DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A GENETIC ALGORITHM APPROACH FOR A REAL LIFE HETEROGENEOUS CAPACITATED VEHICLE ROUTING PROBLEM

by

Bircan ÇİÇEKDEŞ

March, 2011

İZMİR

A GENETIC ALGORITHM APPROACH FOR A REAL LIFE HETEROGENEOUS CAPACITATED VEHICLE ROUTING PROBLEM

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of Dokuz Eylül University

In Partial Fulfillment of the Requirements for the Degree of Master of Science

in

Industrial Engineering, Industrial Engineering Program

by

Bircan ÇİÇEKDEŞ

March, 2011

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A GENETIC ALGORITHM APPROACH FOR A REAL LIFE HETEROGENEOUS CAPACITATED VEHICLE ROUTING PROBLEM"** completed by **BİRCAN ÇİÇEKDEŞ** under supervision of **ASSOCIATE PROF. DR ŞEYDA TOPALOĞLU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

…….

Assoc. Prof. Şeyda TOPALOĞLU

Supervisor

…… ……..

Asst. Prof. Serdar TAŞAN Asst. Prof. Adil ALPKOÇAK

(Jury Member) (Jury Member)

Prof.Dr. Mustafa Sabuncu

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

By getting the chance, I would like to express my gratefulness and give many thanks to my supervisor Associate Prof. Dr. Şeyda Topaloğlu for her guidance, encouragement, patience and continuous support throughout this study.

Inspiration, motivation and endless support of my parents deserve inexpressibly great thanks and take a very special place in my life.

Regardless of difficulties, they have always been right beside me leading the way forward and given countenance to me with their trust.

Cemal, my brother, also owns a big contribution in this project with his patience and great support during computer programming stage. I cannot forget to mention the moral support of my younger brother, Malik. I give them my sincere thanks for all of these as well as their cordial brotherhood.

As worthy leaders for all of us, I would like to thank all instructors and teachers for their contributions throughout my education.

Necessary thanks also to all people who has been helpful for my study. With sincere thanks again, I dedicate this thesis to my family.

Bircan ÇİÇEKDEŞ

A GENETIC ALGORITHM APPROACH FOR A REAL LIFE HETEROGENEOUS CAPACITATED VEHICLE ROUTING PROBLEM

ABSTRACT

The purpose of this study is to understand the supply chain and logistics management generally and then practise a real life application with the most important problem of this discipline. After giving a brief information about supply chain and logistics management, application areas and core problems are introduced. As the case study, the daily distribution planning problem of an automotive company is tackled and designing an efficient solution algorithm for the decision maker is aimed. At the beginning, the definition of real problem is made and all the constraints are put forth for consideration. By examining the existing system and distribution planning process, we realized that we encountered with heterogeneous capacitated vehicle routing problem. First, the mathematical model of the problem was formulated as mixed-integer programming. Then, another literature survey was done for selecting an efficient solution algorithm or heuristic which can give optimum results. Next, genetic algorithm was decided to deal with and the research was focused on related papers. A specific genetic algorithm was developed for solving the problem and programmed in MatLab language. The experimental results showed that the proposed algorithm performs well and produces high-quality solutions which also satisfy the performance target of the company by consuming shorter run-time. The proposed genetic algorithm provides decision maker the opportunity of evaluating alternative distribution plans, as well as saving cost and time.

Keywords: Supply Chain and Logistics, Vehicle Routing, Genetic Algorithm

HETEROJEN KAPASİTELİ GERÇEK HAYAT ARAÇ ROTALAMA PROBLEMİ İÇİN GENETİK ALGORİTMA YAKLAŞIMI

ÖZ

Çalışmanın amacı, genel olarak tedarik zinciri ve lojistik yönetimini anlamak ve sonrasında bu disiplinin en önemli problemi üzerinde, bir gerçek hayat uygulaması yapmaktır. Tedarik zinciri ve lojistik yönetimi kısa bilgiler ile tanıtıldıktan sonra, uygulama alanları ve temel problemleri verilmiştir. Vaka çalışması olarak, bir otomotiv firmasının günlük dağıtım problemi ele alınmış karar verici için etkin bir çözüm algoritması tasarlanması amaçlanmıştır. Başlangıçta, gerçek problemin tanımlaması yapılmış ve ele alınacak tüm kısıtlar ortaya konmuştur. Mevcut sistemi ve dağıtım planlama sürecini incelendiğimizde, heterojen kapasiteli araç rotalama problemi ile karşı karşıya olduğumuzu anladık. İlk olarak, problemin matematiksel modeli karma tamsayılı programlama ile formüle edildi. Sonrasında, optimum sonuçlar elde edilmesini sağlayacak etkin bir çözüm algoritması ya da sezgisel metod belirlenmesi için yeni bir literatür taraması yapılmıştır. Ardından, genetik algoritma çalışılmasına karar verilmiş ve araştırmalar konuyla ilgili çalışmalar üzerine yoğunlaştırılmıştır. Nihayetinde, gerçek hayat probleminin çözümü için özgün bir genetik algoritma geliştirilerek, MatLab dilinde programlanmıştır. Deneysel sonuçlar, geliştirilen algoritmanın iyi performans gösterdiğini ve daha kısa sürelerde kaliteli sonuçlar üreterek, söz konusu firmanın performans hedeflerini de karşıladığını göstermiştir. Öngörülen genetik algoritma, karar vericiye alternatif dağıtım planlarını değerlendirme imkanı sunmakta ve ayrıca zaman ve maliyet tasarrufu sağlamaktadır.

Anahtar Kelimeler: Tedarik Zinciri ve Lojistik, Araç Rotalama, Genetik Algoritma

CONTENTS

Page

MSc THESIS EXAMINATION RESULT FORM..... ii

ACKNOWLEDGEMENTS..... iii

ABSTRACT..... iv

ÖZ..... v

CHAPTER ONE – INTRODUCTION TO SUPPLY CHAIN AND LOGISTICS MANAGEMENT..... 1

1.1 Purpose of the Study……... 4

CHAPTER TWO – DISTRIBUTION PLANNING AND MODELS IN LOGISTICS MANAGEMENT..... 5

2.1 Basic Application Areas of Models…….. 5

2.2 Modeling Views of Application Areas……..……..... 6

2.3 Models In Distribution Planning.....…….……..... 8

2.3.1 Facility Location Models...……..... 9

2.3.2 Production/Distribution Models…..……... 10

CHAPTER THREE – VEHICLE ROUTING PROBLEMS.……..... 11

3.1 Definiton and Classification of Vehicle Routing Problem (VRP)…..…... 11

3.1.1 Capacitated VRP (CVRP)..... 12

3.1.2 Multiple Depot VRP (MDVRP)..... 12

3.1.3 VRP with Time Windows (VRPTW)..... 13
3.1.4 Stochastic (Dynamic) VRP (SVRP)..... 13
3.1.5 Periodic VRP (PVRP)..... 13
3.1.6 Split Delivery VRP (SDVRP)..... 13
3.1.7 VRP with Backhauls (VRPB)..... 14
3.1.8 VRP with Pick Up and Delivery (VRPB)..... 14
3.2 VRP Extensions and Application Areas ……..... 14

CHAPTER FOUR – SOLUTION METHODS FOR VRP..... 16

4.1 Mathematical Modeling..... 17
4.2 Heuristic Methods..... 17
4.2.1 Constructive Heuristics..... 17
4.2.2 Two Phase Heuristics..... 18
4.2.3 Local Search Improvement Heuristics..... 18
4.3 Meta-heuristics..... 19
4.3.1 Memory-less Meta-heuristics..... 19
4.3.2 Memory-based Meta-heuristics..... 20
4.4 Combination of Methods..... 21

CHAPTER FIVE – REVIEWED VEHICLE ROUTING PROBLEMS AND MODELS……..... 22

CHAPTER SIX – VEHICLE ROUTING PROBLEM OF THE AUTOMOTIVE COMPANY……..... 34

6.1 Problem Definition……..... 34

6.1.1 Problem Characteristics and Constraints……..... 35
6.1.2 Objectives……..... 35
6.2 Mathematical Model Elements……..... 36
6.2.1 Notations……..... 36
6.2.2 Decision Variables…..... 37
6.3 MIP Formulation of Developed Model……..... 37
6.4 Extended Model With Lifted Subtour Breaking Constraints..... 40
6.5 Verification and Solution of The Model……..... 40

CHAPTER SEVEN – GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEMS……..... 42

7.1 Introduction to Genetic Algorithms……..... 42
7.2 Structure of a Simple Genetic Algorithm.……..... 46
7.3 Operators of Genetic Algorithm.……..... 48
7.3.1 Selection……..... 48
7.3.2 Crossover……..... 49
7.3.3 Mutation……..... 49
7.4 Reviewed Genetic Algorithm Studies……..……..... 50

CHAPTER EIGHT – PROPOSED GENETIC ALGORITHM APPROACH……..... 57

8.1 Structure of Proposed Genetic Algorithm……..... 57
8.1.1 Chromosome Representation……..…..... 57
8.1.2 Initial Population……..……..... 58
8.1.3 Evaluation of Fitness……..... 59

8.1.4 Selection……..... 59
8.1.5 Crossover Procedure……..... 60
8.1.6 Mutation Procedure……..... 61
8.2 Experiment With a Small-sized Problem……..... 66
8.2.1 Initial Population…..……..... 67
8.2.2 Evaluation of Fitness (F)……..... 70
8.2.3 Selection……..... 70
8.2.4 Crossover……..... 70
8.2.5 Mutation……..... 71

CHAPTER NINE – SOLVING THE REAL-LIFE VRP WITH PROPOSED GENETIC ALGORITHM……..... 74

9.1 Programming of Genetic Algorithm in MatLab……..... 74
9.2 Experimental Runs and Computational Results……..... 76
9.3 A Real Problem Instance Solved by Proposed GA……..... 82
9.4 Performance Evaluation of Our Genetic Algorithm……..... 84
9.4.1 Performance Criterions of Reviewed Studies……..... 84
9.4.2 Performance Criterions of Our GA……..... 85
9.5 Features of Our Study and Other Application Areas…..... 91

CHAPTER TEN – CONCLUSIONS ……..... 92

10.1 Performed Study……..... 92
10.2 Future Research……..... 93

REFERENCES ……..... 94

CHAPTER ONE

INTRODUCTION TO SUPPLY CHAIN AND LOGISTICS MANAGEMENT

Supply chain management (SCM) is coordinating all information flow and activities to improve the way of a company that provides all required resources it needs to represent a product or service, then manufactures and delivers that product or service wherever needed by customers or other companies. SCM includes many business processes such as purchasing, financing, material and production planning, controlling, warehousing, inventory control, distribution and delivery.

SCM definition by the Council of Supply Chain Management Professionals (CSCMP) is as follows: Supply Chain Management encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third-party service providers, and customers. In essence, supply chain management integrates supply and demand management within and across companies.

Supply Chain Management is an integrating function with primary responsibility for linking major business functions and business processes within and across companies into a cohesive and high-performing business model. It includes all of the logistics management activities noted above, as well as manufacturing operations, and it drives coordination of processes and activities with and across marketing, sales, product design, finance and information technology (CSCMP Glossary, 2010).

Logistics is the term which encompasses all the operations of presenting the right product or service, at the right place, in the right time, with the right way, in the right quantity and quality at the right price for the right consumer. So, one can state that SCM is a wider and more integrated concept than logistics.

Logistics is an important component of business strategy. It has the potential to improve a company's competitive position through capability in delivery speed, reliability, responsiveness, and low cost distribution, especially for global manufacturing companies (Zhang, 2007).

Logistics Management is that part of supply chain management that plans, implements, and controls the efficient, effective forward and reverse flow and storage of goods, services, and related information between the point of origin and the point of consumption in order to meet customers' requirements. Logistics management activities typically include inbound and outbound transportation management, fleet management, warehousing, materials handling, order fulfillment, logistics network design, inventory management, supply/demand planning, and management of third party logistics services providers. To varying degrees, the logistics function also includes sourcing and procurement, production planning and scheduling, packaging and assembly, and customer service. It is involved in all levels of planning and execution-strategic, operational, and tactical. Logistics management is an integrating function which coordinates and optimizes all logistics activities, as well as integrates logistics activities with other functions, including marketing, sales, manufacturing, finance, and information technology (CSCMP Glossary, 2010).

As Christopher (2005) states in his book, it must be recognized that the concept of supply chain management, whilst relatively new, is in fact no more than an extension of the logic of logistics. Logistics management is primarily concerned with optimizing flows within the organization, whilst supply chain management recognizes that internal integration by itself is not sufficient. Therefore, SCM is also referred as Value Chain Management with combining every entity and process from the origin point of demand to the end point of customer level. The scope of logistics spans the organization, from the management of raw materials through to the delivery of the final product.

In the middle of these days' highly competitive marketplace, supply chain and logistics management also attracts more attention and becomes a wider area of interest. Because to maintain its existence in the market, a firm which either manufactures and distributes its products to customers or keeps the goods in depots fed by a central warehouse and delivers them to demand points, must construct the best network and transportation routes to accomplish those activities effectively and reduce the costs.

Sometimes the companies, which do not have enough resources to set up warehouses for storage, inventory control and distribution of their products, collaborate with logistics service providers to get customer demands fulfilled on time; in other words, they choose outsourcing of the logistics processes for agility. Because the business turns from supplier-centric to customer-centric structure and the goal of cost minimization is now being combined with the attainment of higher levels of customer responsiveness.

Globalization of markets and developments in business technologies have rapidly changed the nature and structure of logistics process. So, the firms which have the best distribution system survive in the market. To implement the best system, supply chains and distribution networks have been modeled using deterministic or stochastic methods by researchers and various optimization methods, heuristic algorithms or simulation are used when good or exact solutions cannot be obtained due to dynamic behaviors of real world.

1.1 Purpose of The Study

The main objectives of this study are examining the supply chain and logistics processes and solving the daily distribution planning problem of an automotive company located in Izmir. To cope with this problem, existing system and company's processes are examined thoroughly before determination of real problem.

After developing a mathematical model for finding the best routes with miminum total cost, a meta-heuristic algorithm is used to gain high-quality solutions in a reasonable amount of time since the company has to come up with this distribution planning problem everyday.

CHAPTER TWO

DISTRIBUTION PLANNING AND MODELS IN LOGISTICS MANAGEMENT

2.1 Basic Application Areas of Models

Logistics strategies includes the business goals, requirements, allowable decisions, tactics, and vision for designing and operating a logistics system. Although some logistics strategies impact decisions throughout the supply chain, for clarity, Ratliff & Nulty (1996) classifies the logistics applications for strategic areas in five categories as illustrated in Figure 2.1 below and describes these as follows:

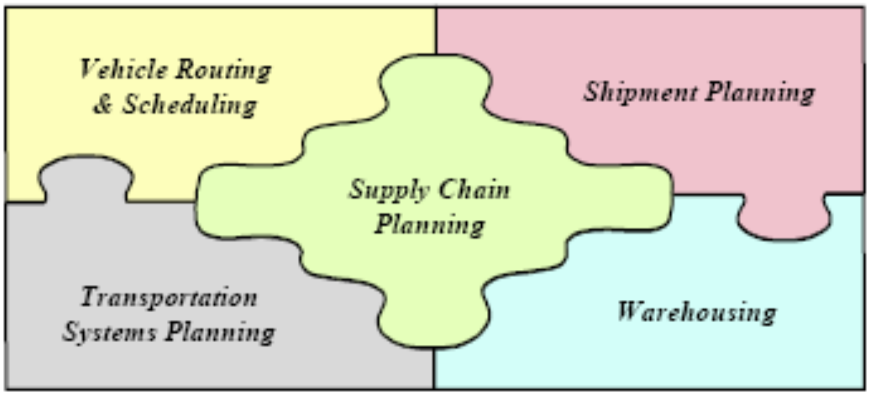


Figure 2.1 Applications in logistics modeling

Supply Chain Planning includes the location, sizing, and configuration of plants and distribution centers, the configuration of shipping lanes and sourcing assignments, the aggregate allocation of production resources, and customer profitability and service issues. *Shipment Planning* is the routing and scheduling of shipments through the supply chain, including freight consolidation and transportation mode selection. *Transportation Systems Planning* includes the location, sizing, and configuration of the transportation infrastructure, including fleet sizing and network alignment.

Warehousing includes the layout design, inventory management and storage/picking operations of distribution centers. *Vehicle Routing and Scheduling* includes the routing and scheduling of drivers, vehicles, trailers, etc. Other applications include dynamic dispatching, customer zone alignment, and frequency of delivery questions.

In this study, we focus on vehicle routing problems (VRP) for modelling a real life distribution planning problem and propose a heuristic solution approach based on genetic algorithm.

2.2 Modelling Views of Application Areas

According to Ratliff & Nulty (1996), analyzing the various logistics strategies requires appropriate modeling views of a logistics supply chain. Because every case in real world needs different executions, different approximates and so, different solutions for the success of those projects. Strategic, tactical, and operational models are three fundamental classes of modeling views. So, the general scope and properties of strategic, tactical and operational model views can be layered as in Figure 2.2.

Klose & Drexl (2004) evaluates the decisions about the distribution system as a strategic issue for almost every company. The problem of locating facilities and allocating customers covers the core components of distribution system design. Industrial firms must locate fabrication and assembly plants as well as warehouses. Stores have to be located by retail outlets. The ability to manufacture and market its products is dependent in part on the location of the facilities. Therefore, we come accross a complex structure comprised of production plants, central depot and/or warehouses, retailers and/or customers on which the operations or services are performed by a fleet of several vehicles.

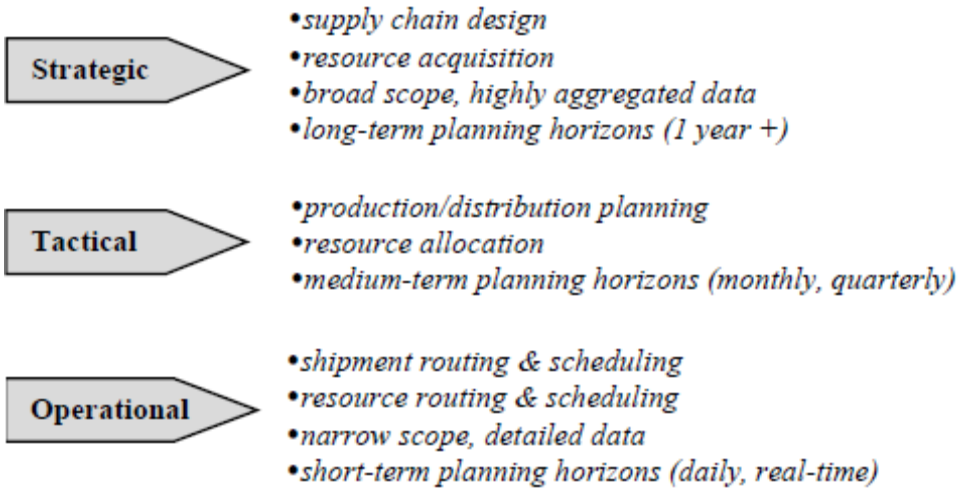


Figure 2.2 General scope and properties of strategic, tactical and operational model views

Kim & Kim (1999) state that effective and efficient logistics management is becoming increasingly important in both private and public organizations. It is a complex activity involving several decision levels, numerous participants with various objectives and a large amount of human and material resources. Indeed, as of the year of 2000, SCM and logistics management had become the core business processes for every manufacturer company.

According to Teng (2005), the collaborative policies in a supply chain may raise higher accuracy in forecasting activities and decrease the risk of over- or under-production/distribution. The end results would be better customer satisfaction, lower inventory cost and a smooth flow of products in the manufacturing supply chain with competitive cost and quality.

The problems which must be tackled by the companies which want to control the transportation, distribution and to make the best possible use of the system's resources, can be generally classified as strategic, tactical and operational problems. Kim & Kim (1999) identify these problem levels as follows:

Strategic problems involve the highest level of management, and its results require large investments and form general strategies of the system. Decisions at strategic level are concerned with physical network design, facility location and resource acquisition. Tactical problems are necessary to ensure appropriate allocation of existing resources to improve performance of the whole system. At this level, decisions need to be made concerning fleet sizes and fleet mix, traffic distribution and empty vehicle movements. Operational problems, in which routes that vehicles will take are determined and vehicles are scheduled and dispatched, are solved in a highly dynamic environment. Here, scheduling means planning the time when each delivery event in a route should take place, while dispatching encompasses both routing (determination of routes) and scheduling and often includes reacting to unforeseen changes in a plan.

2.3 Models In Distribution Planning

Klose & Drexl (2004) reminds that the problem of designing distribution network and locating facilities is not new to the operations research community; the challenge of where to best site facilities has inspired a rich, colorful and ever growing body of literature. To cope with the multitude of applications encountered in the business world and in the public sector, an ever expanding family of models has emerged.

Most of the studies in literature mention that distribution network design problems involve strategic decisions which influence tactical and operational decisions. In particular, they involve facility location, transportation and inventory decisions, which affect the cost of the distribution system and the quality of the customer service level. So, they are core problems for each company.

As stated above, distribution network design problems involve a lot of integrated decisions, which are difficult to consider all together. Generally, some simplifying assumptions are adopted in the literature and only some aspects related to the complex network decisions are being modeled.

Distribution network design problems involve analysis of both the optimization of the goods flow and the improvement of the existing network. More precisely, these problems consist of determining the best way to transfer goods from the supply to the demand points by choosing the structure of the network (layers, different kinds of facilities operating at different layers, their number and their location), while minimizing the overall costs.

