AN ABSTRACT OF THE THESIS OF

Hyunpae Lim for the degree of Master of Science in Industrial Engineering presented on August 22, 2007.

Title: A Genetic Algorithm for the Vehicle Routing Problem with Heterogeneous Vehicles from Multiple Depots, allowing Multiple Visits

Abstract approved:

Shiwoo Lee

In the thesis, an application of a genetic algorithm (GA) is considered to solve the vehicle routing problem (VRP) which involves heterogeneous vehicles to serve known customer demands from multiple depots achieving the minimum delivery cost, where each customer must be satisfied by one or more visit(s), and each vehicle must make at most one visit to any particular customer. Vehicles can be unused. The problem involves optimizing the routes for all vehicles which are to serve a certain number of customers from multiple depots, allowing multiple visits. These conditions are generalized from the classical VRPs, which only involve one depot and one visit to each customer.

The VRP is one of combinatorial optimization problems which are difficult to obtain an optimal solution through the classical optimization methods owing to the high computational complexity. The GA is a randomized global search algorithm to solve problems by imitating processes observed during natural evolution. It has been a widespread application to various combinatorial optimization problems such as traveling salesman problem, scheduling problem and VRP. The performance of GA is subject to the process parameters such as population size, crossover rate, termination condition, and mutation policy. For the generalized VRP under considerations, the influences of the process parameters in the proposed GA are examined by Taguchi method which is known as a robust design tool for optimizing the process parameters.

The proposed GA is the first effort to solve the generalized VRP, which allows the multiple depots, multiple visits and heterogeneous vehicles. A real-life example problem of 35 US cities and 3 depots has been proposed to measure the performance of the proposed GA. In addition, 4 benchmark problems from the prior works only allowing one depot, one visit and homogeneous vehicles has been tested. The proposed GA outperforms the prior works by generating the equal to or the better solutions than the best known solutions. The computational results obtained from the performance comparisons show that the proposed GA is an effective and feasible method for solving the VRP with heterogeneous vehicles from multiple depots, allowing multiple visits to customers.

A Genetic Algorithm for the Vehicle Routing Problem with Heterogeneous Vehicles from Multiple Depots, allowing Multiple Visits

by

Hyunpae Lim

A THESIS

submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Master of Science

Presented August 22, 2007

Commencement June 2008

Master of Science thesis of Hyunpae Lim presented on August 22, 2007.

APPROVED:

Major Professor, representing Industrial Engineering

Head of the Department of Industrial and Manufacturing Engineering

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Hyunpae Lim, Author

ACKNOWLEDGEMENTS

I would like to thank my major professor, Prof. Shiwoo Lee for his invaluable guidance, support and constant encouragement over the past two years which lead me to my successful study. I have learnt a lot working with him. His knowledge and enthusiasm for research have been invaluable sources of inspiration for me.

I am deeply indebted to Kyoungchoel. I would like to thank Chatpong, Weerakit, Tom, Nathan, Melissa and Robin for their timely support and academic matters.

Finally, special thanks are due to my parents and my wife, Seungkyoung. Without their constant understanding, patience and support, this work would not have been completed. I cannot express in words all the love I have for them.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure Page

LIST OF TABLES

LIST OF TABLES (Continued)

Chapter 1. Introduction

The vehicle routing problem (VRP) is a problem in which a set of routes for a fleet of vehicles based at one or several depot(s) must be determined for a certain number of geographically dispersed customers. The objective of the VRP is to minimize the total distance traveled by all vehicles, which can be considered as delivery costs. Recently, with the increase in fuel prices, the importance of minimizing delivery costs has been emphasized as a key factor which can reduce the total costs of production and distribution. Thus, the VRP has received an enormous amount of attention from industries.

The classical VRP consists of a certain number of customers with known demands at predetermined locations, served by a fleet of vehicles with a homogeneous capacity from a depot. In the VRP, vehicles dispatched from a single depot must deliver the required amounts of the goods to all customers, satisfying all demands and finally return to the depot. Ideally, the vehicle routes are designed in such a way that each customer is visited only once by exactly one vehicle and the total demands of all customers on one particular route must not exceed the capacity of the vehicle. In the real world, however, the constraints which include homogeneous vehicles, a single depot and one allowed visit to customers are unrealistic. Therefore, the objective of the thesis is to develop a GA to find a set of all routes which minimizes the total distance traveled by heterogeneous vehicles from multiple depots, allowing multiple visits.

