution	Computational time (sec.)
run 1	15627	445.57
run 2	15587	406.78
run 3	15627	365.54
run 4	15627	406.56
run 5	15873	464.42
run 6	15587	421.35
run 7	15610	411.78
run 8	15177	407.7
run 9	15162	405.54
run 10	15327	420.89

Table 5.5 Results for instance mr101_25 for Population Size=500 & Number of Generation=850

	Objective value for the best solution	Computational time (sec.)
run 1	15162	769.78
run 2	15162	737.87
run 3	15162	747.92
run 4	15327	756.87
run 5	15162	773.37
run 6	15627	766.31
run 7	15516	771.45
run 8	15327	770.14
run 9	15162	764.28
run 10	15362	763.39

Table 5.6 Results for instance mr101_25 for Population Size=600 & Number of Generation=1000

	Objective value for the best solution	Computational time (sec.)
run 1	15162	844
run 2	15320	845.64
run 3	15162	915.62
run 4	15162	934.31
run 5	15162	928.96
run 6	15177	902.57
run 7	15138	860.57
run 8	15433	830.32
run 9	15627	839.56
run 10	15162	829.75

Problem instance mrc207_25

The instance is the extended version of Solomon's instance rc207 with a higher number of potential facility sites. It contains both random and clustered locations for 25 customers. The instance represents the case where the vehicles have high capacities and the planning horizon is neither too long nor too short. The results related with this instance are in Table 5.7 and 5.8.

Table 5.7 Results for instance mrc207_25 for Population Size=500 & Number of Generation=850

	Objective value for the best solution	Computational time (sec.)
run 1	11254	736.46
run 2	11254	706.29
run 3	11254	677.03
run 4	11254	681.79
run 5	11782	705.32
run 6	11254	728.82
run 7	11254	673.45
run 8	11254	704.12
run 9	11254	698.09
run 10	11254	702.65

Table 5.8 Results for instance mrc207_25 for Population Size=600 & Number of Generation=1000

	Objective value for the best solution	Computational time (sec.)
run 1	11254	805.12
run 2	11254	748.68
run 3	11254	748.65
run 4	11254	712.06
run 5	11254	964.79
run 6	11254	768.42
run 7	11254	836.89
run 8	11254	868.04
run 9	11254	713.07
run 10	11254	846.37

Above are the results on some arbitrarily chosen instances. The parameter pair (population size, number of generations) was first taken as (100,250) and seemed to reach good solutions. However the 4th run showed an important deficiency of small population size especially with our crossover operator. Unlike a single / two point crossover operator where the crossover point probably changes each time and a different offpring is generated even if the same individuals are chosen as parents; if the two parents are the same, the problem specific crossover operator gives the same offspring so it needs to choose different parents for diversity. Since the size of the search is limited, it was expected that the runs would last short as it was the case with the first three runs. The probabilistic nature of GA allowed some of the runs to result in a normal way, but in the 4th run, an important issue was observed. Because of the criteria to accept a new individual to the population (that is, the new individual to be inserted must not exist in the population) that should be better than the worst individual in the population, it was difficult to find appropriate new individuals with reproduction in that small population size especially after the generation 200. The 4th run did not finish but was terminated at the end of 3600 seconds, which is at least 10 times the reasonable duration, and we decided that those parameters are not appropriate. Then some trials with (300, 500), (500,850) and (600, 1000) were realized. Being both reliable enough for a problem of this size and giving satisfactory results; it was decided that the runs for 25 customer problems would be done with population size of 500, number of generations of 850.

Problem instance mr101_50

This is the 50-customer version of instance r101; all the other characteristics being the same. The results on this 50-customer instance are presented in Table 5.9 and 5.10.

Presenting some of the results below; with the same criteria to decide, *population size 1000 & number of generations 2000 for 50 customer problems* and *population size 4000 & number of generations 10000 for 100 customer problems* were accepted. The non-linearity of the increase of parameters versus the number of customers can easily be associated with the non-linear increase in the problem size and the search space when the number of customers increases.

Table 5.9 Results for instance mr101_50 for Population Size=1000 & Number of Generation=1750

	Objective value for the best solution	Computational time (sec.)
run 1	28903	2926.43
run 2	29066	2829.42
run 3	28831	2832.84
run 4	28785	2878.87
run 5	29218	2879.89
run 6	28553	2885.42
run 7	29172	2852.06
run 8	27710	2810.54
run 9	28729	2849.53
run 10	27068	2825.51
Average of 10 runs	***28603.5***	

Table 5.10 Results for instance mr101_50 for Population Size=1000 & Number of Generation=2000

	Objective value for the best solution	Computational time (sec.)
run 1	27628	3006.42
run 2	28057	2981.9
run 3	28150	3066.73
run 4	28031	2959.29
run 5	27966	3020.92
run 6	27345	2972.31
run 7	26853	3004.48
run 8	28125	2959.71
run 9	28734	3055.51
run 10	27840	3056.17
Average of 10 runs	***27872.9***	

5.3.2 The Effect of Problem Specific Operators versus Classical Ones

The crossover and mutation operators proposed in this study possess problem specific features. It is always emphasized that these problem specific operators can reach better solutions by combining the nature of GAs and the effects of problem related knowledge. As the location routing problem with time windows is a complex and difficult problem, we work on problem specific crossover and mutation operators and embed them into the GA framework. As these operators proposed here are not applied in any earlier study, we think that it would be informative to evaluate their efficiency.

For this purpose; we have coded a classical crossover and a classical mutation operator and realized some runs with them. The crossover operator is a single point crossover operator. The crossover point is chosen randomly and the right and left parts of the parents are exchanged to produce two offspring. The mutation operator selects a gene randomly and changes it with another randomly chosen feasible facility for the related customer. In order to remain in accordance with the problem specific operator, in each generation a single new individual should be inserted to the population. So after the crossover is realized, each of the offspring goes through the mutation phase and may be mutated with regard to the mutation rate. Then better of the two offspring is accepted as the new individual and inserted to the population. The checks for not inserting a duplicate of an already existing individual in the population and for inserting a new individual if it is better than the worst one in the population are still valid.

The experimental trials are realized on instance mr101_25. The results of the alternative GA algorithm with already known operators are so far from our results, so the following 5 runs were decided to be sufficient to come up with an idea on the problem specific operators. The data related to 10 runs of the same instance with our operators are already presented above in Table 5.5. The results of 5 runs with the known operators are shown below. Population size is 500 and the number of generations is 850 in these trials.

Table 5.11 Results Obtained with the General Crossover and Mutation Operators for Instance mr101_25

	Objective value for the best solution	Computational time (sec.)
run 1	19835	1203.50
run 2	20998	1269.09
run 3	20080	1259.37
run 4	19521	1224.09
run 5	21458	1227.29

While the results with our operators change between 15162 and 15627 as given in Table 5.5, the results with the general operators change between 19521 and 21458. It is obviously seen that the known operators do not perform well for our problem environment. The evolutions of the averages of the populations when treated with those operators are also examined.

The Figures 5.1 and 5.2 present the improvement in a population with the problem specific or the generally known operator. The graphs for other runs of the same classification are similar to those presented here.

With the generally used operators, the best individual of the population is hardly improved through generations and the improvement in the average of the whole population can be observed from the below figures. When the nature of the routing issue is considered, it is seen that any arbitrarily chosen allocation scheme would not be promising to decrease routing related costs. By integrating the problem specific knowledge, we have supported the general flow of GA to produce offspring that are probable to represent good solutions in terms of total costs and as seen from the difference between the results, this highly improves the performance of the genetic algorithm application on location routing problems with time windows.

Choosing good parents does not mean much with a one-point crossover because with regard to the crossover point, all the good allocations in the parents that lead to low costly routes may deteriorate and very costly routes can be constructed in the offspring. But in our crossover operator, we have tried to produce routes using the knowledge carried by the parents for neighborhoods of customers. Consequently the application we have proposed seems effective and also supports the generally accepted theory that problem specific information improves the effectiveness of genetic algorithms.