2.3.1 Facilitiy Location Models

These models involve location of facilities and construction of distribution routes. In this kind of problems, the researcher/decision maker can start form locating the facilities and/or depots and then construct the best routes for dispatching of goods; or construct a model which does both of these tasks simultaneously to find the best solutions.

Klose and Drexl(2004) gives that the Location-Allocation Models cover formulations which range in complexity from simple linear, single-stage, single-product, uncapacitated, deterministic models to non-linear probabilistic models. Algorithms include, among others, local search and mathematical programming based approaches. Some researchers modelled many location problems as mixed integer programming models (MIP) by starting with a given set of potential facility sites. Apparently, network location models differ only gradually from MIP models, because the former ones can be stated as discrete optimization models. Yet network location models explicitly take the structure of the set of potential facilities and the distance metric into account while MIP models just use input parameters without asking where they come from.

Facility location models have various applications: locating the warehouse(s) with minimum average distance to market within a supply chain, locating dangerous or toxic materials at maximum distance to public, locating automatic teller machines of banks at best points to provide a better customer service, etc.

2.3.2 Production/Distribution Models

Teng (2005) mentioned about widely usage of MIP models to formulate the production/distribution problems for three decision levels in a decision-making process for SCM which include strategic, tactical, and operational levels. At the lowest decision level, operational decisions are mostly for handling daily activities**.**

Calvete, Gale, Oliveros, & Valverde (2005) explains that the problem of physical distribution of goods to customer locations is specially relevant since it accounts for a large proportion of the overall operational costs of a producer. Hence, effective and efficient management of transportation and distribution of goods is becoming increasingly important both from the point of view of theoretical research and from the point of view of practical applications.

These problems are generically known as **Vehicle Routing Problem** (VRP) and all its extensions have been worked widely by many researchers over the past two decades. As also given in study of Calvete & others (2005), the classical VRP consists of determining the best set of routes for a fleet of vehicles originating and terminating at a single central depot, to distribute goods to a set of customers geographically dispersed, while minimizing the total travel distance/time or the total distribution cost.

Vehicle routing problems are examined in next chapter of the study.

CHAPTER THREE

VEHICLE ROUTING PROBLEMS

3.1 Definition and Classification of Vehicle Routing Problem

According to description by Toth & Vigo (2002), the vehicle routing problem (VRP) embraces a whole class of complex problems, in which a set of minimum total cost routes must be determined for a number of resources (i.e., a fleet of vehicles) located at one or several points (e.g., warehouses), in order to service efficiently a number of demand and/or supply points.

These problems (VRPs) are central to logistics management both in the private and public sectors. They consist of determining optimal vehicle routes through a set of users, subject to side constraints. The most common operational constraints impose that the total demand carried by a vehicle at any time does not exceed a given capacity, the total duration of any route is not greater than a prescribed bound, and service time windows set by customers are respected. In long-haul routing, vehicles are typically assigned one task at a time while in short-haul routing, tasks are of short duration (much shorter than a work shift) and a tour is to be built through a sequence of tasks (Ghiani, Guerriero, Laporte, & Musmanno, 2003).

Baker & Ayechew (2003) gives the VRP description as follows: The basic 'Vehicle Routing Problem (VRP)' consists of a number of customers, each requiring a specified weight of goods to be delivered. Vehicles despatched from a single depot must deliver the goods required, then return to the depot. Each vehicle can carry a limited weight and may also be restricted in the total distance it can travel. Only one vehicle is allowed to visit each customer. The problem is to find a set of delivery routes satisfying these requirements and giving minimal total cost.

Based on modelling structure, the objective can be minimization of the total distance travelled or minimization of the number of vehicles used and the total distance for this number of vehicles.

From the view of problem features; VRP have several different variants in accordance with the application area and the problem case. We can specify these variants as broadly classified in literature:

3.1.1 Capacitated VRP (CVRP)

This variant of VRP accomodates the vehicles with limited capacity and has extensions according to vehicle types in the fleet. Homogeneous VRP, as the classical CVRP, represents the vehicles with equal capacities whilst Heterogeneous capacitated VRP (HCVRP) has a fleet of vehicles with varying capacities, fixed costs (rental cost etc.) and variable costs (e.g. travelling, distance). Heterogeneous Fixed Fleet VRP has a fleet with fixed number of vehicles of more than one type and hence with different capacities.

On the other side, if HCVRP includes also the decision of fleet composition, it is called Mixed Fleet HCVRP. Each vehicle type is characterised by its capacity, fixed cost and variable travel cost. Each route starts and finishes at the central depot and the objective is usually minimization of the total cost for serving all customers or nodes. Hence, the total cost is consisted of the sum of all fixed costs and the variable cost of the distance travelled by each vehicle.

In all CVRP types, customer demands are deterministic and known, deliveries cannot be split on different vehicles and the sum of the demands on each route does not exceed the capacity of the vehicle assigned to that route. Therefore, each customer can be served by only one vehicle and the routes must be constructed in respect to capacity constraints.

3.1.2 Multiple Depot VRP (MDVRP)

Multiple Depot VRP (MDVRP) is used when the customers get their deliveries from several depots. Thus, there are several depots from which each customer can be served. Similarly, the vehicles can be originated in any of depots but each vehicle must leave from and return to the same depot.

3.1.3 VRP with Time Windows (VRPTW)

VRP with Time Windows (VRPTW) is dealt when there is a time window (start time, end time, service time) associated to each customer. So, there is a specified temporal window of opportunity in which vehicles can visit each demand node and leave the node. Each customer has a service time that the vehicle must wait to unload the goods. This variation can naturally be combined with the CVRP.

3.1.4 Stochastic (Dynamic) VRP (SVRP)

Stochastic VRP (SVRP) is used if any of service time, travel time, set of customers to be served, customer demand or combination of these VRP components has a random behaviour. These are usually assumed to follow a given probability distribution or the unknown input has to become known/updated during the run and solution is constructed accordingly.

3.1.5 Periodic VRP (PVRP)

If the delivery is made in some days, then the case is Periodic VRP (PVRP). In PVRP, the classical VRP is generalized by extending the planning period to T days where not all customers require delivery on every day in this period. The PVRP is consisted of two classical problems: the assignment problem and VRP.

Delivery days have to be assigned to each customer and vehicle routes have to be designed for each day of the period, so the total distribution cost is minimized.

3.1.6 Split Delivery VRP (SDVRP)

Split Delivery VRP (SDVRP) is a variant of VRP where the customer demands may be greater than the vehicle capacity. SDVRP allows deliveries to be split, so a customer may be served by more than one vehicle. SDVRP has a potential to use fewer vehicles thereby reducing the total distance traveled by the fleet and reduce the total cost.

3.1.7 VRP With Backhauls (VRPB)

VRP with Backhauls (VRPB) involves both delivering goods to customers (linehaul) and collecting defective goods, materials for recycling or return materials from customers or vendors (backhaul) on the route. Here, the critical assumption is that each vehicle must collect something from the customers after all deliveries are done. In VRPB models, the delivery quantities and quantities to be collected are known in advance.

3.1.8 VRP With Pick-Up and Delivery (VRPPD)

In VRP with Pick-Ups and Deliveries (VRPPD), the vehicles pick up goods from some locations and deliver them to the customers. So, the goods are not stored in the depots, but they are distributed over the nodes of the distribution network. A transportation decision specifies the size of the load to be transported, the location from where it will be picked up with a pick up time window and the location where the picked up goods to be delivered with a delivery time window. Therefore, these problems always include time windows for pick-up and/or delivery.

3.2 VRP Extensions and Application Areas

Although the VRP has been extensively used in the area of transportation science since 1960s, many managers have started employing VRP models during the last years for effective decision-making.

According to Tarantilis, Ioannou & Prastacos (2005), the following are some examples of the multitude of VRP applications in manufacturing and service operations management:

- Routing of automated guided vehicles, which are considered as one of the most appropriate modes for material handling in contemporary flexibly automated production environments
- Minimization of the distribution costs in a multifacility production system

- Determination of vehicle routes for material delivery within the premises of a plant operating under a Just-In-Time philosophy.

- Sequencing of the operations in single or multi-feeder printed circuit board manufacturing

- Scheduling wafer probing

- Rolling batch planning

CHAPTER FOUR

SOLUTION METHODS FOR VRP

The VRP is, mathematically, a combinatorial optimization problem. In 1980s it was proved that this problem type is non-deterministic polynomial-time (NP)-hard. Namely, the polynomial equation model of the VRP cannot be directly established to determine its optimal solution, and solving time for the VRP grows exponentially with the increase in distribution points. The mathematical programming technique can only be adopted to solve VRPs with small numbers of distribution points. With a large number of distribution points, even the fastest computer is incapable of performing the exhaustive computations required to determine an optimal solution (Wang & Lu, 2009).

Therefore, many researchers had developed metaheuristic algorithms, such as genetic algorithms (GAs), ant colony systems and particle swarm optimization in order to obtain a close-to-optimal solution for the VRP. Some successful metaheuristic algorithms have recently been developed and with many studies, GAs have been proved to be capable of solving VRPs.

Baker & Ayechew (2003) remind that VRPs of realistic size are tackled using heuristics. The tabu search implementations have obtained the best known results to benchmark VRPs. Various authors have reported similar results, obtained using tabu search or simulated annealing. However, it has been observed that such heuristics require substantial computing times and several parameter settings. Ant colony optimisation is another recent approach to difficult combinatorial problems with a number of successful applications reported, including the VRP. Genetic algorithms (GAs) have seen widespread use amongst modern metaheuristics, and several applications to VRPs incorporating time windows have been reported. Hybrid approach to vehicle routing using neural networks and GAs has also been reported.

More generally, VRP solution approaches can be classified as follows:

4.1 Mathematical Modeling

Mathematical modeling approaches provide exact solutions. Nevertheless, VRP is NP-Hard and computationally too complex to obtain optimum solutions. Therefore, these methods usually consume very long solution time. Some of these exact solution approaches are branch and bound algorithm, dynamic programming and Lagrangean relaxation procedure.

4.2 Heuristic Methods

Heuristic procedures have been widely used for solving combinatorial optimization problems, so for the VRP. Heuristic applications usually limit the exploration of the search space, but aim obtaining a good solution in a reasonable amount of time. There are three basic heuristics categories used for solving the VRP:

4.2.1 Constructive Heuristics

Constructive heuristics begin with infeasible route assignments by using problem data and the initial solution is constructed step by step. So, the solution is obtained at the end of the procedure. *Nearest Neighbour Search* starts with any node at the beginning, then finds the closest to the last-added node at each step.

Finally, the procedure completes the route with all the nodes included but requires high computational time. *Savings Procedure* builds an initial solution that can be infeasible, by calculating savings generated by a new route configuration. On the other side, this procedure may produce sub-optimal routes.

4.2.2 Two-phase Heuristics

Two-phase heuristics first cluster nodes into feasible routes and then construct actual routes, using feedback loops between these two stages. *Cluster-first, Route-second Procedure* starts with clustering the nodes and determine feasible routes for each cluster, as it is the principle of this heuristic. The application of this procedure is difficult if the vehicles have different capacities. Sweep algorithm and Fisher and Jaikumar Algorithm are commonly used heuristics of this method. In *Route-first, Cluster-second Procedure* solution process starts with a large route, usually infeasible, and partitiones this initial route into smaller clusters. Therefore, it is not suitable for small problems.

4.2.3 Local Search Improvement Heuristics

This kind of heuristics are iterative search procedures which start from an initial feasible solution and improve the solution progressively by applying a series of local modifications. The way in which insertions are performed is one of the important feature of neighbourhood development: one could use random insertion, could continue with inserting the node at the best position in the target route or could use more complex insertion method which may include a partial re-optimization of the target route.

Insertion Procedure constructs a solution by determining the cheapest insertion of a node into the route, however it may result with sub-routes. The examples are nearest insertion, cheapest insertion, farthest insertion, quick insertion and the convex hull insertion algorithms.

Improvement Procedure starts with an initial route. It examines all the routes that are neighbouring to it and aims finding a short route. When there is no neighbour route which is shorter than the original one, the process terminates. This procedure usually requires long computation time.

4.3 Meta-heuristics

Meta-heuristics are known to be the most promising and effective solution methods for the VRP and solving hard optimization problems. They perform a deep exploration in the most promising regions of the solution space unlike the other heuristics which perform a limited search. Usually, some sophisticated neighbourhood search rules, memory structures and recombination of solutions are combined with meta-heuristics to strengthen the algorithm.

Meta-heuristics apply the principles of intensification and diversification during the search. Intensification provides more detailed exploration of the most promising areas of the solution space while diversification avoids getting caught by a local optimum solution. Hence, the quality of solutions produced by meta-heuristics is usually much higher than those obtained by classical heuristics. But, they usually require finely tuned parameters for effective search.

Meta-heuristics have two types as memory-less and memory-based methods based on the use of previously exploited areas of the solution space:

4.3.1 Memory-less Meta-heuristics

The most important memory-less meta-heuristic methods are Simulated Annealing (SA) and Threshold Accepting (TA). SA is inspired from the physical annealing process emanating in statistical mechanics. It locates a good approximation to the global optimum of a given function in a large search space by local search but accepts, under control of the objective function, the non-improving solutions as well. This feature provides SA to escape from a low quality local optimum. TA is a variant of the SA. Specifically, it overcomes getting stuck at a local optima by accepting worse solutions, which lead to deterioration of objective function, but not worse than previous threshold.

4.3.2 Memory-based Meta-heuristics

Memory-based meta-heuristics continue exploration by exploiting the previously examined area of the solution space via list of solutions kept within a limited memory. Main types of these methods are the Tabu Search (TS) algorithms, the Adaptive Memory Based Algorithms (AMBA), Genetic Algorithm (GA) and Ant Colony Optimization (ACO).

TS is a local search meta-heuristic which explores the solution space, by moving at each iteration, from a current solution to the best solution in its neighbourhood. In this method, the current solution may deteriorate when moved from one iteration to another. Thus, to avoid visiting the same solutions, recently explored solutions are temporarily memorised and stored in a list. The stored attributes are declared tabu or forbidden and the corresponding list stored is the tabu list. The tabu status of an attribute stored in the tabu list is overridden when, e.g. a tabu solution is better than any previously examined solution. This special condition is known as aspiration criterion. Common enhancements to the basic TS methodology include intensification and diversification (Ganesh, Nallathambi, Narendran, 2007).

The AMBA constitute one of the most powerful tools for the diversification and intensification of the search process. According to this concept, an adaptive memory keeps track of the best components of the solutions visited during the search. After that, a provisional solution is created by combining the best components and improved by a local search method. Finally, the improved solution is included or updated in the adaptive memory by considering components of the improved solution as candidates for replacing old components stored in the adaptive memory.

GAs are population-based algorithms that simulate the evolutionary process of species that reproduce. A GA causes the evolution of a population of individuals encoded as chromosomes by creating new generations of offspring through an iterative process that continues until some convergence criteria are met.

At the end of this process, it is expected that an initial population of randomly generated chromosomes will improve and be replaced by better off-springs. The best chromosome obtained by this process is then decoded to obtain the solution (Ganesh, Nallathambi, Narendran, 2007).

ACO is designed to simulate the ability of ants to determine shortest paths between food sources and their nest. During the search process, ants mark the paths they travel by laying down pheromone trails. The pheromone guides other ants to the source of food. After a while, the shortest path to the source will have the largest amount of pheromone as the trail grows by the ants which follow this path. The exploration area of the ants corresponds to the set of feasible solutions and the intensity of pheromone represents the objective function in the ACO framework.

4.4 Combination of the Methods

To cope with combinatorial optimization problems, the researcers have also developed many hybrid approaches by combining two or more of above described methods. Some of these combinations are reported to perform high quality solutions and sometimes best-known solutions at low computational time.

CHAPTER FIVE

REVIEWED VEHICLE ROUTING PROBLEMS AND MODELS

In this section, we give a short brief of reviewed studies and bring the papers together by constructing a summarized table.

Calvete & others (2005), investigate the use of goal programming to model the vehicle routing problem in their work. They consider a general medium-sized delivery problem with soft and hard time windows, a heterogeneous fleet of vehicles and multiple objectives. Its purpose is to investigate the use of goal programming to model the problem and propose an approach to solve it providing an optimal solution in a reasonable computational time in terms of real applications. Mainly, the problem studied involves designing a set of routes for a fleet of vehicles based at one central depot that is required to serve a number of geographically dispersed customers, while minimizing the total travel distance or the total distribution cost. Each route originates and terminates at the central depot and customers demands are known.

In many practical distribution problems, besides a hard time window associated with each customer, defining a time interval in which the customer should be served, managers establish multiple objectives to be considered, like avoiding underutilization of labor and vehicle capacity, while meeting the preferences of customers regarding the time of the day in which they would like to be served (soft time windows). The inclusion of multiple objectives and specific constraints in the general setting of VRP makes the model more complicated and highlights the difficulties regarding the time needed to solve it optimally with standard optimization software. To solve the model, an enumeration-followed-by-optimization approach is proposed in this study which first computes feasible routes and then selects the set of best ones. According to the computational results that they obtained, their approach is adequate for medium-sized delivery problems. The efficiency of the procedure lies on the number of feasible routes which have to be dealt with.

Lau, Sim & Teo (2003) introduce a variant of the vehicle routing problem with time windows where a limited number of vehicles is given (m-VRPTW). According to their scenario, a feasible solution is one that may contain either unserved customers and/or relaxed time windows. Authors provide a computable upper bound to the problem. In order to solve the problem, a tabu search approach is proposed which is characterized by a holding list and a mechanism to force dense packing within a route. They also allow time windows to be relaxed by introducing the notion of penalty for lateness. In their approach, customer jobs are inserted based on a hierarchical objective function that captures multiple objectives. In conclusion, obtained computational results on benchmark problems show that their approach yields good solutions that are competitive to best published results on VRPTW. On m-VRPTW experiments, the approach produces solutions very close to computed upper bounds. Moreover, it is seen that as the number of vehicles decreases, the routes become more densely packed monotically. This prove that the proposed approach is good from both the optimality as well as stability point of view.

Jozefowiez, Semet & Talbi (2008) present an overview of what multi-objective optimization can bring to vehicle routing problems, as well as the possible motivations for applying multi-objective optimization on vehicle routing problems and the potential uses and benefits of doing so. This is illustrated by two problems representing the two main aspects of multi-objective vehicle routing problems and a general optimization strategy. According to authors, academic vehicle routing problems need adaptations for real-life applications. These adaptations are mostly additions of new constraints and/or parameters to a basic problem. Another way to improve the practical aspects of vehicle routing problems is to use several objectives. Therefore, they describe two extended problems as: the vehicle routing problem with route balancing and the bi-objective covering tour problem. A two-phased approach based on the combination of a multi-objective evolutionary algorithm is also proposed and single-objective techniques that respectively provide diversification and intensification for the search in the objective space are described.

Finally, examples of implementation of their method are provided on the two problems. The experimental results are compared with other methods and best-known solutions. The results show that their method gives very good solutions.

Kek, Cheu & Meng (2008) propose two new distance-constrained capacitated vehicle routing problems (DCVRPs) to investigate and study potential benefits in flexibly assigning start and end depots. The first problem, DCVRP-Fix is given as an extension of the traditional symmetric DCVRP, with additional service and travel time constraints, minimization of the number of vehicles and flexible application to both symmetric and asymmetric problems. The second problem, DCVRP-Flex is described as a relaxation of DCVRP-Fix to enable the flexible assignment of start and end depots. According to their study, this give vehicles the freedom to start and end their tour at different depots, while allowing for intermediate visits to any depot (for reloading) during the tour. Network models, integer programming formulations and solution algorithms for both problems are developed and presented in their paper. Authors introduce a new and original idea of relaxing the vehicle assignment constraint with potential time and cost savings. With a keen focus on global express and logistics operations (where capacity and range of vehicles are the key defining factors of operational efficiency), that relaxation is applied on the DCVRP. An analytical comparison of both problems is carried out as a case study, considering the impact of depot locations and problem symmetry. They gained results which show a generation of cost savings up to 49.1% by DCVRP-Flex across the evaluated cases. It is explained that a significant portion of this is sourced by the flexibility to reload at any depot while the rest of it is derived from the flexibility to return to any depot. DCVRP-Flex's adaptability and superior performance over DCVRP-Fix is also found out by their study.

Francis & Smilowitz (2006) introduce a continuous approximation model for the period vehicle routing problem with service choice (PVRP-SC) which is a variant of the period vehicle routing problem (PVRP). In PVRP routes are designed for a fixed fleet of capacitated vehicles each day of a t-day period to visit customers exactly a pre-set number of times.

The model is used in the strategic analysis of the benefits of service choice and the sensitivity of these benefits to various parameters. Results obtained with the model also answer tactical questions relating to the service mix of customers and vehicle fleet planning. Their research provides practitioners with a tool to analyze efficiencies in distribution operations arising from service choice, without requiring extensive computations and detailed data collection typical of discrete models for periodic vehicle routing problems. They present the discrete formulation of the PVRP-SC and the continuous formulation. It is shown that the continuous approximation model can be solved easily with a few modifications. The modified continuous approximation model can yield solutions for large instances which may arise in the strategic planning phases of periodic distribution systems. Thus, the continuous model is not suggested as a replacement for the discrete modeling approach by the authors; however it can be used to estimate costs and develop design guidelines. They focus on the use of the continuous approximation method for strategic decision making, estimation and the selection of parameters. So, the operational/tactical decisions can be made using the discrete method after parameters have been chosen. Geographic decomposition and variable substitution is used to reduce formulated model into a simple problem that can be solved easily. Then, the continuous approximation solution method is implemented and obtained results are very close to the best known solutions.