The VRP is one of combinatorial optimization problems belonging to the nondeterministic polynomial-time hard (NP-hard) class [Bodin et al, 1983] which cannot be solved to optimality within polynomially bounded computational time [Falkenauer, 1996]. Many different approaches have been developed to solve the NP-hard problems. As one of these approaches, genetic algorithm (GA) is one of the widely used computational methods, and has been successfully implemented in a wide variety of problem domains due to its robustness and flexibility [Berger and Barkaoui, 2003].

The GA is an adaptive heuristic search algorithm inspired by the *Theory of Natural Selection* by Charles Darwin, and has been extensively used to tackle many combinatorial problems, including various VRPs. In the GA, a population of chromosomes (individuals) or solutions is maintained during the evolution, in which selection, crossover, and mutation take place. The quality of each solution is evaluated by a fitness function which represents individual's survivability in the wild. This fitness determines the individuals for the crossover or mating, which produces offspring in the next generation. The mutation is also used to prevent the local convergence by diversifying the search space. The average quality of the population gradually improves as new and better solutions are generated and worse solutions are removed.

In the thesis, addressed is the VRP which has heterogeneous vehicles from multiple depots, allowing multiple visits to customers. Based on a rigorous review of literature, this thesis is the first study to solve the generalized VRP, which relaxes the constraints of vehicle's homogeneity, the number of depots and the number of visits. The problem has been approached by developing a mathematical model and an efficient implementation of the GA.

1.1 Objective of the thesis

The first objective of the thesis is to generalize the VRPs by removing the constraints of the number of vehicles, the number of depots, and the number of visits allowed to each customer. The second is to develop and validate a mixed integer programming (MIP) model to achieve the optimality. The third is to develop and validate a GA to effectively and efficiently solve the medium or large VRPs with heterogeneous vehicles from multiple depots, allowing multiple visits. The fourth is to apply the Taguchi robust design method to optimize the process parameters of the proposed GA. The fifth is to develop the new mutation policy to better diversify the search space of the proposed GA.

1.2 The problem statement

In a network, where nodes are customers or depots and the links are the roads, the goods are to be delivered to customers by a fleet of heterogeneous vehicles from multiple depots. The objective of the VRP under consideration is to find a set of all vehicle routes which minimize the total distance traveled with the following constraints. The locations of the customers and the depots are known. The demands of all customers and the capacities of the heterogeneous vehicles in each depot are known. Each vehicle starts from and ends at a depot. Each customer cannot be served only once by one vehicle from a depot but also by other vehicles until his or her demand is satisfied. A customer can be visited in multiple times, but not by a single vehicle, which means a vehicle can visit a customer only once. The quantity of the goods delivered to one or more customer(s) by a vehicle must not exceed the capacity of the vehicle.

1.3 Organization of the thesis

Chapter 1 introduces the problem and the objective of this research. In Chapter 2, the literature review of VRPs, the GA and Taguchi method are presented. Chapter 3 presents a MIP mathematical model for the VRP with heterogeneous vehicles from multiple depots, allowing multiple visits. It also proposes a GA to solve the VRPs with medium or large numbers of nodes under consideration. Taguchi method to optimize the process parameters of the proposed GA has been presented. In Chapter 4, various computational results are summarized. Finally, conclusions and future research are discussed in Chapter 5.