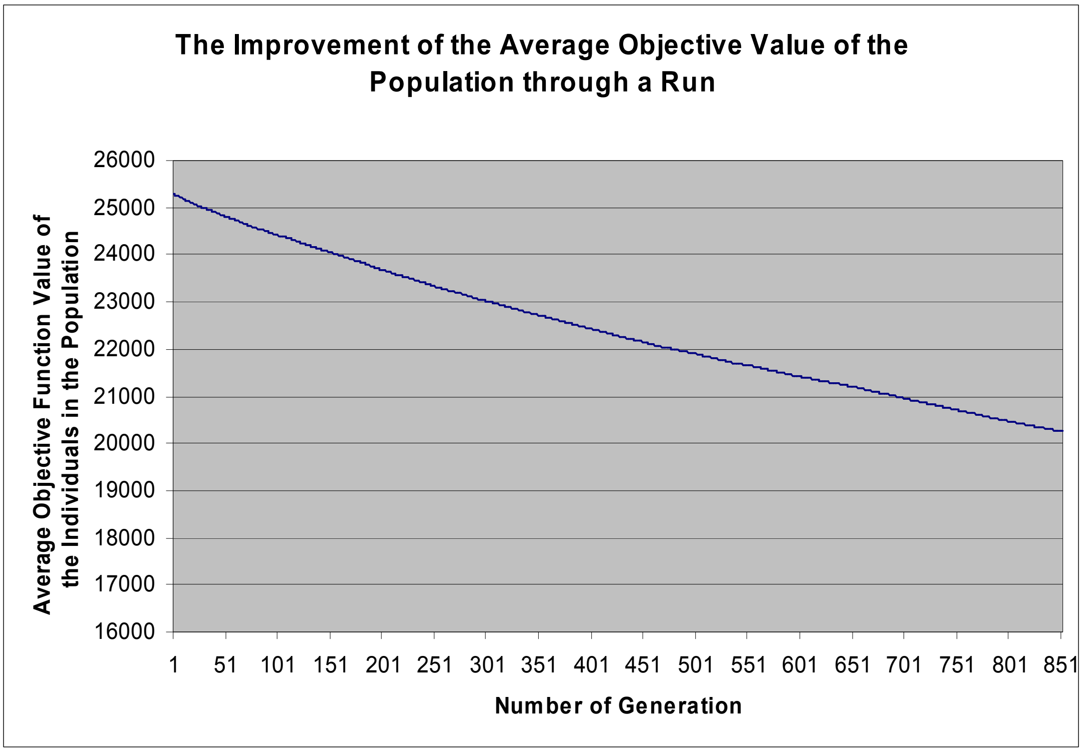


Figure 5.1 The Evolution of the Population through Run4 with General Operators

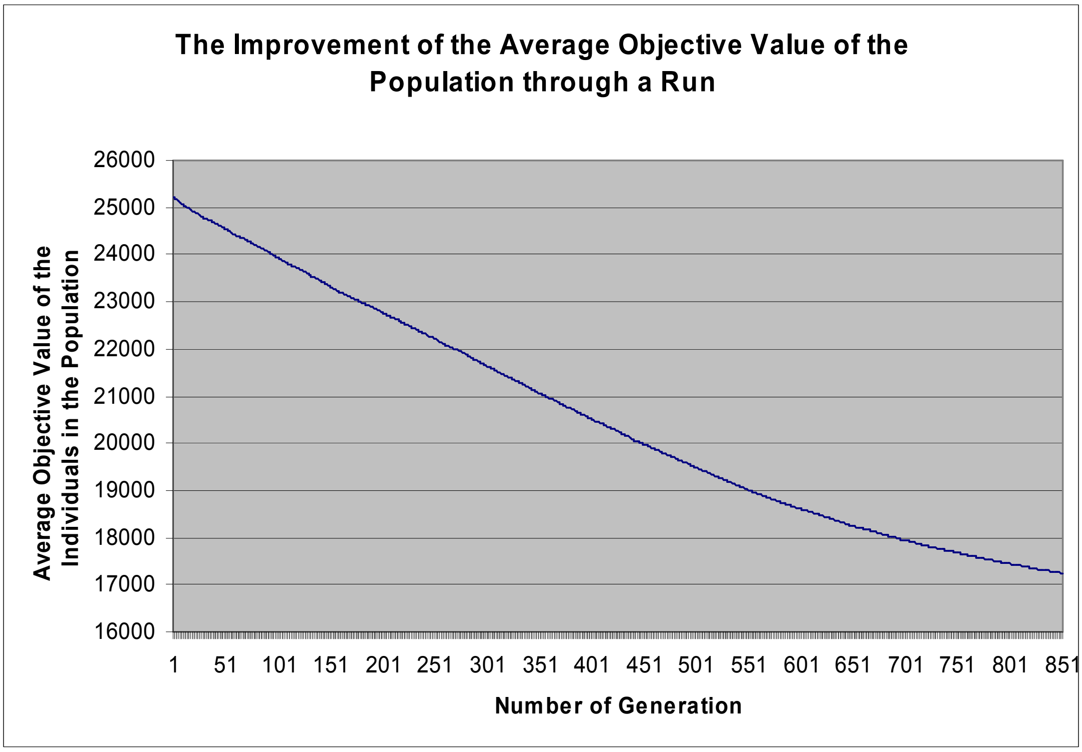


Figure 5.2 The Evolution of the Population through Run1 with Problem Specific Operators

5.3.3 Trials on Obtaining the Optimal and Comparison with Optimal for an Example Case

Before deciding on the sequential approach as the comparison method, we tried to obtain the optimal for the test problems and compare our results with them. However, it has been observed that with the software and hardware configuration available, it is not possible to solve problems of size 25 customers or more to optimality because of the number of variables and constraints. In addition there was a significant increase in the computational time even with a small increase in the number of system elements. Besides all, with the aim of examining some results obtained by solving the mathematical model along with the results of the proposed approach and being able to comment on the computational time for larger problems, the below trials on some small problems are realized.

The instance mr101 is chosen as the baseline case. The small test instances are generated choosing the first desired number of customers and potential sites from this instance. The cost parameters are the same with the 25 customer problems, but the only difference is that the capacity limits for three types of facilities are adjusted according to the total demand in the system. Another issue to mention about the problems below is related to the number of vehicles. Although there is no limit on the number of vehicles in our problem environment, the below instances are generated varying the number of vehicles. It is observed that increasing the number of vehicles has a significant effect on the computational time, as increasing the number of customers or potential sites has. To examine this effect, the small instances are generated as in Table 5.12 :

Table 5.12 The Variation of the Computational Time with regard to the Problem Size

Instance	Number of customers	Number of facilities	Number of vehicles	CPU time (sec.)	Status
1	8	2	4	344.96	Optimum obtained
2	8	2	8	237284.6	Error: Out of memory. A feasible integer solution with the gap 27.74% is obtained
3	8	3	4	3216.38	Optimum obtained
4	8	3	8	249713.3	Error: Out of memory. A feasible integer solution with the gap 23.14% is obtained
5	10	3	10	341016.4	Error: Out of memory. A feasible integer solution with the gap 52.33% is obtained

It is obvious that for problems of size 25 customers or more, this is not a valid comparison method. Yet, instances 2, 4 and 5 where we could obtain a feasible solution are solved also with the GA based heuristic and the sequential approach for comparison. The results are presented in Table 5.13. Though this comparison is presented to gain further insight, it must still be considered that a comparison of the optimum, sequential method and GA based heuristic for a problem of very small size in fact might not be totally sufficient to predict the gap among them for large problems.

Table 5.13 The Comparison of the Results Obtained by GA-based Heuristic with CPLEX Results and with Sequential Approach Results

Instance	Result by CPLEX	MIP Best Bound	The gap of the solution by CPLEX	CPU time for CPLEX solution (sec.)	Result by the sequential approach	Result by the best run of the GA based heuristic	Average CPU time for a GA run (sec.)
2	7518.94	5433.53	27.74%	237284.6	6553	6553	15.87
4	6909.21	5310.69	23.14%	249713.3	6526	6526	182.24
5	8204.59	3911.15	52.33%	341016.4	8153	8153	330.6

The runs for instance 2 are realized with (population size=15, number of generation=20); the runs for instance 4 are realized with (population size=400, number of generations=500) and (population size=400, number of generations=600); the runs for instance 5 are realized with (population size=300, number of generations=300) and (population size=500, number of generations=500). Here it is worth noting that the number of feasible solutions is significantly small compared to the number of feasible solutions for a 25-customer problem, so it is possible to work with smaller populations. Especially for instance 2; a population size within 2n individuals is sufficient where n is the length of the chromosome.

The results also show that the GA based algorithm is giving satisfying solutions compared to the results obtained by the mathematical model. In addition, as observed from the related columns, the results obtained by the sequential approach and the GA-based heuristic are the same.

5.3.4 Comparison of the Algorithm with the Sequential Approach

In order to evaluate the quality of our solution approach, the results of the runs we realized with the GA based approach are compared with the results obtained with the sequential method explained in detail in section 5.2. The Tables 5.14, 5.15 and 5.16 summarize this comparison by grouping the problems according to the problem size.

In those tables, “improvement by the GA based heuristic” is calculated as “(Obj. value by the Seq. App.) - (Obj. value of the best GA run)” and the “%improvement by the GA based heuristic” is calculated as “((Imp. by the GA based heuristic) / (Obj. value of the Seq. App.))*100” .

Moreover, because of the probabilistic nature of genetic algorithms, each problem is solved more than once, 10 or 20 times. These runs and the results are all recorded. The terms “best GA run” and “average cost value obtained by the GA runs” are used for representing the run with the smallest objective value and the average of the all runs related to that problem, respectively.

To observe the differences between the results, the Figures 5.3, 5.4 and 5.5 can also be examined.

As it can be observed from the above tables and figures; the approach we have proposed has proved to be successful regardless of the problem characteristics and the problem size. As expected, the computational time of a run does not increase in a linear proportion when the customer size is doubled. It is observed that when the customer number increases from 25 to 50, the duration of a run approximately increases 4 times; and when the customer number increases from 50 to 100, the duration of a run increases approximately 8 times.

Table 5.14 Results for Problems with 25 Customers

Instance	Result by the sequential approach	Result of the best GA run	Improvement by the GA based heuristic	% improvement by the GA based heuristic	Average CPU time for a GA run (sec.)	Average cost value by the GA runs
mr101_25	15717	15162	555	3.53	762.14	15296.9
mr205_25	13172	11453	1719	13.05	644.12	11636.2
mc109_25	10842	10333	509	4.69	737.80	10336.5
mc206_25	11348	10869	479	4.22	714.75	10878.2
mrc103_25	13623	13113	510	3.75	762.29	13113.0
mrc207_25	13199	11254	1945	14.73	701.70	11306.8

Table 5.15 Results for Problems with 50 Customers

Instance	Result by the sequential approach	Result of the best GA run	Improvement by the GA based heuristic	% improvement by the GA based heuristic	Average CPU time for a GA run (sec.)	Average cost value by the GA runs
mr101_50	28128	26853	1275	4.53	3003.26	28184.2
mr205_50	26083	22645	3438	13.18	3101.43	24368.3
mc109_50	22452	21157	1295	5.77	3047.07	22402.1
mc206_50	23995	19795	4200	17.51	3021.49	22641.2
mrc103_50	26105	24430	1675	6.42	3116.29	25135.8
mrc207_50	24776	21445	3331	13.45	3046.82	23475.7

Table 5.16 Results for Problems with 100 Customers

Instance	Result by the sequential approach	Result of the best GA run	Improvement by the GA based heuristic	% improvement by the GA based heuristic	Average CPU time for a GA run (sec.)	Average cost value by the GA runs
mr101_100	50897	47013	3884	7.63	23985.87	48111.6
mr205_100	46507	48937	-2430	-5.22	25327.88	52666.7
mc109_100	46070	44474	1596	3.46	24850.74	46065.4
mc206_100	46154	40230	5924	12.83	24126.55	46141.9
mrc103_100	52829	47827	5002	9.47	25116.66	49831.0
mrc207_100	51267	46738	4829	9.36	24817.75	49069.8