Santos, Duhamel & Aloise (2008) describe a multi-objective optimization problem for mobile-oil recovery of wells in a petrol field. The goal is finding a set of wells to be pumped in a working day to maximize the oil extraction and to minimize the travel time. The two objectives are opposite, one pushing to increase profit and the other to reduce costs. The researchers give several formulations for the mobile-oil recovery optimization problem (MORP) using a single vehicle or a fleet of vehicles. The formulations are also strengthened by improving the subtour elimination constraints (SECs). They proved the optimality for instances close to reality with up to 200 nodes. Comparison among the proposed formulations with different SECs are measured in terms of time to prove optimality and of linear relaxation.

Their computational experiments show that the time window restriction plays a key role in computing an optimal solution: the smaller the time window, the easier the problem to solve.

Baldacci, Battara & Vigo (2008) give an overview of approaches from the literature to solve heterogeneous vehicle routing problems (VRP). In particular, they classify the different variants described in the literature. Main integer programming formulations are given, first. As no exact algorithm has been presented for any variants of heterogeneous VRP, the lower bounds and the heuristic algorithms proposed are reviewed by the authors. Computational results, comparing the performance of the different heuristic algorithms on benchmark instances, are also discussed in this paper.

Tzur, Francis & Smilowitz (2008) work on a generalized type of vehicle routing problem (VRP). The period vehicle routing problem (PVRP) is a variation of the classic VRP in which delivery routes are constructed for a period of time (for example, multiple days). In this paper, they consider a variation of the PVRP in which service frequency is a decision of the model. Authors refer to this problem as the PVRP with service choice (PVRP-SC). They explore the modeling issues which arise when service choice is introduced to the model and suggest efficient solution methods. Contributions are made both in modeling this new variation of the PVRP and in introducing an exact solution method for the PVRP-SC. In addition, a heuristic variation of the exact method to be used for larger problem instances is proposed. The computational tests show that adding service choice can improve system efficiency and customer service.

Thammapimookkul & Charnsethikul (2005) are concerned with the Vehicle Routing Problem (VRP) arisen for an ATM scheduling, a real-world routing-scheduling problem. In this research, they reformulate the problem with simultaneous consideration of two objectives; minimizing the total traveling time and minimizing traveling time of the longest tour. Many heuristic and optimization approach have been also reviewed.

They propose the “Re-routing ATMs” methodology in order to maximize efficiency measured by two objectives; minimizing the total traveling time and minimizing traveling time of the longest tour simultaneously. A new heuristic is developed and proposed. The heuristic is an appropriate extension and modification of Clark-Wright procedure. The heuristic performance was found to be efficient through a number of tests. Results obtained for the case study illustrate some improvement in both total cost and workload balancing for each tour.

Georgapoulos & Mihiotis (2004) enriched ABC analysis for customers and for products with elements besides annual turnover, which will make ABC classification more representative of the priority level each customer and each product should get. To do so, an evaluation index is designed for every customer and for every product. The problem of planning the routes for the distribution is expressed generally like this: a set of customers being in known positions, with known orders of our products, must be served by the company’s store room using the means of distribution that the company owns. So, the aim is to plan the routes that our lorries will follow in order to minimise the necessary time. They applied a plug-in of MS Excel which uses branch-and-bound method to solve the problem. The target of the paper is constructing the evaluation index to represent the importance that every customer and every product holds for the company. Besides the turnover, they used elements such as the image, the policy, the profit’s rate, the kind of the store and the necessity for special deliveries.

Fisher, Jörnsten & Madsen (1997) deal with the vehicle routing problem with time window constraints (VRPTW). They describe two optimization methods for vehicle routing problems with time windows. These are a K-Tree relaxation with time windows added as side constraints and a Lagrangian decomposition in which variable splitting is used to divide the problem into two subproblems; a semi-assignment problem and a series of shortest path problems with time windows and capacity constraints. Optimal solutions to problems with up to 100 customers are presented in this paper.

Kunnathur, Nandkeolyar & Li (2005) address the problem of partitioning and transporting a shipment of known size through an n-node public transportation network with known scheduled departure and arrival times and expected available capacities for each departure. The objective is to minimize the makespan of shipping. The problem while practical in its scope, has received very little attention in the literature perhaps because of the concentration of research in vehicle routing without regard to partitioning and partitioning without regard to routing. A general non-linear programming model is developed for the partitioning and transportation problem. The model is then converted into a linear model through the Routing First and Assignment Second approach. This approach is different from the general decomposition approaches since they normally do not guarantee optimality. However, the linear model still involves a large number of constraints, and solution is not attempted here. Instead, three heuristics are proposed for solving the problem in this study. Two of the heuristics use iterative techniques to evaluate all possible paths. The third heuristic uses a max-flows approach based upon aggregated capacities to reduce the size of the network presented to the other heuristics. This allows for a good starting point for other heuristics, and may impact the total computational effort. According to writers, the heuristics developed in their paper perform well because in the case of networks that are not congested, they find the optimal solution.

Kim & Kim (1999) consider a multi-period vehicle scheduling problem (MPVSP) in a transportation system where a fleet of homogeneous vehicles delivers products of a single type from a central depot to multiple (N) retailers. In this study, the problem is formulated as a mixed integer linear program. The objective of the MPVSP is to minimize transportation costs for product delivery and inventory holding costs at retailers over the planning horizon. To solve a MPVSP of large size in a reasonable computation time, a two-phase heuristic algorithm is suggested based on a kth shortest path algorithm. In the first phase of the algorithm, the MPVSP is decomposed into N single-retailer problems by ignoring the number of vehicles available.

The single-retailer problem is formulated as the shortest path problem and several good delivery schedules are generated for each retailer using the k-th shortest path algorithm assuming the exact requirement policy is used in the system. In the exact requirement policy, replenishments occur only when the inventory level is zero. In the second phase, a set of vehicle schedules is selected from those generated in the first phase. The vehicle schedule selection problem is a generalized assignment problem and it is solved by a heuristic based on the k-th shortest path algorithm. Computational experiments on randomly generated test problems shows that the suggested algorithm gave near optimal solutions in a reasonable amount of computation time.

Kara & Bektaş (2006) extend the classical multiple traveling salesman problem (mTSP) by imposing a minimal number of nodes that a traveler must visit as a side condition and formulate an integer programming model. They consider single and multi-depot cases and propose integer linear programming formulations for both, with new bounding and subtour elimination constraints. It is shown that the formulation with additional restrictions may easily be adapted to special cases and multidepot situations. In their study, the authors present that several variations of the multiple salesman problem can be modeled in a similar manner. Computational analysis shows that the solution of the multi-depot mTSP with the proposed formulation is significantly superior to previous approaches.

Chen, Hsueh & Chang (2006) describe a different variaion of vehicle routing problem. In their paper, the real-time time-dependent vehicle routing problem with time windows (RT-TDVRPTW) is studied and formulated as a series of mixed integer programming model, which accounts for real-time and time-dependent travel times, and real-time demands. They note that both real-time and time-dependent travel times and real-time demands are uncertain in advance. In addition to vehicles routes, departure times are treated as decision variables, with delayed departure permitted at each node serviced.

A heuristic comprising route construction and route improvement is proposed within which critical nodes are defined to delineate the scope of the remaining problem along the time rolling horizon and an efficient technique for choosing optimal departure times is developed. This a kind of 'anytime' algorithm comprising route construction and route improvement is validated for the RT-TDVRPTW. Fifty-six problems created by Solomon were taken with minor modifications, upon which two scenarios were further differentiated for testing. The results obtained were compared with a specially designed benchmark solution, and the superiority of this model was clearly justified. A real application to a professional logistics company in Taiwan is also provided in the paper.

Tuzun & Burke (1999) introduces a novel two-phase architecture that integrates the location and routing decisions of the location routing problem (LRP). Their two-phase approach coordinates two tabu search (TS) mechanisms - one seeking a good facility configuration, the other a good routing that corresponds to this configuration - in a computationally efficient algorithm. First introduced in their study, the two-phase approach offers a computationally efficient strategy that integrates facility location and routing decisions. This two-phase architecture makes it possible to search the solution space efficiently, thus it provides good solutions without excessive computation. An extensive computational study shows that the TS algorithm achieves significant improvement over a recent effective LRP heuristic.

Yaman (2005) consider the heterogeneous vehicle routing problem where one can choose among vehicles with different costs and capacities to serve the trips. There are six different formulations developed in this paper: the first four based on Miller-Tucker-Zemlin sub-tour elimination constraints and the last two based on flows. The author compares the linear programming bounds of these formulations and derive valid inequalities, also lift some of the constraints to improve the lower bounds. What observed by computational results is that valid inequalities are more useful in disaggregated formulations and that the lower bounds obtained from the strong formulations and the heuristic solutions in the literature are of good quality.

Campbell (2006) propose and formulate the vehicle routing problem with demand range (VRPDR), a new variation on the traditional vehicle routing problem. An integer programming model is formulated in this paper. In the VRPDR, the delivery quantity for each customer is allowed to vary from its original size by a predefined amount. By adding this limited flexibility to the problem, there is potential to generate significant savings in the total distance traveled. The author addresses issues such as bounding the impact of a given flexibility on total distance and provide empirical results to illustrate "typical" behavior. According to computational results, flexibility appears to on average have a better and faster payoff when there is a wide variety in the initial delivery quantities requested by customers. Across the scenarios, the savings in distance costs are almost always less than the associated flexibility value, but still represent significant improvements. In many cases, only a fraction of the potential reduction in total delivery volume is required to create the improvements.

Main characteristics and solution approaches of reviewed papers referred above are summarized as in following table (Table 5.1):

Table 5.1 Summary of reviewed studies on VRP

Author(s)	Paper Name	Model Definition	Objective(s)	Fleet	Depot & Nodes	Constraints	Sub-tour Constraints	Solution Algorithm / Heuristic Method
H. I. Calvete, C. Gale, M. Oliveros, B. S. Valverde	A Goal Programming Approach To Vehicle Routing Problems With Soft Time Windows	VRPTW	Min. operational costs Satisfy time windows Utilization of vehicles & labor	Heterogeneous Capacitated m-vehicles	Single Depot n-customers with Known demands	Vehicle Capacity Time windows	Time windows	Enumeration-followed By-optimization (EBFO) & Goal Programming
Hoong Chuin Lau Melvyn Sim Kwong Meng Teo	Vehicle routing problem with time windows and a limited number of vehicles	m-VRPTW	Max total number of customers served Min total distance travelled	Homogeneous Capacitated m-vehicles	Single Depot n-customers with different demands	Vehicle Capacity Time windows	Timing	Tabu Search Approach with a holding list Upper bound computation
Nicolas Jozefowiez Fr´ed´eric Semet El-Ghazali Talbi	From Single-Objective to Multi-Objective VRPs: Motivations, Case Studies and Methods	CVRP, VRP Route Balancing	Min total distance travelled & Min the difference between the length of the longest & shortest tour	Homogeneous Capacitated m-vehicles	Single Depot n-customers with different demands	Vehicle Capacity Distances for route balancing	Dantzig et al. ISI subsets	Two-phase strategy for Multi-obj.opt: intensification and diversification
Alvina G.H. Kek Ruey Long Cheu Qiang Meng	Distance-constrained capacitated VRPs with flexible assignment of start&end depots	DCVRP-Fix DCVRP-Flex	Min total travel cost Min vehicle utilization costs (staff cost per shift)	Homogeneous Capacitated k-vehicles	Multiple depot n-customers	Service time, range, capacity	Miller-Tucker-Zemlin	MIP with ILOG OPL Branch and Bound Algorithm (node selection)
Peter Francis Karen Smilowitz	Modeling techniques for Periodic VRPs (with service choice)	PVRP-SC	Balancing arc travel times, (min) stopping times and demand-weighted service benefit	Homogeneous Capacitated k-vehicles	Single Depot n-customers with different demands	min no. of visits vehicle capacity service schedule	Dantzig et al. ISI subsets	continuous approximation geographic decomposition & variable substitution (AMPL)
Andr´ea C. Santos Christophe Duhamel Dario J. Aloise	Modeling the Mobile Oil Recovery Problem as a Multiobjective VRP	MO-VRP	Max oil production at wells Min working time	uncapacitated	Multiple depot Multiple wells (source nodes)	Total operation time, travel time	Miller-Tucker-Zemlin (total load)	CPLEX 11 Optimization (linear Relaxation with MTZ and time windows)
Roberto Baldacci Maria Battarra Daniele Vigo	Routing a Heterogeneous Fleet of Vehicles	Mixed Fleet VRP	Min total travel costs and fixed costs of rental or capital amortization for each vehicle	Heterogeneous m-type of k-vehicles	Single Depot n-customers with different demands	Vehicle Capacity Max. no of each vehicle type	Miller-Tucker-Zemlin (total load)	Many heuristics & solution times are compared
Peter Francis Karen Smilowitz Michal Tzur	The Period Vehicle Routing Problem and its Extensions	PVRP PVRP-SC	Min cost of visiting the nodes	Homogeneous Capacitated k-vehicles	Single Depot n-customers	Range, schedule feasibility, daily vehicle usage	ISI subsets	Many heuristics & solution algorithms are referred
Titipom Thammapi-mookkul, Peerayuth Chamsethikul	A Bi-Criteria Vehicles Routing Problem	VRP for ATMs	Min total travelling time & Min the traveling time of the longest tour	Homogeneous Capacitated m-vehicles	N locations to be visited	Vehicle capacity tour lenght	Miller-Tucker-Zemlin (total load)	An extension of Clarke&Wright Algorithm is developed
A. Georgakopoulos A. Mihiotis	Distribution networkdesign : an integer programming approach	VRP	Min total distance travelled	Capacitated m-vehicles	Single Depot n-municipalities	Vehicle capacity Delivery time Serving orders	Miller-Tucker-Zemlin	Developed a new software (ABC analysis, database for customers, order priorities

Table 5.1 (continued) Summary of reviewed studies on VRP

Author(s)	Paper Name	Model Definition	Objective(s)	Fleet	Depot & Nodes	Constraints	Sub-tour Constraints	Solution Algorithm / Heuristic Method
Marshall L. Fisher Kurt 0. Jörnsten Oli B. G. Madsen	Vehicle Routing With Time Windows: Two Optimization Algorithms	VRPTW	Min total travel cost	Heterogeneous Capacitated k-vehicles	Single Depot n-customers	Service time, Vehicle capacity	Time windows	The Lagrangian Relaxation/ Variable Splitting Approach, & The K-tree Approach
Anand S.Kunnathur, Udayan Nandkeol-yar & Daowei Li	Shipment partitioning and routing to minimize makespan in a transportation network	Non-Linear Routing & Assignment	Min the arrival time of the last partition at node-n (min makespan of a public transportation network)	Capacitated (cargo size, partition size)	u-partitions of cargo divided, n-nodes	Arrival times, schedule, cargo capacity		Iterative Kth-Shortest Path, Overloaded Iterative Kth-SP Sequence Max-Flows-IKSP
Jung-Ug Kim Yeong-Dae Kim	A decomposition approach to a multi-period vehicle scheduling problem	MPVSP	Min transportation costs for product delivery & inventory holding costs at retailers	Homogeneous fleet	Single Depot n-retailers	inventory levels, Vehicle capacity	Delivery scheduling	2 phase K-th shortest path algorithm:decompose MPVSP &schedule selection (assign)
Imdat Kara Tolga Bektas	Integer LP formulations of multiple salesman problems and its variations	mTSP	Min total travel cost	m-salesmen	Single Depot n-nodes	No.of nodes to be visited, m-salesman	Miller-Tucker-Zemlin (no.of visits)	New subtour elimination and bounding constraints solved with CPLEX 6.0 software
Huey-Kuo Chen Che-Fu Hsueh Mei-Shiang Chang	The real-time time-dependent vehicle routing problem (with time windows)	RT-TDVRPTW	Min travel times, waiting times before service & departure at all nodes	Heterogeneous Capacitated k-vehicles	Single Depot n-nodes	Service time, arrival&departure times, capacity	Time windows	Calculation of insertion cost, Choice of departure time & Route construction
Dilek Tuzun Laura I. Burke	A two-phase tabu search approach to the location routing problem	LRP	Min total travel costs, acquisition of the vehicles, locating & operating the depots	Heterogeneous Capacitated k-vehicles	n-customers m-candidate facilities	Vehicle capacity	Miller-Tucker-Zemlin (no.of facility)	Two phase tabu search: Routing (Clarke&Wright) and location (swap&add moves)
Hande Yaman	Formulations and Valid Inequalities for the Heterogeneous VRP	HVRP	Min total travel & fixed vehicle cost	Heterogeneous Capacitated k-vehicles	Single Depot n-nodes	Vehicle capacity	MTZ & ISI Cardinality	Improving lower bounds with derived inequalities & lifted constraints
A.Melissa Campbell	The vehicle routing problem with demand range	VRPDR	Min total delivery costs	Homogeneous Capacitated m-vehicles	Single Depot n-nodes	Demand range of customers & Vehicle capacity	ISI subsets Cardinality	Savings changes:Customer with cheapest insertion cost joined to schedule iteratively

CHAPTER SIX

VEHICLE ROUTING PROBLEM OF THE AUTOMOTIVE COMPANY

6.1 Problem Definition

We consider the daily delivery problem of car parts and accessories imported from Europe in order to satisfy customer demands in Turkey. The parts are stored in a main warehouse in Izmir and shipped from this central depot to customers (dealers) which take place over the country in different cities.

The main purpose of our problem is designing a distribution network for the daily delivery of car parts to these dealers within predefined service time with the objective of minimizing the total cost comprised of driver cost, rental cost of vehicles and travelling cost.

The company uses a fleet of hired vehicles. The dealers send their orders via an online system to the central depot. The order data are used to create job commands for order picking in the main warehouse. After the orders are packed and sorted by dealer cities, they are loaded onto different vehicles according to total weight at each demand point. The orders are shipped daily and the vehicles return to the central depot after serving customers; therefore they don't wait after unloading the goods.

Customers demands are known and they must be served within the maximum time period after they are loaded, which is three days (72 hours) for this company, as agreed upon. By providing minimum travel distances for each vehicle assigned to a route, it will be possible to accomplish deliveries with minimum total cost and within the required service time. Hence, instead we represent the travelling cost in our model which depends travelling distance.

6.1.1 Problem Characteristics and Constraints

- The company hires vehicles which have different loading capacities, so we consider a heterogeneous fleet consisting different type of vehicles.

- Each vehicle starts its travel after being loaded at the central depot and it returns to the central depot after serving the customers.

- Service times of customers (unloading times), travelling times (from each delivery point to another) and distances are known.

- Each customer must be served by only one vehicle, so the hired vehicles must have enough capacity for delivering at least one customer's demand.

- The total demand of customers on one particular route must not exceed the capacity of the vehicle assigned to that route.

- There is no limitation on the number of vehicles that can be hired from each type.

- The maximum possible fleet size is determined according to the hiring cost and total freight of each day.

- The maximum time which each driver must deliver the goods to customers assigned on his route (maximum service time to deliver the customer demand) is given.

6.1.2 Objectives

As we scrutinized the papers on vehicle routing problem and made a literature survey, we found out that the daily distribution problem of the company can be classified and modelled as heterogeneous capacitated vehicle routing problem (HCVRP). Therefore, we developed a mathematical model and formulated it as a mixed-integer programming (MIP) model. In proposed HCVRP model, our objective is to minimize the total cost which consists of:

- The fixed cost of hiring (or activating) a vehicle,

- The driver costs pertaining to their total working hours (per shift),

- The total traveling costs of vehicles.

6.2 Mathematical Model Elements

6.2.1 Notations

N= {1,..n} is the set of nodes including central depot and customers.

i= 1,..n nodes

i= 1 represents central depot

j,r = 2,..n customers (demand nodes)

$\mathbf{Q}_i$= Demand of customer *i* (quantity to be shipped to node *i*)

$\mathbf{U}_i$= Unloading time at customer *i*

F= Expected maximum size of fleet

$\mathbf{t}_{ij}$= The time taken to travel from node *i* to node *j* through arc (*i,j*) .

$\mathbf{A}_\mathbf{k}$= Maximum capacity of vehicle ***k*** (at least capable for one customer assigned)

$\mathbf{c}_k$= Cost of using vehicle ***k*** (activation cost)

$\mathbf{c}^k_{ij}$= Cost of shipping (traveling) from node *i* to node *j* associated with vehicle ***k***

$\mathbf{t}_{max}$= Maximum time that a driver must transport the orders loaded onto his vehicle (maximum service time which is 72 hours for this case)

$\mathbf{D}^k_i$= Total demand (load) of vehicle *k* on its route until node *i*
(including the demand of customer *i*)

$\mathbf{t_k}$ = Total working time (travel time) for the driver of vehicle ***k***.

s= Shift duration in hours (in this study, 8 hours)

$\mathbf{w_k}$= Driver cost per shift for vehicle ***k*** (working cost for each 8 hour-shift)

6.2.2 Decision Variables

$$\mathbf{x}^{k}_{ij} = \begin{cases} \mathbf{1,} \text{ If vehicle } \boldsymbol{k} \text{ travels directly from node } i \text{ to node } j \\ \mathbf{0}, \text{ Otherwise} \end{cases}$$

$$\mathbf{y}_{k} = \begin{cases} \mathbf{1,} \text{ If vehicle } \boldsymbol{k} \text{ is used (hired to transport dealer orders)} \\ \mathbf{0}, \text{ Otherwise} \end{cases}$$

6.3 MIP Formulation Of Developed Model

The problem is stated as a mixed-integer programming (MIP) model and the formulation of developed model is given below:

$$\textbf{\textit{Minimize}} \quad \sum_{k=1}^{F} c_k y_k + \sum_{k=1}^{F} \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{j=1}^{n} c_{ij}^{k} x_{ij}^{k} + \sum_{k=1}^{F} w_k t_k \;\Big/ s$$

The objective function formulates minimization of hiring costs of vehicles, shipping cost (freight) and driver costs including returns to central depot.