Chapter 2. Literature review

2.1 Vehicle routing problem

2.1.1 Description of the classical VRP

The VRP was introduced initially by Dantzig and Ramser [1959], and it has been widely studied since. They described a real-world application concerning the delivery of gasoline to service stations. In addition, they proposed the first mathematical programming formulation and an algorithmic approach for the solution of the problem. Fisher [1994] describes the problem as finding the efficient use of a fleet of vehicles that must make a number of stops to deliver passengers or products. The term "*customer*" is used to denote the stops to make. Every customer has to be assigned to exactly one vehicle in a specific order. That is done with respect to the capacity of vehicles in order to minimize the total cost. The classical VRP consists of a set of customers with known demands at predetermined locations and a set of vehicles with a homogeneous capacity. The vehicles are dispatched from and return to a central location referred to as a depot. The VRP is to service all customers without overloading the vehicle, while minimizing the total distance traveled. Figure 2.1 shows an example of a classical VRP with 3 vehicles, 9 customers and a single depot. In Figure 2.1, node 0 in the box denotes the depot, nodes 1 to 9 in the circle are the customers, and the arrows represent visiting orders to customers for each vehicle from the depot.

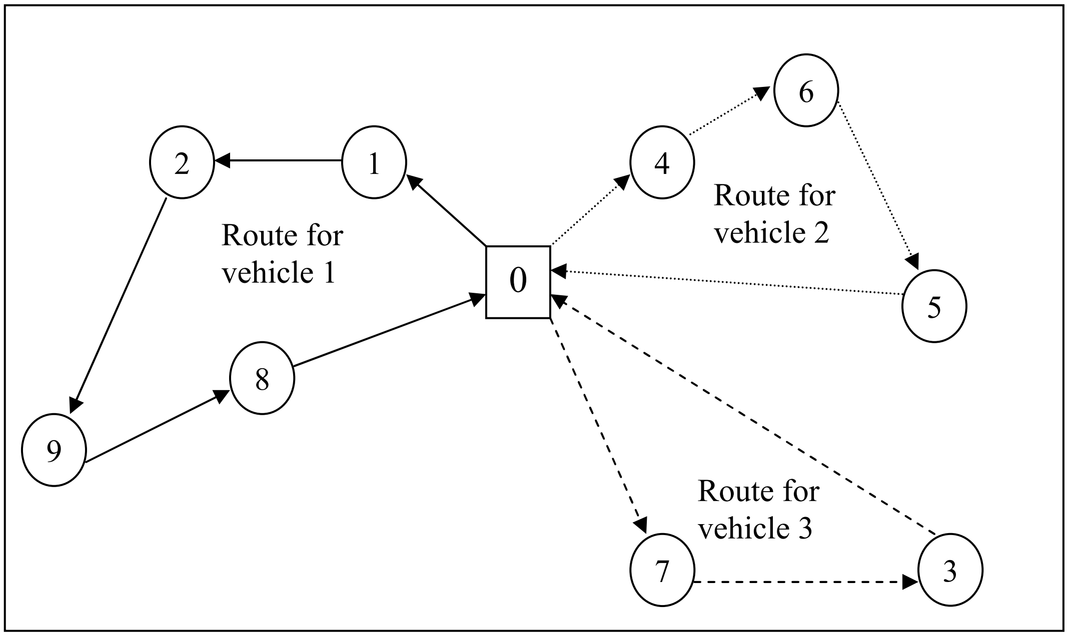

Figure 2.1 An example of a classical VRP with 3 vehicles, 9 customers, and a single depot

2.1.2 Classification of VRPs

A particular case of the VRP arising when only one vehicle is available at a depot and no additional operational constraints are imposed, i.e., traveling salesman problem (TSP), is extensively described by Lawler et al. [1985]. The TSP has one vehicle, one depot and multiple customers. The customer demands should be satisfied with one visit by the vehicle. Thus, the TSP is generally considered as a special case of the VRP.

Another version of the VRP is capacitated VRP (CVRP). The CVRP has n customers and a single depot, which has a number of vehicles, with identical delivery capacity, to satisfy customer demands. The vehicles must accomplish the delivery with the minimum total travel cost, where the cost is the distance d_{ij} from node i to j, where i and j = {0, 1,…, n}, where 0 stands for a single depot and n is the number of customers. Some studies considered the heterogeneous vehicles in order to reduce the delivery cost by dispatching the appropriate vehicles to the routes. The applications of the CVRP can be found in Ball et al. [1995], Fisher [1995], Desrosiers et al.[1995], Osman [1993b], Laporte [1992], Golden and Addad [1995] and Toth and Vigo [2002].