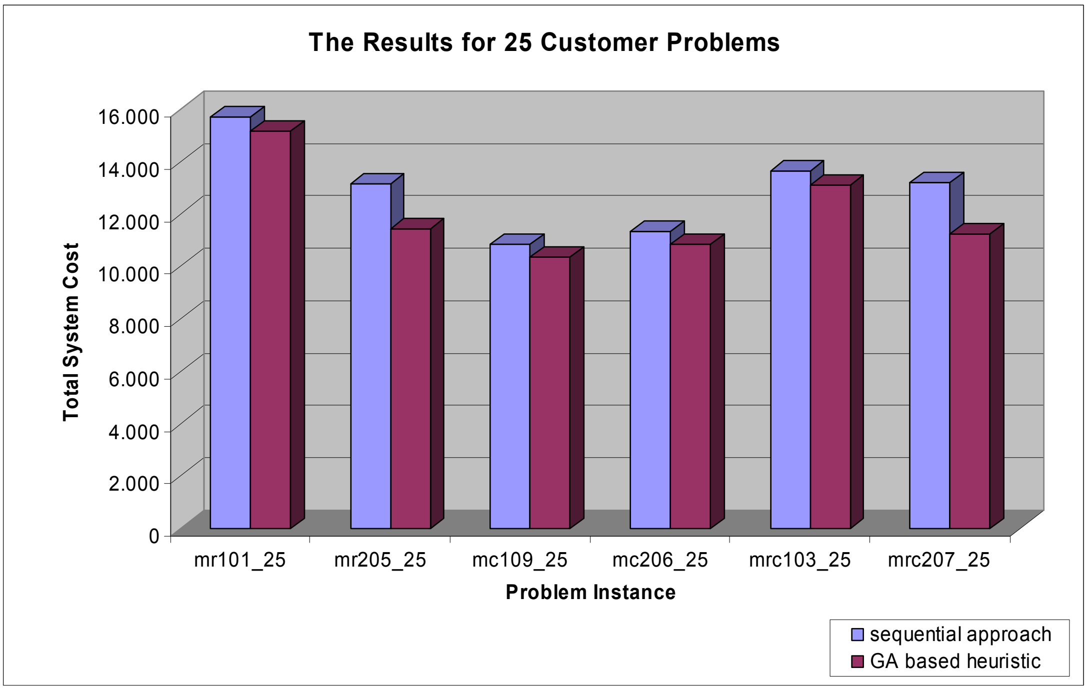


Figure 5.3 The Comparison of Results for Problems with 25 Customers

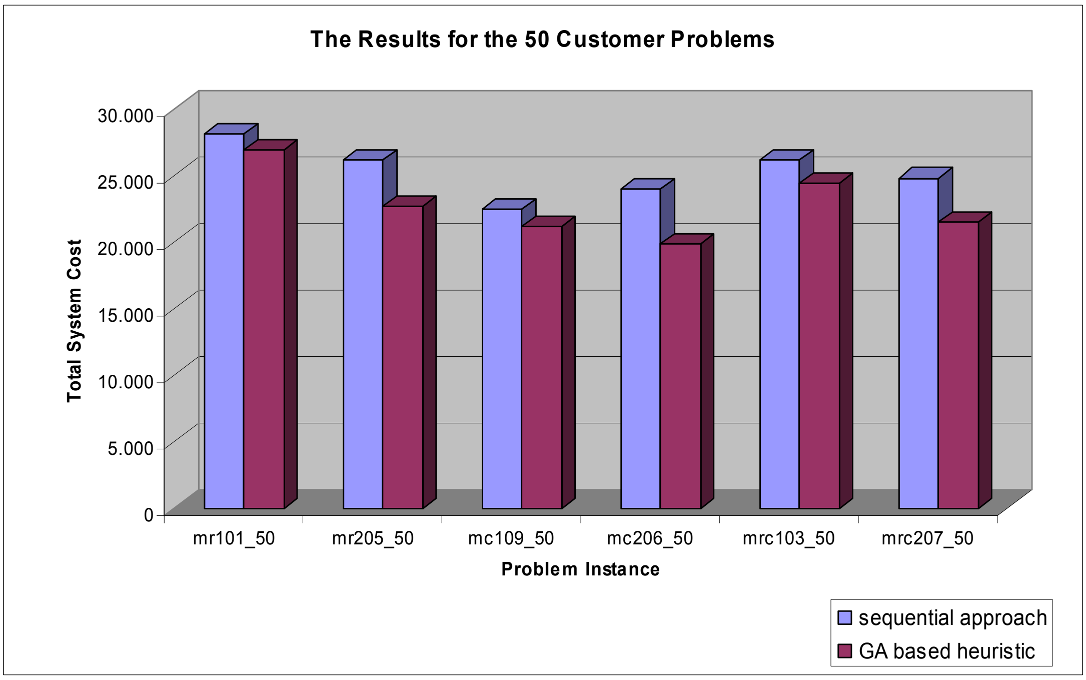


Figure 5.4 The Comparison of Results for Problems with 50 Customers

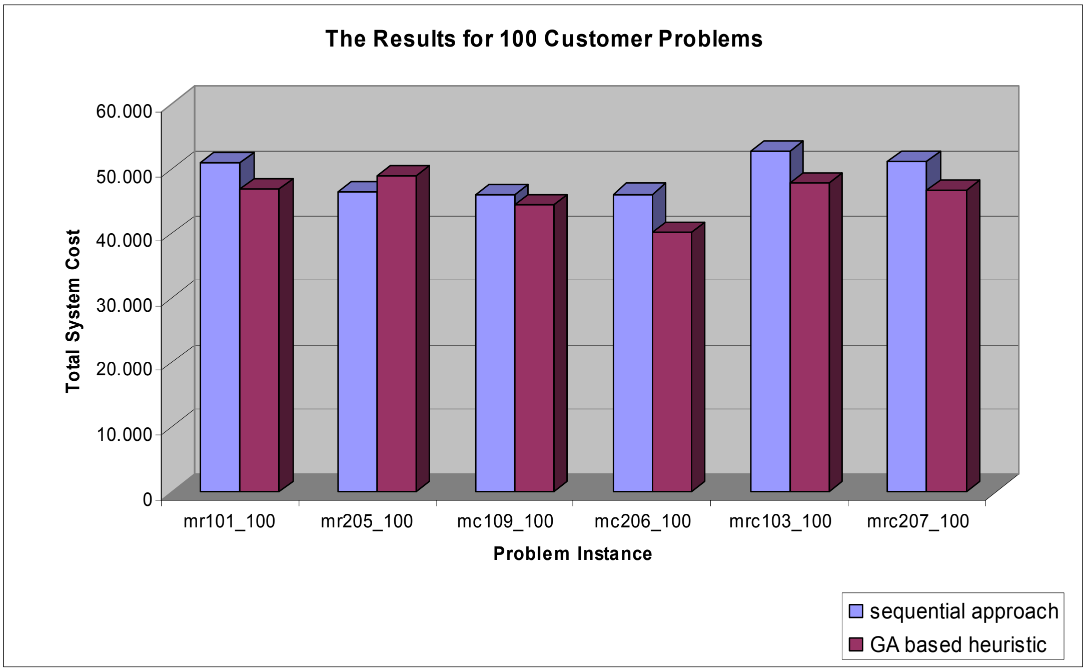


Figure 5.5 The Comparison of Results for Problems with 100 Customers

This is due to the increase in the number of possible feasible solutions as well as the increase in the number of customers. But by adjusting the parameters, that's increasing the population size and maximum number of generations, it has been possible to handle 100 customer-15 facility problems with an integrated approach while those problem sizes are impossible to solve by the mathematical models that contain every aspect of the problem.

It is also significant that our algorithm is capable of giving better results than the sequential approach regardless of the changes in the problem environment such as the location of customers (random or clustered), the tightness of time windows or the vehicle capacities.

Another analysis on the results would be to observe the composition of the total cost value.

Table 5.17 presents the analysis of the total cost values of both methods decomposed to different types of costs. The supplier related costs consist of the transportation costs from the supplier to the facilities, the facility related costs are the fixed and variable facility costs and the routing related costs include the cost of the routes from facility sites to customers.

Observed from the table, in most of the instances, the ratios of supplier and facility related costs in sequential approach are higher than the same ratio in GA solutions.

As explained in 5.2, the supplier and facility related costs are obtained by the solution of the location-allocation model which minimizes the total system costs, including the direct shipment costs to customers in addition to fixed and variable facility costs and the transportation costs from the supplier to facilities. As the direct shipment costs are a significant part of the objective function, to minimize the overall, the solutions of models mostly include higher number of open facilities, some of them being the ones away from the supplier but near to facilities. Consequently, the trade off between "the direct shipment costs" and "supplier and facility related costs" to minimize the total causes an increase in the "supplier and facility related costs". This effect is due to ignoring the routing issue simultaneously. But in the GA solutions, as the total cost is evaluated considering the routing issue at the same time with location-allocation, better objective values could be obtained.

Table 5.17 Analysis of the Total Cost Values for Both Methods

	GA					Sequential Approach				
Instance	**Supplier and facility related costs**	**Supplier and facility cost % in total**	**Routing related costs**	**Routing cost % in total**	**TOTAL COST**	**Supplier and facility related costs**	**Supplier and facility cost % in total**	**Routing related costs**	**Routing cost % in total**	**TOTAL COST**
mr101_25	7359	**49**	7803	**51**	**15162**	7882	**50**	7834	**50**	**15717**
mr205_25	5896	**51**	5557	**49**	**11453**	7882	**60**	5289	**40**	**13172**
mc109_25	6634	**64**	3699	**36**	**10333**	7677	**71**	3165	**29**	**10842**
mc206_25	7630	**70**	3239	**30**	**10869**	7676	**68**	3672	**32**	**11348**
mrc103_25	10469	**80**	2644	**20**	**13113**	10673	**78**	2950	**22**	**13623**
mrc207_25	7544	**67**	3710	**33**	**11254**	10673	**81**	2525	**19**	**13199**
mr101_50	14505	**54**	12348	**46**	**26853**	17410	**62**	10718	**38**	**28128**
mr205_50	12351	**55**	10294	**45**	**22645**	17410	**67**	8672	**33**	**26083**
mc109_50	15246	**72**	5911	**28**	**21157**	17628	**79**	4824	**21**	**22452**
mc206_50	14434	**73**	5361	**27**	**19795**	17903	**75**	6093	**25**	**23995**
mrc103_50	15901	**65**	8529	**35**	**24430**	20522	**79**	5583	**21**	**26105**
mrc207_50	13779	**64**	7666	**36**	**21445**	20522	**83**	4254	**17**	**24776**
mr101_100	28232	**60**	18781	**40**	**47013**	34651	**68**	16246	**32**	**50897**
mr205_100	33926	**69**	15011	**31**	**48937**	34651	**75**	11856	**25**	**46507**
mc109_100	31997	**72**	12477	**28**	**44474**	36726	**80**	9344	**20**	**46070**
mc206_100	30225	**75**	10005	**25**	**40230**	37007	**80**	9147	**20**	**46154**
mrc103_100	32457	**66**	15370	**31**	**47827**	39734	**75**	13095	**25**	**52829**
mrc207_100	31893	**68**	14845	**32**	**46738**	39734	**78**	11533	**22**	**51267**

5.3.5 More Research on Genetic Algorithms

Some further observations on the application of the GA are discussed below.

5.3.5.1 The behavior of genetic algorithms within a run

It is discussed in section 5.3.4 that, in almost all instances of different sizes, the results of the proposed solution methodology are satisfactory when compared to the sequential approach with which we have also solved our problems. Besides evaluating the results, we have also examined the improvement of the objective function through a run. The progresses of the best individual in the population, the worst individual in the population and the average objective value of the population are presented in Figure 5.6.

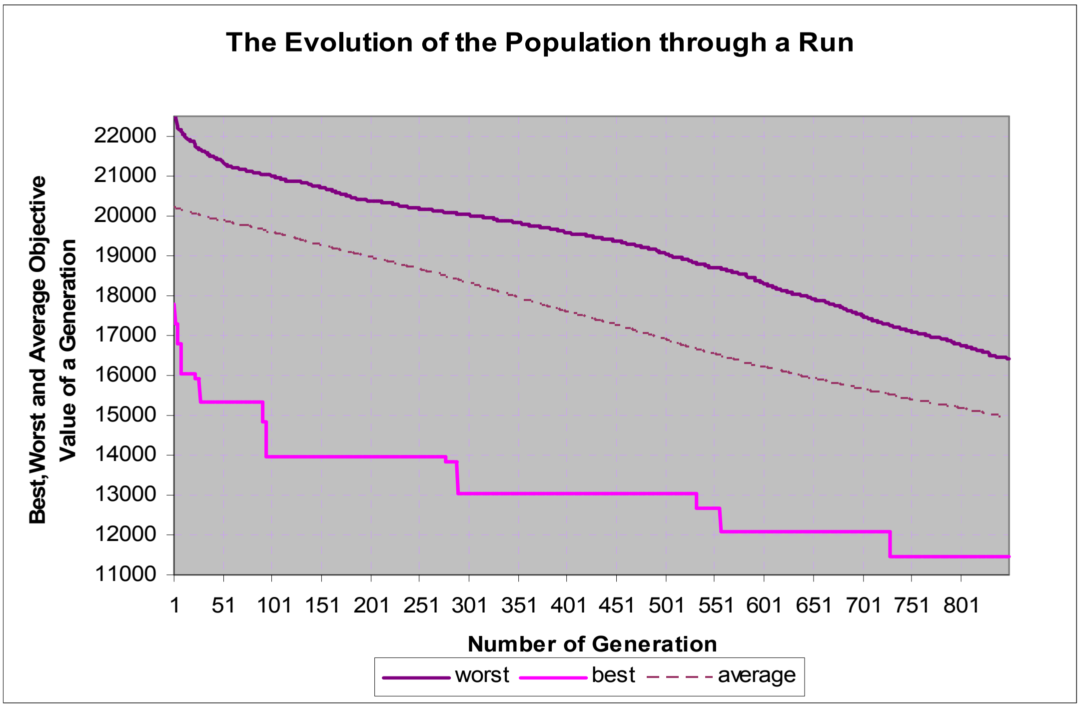


Figure 5.6 The Evolution of the Population with regard to Minimum, Maximum and Average Objective Values

The representative graph is for a 25-customer run of instance mr205. Trial with the least number of generations is chosen on purpose for the ease of examining the graph. It is observed that the fundamental behavior is similar for other instances or for larger problems.