Subject to:

$$\sum_{i=1}^{n} \sum_{\substack{j=2 \\ i \neq j}}^{n} Q_j x_{ij}^{k} \leq A_k y_k \qquad \text{k=1,..F} \qquad (1)$$

Constraint (1) states that total load of each vehicle carrying customer demands on its route can not exceed its capacity.

$$\sum_{i=1}^{n}\sum_{\substack{j=2\\ i\neq j}}^{n}(U_j + t_{ij})x_{ij}^{k} \leq t_{\max} y_k \qquad \text{k=1,..F} \qquad (2)$$

The second constraint ensures that the service time can not be larger than the maximum time that the driver must transport the goods to customers assigned on his route.

$$\sum_{j=2}^{n} x_{1j}^{k} = y_k \qquad \text{k=1,..F} \qquad (3)$$

$$\sum_{i=2}^{n} x_{i1}^{k} = y_k \qquad \text{k=1,..F} \qquad (4)$$

Constraints (3) and (4) indicate that each vehicle departs from the central depot and turns back. These two constraints can also be joined together by equalizing the left-hand sides for each vehicle k.

$$\sum_{i=1}^{n}\sum_{\substack{j=2\\ i\neq j}}^{n}(U_j + t_{ij})x_{ij}^{k} + \sum_{j=2}^{n} t_{j1}x_{j1}^{k} = t_k \qquad \text{k=1,..F} \qquad (5)$$

Constraint (5) calculates the total working duration (including the returning time to the central depot) for the driver of each vehicle.

$$\sum_{i=1}^{n}\sum_{k=1}^{F} x_{ij}^{k} = 1 \qquad \text{i≠j; j=2,..n} \qquad (6)$$

Constraint (6) provides that each customer j is visited once, so they can be served by only one vehicle.

$$\sum_{\substack{i=1 \\ i\neq r}}^{n} x_{ir}^{k} - \sum_{\substack{j=1 \\ j\neq r}}^{n} x_{rj}^{k} = 0 \qquad \text{r=2,..n; k=1,..F} \qquad (7)$$

Constraint (7) is the formulation of typical flow conservation equation that ensures the continuity of each vehicle route. This equation implies that every demand point (customer) entered by a vehicle should be left by the same vehicle.

$$D_j^k \geq D_i^k + Q_j - A_M(1 - x_{ij}^k) \qquad \forall\ (i,j),\ i\neq j;\ i,j=2,..n;\ k=1,..F \qquad (8)$$

$$D_j^k \geq Q_j + \sum_{i=2}^{n} Q_i x_{ij}^k \qquad \forall i \in N,\ i\neq j;\ j=2,..n;\ k=1,..F \qquad (9)$$

Subtour elimination constraint is modelled in (8) according to Miller-Tucker-Zemlin formulations. In relation to this, computation of each D^k_i *is stated in (9) and replace weaker requirements* $D^k_i \geq Q_i$ *for all* $i \in N$.

The Big-M value is represented as $\boldsymbol{A_M}$ *and it must not have a smaller value than the loading capacity of the largest vehicle. For* $x^k_{ij}=1$ *in constraint (8), then* $D^k_j \geq D^k_i + Q_j$ *as required. However, if* $x^k_{ij}= 0$, *then* $D^k_j \geq D^k_i + Q_j - A_M$ *which is also satisfied since* $D^k_j \geq Q_j$ *and* $D^k_i \leq A_M$ *(Yaman, 2006).*

$$y_k \in \{0,1\} \qquad \text{k=1,..F} \qquad (10)$$

$$x_{ij} \in \{0,1\} \qquad \text{i,j=1,..n} \qquad (11)$$

The last two constraints (10)–(11) indicate the binary decision variables used in the model.

6.4 Extended Model With Lifted Subtour Breaking Constraints

We adapted the lifted Miller-Tucker-Zemlin (MTZ) subtour elimination constraints by replacing the equation (8) with the following formulation (8i) in order to strenghten our model:

$$D_i^k - D_j^k + A_M x_{ij}^k + (A_M - Q_i - Q_j) x_{ij}^k \leq A_M - Q_j$$

for $\forall (i, j)$, i≠j; i,j=2,..n; k=1,..F (8i)

6.5 Verification and Solution of The Model

Both of the mathematical models presented in 6.3 and 6.4 were verified in Lingo 9.0 optimization programme and optimal solutions were obtained in a reasonable amount of runtime for small problem sizes.

As seen from the Table 6.1, when number of nodes (N, customers) is more than 12 cities, solution time increases dramatically although the model seems still feasible. The model is incapable of obtaining optimal solutions in reasonable computation time. Especially the problems tested for N=20 or more consume indeed very long runtime in Lingo optimization program with branch and bound solution method; nevertheless no optimal or feasible solution is reached.

Therefore, we need to define a **stopping criteria** for performance evaluation purpose of our model. Considering the computation time of test runs; this criteria was decided to be 3,000,000 iterations which means about 75 minutes of maximum runtime for large-sized problems.

The gap between the best objective value found and the theoretical lower bound value when optimization was interrupted according to stopping critera is exhibited in the table for some different test runs.

Table 6.1 Lingo 9.0 results obtained for test problems

No. Of Cities	Fleet Size	Lower Bound (LB)	Lingo Best Objective (BO)		Runtime (minute)	Gap % (BO-LB)
8	11	4335.2	4335.2	Optimum	0.42	0
9	15	6394.1	6394.1	Optimum	5.2	0
10	10	4245	4245	Optimum	17.9	0
12	14	7296.5	7617.2	Feasible	67.3	4.40 %
14	16	7226.6	8026.2	Feasible	75	11.06 %
15	9	6686	7524	Feasible	75	12.53 %
16	18	8996.5	9897.8	Feasible	75	10.02 %
18	19	7751	8569	Feasible	75	10.55 %
20	17	11841	13138.6	Feasible	75	10.96 %
25	18	15554	18238.1	Feasible	75	17.26 %
30	18	16176	---	Unknown	75	---

We also used the extended model 6.4 with lifted sub-tour breaking constraints in Lingo program to solve the problems for the same input data and checked the solution times. The extended model gives optimum solution results in short computation time for N<12, we could not define any difference regarding runtime when compared to the original model in section 6.3. On the other side, since we could not obtain any optimum results for N>12 runs with the extended model, we are not able to evaluate the advantages of lifted-MTZ constraints over the first model 6.3.

Due to very long computation times required for obtaining optimal solutions and even having no feasible results, as in the case N=30, the need for a heuristic solution method becomes obvious. Therefore, it is our focus to develope and implement a heuristic algorithm for real-sized problems in order to gain optimum or near-optimum results in reasonable solution times.

CHAPTER SEVEN

GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEMS

7.1 Introduction To Genetic Algorithms

In recent years, three main developments have contributed to the acceleration and quality of algorithms relevant in a real-time context. The first is the increase in computing power due to better hardware and software programs. The second is the development of powerful metaheuristics whose main impact has been mostly on solution accuracy even if this gain has sometimes been achieved at the expense of computing time. The third development has arisen in the field of parallel computing. The combination of these three features has yielded a new generation of powerful algorithms that can effectively be used to provide real-time solutions in dynamic contexts (Ghiani & others, 2003).

As one of computer-aided heuristic solution approaches, Genetic Algorithms (GAs) were first invented by Holland (1975) and developed by his colleagues and students at the University of Michigan. After them, Goldberg (1989) had provided some further developments. Holland had tried to investigate how adaptation performs in nature and to develop methods in which this could be imported into computer systems. Thus, according to this idea, GAs are built on the principle of natural selection while evolving solutions for the problem being dealt with.

Mainly, GAs are search methods that performs the processes discovered in natural biological evolution. The algorithm operates on an initial population which is constituted by potential or feasible solutions (usually referred as individuals) and focuses on finding those solutions that approach some specification limits or meet the required criteria. To provide this, the algorithm employs the surviving law for the fittest (the one closest to the target limits or critera) solution for obtaining better and better approximations.

Every generation creates a new set of approximate solutions by the process of selecting individuals (these are supposed to be the parents) according to their level of fitness in the problem domain and by reproducing new individuals (called as offsprings) from selected parents with execution of genetic operators (mutation and crossover) inspired from nature of biology. This reproduction rises the evolution of populations which have individuals that are better suited to their environment than their parents, same as in the natural adaptation. Because, the fitter individuals have tendency of yielding offsprings with better quality, in other words, recombinating them can provide better solutions to the problem.

The evolution and creation of new generations give GAs the advantage of operating in many different areas of the search space simultaneously, therefore there is a smaller possibility that solution set (population) may converge to a local optimum values. As Kim, Tang and Kwong (1999) implied the strength of GAs in their book:

" The GA works on the Darwinian principle of natural selection. Whether the criteria of concern be nonlinear, constrained, discrete, multimodal or even NP-Hard; the GA is entirely equal to the challenge. "

A basic formulation of how a GA works is shown in Figure 7.1 below according to Reeves & Rowe (2003). The details, under which crossover and mutation operators are performed or how a new population is created and then by which termination condition this process is stopped, depends on user-specified specifications. Hence, we can state that GAs are significantly more complicated than the neighbourhood search methods since the internal procedures are developed particularly by the researcher according to the structure of his own problem.

```
Choose an initial population of chromosomes;
while termination condition not satisfied do
    repeat
        if crossover condition satisfied then
        {select parent chromosomes;
        choose crossover parameters;
        perform crossover};
        if mutation condition satisfied then
        {select chromosome(s) for mutation;
        {choose mutation points;
        perform mutation};
        evaluate fitness of offspring
    until sufficient offspring created;
    select new population;
endwhile
```

Figure 7.1 A structure of a simple GA

As described until here, GAs operate on a population comprised of individuals rather than working with a single solution. This also allows the researchers to perform parallel processing when dealing with more large and complex problems. The population of individuals means a set of feasible solutions and each individual has a representation, so they are also known as **chromosome**, encoding a possible solution in a given problem space, which is also referred as the search space or state space, that comprises all possible solutions.

The initial population is usually generated randomly in standard genetic algorithms. At every evolutionary step, called as ***generation*** in most of time, the individuals in the current population are compared and evaluated according to their fitness value for the objective function or target criteria of the problem.

Crossover is the exchange of encoded solution parts (***genoms or genes***) between two selected individuals to produce offsprings. ***Mutation*** means a permanent change (altering some genes) in chromosomes which leads to diversity in the population and helps preventing premature convergence, in other words, this variation avoids getting stuck at a local optimum value. ***Fitness value*** represents the quality level or proximity to target specifications of an ivdividual. Thus, individuals or offsprings with better fitness values are transformed into next generation.

The aim of GAs is to reach to a high quality solution for the given problem by using the genetic operators at every reproduction step. Although GAs do not always converge to the global optimum solution, this processing advantage lets GAs have efficient search techniques. Another main advantage of a GA is its capability of manipulating encoded numerous chromosomes (different possible solutions) by parallel processing simultaneously. Since the whole solution space is searched at the same time, it seriously reduces the possibility of getting stuck in local optimum.

Wang & Lu (2009) states that GAs focus on conversion of genes through genetic evolution, thereby solving complex optimization problems. Accordingly, offspring are bred based on chromosome evolution using operators incorporated into GA reproduction, mutation, and crossover to form a search mechanism. This search mechanism, corresponding to offspring breeding, is guided by probability. According to Goldberg (1989), the attributes of genetic algorithms can be summarized as follows:

1) GA calculations are based on coded parameters, rather than the parameter values.

2) GA posses highly parallel search capability which avoids becoming trapped in a local optimum.

3) GAs have no complex mathematical formulas – only the fitness function must be calculated.

4) GAs have no specific rules for guiding the search direction for an optimum, rather, a random search mechanism uses the probability rule.

As some additional specifications and characteristics of GAs, we can also present the following considerations given to the use of GAs by Townsend (2003):

- a robust search technique is required
- "close" to optimal results are required in a "reasonable" amount of time
- parallel processing can be used when the fitness function is noisy
- an acceptable solution representation is available

- a good fitness function is available
- it is feasible to evaluate each potential solution
- a near-optimal, but not optimal solution is acceptable
- the state-space is too large for other methods
- no gradient information is required
- free of restrictions on the structure of the evaluation function
- fairly simple to develop
- do not require complex mathematics to execute
- able to vary not only the values, but also the structure of the solution
- get a good set of answers, as opposed to a single optimal answer
- make no assumptions about the problem space
- probability and randomness are essential parts of GA
- can be hybridized with conventional optimization methods
- potential for executing many potential solutions in parallel

7.2 Structure of a Simple Genetic Algorithm

As briefly described above, each feasible solution is encoded as a chromosome (individual) and each chromosome has a measure of fitness (fitness factor) via a fitness (evaluation or objective) function. The fitness of a chromosome represents its quality degree which also determines its ability to survive and produce offsprings.

After the initial population (solution set with a finite fixed number of chromosomes) is constructed, they are checked at each generation whether all required specifications or constraints are satisfied. If the termination criteria is not reached, then the process of creating new generations continues.

Parent chromosomes are selected according to their fitness and combined to reproduce superior offsprings by crossover operator. Then, new population including offsprings will undergo mutation with a predefined proportion and this process will be followed by fitness calculation again. If the offsprings are fitter than their parents, they will be inserted into the population by replacing the parents. Finally, the population will be transformed into a new generation.

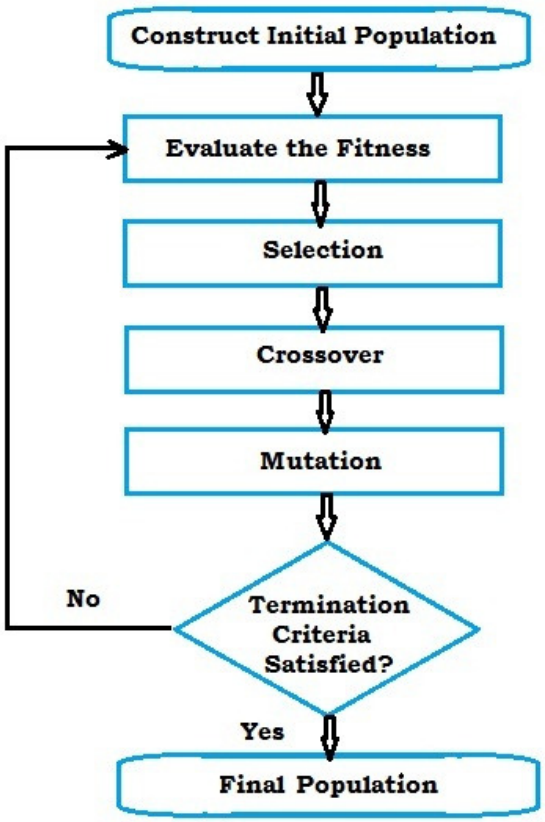


Figure 7.2 A simple GA flowchart

We give the simple flowchart for GAs in above Figure 7.2. GAs can simply work as follows according to description of Townsend (2003):

[**Start**] Generate an initial population according to problem specifications.

[**Fitness**] Evaluate the fitness of each chromosome in the population. Fitness value of a chromosome shows its proximity to the optimal solution.

[**New population**] Create a new population by repeating the following steps until the new population is complete.

[**Selection**] Select two parent chromosomes from the population according to their fitness (the better fitness, the bigger chance to be selected for producing offsprings).

[**Crossover**] With a crossover probability $\mathbf{P_c}$, cross over the parents, at a randomly chosen point, to form two new offspring (children). If no crossover is performed, an offspring is the exact copy of parents.

[**Mutation**] With a mutation probability $\mathbf{P_m}$, mutate two new offsprings.

[**Accepting**] Place new offsprings into the population if they have better fitness.

[**Replace**] Replace the old generation with the new generated population for a further run of the algorithm.

[**Test**] If the end condition is satisfied, stop, and return the best solution in current population.

7.3 Operators Of Genetic Algorithm

A basic genetic algorithm have three genetic operators as selection, crossover and mutation. Herebelow, we present more detailed information with regard to these operators.

7.3.1 Selection

Selection mechanism is important to define which chromosomes in the population will reproduce and survive while creating new generations. The fitter chromosomes have higher probability of being selected and they will be able to pass their characteristics on to offsprings before being dropped out of the population. There are two commonly used selection methods in GAs: roulette-wheel selection and tournament method.

The roulette-wheel method is a proportional selection process according to the relative fitness value of individuals. So, the probability of choosing an individual is directly proportional to its fitness value and highly fit chromosomes will have a greater probability to be selected to form the next generation through crossover and mutation.

The drawback of this method can be its use of the fitness values directly. Thus, when a solution has a very small fitness value compared to the others, it will result in very low probability of being chosen. Tournament method is based on defining the parents by choosing the best individuals from the subgroups of the population.

The subgroups consists of two chromosomes at least and the individuals compete for selection like in a tournament. The best individual within each subgroup is selected for reproduction.

7.3.2 Crossover

The crossover operator involves the swapping of genetic materials in one parent with the corresponding genes of other. In other words, a pair of solutions are recombined in a certain way generating one or more offsprings. Due to this gene exchange, the offsprings carry some of the characteristics of their parents and these characteristic are passed on to the future generations.

The crossover operator roughly imitates biological recombination between two haploid (single chromosome) organisms, as in following example (Townsend, 2003):

Parent A = a1 a2 a3 a4 | a5 a6 and *Parent B* = b1 b2 b3 b4 | b5 b6

The swapping of genetic material between the two parents on either side of the selected crossover point, represented by "|" produces the following offsprings:

Offspring A' = a1 a2 a3 a4 | b5 b6 and *Offspring B'* = b1 b2 b3 b4 | a5 a6

7.3.3 Mutation

Mutation is applied to a single chromosome by predefined probability and provides small random changes in the solution which will create variation by adding some new characteristics to the population and which could not be supplied by the crossover operator.

This operator usually randomly alters one or more genes at the selected positions in a chromosome. By maintaining a sufficient genetic diversity in the population, it enhances the ability of searching in entire solution space and improves the possibility of obtaining near optimal solutions to the problem.

Reeves & Rowe (2003) calls attention to mutation operator in their book, as it needs more careful consideration while deciding the genes to be altered. Since there are several possible values for each gene, mutation is no longer a simple matter. If we decide to change a particular gene, we must provide some means of deciding what its new value should be. This could be a stochastic choice, but if there is some ordinal relation between gene values, it may be more sensible to restrict the choice of genes that are close to the current value.

An example mutation applied at genes 3 and 7 can be exhibited as follows:

Parent: 11**0**010**1**1100 -> Offspring: 11**1**010**0**1100

Here, determining the size of the population is a crucial factor. Choosing a population size too small increases the risk of converging prematurely to local optimum since the population does not have enough genetic material to sufficiently cover the state space by operators. A larger population has a greater chance of finding the global optimum at the expense of more CPU time (Townsend, 2003).

7.4 Reviewed Genetic Algorithm Studies

Before developing a genetic algorithm (GA) to solve the heterogeneous capacitated vehicle routing problem (HCVRP) which was modelled in section 6.3, we searched the literature and reviewed the studies on GA usage for solving VRPs. It was seen that there is scarce of GA implementation on HCVRP papers and they do not show similarity since we are interested in a real life problem with its own characteristics.

We hereby summarize the GA papers closest to our study and then present Table 7.1 to give a short brief of their features:

Baker & Ayechew (2003) consider the application of a genetic algorithm to the basic vehicle routing problem, in which customers of known demand are supplied from a single depot. Vehicles are subject to a weight limit and in some cases, limit on the distance travelled. Only one vehicle is allowed to visit each customer. The problem is to find a set of delivery routes satisfying these requirements and giving minimal total cost. In practice, this is often taken to be equivalent to minimizing the total distance travelled, or to minimising the number of vehicles used and then minimizing total distance for this number of vehicles. The authors propose a GA which generates the initial population by sweep approach from the customers sorted according to increasing order of their distance to central depot and reproduces new solutions by the binary tournament method. Thus, two individuals are chosen from the population at random. The one with the better fitness value is chosen as the first parent. The process is repeated to obtain a second parent. Offsprings are produced from the two parent solutions using a standard 2-point crossover procedure. By applying a simple mutation to new offspring, in which two genes are selected at random and their values are exchanged, two randomly chosen customers are switched between vehicles except in cases where the two customers happen to be on the same vehicle. In the end of paper, computational results are given for the pure GA which is put forward. Further results are given using a hybrid of this GAwith neighbourhood search methods, showing that this approach is competitive with tabu search and simulated annealing in terms of solution time and quality.

Lacomme, Prins & Sevaux (2006) present a multi-objective genetic algorithm for capacitated arc routing problem (CARP) of which objectives are the minimization of total duration of trips and makespan of the longest trip. They use a general framework of multi-objective genetic algorithm (MOGA) called non-dominated sorting GA second version (NSGA-II) and describe how to adapt it for this bi-objective CARP. Initial population is constructed by including both random solutions and 3 good results obtained by heuristics. Then, order crossover and the developed ‘Split’ procedure are adapted to the algorithm. They also use a local search procedure as mutation operator.