The vast majority of papers have been published on a classical single-depot capacitated VRP (SDCVRP) [Laporte et al., 1984; Laporte, 1992; Toth and Vigo, 2002] and only a few papers were found dealing with problems known as multiple-depot capacitated VRP (MDCVRP) [Chao et al., 1993; Renaud et al., 1996; Toth and Vigo, 2002]. The MDCVRP is an extension of the classical VRP with vehicles starting from different depots. The constraints of the MDCVRP are similar to the ones of the VRP, except the requirement which each vehicle starts from and finishes the delivery at the same depot. In the case of multiple depots, if the customers are clustered around the depots, then the problem can be modeled as a set of independent SDCVRPs. However, if the customers and the depots are intermingled, the MDCVRP should be solved. While a large number of papers have been published on the classical SDCVRP, there have been a few dealing with the MDCVRP, where each vehicle starts and finishes its route at the same depot. The applications of the MDCVRP can be found in Laporte et al. [1984], Tillman and Hering [1971], Chao et al. [1993], and Renaud et al. [1996], all using adaptations of classical SDCVRP procedures.

Most prior research on the VRP has considered vehicles with homogeneous or heterogeneous capacities and one or more depot(s) while only allowing a visit to each customer. Having more than one allowed visit to a customer has not been widely considered in the literature. However, if multiple visits are allowed in the VRP, it is intuitive that they may reduce the number of vehicles used to satisfy the customer demands. Let assume that there are five customers, each of who has the demand as much as a little more than half of homogeneous vehicle's capacity. To satisfy all demands by only one visit to each customer, 5 vehicles are necessary. Only three vehicles might be needed to solve the problem. Shin and Kang [1991] introduced and solved the VRP allowing multiple visits to a customer using a heuristic method.

2.1.3 Solution methods for VRPs

The VRP requires the determination of the optimal set of routes to be completed by a fleet of vehicles to serve a given set of customers. The VRP is one of the most important and studied combinatorial optimization problems in academia and industries. The classical VRP introduced in Subsection 2.1.2 is relatively simple. In real life, the VRP can have many more complications, such as asymmetric distances, multiple depots, heterogeneous vehicles and time-windows of each customer. These possible complications make the problem more difficult to solve.

The VRP is a NP-hard problem [Bodin et al., 1983]. NP-hard problems are difficult to solve, and no optimal algorithm which is able to solve the problem in polynomial time has been found [Falkenauer, 1996]. Finding an optimal solution to a NP-hard problem is usually very time consuming or even impossible. Due to this nature of the problem, it is not realistic to use optimal solution methods to solve large problems. For small problems with only a few customers, the branch-and-bound method has been used [Pereira et al., 2002]. Most approaches for large problems are based on heuristics which are approximation algorithms that aim at finding good feasible solutions quickly [Laporte et al., 2000].

Many models and algorithms were proposed to obtain the optimal or approximate solution of the different versions of the VRP. A thorough classification was given in Desrochers et al. [1990]. Laporte and Novert [1987] presented an extensive survey that was entirely devoted to exact methods for VRPs. Other surveys were reported by Christofides et al. [1979], Magnanti [1981], Bodin et al. [1983], Christofides [1985], Laporte [1992], Fisher [1995], Toth and Vigo [1998], and Golden et al. [1998]. They could be divided into two main classes: classical heuristics mostly from between 1960 and 1990, and metaheuristics from 1990 [Laporte et al., 2000].

The classical heuristics can be divided into three groups: construction methods, two-phase methods, and improvement methods [Laporte and Semet, 1999]. Construction methods gradually build a feasible solution by selecting arcs based on minimizing cost. The two-phase method divides the problem into two stages: clustering customers into feasible routes disregarding their order, and constructing routes. One of two-phase methods is the sweep algorithm in Laporte et al. [2000]. Improvement methods start with a feasible solution and try to improve it by exchanging arcs or nodes within or between the routes. The local search algorithms developed by Aarts and Lenstra [1996] belong to the improvement heuristics. The advantage of the classical heuristics is that they have a polynomial running time [Laporte et al., 2000]. When using them, one is better able to provide good solutions within a reasonable amount of time [Cordeau et al., 2002]. On the other hand, they only perform a limited search in the solution space. Therefore, they have a risk of resulting in a local optimum.