The parameters (population size, number of generations) are adjusted as defined in section 5.3.1. As explained in that section, increasing the parameters does not improve the objective value of the best individual obtained.

In this figure, the ongoing progress through a run is summarized.

As expected, the improvement in the objective of the best individual is as a step function. While the process continues, the population gets better as the average and the worst individual always improves, but the best of the population may not change in each generation. Besides being linear, concave or convex which are the characteristics that may change from a run to another with regard to the initial population or pairs of parents chosen for each reproduction, the general behavior representing the improvement also verifies that the GA has worked without any unexpected deficiencies.

The borders of the search space that the population occupies and we have worked on through the run are drawn by the lines standing for the best and worst individuals of each generation. The dashed line between these represents the average of the population, which is always in the expected range.

5.3.5.2 A further study on initial population generation

When the problem size gets larger, it is obvious that the search space also gets larger. We have a very large search space for the 100 customer, 15 potential facility sites problem. Although we have worked with a large population and increased the number of generations to terminate the run, we want to see the effect of starting with a better population. For this reason we have modified the function for generating the initial population in the algorithm and made some trials to evaluate the effect of this modification.

Instead of generating individuals up to a number equal to the population size, replacing the duplicates with new ones, checking the population for covering all the alleles of all genes and starting the reproduction phase with that population; we have developed a modification by generating individuals as many as two times the population size, deleting the duplicates and then taking the best individuals up to the population size before passing to reproduction. This has given us the chance to work with a better population.

On test instance mr101_100, we have realized 5 runs with this modified version of the algorithm. The results of these 5 runs compared with the results of 5 runs obtained by the non-modified version are below.

Table 5.18 Results Obtained with both the Modified and the Non-Modified Initial Population Generation

	Modified Initial Population Generation		Non-Modified Initial Population Generation	
Runs	Objective value for the best solution	Computational time (sec.)	Objective value for the best solution	Computational time (sec.)
run1	47545	30872.96	48129	24676.76
run2	48810	31132.37	47785	23740.81
run3	47911	31222.67	47894	23994.09
run4	48534	31164.95	47497	24554.68
run5	47911	31167.12	47013	24061.35

Although it may be thought that after a greater number of runs, better results with the modified version could also be obtained, the results we got after 5 runs were not as good as we expected. Starting the procedure with a better population, a significant improvement on the cost value for the best solution was expected. However, as seen from the results, we did not observe this improvement. Moreover, the version working with a randomly generated initial population has reached even better results in shorter time. As a greater number of individuals are generated and evaluated to be sorted in the modified version, the computational time of a run increased.

It is possible to comment on this issue that not working with a more limited upper region of the solution space but working with a more diversified initial population instead has given better results with our crossover operator. Consequently, it was not preferred to work with this modification through the runs for other instances.

CHAPTER 6

CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

In this study, we have searched the solution techniques for location routing problems with time windows and shift times. First, we have presented both non-linear and linear mathematical formulations for the problem. We try to represent a realistic environment for the LRP, that is frequently encountered in real life distribution systems. Observing that solving the mathematical models and thus reaching the optimum solutions for problems of reasonable size is not possible, we have focused on the application of metaheuristics.

In our solution methodology, we have developed a solution approach based on genetic algorithms including a number of problem specific characteristics. Through this work, we have tried to avoid constructing a multi-phase approach or decomposing the LRPTW and shift times into subproblems and then solving them sequentially. In our GA approach, one of the subproblems of the main problem VRPTW has posed great difficulty to handle within the general framework. Through the entire search on solution techniques for VRPTW, it has been observed that it is worth to work on VRPTW solution techniques, which produce good results within short computational times because most of the proposed solutions based on heuristics require a great computational effort. But as our main concern is not VRPTW, but LRP with time windows; we have searched, found and used some predefined libraries that could be embedded to our code.

In our computational studies, we have tested our solution approach on a number of test problems developed from those of Solomon. As well as developing a solution approach, the needs for an appropriate comparison method and appropriate test instances also has formed an important part of the study.

In addition to evaluating the efficiency of the proposed approach for LRPTW and shift times, we have also considered the determination of some GA related parameters with regard to the problem size and problem characteristics. It is observed that the representation scheme for a solution of the problem also has importance in determining the size of the search space and this has a significant effect on GA parameters like the population size and the duration of the search. The possibility of representing a complete solution including the routes was also one of the issues we were concerned at the beginning of the study, but then ceased from because we did not come up with an appropriate chromosome representation. From the experimental studies, we have concluded that such a representation would also be inappropriate, because there would be a huge number of alternatives that needed to be searched, decreasing the efficiency.

Also it can be concluded that problem specific operators in a GA application improve the performance of the algorithm. This has also been encountered in some studies in the literature with or without some numeric applications to test its validity. In our study we have also observed that the problem specific crossover and mutation operators support this idea.

Another conclusion drawn from the experimental work is the change in the duration of a run with regard to the problem size. The problem size has one direct and one indirect effect on the duration of a run. First, the increase in the number of customers and in the number of potential sites mostly results in a higher number of open facilities which increases the computational time of the fitness function for a single solution. Second, in an indirect way, because the problem size and the search space get larger, the parameters, namely the population size and maximum number of generations of GA increase, and as a consequence the duration of a run lengthens.

Having discussed the main conclusions above in detail, when we think about the general contribution of this study; it can be first pointed out that we have worked on an environment, which is rarely considered even in LRP literature. The LRP itself being very complex, the time window and shift time issues are not included in the literature to the best of our knowledge. Moreover, we did not prefer to handle the problem by a multi-phase sequential approach and developed an integrated approach based on metaheuristics. Although the GA is widely used and results in satisfactory solutions for different types of problems, we encounter only a single GA application for LRP but for a very simple environment where the reliability of the method and the experimental work seems unsatisfactory (Su, 1998). So our study has been a leading one when both the problem environment and the solution approach are thought. The problem specific issues integrated within the general GA framework starting from the initial population generation to the operators are also contributions in the study.

We have tested the performance of our GA approach against the sequential approach which is known to be a good upper bound in the literature. Our GA approach has outperformed the sequential approach in 17 of the 18 instances by an amount ranging from 3.5% to 17.5% in the objective function value.

Further Research Issues:

For further research on the solution methodology proposed, one can construct an efficient GA application for the VRPTW as well and embed it within the general GA framework for the LRPTW. In this way, two GA applications can be integrated and a pure GA application can thus be generated for the LRPTW and shift times.

Based on the observation drawn in this study that problem specific operators bring out better solutions than the generally known operators, other problem specific operators apart from these may be generated and evaluated. In addition, other improving problem specific characteristics may be included in different phases of the algorithm.

Besides improving the GA application for the LRP, one can work on other solution methodologies for this problem type. During our researches, we have observed that the constraint programming based approaches are efficient for complex problems such as VRPTW. A solution methodology based on constraint programming can be considered for LRP with time window as well.

In our study, we have used a mathematical formulation based benchmark technique which is the upper bound obtained by the sequential approach. Studies on the performance analysis techniques of LRP with time windows and shift times which are NP-hard problems can be carried out; for example worst-case analysis, average-case analysis or other mathematical formulation based techniques such as a good lower bound could be developed.

Other aspects in the location-routing framework may also be to consider as proposed in Min (1998). Stochasticity in customer demands and travel times; multiple periods reflecting the changing nature of LRP parameters over time to the models, multiple objectives such as maximizing system benefits in addition to minimizing total system cost may be some of those aspects. Emphasizing multiple layers for including both inbound and outbound flows and considering the inventory decisions related to locations and routes are also further steps in system-wide approaches.

REFERENCES

Alp, O., Erkut, E. and Drezner, Z., 2003. An Efficient Genetic Algorithm for the p-Median Problem. *Annals of Operations Research* **122**, 21-42.

Ambrosino, D. and Scutella, M.G., 2005. Distribution Network Design: New Problems and Related Models. *European Journal of Operational Research* **165**, 610-624.

Backer, B., Furnon, V., Shaw, P., Kilby, P. and Prosser, P., 2000. Solving Vehicle Routing Problems Using Constraint Programming and Meta heuristics. *Journal of Heuristics* **6**, 501-523.

Ball, M., Magazine M., 1981. The Design and Analysis of Heuristics. *Networks* **11**, 215-219.

Ballou, R.H., 1990. A Continued Comparison of Several Popular Algorithms for Vehicle Routing and Scheduling. *Journal of Business Logistics* **11** (1), 111-127.

Bard, J.F., Kontoravdis, G. and Yu G, 2002. A Branch-and-Cut Procedure for the Vehicle Routing Problem with Time Windows. *Transportation Science* **36** (2) , 250-269.

Beasley, J.E., 1990. OR-Library-Distributing Test Problems by Electronic Mail. *Journal of Operational Research Society* **41** (11), 1069-1072.

Beasley, D., Bull, D.R. and Martin, R.R., 1993. An Overview of Genetic Algorithms-Fundemantals. *University Computing* **15** (2), 58-69.

Beasley, J.E. and Chu, P.C., 1996. A Genetic Algorithm for the Set Covering Problem. *European Journal of Operational Research* **94**, 392-404.

Blanton, J.L. and Wainwright, R.L., 1999. Multiple Vehicle Routing with Time and Capacity Constraints Using Genetic Algorithms in in *Foundations of Genetic Algorithm- 5 by Banzhaf W.,* Morgan Kaufmann Publishers, San Francisco, 452-459.

Bookbinder, J.H. and Reece, K.E., 1988. Vehicle Routing Considerations in Distribution System Design. *European Journal of Operational Research* **37**, 204-213.

Bramel, J. and Simchi-Levi, D., 1995. A Location Based Heuristic for General Routing Problems. *Operations Research* **43** (4), 649-660.

Campbell, J.F., 1994a. Integer Programming Formulations of Discrete Hub Location Problems. *European Journal of Operational Research* **72** (2), 387-405.

Cappanera, P., Gallo, G. and Maffioli, F., 2004. Discrete Facility Location and Routing of Obnoxious Activities. *Discrete Applied Mathematics* **133**, 3-28.

Chien, T.W., 1993. Heuristic Procedures for Practical Sized Uncapacitated Location-Capacitated Routing Problems. *Decision Sciences* **24** (5), 995-1021.

Christofides, N., Mingozzi, A. and Toth, P., 1979. The Vehicle Routing Problem. Combinatorial Optimization.. Wiley, New York.

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

Cooper, M.C., 1983. Freight Consolidation and Warehouse Location Strategies in Physical Distribution Systems. *Journal of Business Logistics* **4** (2), 53-74.

Correa, E.S., Steiner, M.T.A., Freitas, A.A. and Carnieri, C., 2004. A Genetic Algorithm for Solving Capaciated p-Median Problem. *Numerical Algorithms* **35**, 373-388.

Çetiner, S., 2003. An Iterative Hub Location and Routing Problem for Postal Delivery Systems. M.S. Thesis, Middle East Technical University, Ankara.

Davis, L., 1989. Adapting Operator Probabilities in Genetic Algorithms in *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Cali, 61-67.

Davis, L., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York.

De Backer, B. and Furnon, V., 1997. Meta-heuristics in Constraint Programming Experiments with Tabu Search on the Vehicle Routing Problem in *Proceedings of the Second International Conference on Metaheuristics (MIC'97)*, Sophia Antipolis, France.