The performance of their MOGA is monitored in three sets of classical CARP instances comprising 81 files. It is seen that this algorithm designed for a bi-objective problem is competitive with state-of-the-art metaheuristics for the single objective CARP, both in terms of solution quality and computational efficiency.

Liu, Huang & Ma (2009) study the fleet size and mix vehicle routing problem (FSMVRP), in which the fleet is heterogeneous and its composition is to be determined. They design and implement a genetic algorithm based heuristic. The FSMVRP consists of determining the composition of vehicles to be used; the route of each vehicle, so that the total cost of delivering to all customers, is minimized while each route starts and ends at the depot. Each customer is visited exactly once, customer demands are satisfied and vehicle capacity is not violated. Firstly, an initial solution is produced by the sweep heuristic and then this is converted into its corresponding chromosome by concatenating its routes. The rest of the population consists of chromosomes that are randomly generated. A randomized procedure is built to produce random permutation of the customer locations. Then, a single parent crossover operator is developed to allow more diversity. Instead of using only simple mutation operators, such as insertion or swap, they revise and implement the edge operations for the capacitated vehicle routing problem (CVRP) as a local search (improvement) strategy for the FSMVRP. On a set of twenty benchmark problems, the proposed algorithm reaches the best-known solution 14 times and finds one new best solution. It also provides a competitive performance in terms of average solution.

Tan, Chew & Lee (2006) model the transportation problem for moving empty or laden containers for a logistic company. Owing to the limited resource of its vehicles (trucks and trailers), the company often needs to sub-contract certain job orders to outsourced companies. Their study presents a transportation solution for trucks and trailers vehicle routing problem (TTVRP) containing multiple objectives and constraints.

The solution to the TTVRP consists of finding a complete routing schedule for serving the jobs with minimum routing distance and number of trucks, subject to a

number of constraints such as time windows and availability of trucks and trailers. It contains useful decision-making information, such as the best fleet size to accommodate a variety of job orders and the trend for different number of trailers available at trailer exchange points, which could be utilized by the management to visualize the complex correlations among different variables in the routing problem. To solve such a multi-objective and multi-modal combinatorial optimization problem, a hybrid multi-objective evolutionary algorithm (HMOEA) featured with specialized genetic operators, variable-length representation and local search heuristic is applied to find the Pareto optimal routing solutions for the TTVRP. In the last section of the paper, detailed analysis is performed to extract useful decision-making information from the multi-objective optimization results, as well as to examine the correlations among different variables, such as the number of trucks and trailers, the trailer exchange points, and the utilization of trucks in the routing solutions. It has been shown that the HMOEA is effective in solving multi-objective combinatorial optimization problems, such as finding useful trade-off solutions for the TTVRP routing problem.

Lorena & Lopes (1997) show in their work that classical GA implementation reaches high quality computational results for the difficult set covering problems (SCP). For each population, all solutions are made feasible using a feasibility operator and performing some local search. The fitness function is directly derived from the objective function of SCP. The reproduction operator used is the biased roulette wheel. The roulette is created with each population structure occupying a sector whose size is proportional to its fitness. The reproduction operator 'runs' the roulette selecting a sector. Structures with high fitness generate more offspring at the next generation. One and two point crossover operators are considered. Also, a kind of dynamic mutation operator is implemented of which rate is calculated by a formula.

In computational tests, all optimal and best known solutions are found for trial incidences with reasonable times for a microcomputer implementation. A subset of the standard set covering instances considered in Beasley and Chu are also examined with very competitive results.

Prins (2004) develops an effective and simple hybrid GA for the VRP without trip delimiters. The proposed GA is hybridized with a local search procedure and it can be adjusted for both homogeneous and heterogeneous VRPs. The classical order crossover is used to obtain new solutions. The initial population consists of both solutions generated by a special procedure called 'Split' and some good results obtained by heuristics. In terms of average solution cost, the algorithm outperforms most published tabu search heuristics on the 14 classical Christofides instances and becomes the best solution method for the 20 large-scale instances. The resulting algorithm is flexible, relatively simple, and very effective when applied to two sets of standard benchmark instances ranging from 50 to 483 customers.

Jeon, Leep & Shim (2007) aim to find out the best solution of the vehicle routing problem simultaneously considering heterogeneous vehicles, double trips, and multiple depots by using a hybrid genetic algorithm (HGA). The proposed HGA considers the improvement of generation for an initial solution, three different heuristic processes, and a float mutation rate for escaping from the local solution in order to find the best solution. The HGA additionally considers the initial population by using both random generation and heuristic techniques for better values, minimization process for an infeasible solution, gene exchange process, route exchange process, and flexible mutation rate. Finally, the proposed HGA with position based crossover and point mutation operators, is compared with the results of both a general and hybrid genetic algorithm as well as existing problems suggested by Eilon and Fisher. It gives better solutions rather than the existing genetic algorithms.

Wang & Lu (2008) primarily focuses on solving a CVRP by applying a novel hybrid genetic algorithm (HGA) capable of practical use for manufacturers.

Their proposed HGA has three stages as following: First, the nearest addition method (NAM) is adapted into sweep algorithm (SA) that simultaneously accounts for axial and radius relationships among distribution points with the depot in order to generate a well-structured initial chromosome population, rather than adopting either the NAM or SA alone. Second, response surface methodology (RSM) is employed to

optimize crossover probability and mutation probability via systematic experiments. Finally, an improved sweep algorithm is applied producing a stir over gene permutations in chromosomes which enhance the exploration diversity of the GA, by avoiding convergence in a limited region and enhancing the search capability of the GA in approaching a close-to-optimal solution. Furthermore, an elitism conservation strategy holding superior chromosomes to replace inferior chromosomes was also performed. As the proposed HGA is primarily used to solve practical problem, benchmark problems with fewer than 100 distribution points from an Internet website were utilized to confirm the effectiveness of the proposed HGA. A real case regarding the mission of local active distribution from armed forces in Taiwan details the analytical process and demonstrates the practicability of the proposed HGA to optimize the CVRP.

Park (2001) proposes a hybrid genetic algorithm (HGAV) incorporating a greedy interchange local optimization algorithm for the vehicle scheduling problem with service due times and time deadlines where three conflicting objectives of the minimization of total vehicle travel time, total weighted tardiness, and fleet size are explicitly treated. The vehicles are allowed to visit the nodes exceeding their service due times with a penalty, but within their latest allowable times. The HGAV employs a mixed farming and migration strategy along with a variant of partially mapped crossover and several mutation operators newly developed while maintaining the solution feasibility during the whole evolutionary process. The HGAV is extensively evaluated on the various types of test problems, including a sensitivity analysis with varied geneticparameters on the weighted sum of three objectives of a solution. The results show that the HGAV always attains better solutions than the BC-saving algorithm, but with a much longer computation time.

Table 7.1 Summary of reviewed GA studies on

Author(s)	Study	Problem Type	Objective(s)	Solution Algorithm	Construction of Initial Population	Population Size (N)	Crossover & Rate (Pc)	Mutation & Rate (Pm)	Stopping Criteria	Performance Criterions
Barrie M. Baker Mossie A. Ayechew	A genetic algorithm for the vehicle routing problem	Capacitated Vehicle Routing (CVRP)	Minimization of total travel distance	Pure & Hybrid Genetic Algorithms	1-Random (by sweep approach) 2-Structured (with generalised assignment)	For small problems:30 For large problems: 50	2-point X-over	Simple 2-gene exchange	Number of generations	Benchmarking with other solution algorithms on OR Library problems
P. Lacomme Christian Prins Marc Sevaux	A genetic algorithm for a bi-objective capacitated arc routing problem	Capacitated Arc Routing (CARP)	Minimizing the total duration of trips & duration (makespan) of the longest trip	Multi-objective Genetic Algorithm (MOGA)	Random & 3 good results obtained by heuristics (mix of initial solutions)	Population of 60 solutions including 3 good results	Order (OX) X-over & a procedure called "Split"	Local Search Procedure	Number of generations	Benchmarking with other solution algorithms on the Gdb problem sets
Shuguang Liu Weilai Huang Huiming Ma	An effective genetic algorithm for the fleet size and mix vehicle routing problems (FSMVRP)	Heterogen-ous Fleet Vehicle Routing	Minimization of total cost of delivering to all customers	Hybrid (with local search) Genetic Algorithm	Random (by sweep) & Heuristic (Clark&Wright savings) algorithms used together	30 solutions	Single parent & Order (OX) X-over	Local Search Procedure Pm: 0,05-0,1	Number of generations	Benchmarking on the 20 instances by Golden
K.C. Tan Y.H. Chew L.H. Lee	A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems	Truck Trailer Vehicle Routing (TTVRP)	Minimizing the total routing distance & finding a complete routing schedule	HMOEA Hybrid multi-objective evolutionary algorithm	Random by an insertion procedure	N=800	Route-exchange X-over Pc: 0,8	Multi-mode mutation & local search Pm: 0,3	Number of generations	On 12 test cases Utilization rate, progress ratio & routing plan
LAN Lorena L de Souza Lopes	Genetic algorithms applied to computationally difficult set covering problems	Set Covering Problem-SCP	Minimization of total cost	Classical GA with dynamic mutation and elitist strategy	Local Search Initialization (biased roulette wheel for reproduction)	N=100	One and two-point X-over Pc: 0,6	Dynamic mutation & rate defined by a formula	Number of generations (M=500)	Test problems generated by the Hall / Fulkerson's rules / procedures
Christian Prins	A simple and effective evolutionary algorithm for Vehicle Routing Problem	Heterogen- & Homogenous Fleet Vehicle Routing	Minimization of total cost & number of vehicles	Hybridized (with local search) GA	3 good solutions by using heuristics & a special procedure called "Split"	N between 25 & 50	Order X-over & various OX numbers	Local Search Procedure Pm: 0,05-0,2	Number of generations	Benchmarking on Christofides instances
Geonwook Jeon Herman R. Leep Jae Young Shim	A vehicle routing problem solved by using a hybrid genetic algorithm	Heterogenous VRP with double trips & multiple depots	Minimizing the total sum of distances moved	Hybrid (with heuristic) Genetic Algorithm	Random generation & heuristic method	N=200	Position based X-over Pc: 0,8	Point mutation Pm: 0,01	Number of generations (M=10000)	Comparison with Eilon Data & Fisher Problems
Chung-Ho Wang Jiu-Zhang Lu	A hybrid genetic algorithm that optimizes capacitated vehicle routing problems	Homogenous Vehicle Routing	Minimizing the total distance	Hybrid Genetic Algorithm	Nearest addition method & Sweep algorithm	N= Number of customers (distribution points)	Linear-order X-over Pc: 0,5-1,0	Simple 2-gene exchange Pm: 0,5-1,0	Fitness value change<0.0001 or reach 10000 generations	Benchmark on Eilon problems & application to a real-life problem
Yang-Byung Park	A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines	Capacitated & time constr-ained vehicle scheduling	Minimum total cost of vehicle schedules	Hybrid Genetic Algorithm	Generation of nodes randomly & insertion of feasible vehicle to nodes	N=90	Variant of Partially mapped XO Pc: 0,4	Developed an integrated mutation type Pm: 0,4	Number of generations (M=1000)	Benchmark on Solomon's problems

CHAPTER EIGHT

PROPOSED GENETIC ALGORITHM APPROACH

8.1 Structure Of Proposed Genetic Algorithm

In this chapter, we construct a genetic algorithm (GA) approach to solve the company's daily distribution problem (HCVRP) which was formulated as a mixed integer programming model in section 6.3 and extended in section 6.4. Herebelow, we describe the features of our GA followed by the flow chart in Figure 8.1 and then give a small example to explain how it works.

8.1.1 Chromosome Representation

Chromosome representation is one of the important decisions while developing a GA because a chromosome represents a feasible solution for the problem and all operations are applied on these solutions. Depending on the computer program which will be used to run the algorithm, chromosome coding may effect the elapsed time until the termination criterion is satisfied.

We preferred a simple matrix representation as it does not require specific encoding and easier to read directly from the computer software which wil be used to program our GA. Basicly, if there are *n* customers (cities) and *m* vehicles (maximum number of vehicles), a solution (chromosome) can be represented as 2x(n-1) matrix since n=1 shows the central depot. For example, with N=10 cities and m=7 vehicles (consisting of different types) relevant solutions would be represented as follows:

	Parent 1									Parent 2								
Customer	2	3	4	5	6	7	8	9	10	2	3	4	5	6	7	8	9	10
Vehicle	2	1	5	3	1	6	7	1	6	4	6	5	7	6	1	3	6	1

8.1.2 Initial Population

We follow the below steps for obtaining a large set of feasible 'L' solutions by a kind of 'sweep' approach which is similar to description of Baker & Ayechew (2003) and then, we choose the best 'P'of these solutions to constitute our initial population at the size (P) preferred:

- Customers (cities) are sorted according to ascending order of their distance (we use the travel costs based on distance) to central depot.

- Based on sweeping algorithm and starting with a vehicle of any type (choosen randomly), the first customer is allocated to the current vehicle and assignment continues in the order of sorted customers unless any of constraints is violated.

- Next customer is allocated to the same vehicle as long as capacity and maximum service time constraints are satisfied.

- When a violation of any constraint occurs, the last customer is moved to a new randomly chosen vehicle of any type.

- Each vehicle to be used for sweeping is picked from the set of maximum number of vehicles randomly for any type and the next one is chosen in the same way when a constraint is violated.

- These steps are repeated until 'L' feasible solutions obtained.

- Finally, the best 'P' of these 'L' solutions are chosen according to their fitness (objective function) value in order to have the initial population.

8.1.3 Evaluation of Fitness

For each chromosome in population, value of objective function (here, total cost) is calculated and recorded as fitness of relevant solution. Since our aim is finding the best routes with minimum total cost while service time and capacity constraints are satisfied, the chromosome with smaller fitness value means a better solution (route) for our HCVRP problem.

8.1.4 Selection

We apply an elitist binary tournament method for the selection of chromosomes that undergo crossover and mutation. According to Reeves & Rowe (2003), one way in which we could easily make a genetic algorithm converge is to keep a record at each time step of the best individual seen so far. Since this sequence of values never decreases and since it is bounded above by the value of the optimum, it must converge.

If there is any procedure built that naturally ensure convergence by letting the best individual always remain in the population, then the GA is elitist. So, our selection procedure works as below:

- Two pairs of chromosomes are chosen randomly from the initial population as reproductive parents.

- The fitness values of these pairs are compared and the first two with the worst values (highest total costs) are eliminated

- The remaining pair of chromosomes undergo crossover to reproduce new offsprings.

8.1.5 Crossover Procedure

As decribed by Golub & Picek (2010), uniform crossover (UX) generalizes definition of cross points to make every gene a potential crossover point. So, according to individual structure it indicates randomly which parent will supply the offspring with which genes. Since UX performs random changes, it disrupts the individual genes with great probability but searches larger problem space. Hence, using this method is predicted to be effective while working on small sized populations.

Therefore, we proceed with UX method to reproduce new solutions and evaluate their fitness before including in the population. The pair of parents selected by binary tournament method are used to obtain new chromosomes by crossover. Our UX operator can be described as follows:

- Two customers (genes) served by different vehicles are selected randomly on one parent and these vehicles are imposed on the corresponding customers of other parent.

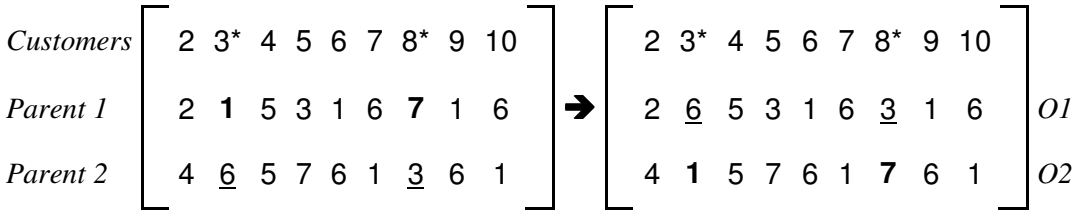
- In order to provide variation and obtain new (different) offsprings, there must be different vehicles assigned to the customers in selected genes (positions) of the other parent.

- The violation of any constaint is immediately checked and if they are all satisfied, fitness value of offsprings (children) are calculated in order to compare with the fitness of parents.

- According to fitness evaluation; this replacement may either include none of the parents or none of the children (when reproduced chromosomes have worse or better fitness values than both parents, or they do not satisfy the capacity and service time constraints).

- The best two among these four chromosomes enter the population and replace the others; so the population size does not change after replication.

Example: A UX operation between customers (city) 3 and 8 is exhibited below.



After the crossover operation, obtained Offspring-1 (O1) seem to have a better fitness value since vehicle-7 is not used anymore. However, let capacity of vehicle-6 be exceeded by demand of customer-3 which means it is not a feasible solution. On the other side, let Offspring-2 (O2) satisfy the capacity and service time constraints while having a better fitness value by removing (de-activating) vehicle-3 and reducing the activation cost.

Hence, fitness values of Offspring-2 and parents are compared. Let the fitness values be in order of Parent-2>Parent-1>Offspring-2. So, Offspring-2 enters the population by replacing Parent-2 while Parent-1 survives. By this way, the population size remains constant after each replication.

8.1.6 Mutation Procedure

We propose the simple gene exchange procedure as mutation operator in order to gain variation among the chromosomes and reproduce new solutions to obtain better objective values. The mutation operator is applied as follows:

- Survived parent(s) and accepted offspring(s) will undergo a simple mutation by selecting their two genes (customers) which are served by different vehicles and swapping these vehicles assigned to those customers.

- If obtained new chromosomes does not violate the capacity and service time constraints, their fitness values are calculated.

- As done during the crossover operation, fitness values of mutated chromosomes are compared with first ones and best two solutions join the population.

- Evaluation of fitness may result with replacing both of the parents or including none of the children (offsprings may have worse or better fitness values than their parents, or they may not satisfy the capacity and service time constraints).

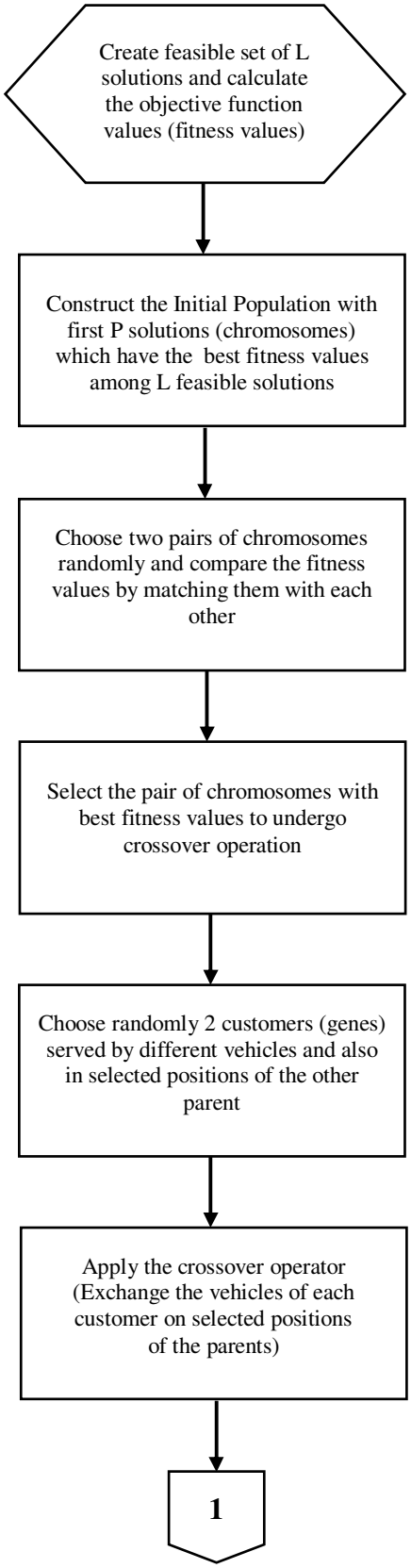
Example: A mutation operation on customers (genes) 4 and 7 is shown below.