During the past few decades, there have been many attempts to solve VRPs quickly and effectively by using metaheuristics such as tabu search (TS), simulated annealing (SA), and GA [Laporte et al., 2000]. The TS and the SA move from one solution to another in the neighborhood until a stopping criterion is satisfied. Many different TS heuristics have been proposed with unequal success. Rochat and Taillard [1995] used the TS heuristic to solve some benchmark VRPs. Osman [1993a] obtained similar results using the SA. The GA maintains a population of good solutions that are recombined to produce new solutions. A considerable research on the GA has been done to solve VRP with time windows (VRPTW) [Berger and Barkaoui, 2003], where each customer has a time window for which the vehicle has to arrive. Berger and Barkaoui presented a new hybrid GA (HGA) to solve the CVRP. The HGA uses two populations of solutions that periodically exchange some chromosomes which are the feasible solutions to the CVRP. The algorithm has shown to be competitive in comparison to the best TS heuristics. However, Renaud et al. [1996] reported that such heuristics require substantial computing times and several parameter settings.

2.2 Genetic algorithm

2.2.1 The background

The *Theory of Natural Selection* was proposed by Charles Darwin in 1859. The theory states that individuals with certain favorable characteristics are more likely to survive and consequently pass their characteristics on to their offspring. Individuals with less favorable characteristics will gradually disappear from the population. In nature, the genetic inheritance is stored in chromosomes made of genes. The characteristics of every organism are controlled by the genes which are passed on to the offspring when the organisms reproduce. Occasionally a mutation causes changes in the chromosomes. Due to natural selection, the population will gradually improve on average as the number of individuals having the favorable characteristics increases.

The GA is a randomized global search algorithm that solves problems by imitating genetic processes observed during natural evolution. The “survival of the fittest” nature of this algorithm lends itself favorably to being extremely robust in its search for optimality [Gen and Cheng, 2000]. Fundamentally, the GA evolves a population of bit strings, or chromosomes, where each chromosome encodes a solution to a particular problem. This evolution takes place through the application of genetic operators which mimic phenomena such as reproduction and mutation observed in nature. The characteristics of the GA that are different from other heuristics, are as follows [Gen and Cheng, 2000]:

- The GA works with coding of the solutions instead of the solutions themselves. Therefore, a well-designed coding or efficient representation of the solutions in the form of a chromosome is required.
- The GA searches from a group of solutions, different from other metaheuristics like the SA and the TS which start with a single solution and move to another solution by some transition. Therefore, the GA does a multi-directional search in the solution space, reducing the probability of finishing in a local optimum.
- The GA only requires objective function value which measures the fitness of chromosomes while many other algorithms require the continuity or differentiability. Many real-life examples contain discontinuous search space.
- The GA is nondeterministic, i.e., it is stochastic in natural decisions, which make the GA more robust.
- The GA is a heuristic because it does not know when it has found an optimal solution.

Generally, the GA has five basic components as summarized below [Gen and Cheng, 2000; Rawlins, 1991].

- A genetic representation of solutions to the problem
- A way to create an initial population of solutions

- An evaluation function rating solutions in terms of their fitness
- Genetic operators that alter the genetic composition of offspring during reproduction
- Values for parameters of the GA

2.2.2 The procedure

The procedure of the traditional GA may be described as follows. The GA starts from some randomly generated initial population which is a set of solutions. Davis [1987] suggests that for research purposes, a good deal can be learned by initializing a population randomly. Moving from a randomly-created population to a well-adapted population is a good test of the algorithm. By doing this, important features of the final solution will have been produced by the search and recombination mechanism of the algorithm, rather than the initialization process. To generate and to search for an optimal solution, a function which evaluates the survivability of solutions is required in the initialization process. This is also called the fitness function, because it ranks each feasible solution in accordance to its fitness value. The fitness function is the most critical part of the GA, as it is the one which decides how much time it takes to find the optimal solution.