Fogarty, T.C., 1989. Varying the Probability of Mutation in the Genetic Algorithms in *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Cali, 104-109.

Galvão, R.D. and ReVelle, C., 1996. A Lagrangean Heuristic for the Maximal Covering Location Problem. *European Journal of Operations Research* **88,** 114-123.

Gen, M. and Cheng, R., 1997. Genetic Algorithms and Engineering Design. John Wiley & Sons, London.

Gendreau, M., Hertz, A. and Laporte, G.,1994. A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science* **40** (10), 1276-1290.

Ghiani, G. and Improta, G., An Efficient Transformation of the Generalized Vehicle Routing Problem. *European Journal of Operational Research* **122**,11-17.

Gillett, B.E. and Miller L.R., 1974. A Heuristic Algorithm for the Vehicle Dispatch Problem. *Operations Research* **22**, 340-349.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston.

Goldberg, D.E., 1989. Sizing Populations for Serial and Paralel Genetic Algorithms in *Proceedings of the Third International Conferene on Genetic Algorithms by Schaffer, J. D.*, Morgan Kaufmann Publishers, San Mateo, 70-79.

Goldberg, D.E. and Deb K., 1991. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms in *Foundations of Genetic Algorithms by Rawlins, G.J.E.,* Morgan Kaufmann Publishers, San Mateo, 69-93.

Golden, B.L., Ball, M. and Bodin, L.D., 1981. Current and Future Research Directions in Network Optimization. *Computers and Operations Research* **8**, 71-81.

Haimovich, M. and Rinnooy Kan, A.H.G., 1985. Bounds and Heuristics for Capacitated Routing Problems. *Mathematics of Operations Research* **10**, 527-542.

Hall, R.W., 1987. Consolidation Strategy: Inventory, Vehicles and Terminals. *Journal of Business Logistics* **8** (2), 57-73.

Hansen, P.H., Hegedahl, B., Hjortkjaer, S. and Obel, B., 1994. A Heuristic Solution to the Warehouse Location-Routing Problem. *European Journal of Operational Research* **76**, 111-127.

Haupt, R.L. and Haupt, S.E., 1998. Practical Genetic Algorithms. John Wiley and Sons, Inc., New York.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI.

Hosage, C.M. and Goodchild, M.F., 1986. Discrete Space Location-Allocation Solutions From Genetic Algorithms. *Annals of Operational Research* **6**, 35-46.

ILOG Dispatcher 4.1 User's Manual, ILOG S.A., Cedex, 2005

ILOG Dispatcher 4.2 Reference Manual, ILOG S.A., Cedex, 2006

ILOG Dispatcher 4.2 Release Notes, ILOG S.A., Cedex, 2006

Jaramillo, H.J., Bhadury, J. and Batta, R., 2002. On the Use of Genetic Algorithms to Solve Location Problems. *Computers and Operations Research* **29**, 761-779.

Khumawala, B.M., 1972. An Efficient Branch and Bound Algorithm for The Warehouse Location Problem. *Management Science* **18**, 718-731.

Koerkel, M., 1989. On the Exact Solution of Large-Scale Simple Plant Location Problems. *European Journal of Operations Research* **39**,157-173.

Ko, H.J. and Evans, G.W., 2006. A Genetic Algorithm Based Heuristic for the Dynamic Integrated Forward/Reverse Logistics Network for 3PLs. *Computers & Operations Research*, Forthcoming.

Kontoravdis, G. and Bard, J. F, 1995. A GRASP for the Vehicle Routing Problem with Time Windows. *ORSA Journal on Computing* **7** (1), 10-23 .

Laporte G., 1992. The Vehicle Routing Problem: An overview of Exact and Approximate Algorithms. *European Journal of Operational Research* **59**, 345-358.

Laporte, G., 1988. Location Routing Problems in *Vehicle Routing: Methods and Studies by Golden B.L., Assad A.A.*, North Holland Publishing, Amsterdam, 163-198.

Laporte, G., Nobert, Y. and Arpi, D., 1986. An Exact Algorithm for Solving a Capacitated Location-Routing Problem. *Annals of Operational Research* **6**, 293-310.

Laporte, G., Nobert, Y. and Taillefeer, S., 1988. Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problems. *Transportation Science* **22** (3), 161-172.

Lin, C.K.Y., Chow, C.K. and Chen, A., 2002. A Location-Routing-Loading Problem for Bill Delivery Services. *Computers & Industrial Engineering* **43**, 5-25.

Min, H., 1996. Consolidation Terminal Location-Allocation and Consolidated Routing Problems. *Journal of Business Logistics* **17** (2), 235-263.

Min, H., Jayaraman, V. and Srivastava, R., 1998. Combined Location-Routing Problems: A Synthesis and Future Research Directions. *European Journal of Operational Research* **108**, 1-15.

Nelson, M.D., Nygard, K.E., Griffin, J.H. and Shreve W.E., 1985. Implemantation Techniques for the Vehicle Routing Problem. *Computers & Operations Research* **12** (3), 273-283.

Or, I. And Pierskalla, W.P., 1979. A Transportation Location-Allocation Model for Regional Blood Banking. *AIIE Transactions* **11** (2), 86-95.

Perl, J., 1983. A Unified Warehouse Location-Routing Methodology. *Ph.D. Dissertation*, Northern University.

Perl, J. and Daskin, M.S., 1984. A Unified Warehouse Location-Routing Methodology. *Journal of Business Logistics* **5** (1), 92-111.

Perl, J. and Daskin, M.S., 1985. A Warehouse Location-Routing Problem. *Transportation Research Part B: Methodological* **19**, 381-396.

Preston, P. and Kozan, E., 2001. An Approach to Determine Storage Locations of Containers at Seaport Terminals. *Computers & Operations Research* **28**,983-995.

Reeves, C.R. and Rowe, J.E., 2003. Genetic Algorithms: Principles and Perspectives, A Guide to GA Theory. Kluwer Academic Publishers, London.

Rousseau, L., Gendreau, M. and Pesant, G, 2002. Using Constrained-Based Operators to Solve the Vehicle Routing Problem with Time Windows. *Journal of Heuristics* **8**, 43-58.

Salhi, S. and Gamal, M.D.H, 2003. A Genetic Algorithm Based Approach for the Uncapacitated Continuous Location-Allocation Problem. *Annals of Operations Research* **123**, 203-222.

Salhi, S. and Rand, G., 1989. The Effect of Ignoring Routes When Locating Depots. *European Journal of Operational Research* **39**, 150-156.

Sambola, M.A., Diaz, J.A. and Fernandez, E., 2005. A Compact Model and Tight Bounds for a Combined Location-Routing Problem. *Computers & Operations Research* **32**, 407-428.

Schaffer, J.D., Caruana, R.A., Eshelman, L.J and Das R., 1989. A Study of Control Parameters Affecting Online Performance of Genetic algorithms for Function Optimization in *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Cali, 51-60.

Schrage, L., 1981. Formulations and Structure of more Complex / Realistic Routing and Scheduling Problems. *Networks* **11**, 229-232.

Shaw, P., 1998. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems in *Principle and Practice of Constraint Programming—CP98* Pisa, Italy, Springer-Verlag LNCS, Berlin.

Smith, J. and Vavak, F., 1999. Replacement Strategies in Steady State Genetic Algorithms : Static Environments in *Foundations of Genetic Algorithms-5 by Banzhaf W.*, Morgan Kaufmann Publishers, San Francisco,219-233.

Solomon, M.M., 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Windows. *Operations Research* **35** (2), 254-265.

Solomon, M.M., (last updated) 24 March 2005. VRPTW Benchmark Problems. http://w.cba.neu.edu/~msolomon/problems.htm (last accessed 22 May 2006)

Srivastava R., 1993. Alternate Solution Procedures for the Location-Routing Problem. *Omega International Journal of Management Science* **21** (4), 497-506.

Srivastava, R. and Benton W.C., 1990. The Location-Routing Problem: Considerations in Physical Distribution System Design. *Computers & Operations Research* **5**, 427-435.

Starkweather, T., McDaniel, S., Mathias, K., Whitley, D. and Whitley C., 1997. A Comparison of Genetic Sequencing Operators in *Proceedings of Seventh International Conference on Genetic Algorithms by Back, T.*, Morgan Kaufmann Publishers, San Francisco, 69-76.

Su, C.T., 1998. Locations and Vehicle Routing Designs of Physical Distribution Systems. *Production Planning & Control* **9** (7), 650-659.

Syswerda, G., 1991. A Study of Reproduction in Generational and Steady-State Genetic Algorithms in *Foundations of Genetic Algorithms by Rawlins, G.J.E.,* Morgan Kaufmann Publishers, San Mateo, 94-101.

Thangiah, S.R. and Salhi, S., 2001. Genetic Clustering: An Adaptive Heuristic for Multi-depot Vehicle Routing Problems. *Applied Artificial Intelligence* **15**, 361-383.

Thangiah, S.R., Vinayagamoorthy, R. and Gubbi, A.V., 1999. Vehicle Routing with Time Deadlines using Genetic and Local Algorithms in *Foundations of Genetic Algorithm- 5 by Banzhaf W.,* Morgan Kaufmann Publishers, San Francisco, 506-513.

Tillman, F., 1969. The Multiple Terminal Delivery Problem with Stochastic Demands. *Transportation Science* **3**,192-204.

Topçuoğlu, H., Corut, F., Ermis, M. and Yılmaz, G., 2005. Solving the Uncapacitated Hub Location Problem Using Genetic Algorithms. *Computers & Operations Research* **32**, 967-984.

Tuzun, D. and Burke, L.I., 1999. A Two-phase Tabu Search Approach to the Location Routing Problem. *European Journal of Operational Research* **116**, 87-99.

Wasner, M. and Zäpfel, G., 2004. An Integrated Multi-Depot Hub-Location Vehicle Routing Model for Network Planning of Parcel Service. *International Journal of Production Economics* **90**, 403-419.

Whitley, D., 1989. The Genitor Algorithm and Selection Pressure : Why Rank-Based Allocation of Reproductive Trials is Best in *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Cali, 116-121.

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

Zanakis, S.H. and Evans, J. R.,1981. Heuristic 'Optimization': Why, When, and How to Use It. *Interfaces* **11** (5), 84-90.

Zhou, G., Min, H. and Gen, M., 2003. A Genetic Algorithm Approach to the Bi-criteria Allocation of Customers to Warehouses. *International Journal of Production Economics* **86**, 35-45.

APPENDIX A

GAMS MODELS

The mathematical formulations translated to GAMS are presented below. The number of customers or potential facility sites and the related numerical data in the models are not included and the emphasis is on the definition of the elements of the model, the objective function and the constraints.