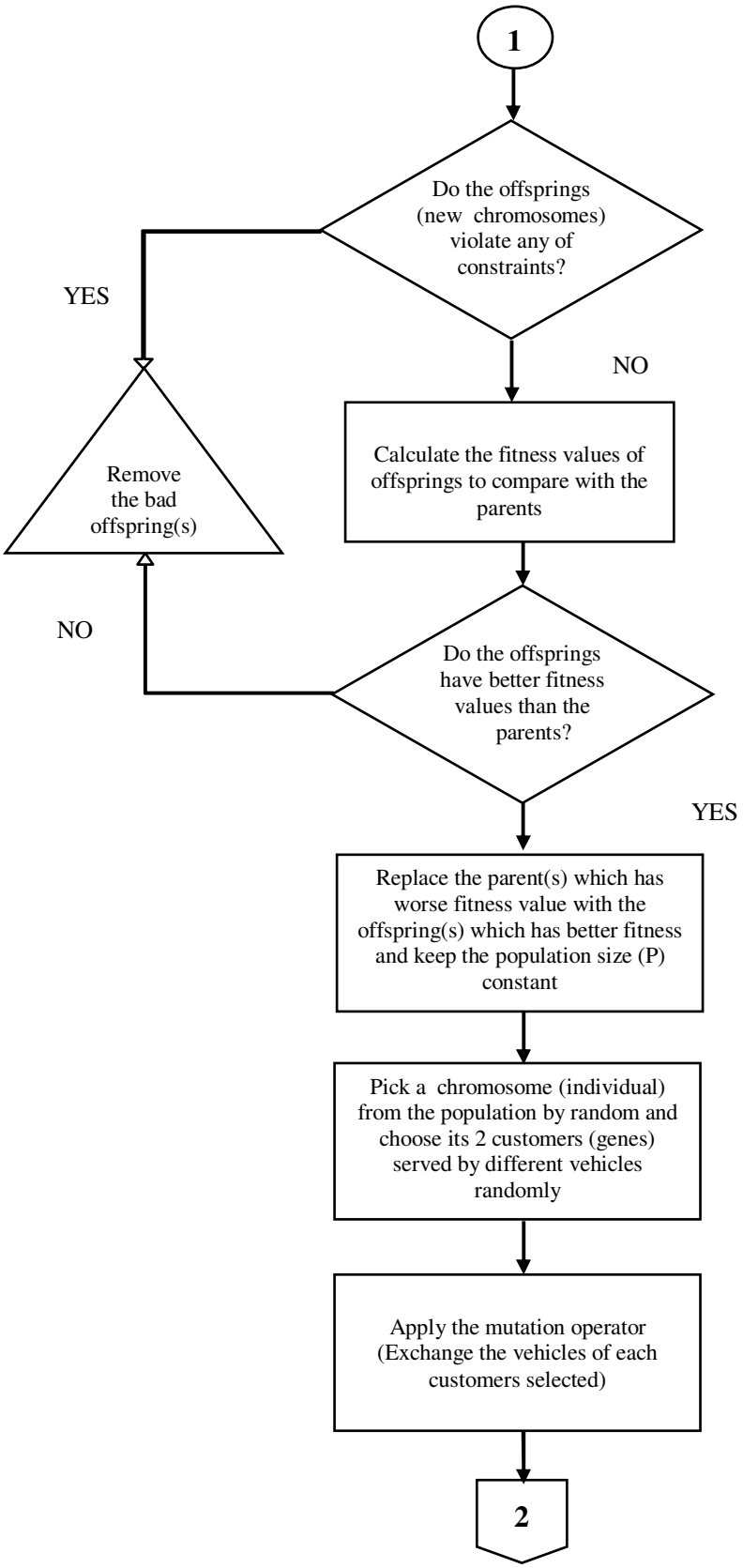
Customers	2	3	4*	5	6	7*	8	9	10		2	3	4*	5	6	7*	8	9	10	
Parent 1	2	1	5	3	1	6	7	1	6	➔	2	1	6	3	1	5	3	1	6	*P1m*
Offspring 2	4	1	**5**	7	6	**1**	7	6	1		4	1	**1**	7	6	**5**	7	6	1	*O2m*

Let all of these chromosomes satisfy the constraints and the fitness values be ordered as Parent-1m>Offspring-2>Parent-1>Offspring-2m. So, the mutation operator did not creat a new better solution (better objective value) for Parent-1m.

However Offspring-2m got a lower fitness value (lower total cost) after vehicle exchange between customers 4 and 7. Thus, Parent-1 survives again and Offspring-2m participates in the population by replacing Offspring-2.

Herebelow, in Figure 8.1 we represent the flow chart of our GA.


Create feasible set of L solutions and calculate the objective function values (fitness values)
Construct the Initial Population with first P solutions (chromosomes) which have the best fitness values among L feasible solutions
Choose two pairs of chromosomes randomly and compare the fitness values by matching them with each other
Select the pair of chromosomes with best fitness values to undergo crossover operation
Choose randomly 2 customers (genes) served by different vehicles and also in selected positions of the other parent
Apply the crossover operator (Exchange the vehicles of each customer on selected positions of the parents)
1


1
Do the offsprings (new chromosomes) violate any of constraints?
YES
NO
Remove the bad offspring(s)
Calculate the fitness values of offsprings to compare with the parents
NO
Do the offsprings have better fitness values than the parents?
YES
Replace the parent(s) which has worse fitness value with the offspring(s) which has better fitness and keep the population size (P) constant
Pick a chromosome (individual) from the population by random and choose its 2 customers (genes) served by different vehicles randomly
Apply the mutation operator (Exchange the vehicles of each customers selected)
2

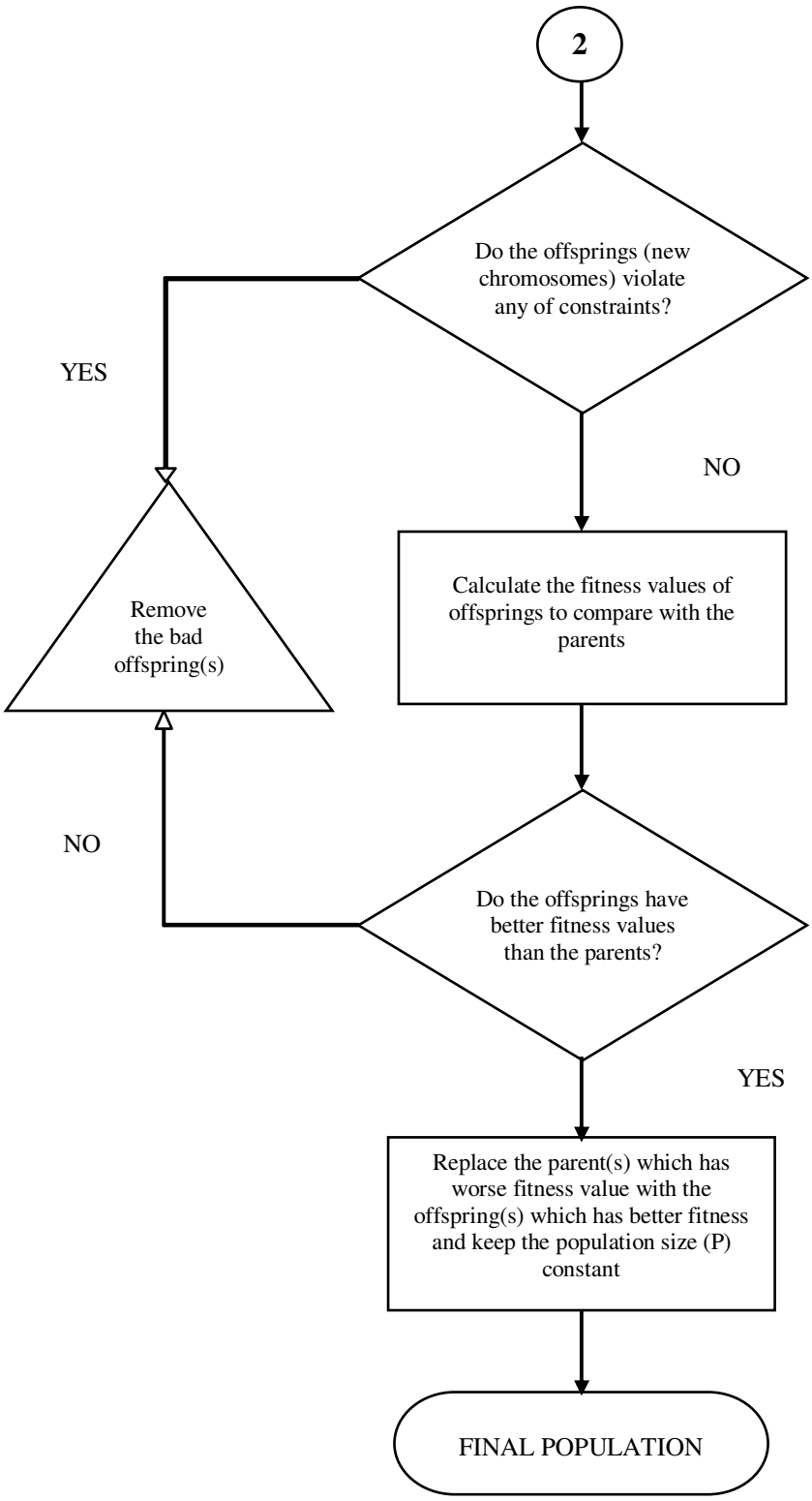


Figure 8.1 Flow chart of proposed GA

8.2 Experiment With A Small-sized Problem

Let us suppose that there are 6 cities to be served by 3 vehicles of different type:

N: 1..7 cities where n=1 represents the central depot

K: 1..3 vehicles, S(shift)=8 hours, t_{max}=72 hours (maximum service time)

Let other data of the problem be given as in following tables.

Vehicles	*Truck (1)*	*Lorry (2)*	*Van (3)*
Capacity (m3)	4000	2500	1500
Driver Cost ($/shift)	100	60	45
Activation Cost ($)	2000	1250	750

Travel Cost ($)	*Izmir*	*Bursa*	*Ankara*	*Afyon*	*Adana*	*Aydin*	*Istanbul*
Izmir	0	90	140	80	230	40	180
Bursa	90	0	90	70	200	110	60
Ankara	140	90	0	60	120	150	110
Afyon	80	70	60	0	140	90	120
Adana	230	200	120	140	0	210	240
Aydin	40	110	150	90	210	0	160
Istanbul	180	60	110	120	240	160	0

Customers	*Izmir*	*Bursa*	*Ankara*	*Afyon*	*Adana*	*Aydin*	*Istanbul*
Demand (m3)	0	700	1200	800	1200	1500	900
Unloading Time (h)	0	0.7	1.2	0.8	1.2	1.5	0.9

Travel Time (h)	*Izmir*	*Bursa*	*Ankara*	*Afyon*	*Adana*	*Aydin*	*Istanbul*
Izmir	0	9	14	8	23	4	18
Bursa	9	0	9	7	20	11	6
Ankara	14	9	0	6	12	15	11
Afyon	8	7	6	0	14	9	12
Adana	23	20	12	14	0	21	24
Aydin	4	11	15	9	21	0	16
Istanbul	18	6	11	12	24	16	0

8.2.1 Initial Population

The cities are sorted and numbered according to their distance (travel cost) to the central depot (Izmir):

Customers	*1* *Aydin*	*2* *Afyon*	*3* *Bursa*	*4* *Ankara*	*5* *Istanbul*	*6* *Adana*
Demand (m3)	1500	800	700	1200	900	1200
Unloading Time (h)	1.5	0.8	0.7	1.2	0.9	1.2

P1: Let Lorry (vehicle-2) be picked by random which is capable of serving City1 and City2. When we calculate to check, we see that it also satisfies time constraint which is 't_{max}=72 hours':

Service Time = Travel Time+Unloading Time ➔ 4+1.5+9+0.8=15.3 hours
Working Time= Service Time+Returning Time ➔ 15.3+8= 23.3 hours
Travel Cost = 40+90+80 = $210

To continue building routes, Truck (vehicle-1) is randomly selected next. It can serve customers (cities) 3,4,5 and 6 with full loading capacity. Calculation to check the constraints:

Service Time = 9+0.7+9+1.2+11+0.9+24+1.2 = 57 hours
Working Time= 57+23 = 80 hours
Travel Cost = 90+90+110+240+230 = $760

Therefore, the first chromosome (solution) of initial population is P1 and its fitness value F(P1) is equal to total cost (value of objective function):

P1:

1	2	3	4	5	6
2	2	1	1	1	1

Total Cost = Activation Costs+Driver Costs+Travel Costs
F(P1)= 2000+1250+60*(23.3/8)+100*(80/8)+210+760 ➔ F(P1)= $5400

P2: Assume that, Van (vehicle-3) is selected randomly. Due to is capacity of 1500m^3, it will only serve City1 and return to the depot. Thus, the time constraint is satisfied as follows:

Service Time = 4+1.5 = 5.5 hours
Working Time= 5.5+4 = 9.5 hours
Travel Cost = 40+40 = $80

Then, let Truck (vehicle-1) be picked next. It can serve cities 2,3,4,5 while satisfying both maximum service time and capacity constrains:

Service Time = 8+0.8+7+0.7+9+1.2+11+0.9 = 38.6 hours
Working Time= 38.6+18 = 56.6 hours
Travel Cost = 80+70+90+110+180 = $530

Finally, Lorry (vehicle-2) is picked to serve City6. Therefore, the relevant time and cost value computation will be as the following:

Service Time = 23+1.2 = 24.2 hours

Working Time= 24.2+23 = 47.2 hours

Travel Cost = 230+230 = $460

Now our second chromosome P2 and its fitness value will be:

P2:

1	2	3	4	5	6
3	1	1	1	1	2

F(P1)= 2000+1250+750+45*(9.5/8)+60*(47.2/8)+100*(56.6/8)+80+530+460

F(P1)= $6184

P3: Initially, Truck (vehicle-1) is chosen randomly. Having capacity of 4000m^3, it can carry the demands of cities 1,2,3. Because City4 has a demand of 1200m^3 which would cause violation of capacity constraints. Let's define the service time and travel cost:

Service Time = 4+1.5+9+0.8+7+0.7 = 23 hours

Working Time= 23+9 = 32 hours

Travel Cost = 40+90+70+90 = $290

Then, Van (vehicle-3) is picked which can only serve City4 with its capacity. The relevant time and cost calculations follow as:

Service Time = 14+1.2 = 15.2 hours

Working Time= 15.2+14 = 29.2 hours

Travel Cost = 140+140 = $280

Next, Lorry (vehicle-2) starts its travel and delivers the demand of City5 and 6. Calculation of service time and cost for vehicle-2 are calculated as below:

Service Time = 18+0.9+24+1.2 = 44.1 hours

Working Time= 44.1+23 = 67.1 hours

Travel Cost = 180+240+230 = $650

Finally, the third member of initial population P3 and its fitness value are:

P3:

1	2	3	4	5	6
1	1	1	3	2	2

F(P3)= 2000+750+1250+45*(29.2/8)+60*(67.1/8)+100*(32/8)+290+280+650

F(P3)= $6302

Now, we have an initial population with three feasible solutions (chromosomes) as P1, P2 and P3 for this small instance.

8.2.2 Evaluation of Fitness (F)

Since the calculations were made as above, inital solutions have the fitness values in the order of: F(P1)= 5400 < F(P2)= 6184 < F(P3)= 6302

8.2.3 Selection

Since this is a small problem and the population size is small, we may directly choose the best two fitness valued chromosomes to have a pair of parents in order to proceed with crossover operator. Hence, P1 and P2 will be used for reproduction.

8.2.4 Crossover

The two cities served by different vehicles are selected as crossover positions and gene exchange is applied for these cities.

Customers	1	2*	3	4	5	6*		1	2*	3	4	5	6*	
Parent 1	2	2	1	1	1	1	➔	2	**1**	1	1	1	**2**	*Offspring 1*
Parent 2	3	**1**	1	1	1	**2**		3	2	1	1	1	1	*Offspring 2*

Now, we will check the offsprings (O1 and O2) if they satisfy the constraints and represent better fitness values.

O1: Lorry (vehicle-2) cannot serve City1 and 6 since the total demand exceeds its capacity.

O2: Truck (vehicle-1) serves to cities 3,4,5,6 with full capacity. Besides, there is no constraint violation which is the same for vehicle-2 and 3. The fitness value of this chromosome is calculated as below:

F(O2)= 2000+750+1250+45*(9.5/8)+80+60*(16.8/8)+160+100*(80/8)+760
F(O2)= $6180 which is better than F(P2)= $6184

As a result, P1 survives and O2 enters the population by replacing P2. So, the new population consists of P1, O2 and P3 chromosomes.

8.2.5 Mutation

Two cities which are served by different vehicles are chosen for gene exchange and after the vehicles are swapped, the calculations are done to check whether all constraints are satisfied.

Cities	1	2*	3	4	5*	6		1	2*	3	4	5*	6	
Parent 1	2	2	1	1	1	1	➔	2	1	1	1	2	1	*Parent 1m*
Offspring 2	3	**2**	1	1	**1**	1		3	**1**	1	1	**2**	1	*Offspring 2m*

P1m (mutated P1) satisfies the capacity and service time constrains. For vehicle-2, total load is (2400m^3) whilst the capacity is 2500m^3. For vehicle-1, total load is 3900m^3 where its capacity is 4000m^3.

Hence, we can calculate the fitness value F(P1m):

Service Time = 4+1.5+16+0.9 = 22.4 hours (Lorry)
Working Time= 22.4+18 = 40.4 hours
Travel Cost = 180+40+160 = $380

Service Time = 8+0.8+9+0.7+9+1.2+12+1.2 = 41.9 hours (Truck)
Working Time= 41.9+23 = 64.9 hours
Travel Cost = 80+90+90+120+230 = $610
F(P1m)= 1250+380+60*(40.4/8)+2000+610+100*(64.9/8) ➔ F(P1m)= $5381

As can be seen, F(P1m) < F(P1) and mutated chromosome will join the population and replace the parent P1.

O2m (mutated O2) satisfies the capacity and service time constrains. The total load for vehicle-3 is 1500m^3 which equals to its capacity. With regard to vehicle-2, total demand is 900m^3 whilst the capacity is 2500m^3. And the capacity of vehicle-1 (4000m^3) is greater than the total load which is 3900m^3.

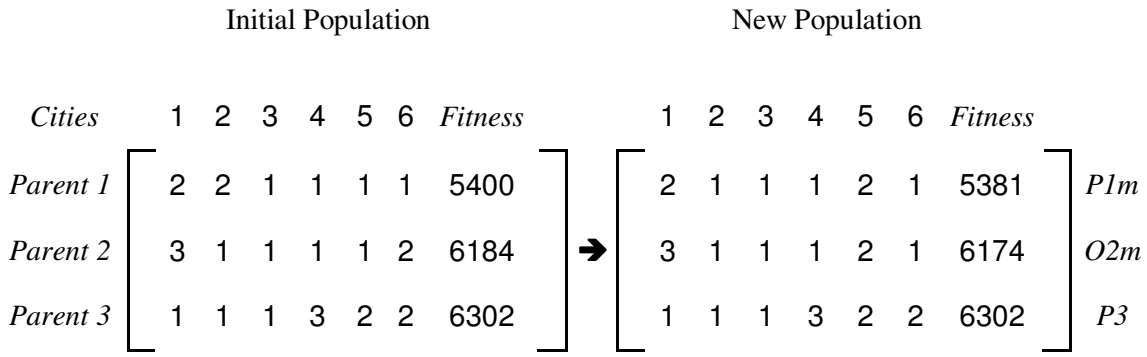
Service Time = 4+1.5 = 5.5 hours (Van)
Working Time= 5.5+4 = 9.5 hours
Travel Cost = 40+40 = $80

Service Time = 8+0.8+9+0.7+9+1.2+12+1.2 = 41.9 hours (Truck)
Working Time= 41.9+23 = 64.9 hours
Travel Cost = 80+90+90+120+230 = $610

Service Time = 18+0.9 = 18.9 hours (Lorry)
Working Time= 18.9+18 = 36.9 hours
Travel Cost = 180+180 = $360
F(O2m)= 1250+360+60*(36.9/8)+2000+610+100*(64.9/8)+750+80+45*(9.5/8)
F(O2m)= $6174

Therefore, mutated offspring will replace its seed O2 and enter the population since it has a better fitness value; F(O2m) < F(O2).

Finally, the new population consists of P1m, O2m and P3 and can be represented in matrix form as following:


Initial Population
New Population
Cities 1 2 3 4 5 6 Fitness
Parent 1 2 2 1 1 1 1 5400
Parent 2 3 1 1 1 1 2 6184
Parent 3 1 1 1 3 2 2 6302
1 2 3 4 5 6 Fitness
2 1 1 1 2 1 5381 P1m
3 1 1 1 2 1 6174 O2m
1 1 1 3 2 2 6302 P3

CHAPTER NINE

SOLVING THE REAL-LIFE HCVRP WITH PROPOSED GENETIC ALGORITHM

9.1 Programming of Genetic Algorithm in MatLab

We developed a program in MatLab 7.0 language regarding the genetic algorithm (GA) described in Chapter 8, which works on producing solutions to the proposed mathematical model for the concerned real-life problem by minimizing the total cost. This GA program has MS-Excel database connections for reading the required data while solving the problem: customer demands, vehicle capacities, activation costs, driver costs, travel costs, travel times and unloading times.

We used the real demand data of past 20 months which were downloaded from the warehouse management system software of the company. The unloading time matrix is calculated by multiplying each demand data with a coefficient used by the company. The travel time matrix was constructed according to the real distances among demand points (customers) and average vehicle speed pre-defined by the company as 65km per hour. In a similar way, the travel cost matrix was also obtained from the real distances by the cost coefficient of the company. MatLab program reads all data from the addressed worksheets of the excel file and generates solutions with GA.

Different numbers of customers are served on different days. Thus, our MatLab programme is designed to re-size the travel time and travel cost matrices by itself in order to make things easier for the decision maker and avoid extra workload of constructing new matrices everyday. Therefore, the program eliminates the unneccessary columns and rows of the matrices according to customers who have zero demand entry on that day. The distribution planner or decision maker only needs to construct other input matrices daily.

Last but not least, the developed GA program allows us to produce many feasible solutions quickly and choose the initial population among. In other words, we are able to have the program first produce a set of feasible solutions, then construct the initial population among those with the best objective function values in accordance with the given population size. As the problem size becomes greater, we have the chance to initiate producing a larger set of feasible solutions and then select the best of these for constituting the initial population. A screen shot of the developed program can be seen in Figure 9.1.