Linear Model for the Environment with Hard Time-Windows

Sets

```
n nodes: customers and potential facility sites
c(n)  customers
f(n)  potential facility sites
s     suppliers
k     vehicles
l     facility types                    ;

alias (n,p,r), (c,ab)                   ;
```

Scalars

```
tc0 transportation cost from supply source to facility sites
tc1 transportation cost among nodes(facilities and customers)
h   time it takes to cross a km
w   the cost of waiting for a unit
Kp  the capacity of a vehicle
```

FV the fixed cost of a vehicle

VF the variable cost for potential sites

CONST a constant big enough ;

Parameters

de(c) demand of customers

FF(l) the fixed costs for different types of facilities

Cp(l) the capacity limits for different types of facilities

e (n) the earliest service starting time for nodes

se(n) the required service time for customers or facility sites

la (n) the latest service time for nodes

d2(f) distance between supply source and potential sites(km)

m the number of customers ; m = card(c);

Table d (n,p) distance between node pairs ;

Binary Variables

x(n,p,k) decides if a node in n precedes another in p on the route of vehicle-k

z(f,l) decides if a facility of type l is opened at that site

zs(f) decides if a site is used to open any type of facility

y(c,f) decides if a customer is assigned to a potential facility site

v(k) decides if a vehicle is used or not

t(n,p,k) auxiliary binary variable ;

Positive variables

q(s,f) the quantity shipped from a supply source to a facility

wt(n) waiting time at a node

at(n) starting time of service at a node

u(c,k) auxiliary variable

rt(f) returning time to a facility ;

Variable o the objective value ;

Equations

tcost "total objective cost of the system"

eq1(c) "each customer assigned to a single DC/vehicle"

eq2(k) "capacity of each vehicle not exceeded"

eq3(c,ab,k) "subtour elimination constraints"

eq4(p,k)"every node entered is left"

eq5(k) "every vehicle is used at most from one fac. site"

eq6(f) "balance equation between input/output for fac. sites"

eq7(f) "capacity control for each fac. site"

eq8(f) "at most one type of facility is opened at a potential site"

eq9(f) "whether the site is used or not is related with whether a facility is opened at that site"

eq10(c,f,k) "a customer can be assigned to a vehicle only if they belong to the same vehicle"

eq11a(c,f,k) "determines the relations between arrival of nodes and returning time to facilities"

eq11b(c,f,k) "determines the relations between arrival of nodes and returning time to facilities"

eq11c(c,f,k) "determines the relations between arrival of nodes and returning time to facilities"

eq12a(n,c,k) "determines the relations between arrival times of nodes"

eq12b(n,c,k) "determines the relations between arrival times of nodes"

eq12c(n,c,k) "determines the relations between arrival times of nodes"

eq13(n) "assures not to violate earliest service starting time for a node"

eq14(c) "assures not to violate latest service time for a customer"

eq15(f) "assures not to violate latest service time for a facility" ;

tcost .. o =e= sum ((f,l), FF(l)*z(f,l)) + sum (k, FV * v(k)) + sum (f , VF * sum (c, de(c)*y(c,f))) + sum((s,f) , tc0 * q(s,f) * d2(f)) + sum ((n,p,k) , tc1 * d(n,p) * x(n,p,k)) + sum(n , w * wt(n));

```
eq1(c) ..      sum ( (n,k), x (n,c,k) ) =e= 1                        ;
eq2(k) ..      sum ( c, de(c) * sum( n, x(n,c,k))) =l= Kp * v(k)    ;
eq3(c,ab,k) ..   u(c,k) - u(ab,k)+ m * x(c,ab,k) =l= m - 1           ;
eq4(p,k) ..    sum ( n, x(p,n,k) ) - sum (n, x(n,p,k) ) =e= 0          ;
eq5(k) ..      sum ( (c,f), x(c,f,k) )  =l= 1                        ;
eq6(f) ..      sum ( s, q(s,f) ) - sum ( c, de(c) * y(c,f) ) =e= 0      ;
eq7(f) ..      sum ( c, de(c) * y(c,f) ) - sum( l, Cp(l) * z(f,l)) =l= 0   ;
eq8(f) ..      sum ( l, z(f,l) ) =l= 1   ;
eq9(f) ..      sum ( l, z(f,l) ) =e= zs(f) ;
eq10(c,f,k) ..  -y(c,f) + sum (r, ( x(c,r,k) + x(r,f,k) ) ) =l= 1     ;
eq11a(c,f,k) .. at(c) + wt(c) + se(c) + d(c,f) * h - rt(f) =l= CONST * t(c,f,k);
eq11b(c,f,k) .. -at(c) - wt(c) - se(c) - d(c,f) * h + rt(f) =l= CONST * t(c,f,k);
eq11c(c,f,k) .. x(c,f,k) =l= CONST - (CONST * t(c,f,k));
eq12a(n,c,k) .. at(n) + wt(n) + se(n) + d(n,c) * h - at(c) =l= CONST * t(n,c,k);
eq12b(n,c,k) .. -at(n) - wt(n) - se(n) - d(n,c) * h + at(c) =l= CONST * t(n,c,k);
eq12c(n,c,k) .. x(n,c,k) =l= CONST - (CONST * t(n,c,k));
eq13(n) ..  at(n) + wt(n) =g= e(n);
eq14(c) ..  at(c) + wt(c) =l= la(c) ;
eq15(f) ..  rt(f) =l= la(f) * zs(f) ;

*exclude diagonal  ref:awktsp.gms

 x.fx(n,n,k) = 0 ;

*depolarin baslangic noktasi oldugunu gostermek icin

 at.fx(f) = 0 ;

model LRP1 / all / ;

solve LRP1 using mip minimizing o ;
```

Non-Linear Model for the Environment with Hard Time-Windows

Sets

n nodes: customers and potential facility sites
c(n) customers
f(n) potential facility sites
s suppliers
k vehicles
l facility types ;

alias (n,p,r), (c,ab) ;

Scalars

tc0 transportation cost from supply source to facility sites
tc1 transportation cost among nodes(facilities and customers)
h time it takes to cross a km
w the cost of waiting for a unit
Kp the capacity of a vehicle
FV the fixed cost of a vehicle
VF the variable cost for potential sites ;

Parameters

de(c) demand of customers
FF(l) the fixed costs for different types of facilities
Cp(l) the capacity limits for different types of facilities
e (n) the earliest service starting time for nodes
se(n) the required service time for customers or facility sites
la (n) the latest service time for nodes
d2(f) distance between supply source and potential sites
m the number of customers ; m = card(c);

Table d (n,p) distance between node pairs ;

Binary Variables

x(n,p,k) decides if a node in n precedes another in p on the route of vehicle-k
z(f,l) decides if a facility of type l is opened at that site
zs(f) decides if a site is used to open any type of facility
y(c,f) decides if a customer is assigned to a potential facility site
v(k) decides if a vehicle is used or not ;

Positive variables

q(s,f) the quantity shipped from a supply source to a facility
wt(n) waiting time at a node
at(n) starting time of service at a node
u(c,k) auxiliary variable
rt(f) returning time to a facility ;

Variable o the objective value ;

Equations

tcost "total objective cost of the system"
eq1(c) "each customer assigned to a single DC/vehicle"
eq2(k) "capacity of each vehicle not exceeded"
eq3(c,ab,k) "subtour elimination constraints"
eq4(p,k)"every node entered is left"
eq5(k) "every vehicle is used at most from one fac. site"
eq6(f) "balance equation between input/output for fac. sites"
eq7(f) "capacity control for each fac. site"
eq8(f) "at most one type of facility is opened at a potential site"
eq9(f) "whether the site is used or not is related with whether a facility is opened at that site"

```
  eq10(c,f,k) "a customer can be assigned to a vehicle only if they belong to the same vehicle"
  eq11(c,f,k) "determines the relations between arrival of nodes and returning time to facilities"
  eq12(n,c,k) "determines the relations between arrival times of nodes"
  eq13(n) "assures not to violate earliest service starting time for a node"
  eq14(c) "assures not to violate latest service time for a customer"
  eq15(f) "assures not to violate latest service time for a facility"      ;

tcost ..  o =e=  sum ((f,l), FF(l)*z(f,l) ) + sum ( k,  FV * v(k) ) + sum (f , VF * sum (c, de(c)*y(c,f) )) + sum((s,f) , tc0 * q(s,f) * d2(f)) + sum ((n,p,k) , tc1 * d(n,p) * x(n,p,k)) + sum( n , w * wt(n));

eq1(c) ..      sum ( (n,k), x (n,c,k) ) =e= 1                          ;
eq2(k) ..      sum ( c, de(c) * sum( n, x(n,c,k))) =l= Kp * v(k)    ;
eq3(c,ab,k) ..   u(c,k) - u(ab,k)+ m * x(c,ab,k) =l= m - 1            ;
eq4(p,k) ..    sum ( n, x(p,n,k) ) - sum (n, x(n,p,k) ) =e= 0          ;
eq5(k) ..      sum ( (c,f), x(c,f,k) )  =l= 1                         ;
eq6(f) ..      sum ( s, q(s,f) ) - sum ( c, de(c) * y(c,f) ) =e= 0      ;
eq7(f) ..      sum ( c, de(c) * y(c,f) ) - sum( l, Cp(l) * z(f,l)) =l= 0   ;
eq8(f) ..      sum ( l, z(f,l) ) =l= 1    ;
eq9(f) ..      sum ( l, z(f,l) ) =e= zs(f) ;
eq10(c,f,k) ..  -y(c,f) + sum (r, ( x(c,r,k) + x(r,f,k) ) ) =l= 1      ;
eq11(c,f,k) .. at(c) * X(c,f,k) + wt(c) * X(c,f,k) + se(c)* X(c,f,k) + d(c,f) * h * X(c,f,k) =l= rt(f);
eq12(n,c,k) .. at(n) * X(n,c,k) + wt(n) * X(n,c,k) + se(n)* X(n,c,k) + d(n,c) * h * X(n,c,k) =l= at(c);
eq13(n) ..  at(n) + wt(n) =g= e(n);
eq14(c) ..  at(c) + wt(c) =l= la(c) ;
eq15(f) ..  rt(f) =l= la(f) * zs(f) ;
```

```
*exclude diagonal  ref:awktsp.gms
 x.fx(n,n,k) = 0 ;
*depolarin baslangic noktasi oldugunu gostermek icin
 at.fx(f) = 0 ;
model LRP1 / all / ;
solve LRP1 using minlp minimizing o ;
```

APPENDIX B

ILOG DISPATCHER

In our work, we decided to use the approach developed by the ILOG team, the ILOG Dispatcher which proposes a method using local search techniques in a constraint programming (CP) framework. Constraint programming issued from the field of artificial intelligence has been helpful in solution of real world problems from a number of different domains as it provides good modeling flexibility (Rousseau, 2002). However, despite some improving studies, the search techniques in constraint programming cause long execution times difficult to handle. That's why the use of heuristics with constraint programming in the same framework offers better solutions. As referred to in Rousseau (2002); studies of DeBacker & Furnon (1997) and Shaw (1998) are examples of efficient techniques combining CP and heuristics.

In the field of Vehicle Routing, based on the above ideas, the ILOG equips developed a solution methodology within Constraint Programming. It can also be described as a specialized case of ILOG Solver (CP solver) enriched by local search techniques and metaheuristics. The basics of ILOG Dispatcher are defined in Backer et al. (2000).

This proposes a method using local search techniques in a constraint programming framework. Because of long execution times, in ILOG Dispatcher constraint programming is not used to search the solution space. Instead local search techniques which are promising in finding good solutions in very short periods are used for the search and constraint programming is used to check the validity of solutions and determine the values of the constrained variables. In addition to these, in order to avoid getting trapped in local optima, metaheuristics such as tabu search and guided local search are enabled to be used. So a complete search method is developed to solve the highly constrained vehicle routing problems. How the algorithm works and the techniques possible to use within the classes and functions of ILOG Dispatcher are as follows. More detailed knowledge on Dispatcher can be obtained from ILOG Dispatcher 4.1 User's Manual (2005), ILOG Dispatcher 4.2 Reference Manual (2006) and ILOG Dispatcher 4.2 Release Notes (2006).