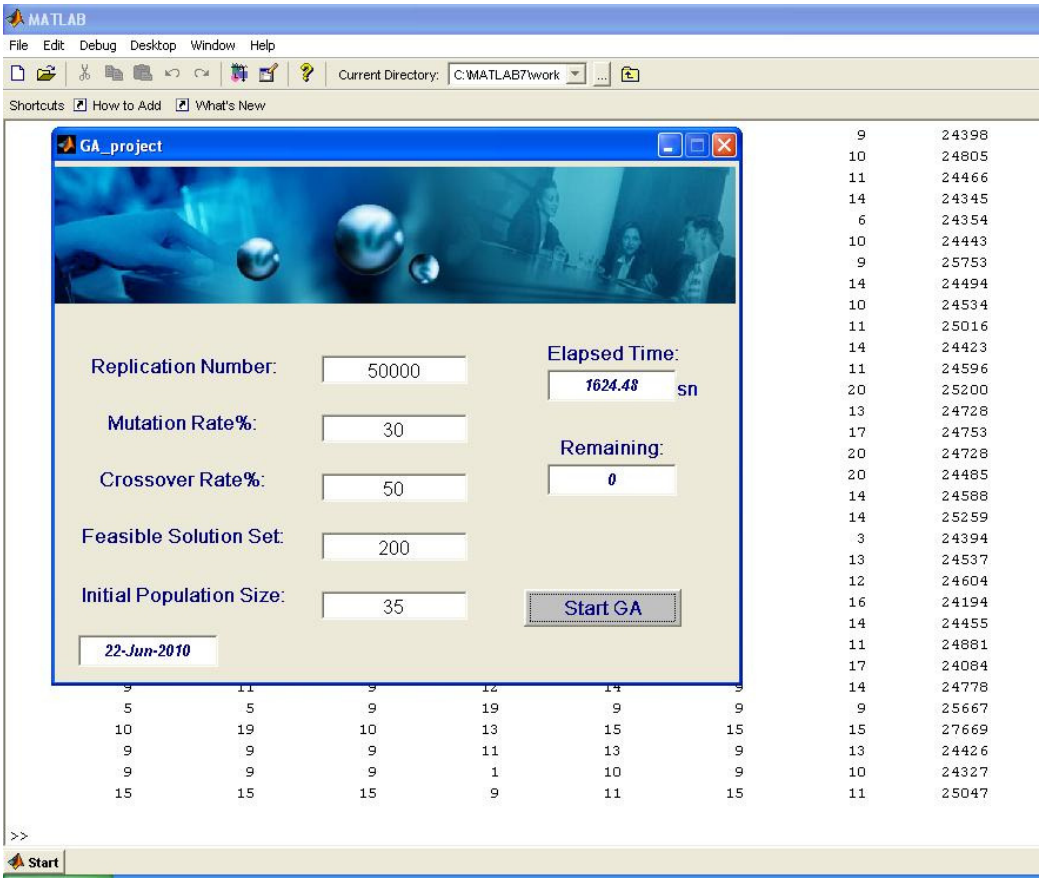


Figure 9.1 Screen shot of the developed genetic algorithm program in MatLab

Taking this advantage gives decision maker the chance of starting his search in more appropriate zones of solution space and this may also provide producing better solutions faster to get closer to the optimum results.

9.2 Parameter Settings and Experimental Runs

The MatLab 7.0 program was run for different sizes of the real-life problem with real input data and obtained solutions were evaluated. For small problem instances, the results were compared with Lingo 9.0 program results as well as theoretical lower bound values. For the problems of real-size, we set forth different performance metrics of which details will be given in Section 9.3.2.

First of all, we shall explain how the initial fleet is determined for each problem instance and which GA parameters were found to be efficient by experimental runs with regard to parameter settings.

Before running the program, the only input that needs to be constructed by the decision maker is the maximum set of vehicles depending on total daily demand. This is defined at first for each different problem and the relevant matrix is introduced to GA program together with other input matrices. The definition of sets for the maximum fleet size which includes different types of vehicles is done according to the simple formulation below:

Number of Vehicle Type-k = Total Daily Demand / Capacity of Vehicle Type-k

For example, let the total daily demand be 5600 units and the vehicles have the loading capacities as Type-1= 4000, Type-2= 2500, Type-3= 1500 units. So, the fleet definition:

5600/4000= 1,4 -> 2 vehicles of Type-1
5600/2500= 2,2 -> 3 vehicles of Type-2
5600/1500= 3,8 -> 4 vehicles of Type-3

Finally, the maximum set of vehicles for this instance will consist of totally 9 vehicles including the above number from each type. The experimental runs for different sizes of real problem were made accordingly.

In the light of reviewed GA studies and papers regarding genetic parameters, we started to test the crossover (P_c) and mutation (P_m) rates in below intervals and replication (reproduction) numbers depending on the problem size. In most of studies as illustrated in Table 7.1, these intervals of genetic operators' rate are found to be efficient for obtaining good-quality solutions and also sometimes best known solutions, as well.

Genetic Operator Rates: P_c= (0.25, 0.7); P_m= (0.05, 0.5)

As proven by VRP experiments of many GA researchers, low P_c and P_m rates usually do not create enough variation in population unless they contain a special algorithm or search procedure and this requires higher replication numbers to reach better solutions. On the other side, very high P_c and P_m rates consume long computation time due to many gene changes and checking calculations of these variations whether they provide indeed better fitness values and do not violate any constraints. Therefore, some test runs must be made for each specific problem at the beginning of experiments and then the parameter settings can be done according to the evaluation of these test results.

Herebelow, we present the test runs of parameter optimization regarding crossover rate, mutation rate, population size and required generation number on two different problem size. daily We used the same daily demand data and fleet composition for each group of test runs but only population size and genetic operator rates were changed.

For deciding the required number of generations, we plotted the evolution graph during each test run with MatLab program and observed the population improvement versus generation number. One of these evolution graphs is exhibited in Figure 9.2.

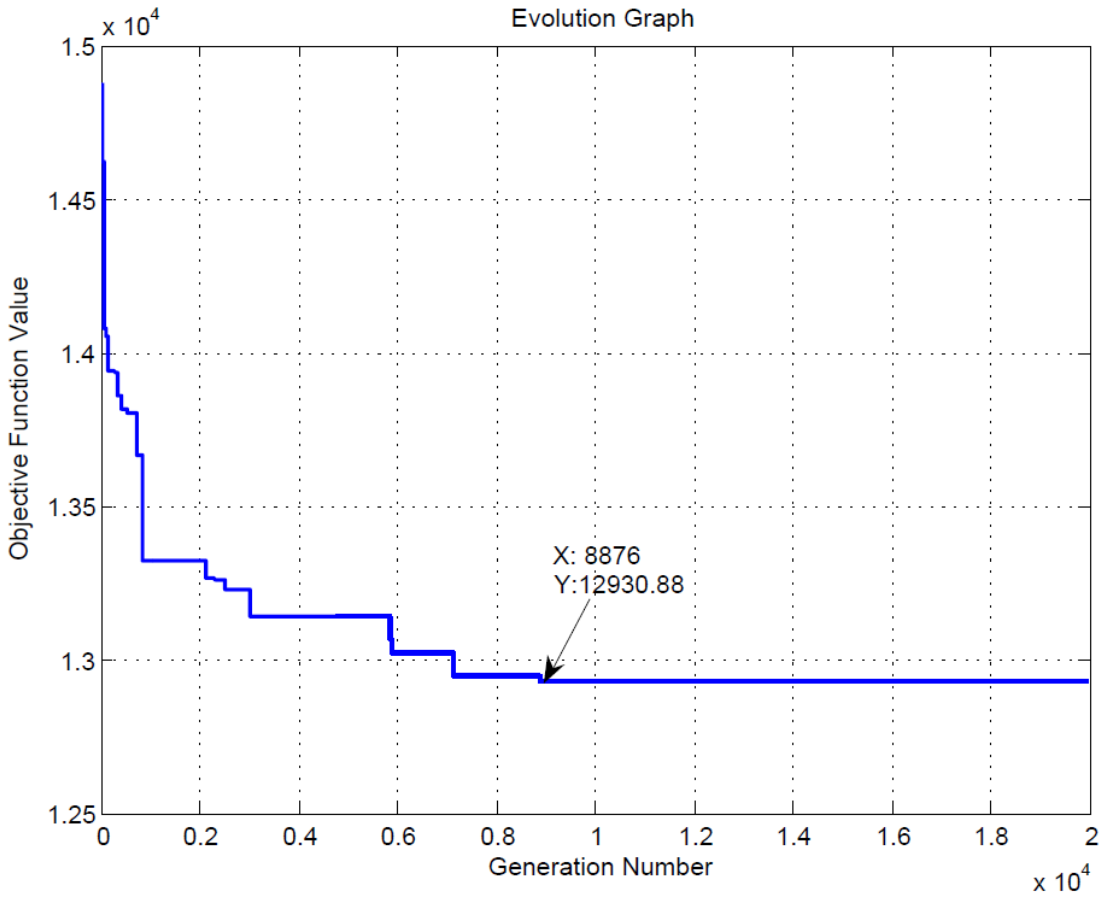


Figure 9.2 Evolution graph of a test instance

Table 9.1 gives the results of medium-size (20 cities) problem while results of large-size instance (30 cities) are shown with Table 9.2 and description of abbrevations used in tables are as follows:

* BOV: Best Objective Value

** Required generation number according to evolution graph which displays the improvement on BOV up to 100,000 generations. This is the last generation during which population shows improvement and after that objective value remains unchanged.

Table 9.1 Parameter settings for medium-size problem instance

Problem No.	Initial Population Size	Crossover Rate	Mutation Rate	Max. Generation Number**	Initial Population BOV* (I)	Final Population BOV (II)	Improvement Rate (I-II)
1	25	25%	5%	10,020	14,875.00	13,123.17	13.35%
2	30	25%	5%	4,430	15,017.00	12,930.88	16.13%
3	35	25%	5%	11,137	14,598.00	12,850.38	13.60%
4	40	25%	5%	8,876	14,875.00	12,930.88	15.03%
5	25	30%	10%	1,268	14,802.00	12,805.33	15.59%
6	30	30%	10%	18,295	14,952.00	12,996.23	15.05%
7	35	30%	10%	8,493	14,672.00	12,979.11	13.04%
8	40	30%	10%	8,132	14,598.00	12,930.88	12.89%
9	25	35%	15%	26,981	14,598.00	13,093.68	11.49%
10	30	35%	15%	14,306	14,875.00	12,904.66	15.27%
11	35	35%	15%	11,288	14,875.00	12,689.44	17.22%
12	40	35%	15%	5,794	14,672.00	12,905.94	13.68%
13	25	40%	20%	2,905	14,802.00	13,100.95	12.98%
14	30	40%	20%	5,103	14,802.00	12,975.23	14.08%
15	35	40%	20%	2,196	14,598.00	12,930.88	12.89%
16	40	40%	20%	15,868	14,672.00	12,606.57	16.38%
17	25	45%	25%	2,447	14,802.00	13,026.32	13.63%
18	30	45%	25%	2,698	14,936.00	12,694.51	17.66%
19	35	45%	25%	8,714	14,875.00	12,873.35	15.55%
20	40	45%	25%	3,251	14,598.00	12,996.01	12.33%
21	25	50%	30%	7,533	14,802.00	12,997.20	13.89%
22	30	50%	30%	2,227	14,875.00	12,987.68	14.53%
23	35	50%	30%	2,187	14,944.00	12,696.71	17.70%
24	40	50%	30%	878	14,936.00	12,930.88	15.51%
25	25	55%	35%	4,417	14,598.00	12,904.66	13.12%
26	30	55%	35%	1,061	14,598.00	12,930.88	12.89%
27	35	55%	35%	2,513	14,672.00	12,608.58	16.37%
28	40	55%	35%	1,021	14,672.00	12,930.88	13.46%
29	25	60%	40%	1,119	14,802.00	12,975.23	14.08%
30	30	60%	40%	11,392	14,802.00	12,696.71	16.58%
31	35	60%	40%	4,992	14,875.00	12,696.71	17.16%
32	40	60%	40%	5,738	14,941.00	12,930.88	15.55%
33	25	65%	45%	1,149	14,598.00	12,930.88	12.89%
34	30	65%	45%	1,307	14,598.00	13,053.22	11.83%
35	35	65%	45%	4,992	14,802.00	12,996.01	13.90%
36	40	65%	45%	10,987	14,672.00	12,975.23	13.08%
37	25	70%	50%	782	14,966.00	12,930.88	15.74%
38	30	70%	50%	3,252	14,598.00	12,930.88	12.89%
39	35	70%	50%	7,832	14,598.00	12,696.71	14.97%
40	40	70%	50%	1,673	14,880.00	12,975.23	14.68%

Table 9.2 Parameter settings for large-size problem instance

Problem No.	Initial Population Size	Crossover Rate	Mutation Rate	Max. Generation Number**	Initial Population BOV* (I)	Final Population BOV (II)	Improvement Rate (I-II)
1	25	25%	5%	18,058	22,915.00	20,872.09	9.79%
2	30	25%	5%	19,803	22,947.00	21,116.11	8.67%
3	35	25%	5%	21,150	23,005.00	21,152.49	8.76%
4	40	25%	5%	67,522	22,609.00	20,749.79	8.96%
5	25	30%	10%	5,494	22,946.00	20,946.68	9.54%
6	30	30%	10%	13,110	22,803.00	20,457.22	11.47%
7	35	30%	10%	14,562	23,407.00	20,871.35	12.15%
8	40	30%	10%	36,994	22,609.00	20,880.26	8.28%
9	25	35%	15%	15,596	22,753.00	20,680.95	10.02%
10	30	35%	15%	25,192	23,020.00	20,788.87	10.73%
11	35	35%	15%	45,396	22,803.00	20,689.93	10.21%
12	40	35%	15%	11,095	22,772.00	20,895.43	8.98%
13	25	40%	20%	21,478	23,020.00	20,599.42	11.75%
14	30	40%	20%	9,393	23,060.00	21,164.74	8.95%
15	35	40%	20%	18,434	22,609.00	20,753.37	8.94%
16	40	40%	20%	41,547	23,426.00	20,558.93	13.95%
17	25	45%	25%	25,396	23,060.00	21,341.24	8.05%
18	30	45%	25%	3,867	23,167.00	20,661.93	12.12%
19	35	45%	25%	16,193	23,377.00	20,777.77	12.51%
20	40	45%	25%	54,323	22,743.00	20,461.14	11.15%
21	25	50%	30%	3,627	22,609.00	21,120.02	7.05%
22	30	50%	30%	7,044	23,423.00	20,901.82	12.06%
23	35	50%	30%	6,524	23,060.00	20,891.98	10.38%
24	40	50%	30%	14,866	23,423.00	20,855.42	12.31%
25	25	55%	35%	16,708	22,829.00	20,908.20	9.19%
26	30	55%	35%	4,678	23,105.00	21,323.38	8.36%
27	35	55%	35%	77,485	23,105.00	20,701.93	11.61%
28	40	55%	35%	4,245	22,683.00	20,759.45	9.27%
29	25	60%	40%	4,887	23,105.00	20,825.52	10.95%
30	30	60%	40%	5,146	23,060.00	20,630.66	11.78%
31	35	60%	40%	6,255	23,342.00	20,913.13	11.61%
32	40	60%	40%	29,403	23,105.00	20,878.19	10.67%
33	25	65%	45%	12,601	22,794.00	21,268.23	7.17%
34	30	65%	45%	11,367	23,060.00	20,922.01	10.22%
35	35	65%	45%	22,354	23,202.00	20,723.10	11.96%
36	40	65%	45%	9,884	22,730.00	20,892.94	8.79%
37	25	70%	50%	6,204	22,946.00	21,201.66	8.23%
38	30	70%	50%	5,889	22,609.00	20,585.90	9.83%
39	35	70%	50%	10,648	22,947.00	20,680.56	10.96%
40	40	70%	50%	8,161	23,060.00	20,953.53	10.05%

Finally, we evaluated the improvement rates per parameter values of each problem instance and revealed the parameter optimization chart below:

Table 9.3 Parameter optimization chart

		20 Cities	Improvement Rate (Average)	Max. Generation Number	35 Cities	Improvement Rate (Ave.)	Max. Generation Number
Population Size		25	13.68%	26,981	25	9.17%	25,396
		30	14.69%	18,295	30	10.42%	77,485
		35	15.24%	11,288	35	10.91%	25,192
		40	14.26%	15,868	40	10.24%	67,522
Crossover - Mutation Rate (%)		25 - 5	14.53%	11,137	25 - 5	9.04%	67,522
		30 - 10	14.14%	18,295	30 - 10	10.36%	36,994
		35 - 15	14.42%	26,981	35 - 15	9.99%	45,396
		40 - 20	14.08%	15,868	40 - 20	10.90%	41,547
		45 - 25	14.79%	8,714	45 - 25	10.96%	54,323
		50 - 30	15.41%	7,533	50 - 30	10.45%	14,866
		55 - 35	13.96%	4,417	55 - 35	9.60%	77,485
		60 - 40	15.84%	11,392	60 - 40	11.25%	29,403
		65 -45	12.93%	10,987	65 -45	9.54%	22,354
		70 - 50	14.57%	7832	70 - 50	9.77%	10648

According to obtained results in above chart, the highest average improvement rates versus population size, crossover and mutation rates reflect the same parameter values for both test problem instances. Therefore, we define the optimum parameters of our GA as follows:

Population Size: 35; Crossover Rate: 0.60; Mutation Rate: 0.40

Besides, we observed that the developed GA does not require high generation numbers for the most of problem instances. In accordance with to Table 9.1 and 9.2 we defined the generation number as below. Further from this point, these values will be used for all problem instances in MatLab program.

For problem size N<20: 15,000 generations; N>20: 30,000 generations

9.3 A Real Problem Instance Solved by Proposed GA

In this section, we represent the data tables of a real problem instance and the relevant output obtained by our GA in MatLab 7.0 program. The demand data belongs to 9th of September, so calculated unloading times and set of vehicles are given, respectively.

Table 9.4 shows customers (cities), relevant demands and unloading times, fleet input (set of vehicles defined according to demands), driver and rental costs in first rows. The next rows include the best solution obtained from the final population followed by fleet composition and route of each vehicle. We also exhibit these routes on the map in below Figure 9.3 to provide a visuality.

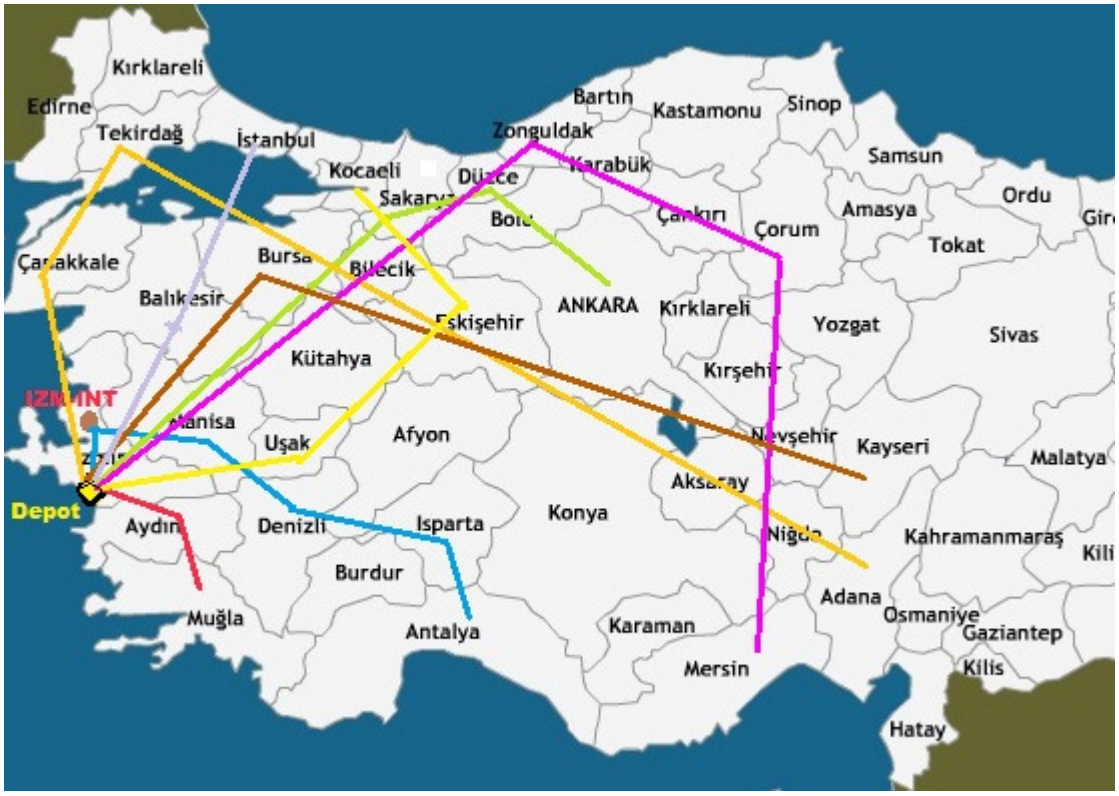


Figure 9.3 View of vehicle routes on the map

Table 9.4 Problem input data and solution output

Problem Data For September																								
City no.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
Sep. 09	DEPOT	IZM-INT	MANIS.	AYD.	B.ESIR	USAK	DNIZLI	MUGL	BURSA	C.KALE	ISPR.	E.SEHIR	ANTLY.	K.ELI	ADPAZ.	TEKRD.	DUZCE	KNYA	ISTNB.	ANK.	ERGLI	CORM	KAYSR	MRSIN	ADA.
Demand	0	235	804	717	469	863	706	3510	302	175	1384	2019	664	1016	1008	632	425	3367	2858	674	37	1004	3240	1429	3520
UnloadT		0.20	0.67	0.60	0.39	0.72	0.59	2.93	0.25	0.15	1.15	1.68	0.55	0.85	0.84	0.53	0.35	2.81	2.38	0.56	0.03	0.84	2.70	1.19	2.93

FLEET	Truck1	Truck2	Truck3	Truck4	Truck5	Truck6	Truck7	Truck8	Lorry1	Lorry2	Lorry3	Lorry4	Lorry5	Lorry6	Lorry7	Lorry8	Lorry9	Van1	Van2	Van3	Van4	Van5	Van6	Van7	Van8
INPUT	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
Capacity	4000	4000	4000	4000	4000	4000	4000	4000	2500	2500	2500	2500	2500	2500	2500	2500	2500	1500	1500	1500	1500	1500	1500	1500	1500
DriverC.	60	60	60	60	60	60	60	60	45	45	45	45	45	45	45	45	45	32	32	32	32	32	32	32	32
RentalC.	1000	1000	1000	1000	1000	1000	1000	1000	650	650	650	650	650	650	650	650	650	450	450	450	450	450	450	450	450

Best Solution From The Final Population (Objective Value= 10,181.17)																								
Cities	DEPOT	IZM-INT	MANIS.	AYD.	B.ESIR	USAK	DNIZLI	MUGL	BURSA	C.KALE	ISPR.	E.SEHIR	ANTLY.	K.ELI	ADPAZ.	TEKRD.	DUZCE	KNYA	ISTNB.	ANK.	ERGLI	CORM	KAYSR	MRSIN	ADA.
Demand	0	282	754	891	319	859	748	1528	414	200	1025	1980	686	1125	1000	654	650	3493	3602	767	36	830	3423	1634	3032
Vehicles		2	2	14	6	1	2	14	5	3	2	1	2	1	12	3	12	7	6	12	11	11	5	11	3

Fleet Composition and Vehicle Routes

Truck1	USAK	E.SEHIR	K.ELI		
Truck2	IZM-INT	MANIS.	DNIZLI	ISPR.	ANTLY.

Truck3	C.KALE	TEKRD.	ADA.
Truck5	BURSA	KAYSR	

Truck6	B.ESIR	ISTNB.	
Lorry3	ERGLI	CORUM	MRSIN

Lorry4	ADPAZ.	DUZCE	ANK.
Lorry6	AYD.	MUGL	

9.4 Performance Evaluation of Our Genetic Algorithm

In this section, we refer to the evaluation criterions of reviewed genetic algorithm papers as illustrated in Table 7.1 and then introduce our performance metrics for the developed GA approach.

9.4.1 Performance Criterions Of Reviewed Studies

The computational results and benchmarking criterions of studies with regard to performance evaluation of the proposed algorithms are summarized in Table 7.1. Most of the researchers evaluate the performance of their GA algorithm with the benchmark problems commonly used in literature or compare the quality of solutions with the results of other heuristics and algorithms like tabu-search, simulated annealing. Hence, there are not common measures for the GA approaches, especially for real-life VRP applications, since the problem characteristics and modeling methods differ from each other.

Hereby the short brief of benchmark problem sets and criterions which are used in the reviewed papers in Section 7.4:

- Test runs for OR library problems (Baker & Ayechew, 2003)
- Experimental runs on GDB problem sets (from Golden, De Armon & Baker) and comparison with other algorithms (Lacomme, Prins & Sevaux, 2006)
- Benchmarking on the problem instances by Golden (Liu, Huang & Ma, 2009)
- Test problems generated by Hall and Fulkerson's rules (Lorena & Lopes, 1997)
- Comparison with Eilon data and Fisher problems (Jeon, Leep & Shim, 2007)
- Benchmarking on Solomon's problems (Park, 2001)
- Utilization rate of vehicles and progress ratio of population (Tan, Chew & Lee, 2006)

9.4.2 Performance Criterions Of Our GA

Since the real life problem with which we are dealing does not match any of the VRP studies in the literature, it is neccessary to design our own performance criteria considering the company.

Currently, we are not able to evaluate the proposed algorithm with the common VRP problems in literature or to apply any of other algorithms proposed in papers. Nevertheless, we defined the following performance criterions to measure the efficiency of our GA approach:

i) For N (customer number) $\leq$ 15 cases, solutions were compared with Lingo 9.0 program results and theoretical lower bound values. As stated in 9.3.1 above, this lower bound is not the optimum and may not be reached most of time. We observed that as the problem size gets larger in Lingo, this theoretical bound diverges to unfavorable values. Thus, it is not a binding criterion.

In Table 9.5, we give the details of experimental runs regarding different sizes of real-life problem and compare with Lingo program output. As explained in Section 6.5, for large-sized problems, Lingo 9.0 does not provide a global or local optimum solution in an acceptable run-time; however it gives a theoretical bound for the best objective value. This lower bound is not the optimum and may not be reached, but it is the lowest objective value predicted (not binding at all) for the relevant problem instance.

Hence, taking this into consideration, we added the lower bound values to our comparison table in order to give an idea about the quality of solutions reached by our GA application. We also define the stopping criteria as 3,000,000 iterations (about 75 minutes as the maximum runtime duration) for solving any problem instance with Lingo 9.0 and if the program does not terminate, we stop the run and evaluate the solutions or lower bound value obtained so far.

However, the stopping criteria of our GA program is the replication (generation) number. MatLab terminates the algorithm and creates an excel file to save the solutions in the last population as soon as the replication number criteria is met. We only needed to stop Lingo program even for these small size instances and noted the bound values. The comparison of experimental runs shows that our GA has a strong structure of producing high-quality solutions in shorter runtimes and stopping criteria is satisfied for all MatLab runs in less than longest runtime of 75 minutes (that is 3,000,000 iterations for Lingo 9.0). Computational results and details of tested problem cases are shown in following Table 9.5.

ii) For performance evaluation of every problem instance; we define the following two main performance metrics to observe:

- Improvement Rate (%) of Population: We compare the best objective value obtained by the proposed GA after replications with the best solution produced for the initial population. This give an idea on the reproduction quality and efficiency of genetic parameters for the relevant instance being run.

- Vehicle Utilization Rate (%): The company aims to have a distribution performance above 80% and this is calculated as follows:

Distribution Performance (%) = Total Demand/Total Vehicle Capacity Usage

We calculate the vehicle utilization (capacity usage) rate for the best solutions gained by our GA and observe whether the performance criterion is met.

As summarized in Table 9.6, the results are pleasing. Our GA performs very good and provides high-quality solutions in a reasonable amount of runtime. In the existing system, defining the routes and fleet requirements takes up to 3-4 hours of the planner staff everyday. So, we can state that our GA approach is inspirational to tackle the distribution planning problem of the company.

Performance evaluation for problem groups of different sizes is also considered and more experiments were done on six customer sets as 15, 20, 25, 30, 35 and 39 cities. We made runs of five different problems for each set. For each of problem sets, all input data (fleet mix, costs, customers etc.) had been used as the same except demands and unloading times. Each of the five problems had demand input (and unloading times, respectively) of five different days chosen randomly. The aim of these experiments was monitoring the performance change of our GA for different problem sizes with different demands.

We observed that distribution performance changes slightly based on demand input. However, developed GA never gives lower performance than company's target. The objective values and performance of obtained best solutions are exhibited in Table 9.6.

Table 9.5 Performance comparison with Lingo 9.0 results for small problems

No. Of Cities	Number of Generations	BOV (I) MatLab	Lingo Lower Bound (LB)	Best Objective Value (BOV) Lingo (II)		BOV % (I-II)	Gap % (I - LB)	MatLab (T) Runtime (minute)	Lingo (T) Runtime (minute)	Runtime (T-TT)
10	15,000	3489.89	3375.73	3478.56	Feasible	0.003 %	3.270 %	6.12	8.43	-37.7 %
10	15,000	3461.70	3220.59	3353.04	Feasible	3.139 %	6.965 %	5.89	7.33	-24.45 %
10	15,000	3053.61	2778.52	3019.59	Feasible	1.114 %	9.009 %	6.55	7.02	-7.18 %
12	15,000	3468.2	3031.27	3440.32	Feasible	0.804 %	12.598 %	7.98	9.10	-14.04 %
12	15,000	3543.6	3165.92	3481.57	Feasible	1.750 %	10.658 %	7.63	9.62	-26.08 %
12	15,000	3528.0	3271.95	3454.88	Feasible	2.073%	7.258 %	7.61	11.2	-47.17 %
15	15,000	3980.92	3548.12	4083.81	Feasible	-2.585 %	7.258 %	6.83	20.43	-199.12 %
15	15,000	4058.40	3600.22	4373.59	Feasible	-7.766 %	11.29 %	5.10	27.55	-440.20 %
15	15,000	4086.30	3625.06	4044.69	Feasible	1.018 %	11.287 %	6.25	24.13	-286.08 %

Table 9.6 Performance evaluation for the problem groups based on demand of five different days

Number of Cities	Problem No.	Population Size	Crossover Rate	Mutation Rate	Initial Population BOV (I)	Final Population BOV (II)	MatLab Runtime (minute)	Improvement Rate (I-II)	Vehicle Utilization Rate
15	1	35	60%	40%	8,234.40	7,912.64	6.84	4.07%	92.27%
	2	35	60%	40%	8,719.50	8,233.35	7.05	5.90%	94.00%
	3	35	60%	40%	8,833.40	7,971.30	6.73	10.82%	97.48%
	4	35	60%	40%	9,118.20	8,375.38	7.20	8.87%	94.70%
	5	35	60%	40%	9,405.30	9,142.84	7.66	2.87%	95.80%
20	6	35	60%	40%	15,641.00	13,857.69	9.47	12.87%	95.73%
	7	35	60%	40%	16,814.00	15,164.03	8.80	10.88%	92.52%
	8	35	60%	40%	16,954.00	15,505.40	7.96	9.34%	98.74%
	9	35	60%	40%	18,079.00	16,746.30	7.84	7.96%	96.98%
	10	35	60%	40%	14,802.00	12,975.23	8.37	14.08%	93.33%
25	11	35	60%	40%	17,311.00	15,325.59	12.16	12.95%	97.30%
	12	35	60%	40%	21,243.00	19,247.48	11.73	10.37%	95.68%
	13	35	60%	40%	21,666.00	19,171.94	13.01	13.01%	95.02%
	14	35	60%	40%	21,701.00	20,019.66	12.70	8.40%	92.71%
	15	35	60%	40%	18,025.00	15,977.55	11.80	12.81%	97.43%
30	16	35	60%	40%	18,025.00	16,008.65	13.57	12.60%	98.63%
	17	35	60%	40%	19,437.00	17,810.60	14.23	9.13%	94.21%
	18	35	60%	40%	21,014.00	19,373.12	15.53	8.47%	95.00%
	19	35	60%	40%	22,665.00	20,050.73	14.18	13.04%	96.45%
	20	35	60%	40%	22,630.00	20,192.94	15.52	12.07%	97.48%

Table 9.6 (Continued) Performance evaluation for the problem groups based on demand of five different days

Number of Cities	Problem No.	Population Size	Crossover Rate	Mutation Rate	Initial Population BOV (I)	Final Population BOV (II)	MatLab Runtime (minute)	Improvement Rate (I-II)	Vehicle Utilization Rate
35	21	35	60%	40%	24,731.00	22,020.03	18.23	12.31%	95.87%
	22	35	60%	40%	24,055.00	22,283.38	17.80	7.95%	96.59%
	23	35	60%	40%	28,208.00	24,146.22	16.20	16.82%	94.49%
	24	35	60%	40%	28,146.00	24,873.64	18.78	13.16%	93.88%
	25	35	60%	40%	23,444.00	20,622.01	17.63	13.68%	96.08%
39	26	35	60%	40%	20,463.00	18,564.73	20.18	10.23%	93.30%
	27	35	60%	40%	20,242.00	18,429.46	19.16	9.84%	92.66%
	28	35	60%	40%	25,696.00	22,875.84	20.80	12.33%	94.09%
	29	35	60%	40%	21,101.00	19,444.18	21.92	8.52%	96.26%
	30	35	60%	40%	24,054.00	21,986.55	22.24	9.40%	95.53%

9.5 Features Of Our Study And Other Application Areas

We have scrutinized many VRP papers in the literature and seen that the heterogeneous fleet construction studies are indeed rare. Additionally, the case studies even on real-life problems among those do not consider the total operation costs and fleet construction at the same time. Besides, considering the solution approaches, the application of GAs to capacitated heterogeneous vehicle routing problems is scarce. Therefore, to the best of our knowledge, the distinguishing features of our study can be described as follows:

- The proposed HCVRP model takes all cost items into account which are respectively driver costs, vehicle activation costs and operation (travelling) costs of vehicles including their return to central depot,

- The developed program allows the decision maker to evaluate many different data at the same time like costs, customer service time, driver shifts, vehicle requirements planning and alternative route,

- Our GA program can be modified to model and solve the other similar distribution and fleet sizing problems of many companies,

- It is also applicable to minimizing the distribution costs of raw material, semi-finished goods or other consumables among workstations or production cells of a multi-facility manufacturing company,

- It can be used to define the vehicle routes for material (waste, recycle products or returning faulty goods) collection from several locations (pick-up points, customers, cities etc.) to a cental warehouse or a plant in terms of reverse logistics.

CHAPTER TEN

CONCLUSIONS

10.1 Performed Study

The purpose of our study is to tackle the daily distribution planning problem of the automotive company and to design an efficient solution algorithm for the decision maker. We first made the definition of the real problem and put forth all the constraints for consideration. By examining the existing system and distribution planning process, we found out that there is more than constructing the best routes to solve the problem.

After a wide literature survey and scrutinizing the papers on vehicle routing, we realized that we encountered with a heterogeneous capacitated vehicle routing problem. First, the problem was formulated as mixed-integer programming model. Then, we started searching for a solution algorithm which can give optimum or near-optimum results.

The mathematical model was encoded in Lingo optimization software and optimum results were obtained for small sized instances. However, the program could not provide solutions for the real-sized problem instances. This led us to seek for heuristic solution methods and other programming tools. Finally, we decided to study on genetic algorithm and focused on related papers.

We developed a genetic algorithm for solving the problem and programmed it in MatLab language. The experimental results were pleasing. Our algorithm performs well and produces high-quality solutions which also satisfy the performance target of the company by consuming shorter run-time. It provides the decision maker the opportunity of evaluating alternative distribution plans by saving cost and time.

10.2 Future Research

As it is known, the vehicle routing problem is classified as NP-hard which means the most difficult problem type to tackle and provide optimum results. It is attracting more attention day by day and inspiring the operations researchers to develope better models and faster solution algorithms.

It may be useful if a hybrid genetic algorithm is designed for solving the problem and observe the improvement on solutions and run-time. Also, by implementing one of other heuristic methods like tabu-search, simulated annealing etc., obtained results can be compared with our GA solutions. In this way, the decision maker gets the advantage of a better evaluation and choosing the best one among alternative solutions.

REFERENCES

Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a Heterogeneous Fleet of Vehicles. *Operations Research/Computer Science Interfaces, 43* (1), 3-27.

Baker M. B., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research, 30*, 787–800.

Calvete, H. I., Gale C., Oliveros M. & Valverde B. S. (2007). A Goal Programming Approach to Vehicle Routing Problems With Soft Time Windows. *European Journal of Operations Research, 177* (3), 1720-1733.

Campbell A. M. (2006). *The vehicle routing problem with demand range*. University of Iowa, Iowa City, USA. Retrieved November 26, 2010 from http://myweb.uiowa.edu/acampbll/papers/vrpdr.pdf

Chen H. K., Hsueh, C. F., & Chang, M. S. (2006). The real-time time-dependent vehicle routing problem. *Transportation Research*, *42* (Part E), 383–408.

Christopher, M. (2005). *Logistics and Supply Chain Management*, (3rd ed.). Harlow: FT Prentice Hall.

CSCMP, Council of Supply Chain Management Professionals (February, 2010). *Supply Chain Management Terms and Glossary*. Retrieved November 26, 2010 from http://cscmp.org/digital/glossary/document.pdf

Fisher, M. L., Jörnsten, K. O., & Madsen, B. G. (1997). Vehicle Routing Problems with Time Windows: Two Optimization Algorithms. *Operations Research, 45* (3), 488-492.

Francis P., & Smilowitz, K. (2006). Modeling techniques for periodic vehicle routing problems. *Transportation Research, 40* (Part B), 872–884.

Francis, P., Smilowitz K., & Tzur, M. (2008). The period vehicle routing problem and its extensions. *Operations Research/Computer Science Interfaces, 43* (1), 73-102.

Ganesh, K., Nallathambi, A. S., & Narendran, T. T. (2007). Variants, solution approaches and applications for Vehicle Routing Problems in supply chain. *International Journal of Agile Systems and Management, 2* (1), 50-75.

Georgakopoulos A., & Mihiotis A. (2004). Distribution network design: An integer programming approach. *Journal of Retailing and Consumer Services, 11,* 41–49

Ghiani G., Guerriero, F., Laporte, G. & Musmanno, R. (2003). Real-time Vehicle Routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, *151* (1), 1-11.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Massachusets: Addison-Wesley.

Golub, M. & Picek, S. (June, 2010). *On the Efficiency of Crossover Operators in Genetic Algorithms with Binary Representation.* Retrieved December 12, 2010 from http://www.wseas.us/e-library/conferences/2010/Iasi/NNECFS/NNECFS-23.pdf

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Michigan (MI): University of Michigan Press.

Jeon, G., Leep, H. R., & Shim, J. Y. (2007). A vehicle routing problem solved by using a hybrid genetic algorithm. *Computers & Industrial Engineering, 53,* 680–692.

Jozefowiez, N., Semet, F., & Talbi, E. G. (2008). From Single-Objective to Multi-Objective Vehicle Routing Problems: Motivations, Case Studies, and Methods. *Operations Research/Computer Science Interfaces, 43* (2), 445-471.

Kara I., & Bektas, T. (2006). Integer linear programming formulations of multiple salesman problems and its variations. *European Journal of Operational Research, 174,* 1449–1458.

Kek, A. G. H., Cheu, R. L., & Meng, Q. (2008). Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling, 47* (200), 140–152.

Kim J. & Kim Y.(1999). A Decomposition Approach To A Multi-Period Vehicle Scheduling Problem. *The International Journal of Management Science*, *27*, 421-430.

Kim F. M., Tang, K. S., & Kwong, S. (1999). *Genetic algorithms: concepts and designs* (2nd ed.). London: Springer-Verlag.

Klose, A. & Drexl, A. (2005). Facility Location Models For Distribution System Design. *European Journal of Operational Research*, *162*, 4-29.

Kunnathur, A. S., Nandkeolyar, U., & Li D. (2005). Shipment partitioning and routing to minimize makespan in a transportation network. *Computers & Industrial Engineering, 48,* 237–250.

Lacomme P., Prins C., & Sevaux, M. (2006). A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers and Operations Research, 33,* 3473–3493.

Lau C. H., Sim, M., & Teo K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research, 148,* 559–569.

Liu, S., Huang, W., & Ma, H. (2009). An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research, 45* (Part E), 434–445.

Lorena, L. & Lopes, L. S. (1997). Genetic algorithms applied to computationally difficult set covering problems. *Journal of the Operational Research Society, 48,* 440-445.

Park Y., B. (2001). A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. *International Journal of Production and Economics, 73,* 175-188.

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

Ratliff, H. D., Nulty, W. G. (October 5, 1996). Logistics Composite Modeling. *Technical White Paper Series of The Logistics Institute at Georgia Tech*. Retrieved November 26, 2010 from http://www.idii.com/wp/tli_logistics_model.pdf

Reeves, C. R., & Rowe, J. E. (2003). *Genetic Algorithms: Principles And Perspectives- A Guide To GA Theory*. New York: Kluwer Academic Publishers.

Santos, A. C., Duhamel, C., & Aloise, D. J. (2008). Modeling the Mobile Oil Recovery Problem as a Multiobjective Vehicle Routing Problem. *Communications in Computer and Information Science, 14* (1), 282-292.

Tan, K. C., Chew, Y. H., & Lee, L. H. (2006). A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research, 172,* 855–885.

Tarantilis C. D., Ioannou, G., & Prastacos, G. (2005). Advanced vehicle routing algorithms for complex operations management problems,. *Journal of Food Engineering, 70,* 455–471.

Teng, S. G. & Jaramillo H., (2005). Linking Tactical and Operational Decision-Making to Strengthen Textile/Apparel Supply Chains. *International Journal of Logistics Systems and Management*, *1* (3), 244-266.

Thammapimookkul T., & Charnsethikul, P. (2005). *A Bi-Criteria Vehicles Routing Problem.* Kasetsart University, Bangkok, Thailand. Retrieved November 26, 2010 from http://ieinter.eng.ku.ac.th/research/optimization/tha05a.pdf

Toth, P. & Vigo, D. (2002). *The Vehicle Routing Problem*. USA: SIAM, Philadelphia Monographs on Discrete Mathematics and Applications.

Townsend, A. A. R. (2003). *Genetic Algorithms–A Tutorial.* Retrieved May 19, 2010 from http://www-course.cs.york.ac.uk/evo/SupportingDocs/TutorialGAs.pdf

Tuzun D., & Burke, L. I. (1999). A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research, 116,* 87-99.

Wang, C. H., & Lu, J. Z. (2009). A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. *Expert Systems with Applications, 36,* 2921–2936.

Yaman, H. (2006). Formulations and Valid Inequalities for the Heterogeneous Vehicle Routing Problem. *Mathematical Programming, 106* (Ser. A), 365–390.

Zhang, L. (September 4, 2007). *Vehicle Routing in Reverse Logistics*. Retrieved December 12, 2010, from http://www.patrec.org/old_patrec/conferences/PATRECForum_Sep2007/presentations/13_Lixi%20Zhang_Vehicle%20Routing%20in%20Reverse%20Logistics.pdf