1. First Solution Heuristics are used to generate a first solution to a routing problem, where the first solution is a "good" feasible solution obtained very quickly. The available first solution heuristics that could be included in the related code are:
   - Enumeration Heuristic

     It builds a solution to the problem using an algorithm that completely explores the search space. Hence, it should be used in small problems.
   - Savings Heuristic

     It is based on the trade-off between more vehicles with shorter routes and fewer vehicles with longer routes. It depends basically on Clarke&Wright's savings algorithm, but improved to handle additional constraints such as time windows.
   - Sweep Heuristic

     This method uses angles calculated regarding a reference point, the location of the depot and the customer locations. As long as the capacity and time window constraints are satisfied, the vehicles are loaded by customers in an increasing angle order.
   - Nearest-to-depot Heuristic

     In this heuristic, the routes are constructed by adding a customer without violating constraints starting from the one nearest to the depot at each iteration. If this is not possible, this route is assumed to be completed and another route is started.
   - Nearest Addition Heuristic

This heuristic also progresses as the nearest-to-depot heuristic. But the difference is that the customers are not added to the routes according to their nearness to the depot but according to their nearness to the last customer on the partial route.

- Insertion Heuristic

This heuristic depends on the positioning of customers with the least cost increase in a feasible way in existing or new routes.

2. After good, feasible first solution is found quickly by one of the first solution heuristics; they are moreover improved by searching the neighborhoods. To perform this search, there exist a number of predefined neighborhoods within Dispatcher libraries. The neighborhoods define a set of solution changes over an existing solution and reach new solutions by this way. The neighborhoods in Dispatcher can be classified and explained as follows:
   - Intra-Route Improvements
     - 2-opt: It reverses a section of a tour by deleting two arcs and replacing them with two others.
     - Or-opt: It moves arcs related with 1, 2 or 3 adjacent nodes and replaces them with other possible alternatives.
   - Inter-Route Improvements
     - Relocate : A visit within a route is replaced within another route by this operator, if the constraints are still satisfied.
     - Cross : This operator exchanges the ends of two routes.
     - Exchange : It swaps two visits from two different routes if the constraints are still satisfied.
   - Other Improvement Operators

   Those operators search the neighborhood by changing whether a visit is performed or unperformed within a route.

3. The third important element in the search is the definition of the solutions that will be accepted. Basic heuristics for the search is strictly accepting a neighbor better than the existing one. Based on this criterion, there exist two search techniques within Dispatcher. *First accept* search takes the first encountered cost decreasing legal solution, whereas *best accept* searches all legal neighborhoods and takes the most improving one. The way how the neighborhoods are created is already defined by the second phase.

4. Along with these search heuristics, metaheuristics can also be included in the solution of the Dispatcher if needed. The basic searches by first accept or best accept stop when there are no improving moves. However, in some cases, it may be profitable to go on searching even improving a worst solution to reach better regions and to prevent getting trapped in local optima. Dispatcher provides two metaheuristics; tabu search and guided local search for this purpose. Tabu Search carries features like tabu lists and aspiration criterion. Guided local search works by making a series of greedy searches by minimizing the true objective of the problem added to a penalty cost for a local minimum. Guided tabu search (combination of simple tabu search and guided local search) is also possible to use.

Among these; in our algorithm we have used the savings heuristic as the first solution heuristic, the defined five neighborhoods for search and first accept strategy.

APPENDIX C

THE PSEUDOCODES FOR THE GA-BASED HEURISTIC

The pseudocodes which are the basis of the code of the algorithm are presented here. First the main flow is given as a whole composed of small procedures. Then each procedure is explained in the order of appearance in the main body of the algorithm.

The first group of procedures bordered by the curly brackets are realized only once at the beginning of the algorithm to read the data and construct initial arrays; generate, test and evaluate the initial population.

<u>Main Body of the Algorithm</u>

```
Procedure ReadData
Procedure FeasAssign
if (there exists at least one customer that can't be assigned to a facility feasibly) then
        display "the problem has no feasible solutions"
        stop the algorithm
else
        Procedure Nearness
        Procedure Nearfac
        Procedure PopGen
        Procedure Dup
        Procedure CoverCheck

        for  k=1 to PopulationSize  do
                Procedure Decod
                for j=CustomerNo to NodeNo
                                if facility-j is closed
                                        display "facility-j in individual-k is closed"
                                else generate the input files for the VRPTW for each
                                     open facility-j
                                     solve the VRPTW for each open facility-j
                                     assign the routing cost to array RCost
                Procedure CalcCost
```

```
endfor

assign a very large number to minf
assign a very large number to globalminf

Procedure PopStatus1
Procedure PopStatus2
Procedure PopStatus3
Procedure Worst
Procedure GlobalCheck

while (NumberOfGeneration =< MaximumGeneration)  do
        NEWGEN:
        Procedure ParentSelect
        Procedure Cross /*output newindiv*/
        generate a random number in [0,1]
        assign this to mut
        if (mut < Pmut)
        Procedure Mutation/*output newindiv*/
        Procedure IndCheck  for the newindiv
        if  (newindiv exists in the population)  then
                return NEWGEN
        else
        for the newindiv
                Procedure Decod
                for j=CustomerNo to NodeNo
                                if facility-j is closed
                                        display "facility-j in newindiv is closed"
                                else generate the input files for the VRPTW for
                                     each open facility-j
                                     solve the VRPTW for each open facility-j
                                     assign the routing cost to array RCost
                Procedure CalcCost
        endfor
        if (fitness of the newindiv < fitness of the worst) then
                        Procedure DelIns
        else return NEWGEN

Procedure PopStatus1
Procedure PopStatus2
Procedure PopStatus3
Procedure Worst
Procedure GlobalCheck

endwhile
```

Procedure ReadData

```
for i=1 to CustomerNo do

        read x coordinate and assign to array ap
        read y coordinate and assign to array or
        read demand and assign to array dem
        read ready time and assign to array ear
        read due time and assign to array lat
        read service time and assign to array ser

endfor

for i=CustomerNo to NodeNo (NodeNo=CustomerNo+FacilityNo) do
        read x coordinate and assign to array ap
        read y coordinate and assign to array or
        read ready time and assign to array ear
        read shift time and assign to array lat
        read service time and assign to array ser
endfor

for i=NodeNo to NodeNo+SupplierNo do
        read x coordinate and assign to array ap
        read y coordinate and assign to array or
endfor

for i=1 to NodeNo do
        for j=1 to NodeNo do
                calculate Euclidean distances among nodes
                assign to array dist
        endfor
endfor

for i=NodeNo to NodeNo+SupplierNo do
        for j=CustomerNo to NodeNo do
                calculate Euclidean distance between suppliers and potential facility
                sites
                assign to array dist
        endfor
endfor

for i=NodeNo to NodeNo+SupplierNo do
        for j=CustomerNo to NodeNo do
                calculate distances' costs by multiplying array dist with unit cost and
                assign to array Mdist
        endfor
```

endfor

for i=1 **to** NodeNo **do**
    for j=1 **to** NodeNo **do**
        calculate Euclidean times among nodes by multiplying array dist with time coefficient to cross a unit distance
        assign to *array Hdist*
    endfor
endfor

display all the arrays defined above in file Outdata

<u>Procedure FeasAssign</u>

for i=1 **to** CustomerNo **do**
    for j=CustomerNo **to** NodeNo **do**
        if ((the time it takes from fac-j to cust-i +the service time at cust-i+the time it takes from cust-i to fac-j =< shift time-j) **and** (the time it takes from fac-j to cust-i +the service time at cust-i=<latest service time-i) **and** (the earliest delivery time at cust-i+the service time at cust-i+the time it takes from cust-i to fac-j=<shift time-j)) **then**
        assign this facility to array FeAl-i which is the set of facilities cust-i can be feasibly assigned
        endif
    endfor
endfor

count the number of facilities each customer can be assigned to
calculate the probability each feasible facility has in order to be chosen for each customer

display the arrays for each customer

<u>Procedure Nearness</u>

for i=1 **to** CustomerNo **do**
    assign each customer as the nearest customer to itself (0^{th} element in the *array near1* related with each customer)
    for j=1 **to** CustomerNo **do**
        if (j is different than i) **then**
            compare the distance cust-i / cust-j and cust-i / the last customer assigned to the *array near1* related with cust-i
            if (greater) **then**
                assign this customer as the last element in the *array near1* of cust-i

else continue to compare $distance_{ij}$ with the distances between cust-i and other customers assigned to *array near1* starting from the end of the array
place customer-j to the appropriate rank in the *array near1*
end else
endif
endfor
endfor

for i=1 **to** CustomerNo **do**
construct *array near2* by deleting each customer from its own *array near1*
endfor

<u>Procedure Nearfac</u>

for i=1 **to** CustomerNo **do**
compare the distances of facilities to the cust-i
find the nearest facility to cust-i
assign it as the i^{th} element of *array nearfac*
endfor

<u>Procedure PopGen</u>

for k=1 **to** PopulationSize **do**
Procedure IndGen
endfor

<u>Procedure Dup</u>

for i=1 **to** PopulationSize **do**
for j=i+1 **to** PopulationSize **do**
compare each gene of individual-i with individual–j
if (all the genes are identical) **then**
delete the duplicate from the population and move the rest of the population towards the beginning of the population by one
update the counter for customer-facility pairs for each gene of the deleted individual
increase nodeleted by one
continue to check the rest of the population for the duplicates of cust-i
endif
endfor
endfor

for i=1 **to** nodeleted **do**
Procedure IndGen
Procedure IndCheck
if (the new individual created by IndGen doesn't exist in the population) **then**
add this individual to the population
else **decrease** the counter for each cust-facility pair of the new individual by one
return to *Procedure IndGen*
endfor

<u>Procedure IndGen</u>

for i=1 **to** CustomerNo **do**
choose a facility from the feasible set of customer-i randomly
assign customer-i to that facility
increase the counter for the above mentioned customer-facility pair by one
endfor

<u>Procedure IndCheck</u>

compare each gene of the new individual with all individuals present in the population
if (all genes are identical to any of the present individuals) **then**
display "identical individual exists"
else **display** "no identical exists"

<u>Procedure CoverCheck</u>

for i=1 **to** CustomerNo **do**
if (a facility that is in the feasible set for that customer(determined by Procedure FeasAssign) is not used in any of the individuals(call it *poor*)) **then**
choose randomly another facility from the feasible set of that customer(call it *exchange*)
if (*exchange* is used once or less than once) **then**
return to the previous step and choose another *exchange*
else **choose** randomly an individual(call it *victim*)
if (*victim* contains *exchange* for the mentioned customer)**then**
replace *exchange* by *poor*
else return to choosing another *victim*
end else
endif

```
endfor
```

Procedure Decod

```
for i=1 to CustomerNo do
        for j=CustomerNo to NodeNo do
                if ( facility-j stands for the gene related with that customer) then
                        assign     Yijk =1
                else   assign      Yijk =0
    endfor
endfor

for j=CustomerNo to NodeNo do
         if  (any Yijk=1 for that facility-j) then
                assign Zjk=1
        else Zjk =0
endfor
```

Procedure CalcCost

```
for j=CustomerNo to NodeNo do
        if (facility-j is open)
                add the routing cost to TotalCost-j
                add the variable facility cost to TotalCost-j
                add the transportation cost from supply source to facility to
                TotalCost-j
                add the facility fixed cost to TotalCost-j regarding the capacity of the
                facility
        endif
endfor
```

Procedure Popstatus (PopStatus1, PopStatus2, PopStatus3, Worst, GlobalCheck)

(PopStatus1)

```
for k=1 to PopulationSize do
        if (fitness of the individual-k < minf) then
                update minf = fitness of the individual-k
                update minfit = minf
         endif
endfor
```

(PopStatus2)

for k=1 **to** PopulationSize **do**
    if (fitness of the individual-k > maxf) **then**
        update maxf = fitness of the individual-k
        update maxfit = maxf
    endif
endfor

(PopStatus3)

for k=1 **to** PopulationSize **do**
    add the fitness of individual-k to sum
endfor

calculate the average fitness of the population by dividing sum with population size

(Worst)

for k=1 **to** PopSize **do**
    if (fitness of individual-k = maxfit) **then**
        assign indexworst = k
        assign Worstindiv = individual-k
    endif
endfor

(GlobalCheck)

if (the best fitness value for that generation (*minf* found in PopStatus1) is better than *globalmin*) **then**
    for k=1 **to** PopulationSize **do**
        Search the population to find the individual with fitness equal to *minf*
    assign indexbest = k
    assign Globalbest = individual-k
endif

Procedure ParentSelect

for i=1 **to** ParentNo **do**
    for j=1 **to** TournamentSize **do**
        choose randomly individuals
        assign them as candidate parents
    endfor
    find the candidate parent with the smallest fitness
    assign this candidate parent as a parent
endfor

<u>Procedure DelIns</u>

display the new individual coming out from IndCheck and its fitness
replace the worst individual in the population with the new individual
replace the fitness of the worst individual in the population with the fitness of new individual
for i=CustomerNo **to** NodeNo **do**
    replace the total demands of facilities in the worst individual with total demands in the new individual
    replace the routing costs in worst individual with routing costs in the new individual
    replace the open/close variables in worst individual with the variables in the new individual
endfor

<u>Procedure Cross /*output newindiv*/</u>

for i=1 **to** CustomerNo **do**
    for j=1 **to** ParentNo **do**
        determine the facilities each customer is assigned to in each parent (by reading the gene value related with each customer)
        construct the *array pos* related with each customer using the above data
    endfor
endfor

for i=1 **to** CustomerNo **do**
    mark each customer as *unassigned*
endfor
for i=1 **to** CustomerNo **do**
    if (a customer is assigned to the same facility in both customers) **then**
        assign this customer to that facility in the offspring
        mark it as *assigned*
        if (the nearest customer to that customer has that facility in a parent) **then**
            assign it to that facility
            mark as *assigned*
            update the *array pos* related to this customer
        endif
    endif
endfor

for j=1 **to** CustomerNo **do**
        if (cust-j is *unassigned*) then
            assign i=0 /*i stands as the nearness index*/
            ASSIG:

```
        assign k as the i^th nearest element in array near2
        if (any facilities possible for cust-j also possible for cust-k)
        then
            if ( both possible facilities are equal )  then
                increase i by 1        /* Pass to the next customer
                                          in the array  near2*/
                 return to ASSIG
            else if ( the 1st elmt. of array pos-j is present in array
                pos-k ) then
                assign cust-j to the first element of  array pos-j
                in the newindiv
              else assign cust-j to the second element of array
                 pos-j  in the  newindiv
        endif
        else if (it is not the most distant neighbor customer)  then
            increase i by 1        /* Pass to the next customer in the
                                                        array near2*/
            return to ASSIG
          else choose the parent with better fitness
              assign cust-j to the facility coming from that parent in
                      the  newindiv
        end else
    endif
endfor
```

<u>Mutation</u>

choose randomly a customer
change the gene related with this customer with the related element of *array nearfac*

APPENDIX D

TEST PROBLEM INSTANCES

In the following, the data sets that we have used in the evaluation of our algorithm are given. The names of the sets start with "m" which stands for "modified"; then the reference Solomon's instance is pointed as r101, rc207; the last two digits show the number of customers included in the instance (i.e. mr101_25, mrc207_100).

The nodes can be classified as customers, potential sites and the supply source in the order of appearance. The columns represent "x coordinate", "y coordinate", "demand", "ready time", "due time" and "service time", respectively.

The example of the data structure is given for a single instance for 3 sizes of problem. For the rest, the followings are valid:

- The customer data is the same as those in Solomon's related instance.
- The potential size locations for the same size problems of different instances are the same. (i.e. for each 25 customer problem, the potential site data is the same as mr101_25)
- The vehicle capacities and the shift times for potential sites are declared under each subtitle.

mr101_25

Vehicle capacity = 200

Shift time = 230

x coord.	y coord.	demand	ready time	due time	service time
41	49	10	161	171	10
35	17	7	50	60	10
55	45	13	116	126	10
55	20	19	149	159	10
15	30	26	34	44	10
25	30	3	99	109	10
20	50	5	81	91	10
10	43	9	95	105	10
55	60	16	97	107	10
30	60	16	124	134	10
20	65	12	67	77	10
50	35	19	63	73	10
30	25	23	159	169	10
15	10	20	32	42	10
30	5	8	61	71	10
10	20	19	75	85	10
5	30	2	157	167	10
20	40	12	87	97	10
15	60	17	76	86	10
45	65	9	126	136	10
45	20	11	62	72	10
45	10	18	97	107	10
55	5	29	68	78	10
65	35	3	153	163	10
65	20	6	172	182	10
30	94		0	230	0
3	27		0	230	0
31	1		0	230	0
85	21		0	230	0

14	76		0	230	0
130	130				

mr101_50

Vehicle capacity = 200

Shift time = 230

x coord.	y coord.	demand	ready time	due time	service time
41	49	10	161	171	10
35	17	7	50	60	10
55	45	13	116	126	10
55	20	19	149	159	10
15	30	26	34	44	10
25	30	3	99	109	10
20	50	5	81	91	10
10	43	9	95	105	10
55	60	16	97	107	10
30	60	16	124	134	10
20	65	12	67	77	10
50	35	19	63	73	10
30	25	23	159	169	10
15	10	20	32	42	10
30	5	8	61	71	10
10	20	19	75	85	10
5	30	2	157	167	10
20	40	12	87	97	10
15	60	17	76	86	10
45	65	9	126	136	10
45	20	11	62	72	10
45	10	18	97	107	10
55	5	29	68	78	10

65	35	3	153	163	10
65	20	6	172	182	10
45	30	17	132	142	10
35	40	16	37	47	10
41	37	16	39	49	10
64	42	9	63	73	10
40	60	21	71	81	10
31	52	27	50	60	10
35	69	23	141	151	10
53	52	11	37	47	10
65	55	14	117	127	10
63	65	8	143	153	10
2	60	5	41	51	10
20	20	8	134	144	10
5	5	16	83	93	10
60	12	31	44	54	10
40	25	9	85	95	10
42	7	5	97	107	10
24	12	5	31	41	10
23	3	7	132	142	10
11	14	18	69	79	10
6	38	16	32	42	10
2	48	1	117	127	10
8	56	27	51	61	10
13	52	36	165	175	10
6	68	30	108	118	10
47	47	13	124	134	10
30	94		0	230	0
3	27		0	230	0
31	1		0	230	0

85	21		0	230	0
14	76		0	230	0
85	41		0	230	0
32	28		0	230	0
54	16		0	230	0
73	41		0	230	0
55	5		0	230	0
130	130				

mr101_100

Vehicle capacity = 200

Shift time = 230

x coord.	y coord.	demand	ready time	due time	service time
41	49	10	161	171	10
35	17	7	50	60	10
55	45	13	116	126	10
55	20	19	149	159	10
15	30	26	34	44	10
25	30	3	99	109	10
20	50	5	81	91	10
10	43	9	95	105	10
55	60	16	97	107	10
30	60	16	124	134	10
20	65	12	67	77	10
50	35	19	63	73	10
30	25	23	159	169	10
15	10	20	32	42	10
30	5	8	61	71	10
10	20	19	75	85	10
5	30	2	157	167	10

20	40	12	87	97	10
15	60	17	76	86	10
45	65	9	126	136	10
45	20	11	62	72	10
45	10	18	97	107	10
55	5	29	68	78	10
65	35	3	153	163	10
65	20	6	172	182	10
45	30	17	132	142	10
35	40	16	37	47	10
41	37	16	39	49	10
64	42	9	63	73	10
40	60	21	71	81	10
31	52	27	50	60	10
35	69	23	141	151	10
53	52	11	37	47	10
65	55	14	117	127	10
63	65	8	143	153	10
2	60	5	41	51	10
20	20	8	134	144	10
5	5	16	83	93	10
60	12	31	44	54	10
40	25	9	85	95	10
42	7	5	97	107	10
24	12	5	31	41	10
23	3	7	132	142	10
11	14	18	69	79	10
6	38	16	32	42	10
2	48	1	117	127	10
8	56	27	51	61	10
13	52	36	165	175	10

6	68	30	108	118	10
47	47	13	124	134	10
49	58	10	88	98	10
27	43	9	52	62	10
37	31	14	95	105	10
57	29	18	140	150	10
63	23	2	136	146	10
53	12	6	130	140	10
32	12	7	101	111	10
36	26	18	200	210	10
21	24	28	18	28	10
17	34	3	162	172	10
12	24	13	76	86	10
24	58	19	58	68	10
27	69	10	34	44	10
15	77	9	73	83	10
62	77	20	51	61	10
49	73	25	127	137	10
67	5	25	83	93	10
56	39	36	142	152	10
37	47	6	50	60	10
37	56	5	182	192	10
57	68	15	77	87	10
47	16	25	35	45	10
44	17	9	78	88	10
46	13	8	149	159	10
49	11	18	69	79	10
49	42	13	73	83	10
53	43	14	179	189	10
61	52	3	96	106	10
57	48	23	92	102	10

56	37	6	182	192	10
55	54	26	94	104	10
15	47	16	55	65	10
14	37	11	44	54	10
11	31	7	101	111	10
16	22	41	91	101	10
4	18	35	94	104	10
28	18	26	93	103	10
26	52	9	74	84	10
26	35	15	176	186	10
31	67	3	95	105	10
15	19	1	160	170	10
22	22	2	18	28	10
18	24	22	188	198	10
26	27	27	100	110	10
25	24	20	39	49	10
22	27	11	135	145	10
25	21	12	133	143	10
19	21	10	58	68	10
20	26	9	83	93	10
18	18	17	185	195	10
30	94		0	230	0
3	27		0	230	0
31	1		0	230	0
85	21		0	230	0
14	76		0	230	0
85	41		0	230	0
32	28		0	230	0
54	16		0	230	0
73	41		0	230	0

55	5	0	230	0
39	86	0	230	0
99	10	0	230	0
8	98	0	230	0
65	43	0	230	0
24	58	0	230	0
130	130			

mr205_25

Vehicle capacity = 1000

Shift time = 1000

mr205_50

Vehicle capacity = 1000

Shift time = 1000

mr205_100

Vehicle capacity = 1000

Shift time = 1000

mc109_25

Vehicle capacity = 200

Shift time = 1236

mc109_50

Vehicle capacity = 200

Shift time = 1236

mc109_100

Vehicle capacity = 200

Shift time = 1236

mc206_25

Vehicle capacity = 700

Shift time = 3390

mc206_50

Vehicle capacity = 700

Shift time = 3390

mc206_100

Vehicle capacity = 700

Shift time = 3390

mrc103_25

Vehicle capacity = 200

Shift time = 240

mrc103_50

Vehicle capacity = 200

Shift time = 240

mrc103_100

Vehicle capacity = 200

Shift time = 240

mrc207_25

Vehicle capacity = 1000

Shift time = 960

mrc207_50

Vehicle capacity = 1000

Shift time = 960

mrc207_100

Vehicle capacity = 1000

Shift time = 960