The second step, a reproductive process allows parent solutions to be randomly selected from the population. Typically, a lower selection pressure is indicated at the start of a search in favor of wide exploration of search space, while a higher selection pressure is recommended at the end to narrow the search space [Gen and Cheng, 2000]. Offspring solutions are made by the reproductive processes using a crossover operator. The offspring solutions are produced which inherit some of the characteristics from each parent. Then, a random mutation could be applied to the offspring with a certain probability. Gen and Cheng [2000] proved that the mutation operator can sometimes play a more crucial role than crossover. Therefore, the crossover and mutation operators need to well-designed in accordance with the problem on hand.

Finally, generation replacement takes place in the third step. The evaluation of the solutions can be related to the objective function value. In the VRPs, the total distance traveled and the level of any constraint violation can be fitness functions. Analogous to biological processes, offspring with relatively good fitness levels are more likely to survive and reproduce with the expectation that fitness levels throughout the population will improve as they evolve. More details can be found in Reeves [1993].

2.2.3 Representation of a solution

The preliminary component involves choosing the right coding schema for the representation of solutions to the problem. Diverse encoding methods have been suggested for different problems to provide efficient implementation of GAs. Depending on symbol used for the bits of the chromosome, the encoding methods can be classified into:

- Binary encoding
- Real number encoding
- Integer of literal permutation encoding
- Data structure encoding

Binary encoding is the most common encoding method because it is easy to create and manipulate. A wide range of problems can use binary encoding, one point crossover and mutation without modification [Davis, 1987]. For efficiency, the other coding methods which are introduced in the following are more favorable in real world.

Real number encoding is appropriate for function optimization problems. It has been widely confirmed that the real number encoding performs better than binary encoding for optimization problems as Eshelman and Schaffer [1993], Michalewicz [1996], and Walters and Smith [1995] reported.

Integer or literal permutation encoding is useful for combinatorial optimization problems. Since the essence of combinatorial optimization problems is the search for a best permutation or combination of items subject to constraints, literal permutation encoding can be good way to be used for this type of problem. For more complex real world problems, an appropriate *data structure encoding* is suggested as the bits of a chromosome to capture the nature of the problem [Gen and Cheng, 2000].

According to the computer data structure, the encoding methods can be classified into two types: one-dimensional encoding and multidimensional encoding. In most practices, one-dimensional encoding has been widely used but some complex problems require multidimensional encodings. Cohoon and Paris [1986] used two-dimensional encoding for the circuit placement problems. Anderson et al. [1991] used a two-dimensional grid type of encoding.

2.2.4 Selection

The selection directs the genetic search toward promising regions in the solution space. The population diversity and the selective pressure are the two most important factors in the genetic search [Michalewicz, 1996]. An increase in the selective pressure decreases the population diversity and vice versa. Thus, they have a strong inverse relationship. Therefore, it is important to maintain the balance when determining a selection method for the GA.

Four commonly used selection methods are as follows:

- Roulette wheel selection
- Tournament selection
- Elitism
- Scaling

In *roulette wheel selection*, probability to be chosen is chromosome's fitness over the total fitness of the population. Each chromosome is assigned a slice of a circular roulette wheel, the size of the slice being proportional to the chromosome's fitness. The wheel is spun N times, where N is the number of chromosomes in the population. On each spin, the chromosome under the wheel's marker is selected to be in the pool of parents for the crossover.

Tournament selection method randomly chooses a set of chromosomes and picks out the best chromosome for reproduction. The number of chromosomes in a competition is called the tournament size. A common tournament size is two and this is called a binary tournament. A random number r is then generated between 0 and 1. If $r < k$, where k is a parameter between 0 and 1, then the fitter of the two chromosomes is selected to be a parent. Otherwise the less fit chromosome is selected. The two chromosomes are then returned to the original population for the next round of selection.

Elitism is an addition to other selection methods that forces the GA to retain number of good chromosomes in each generation. Without elitism, good chromosomes can be lost if they are not selected to reproduce or if they are destroyed by crossover or mutation.

Scaling method has been proposed to prevent quick convergence to local optima. The scaling method maps raw fitness values of all chromosomes in a population to scaled fitness values which are positive real values. The selection process will be performed based on scaled fitness values. There are many scaling methods proposed in the literature of the GA. Scaling parameters are known problem-dependent [Gen and Cheng, 2000]. One of the commonly used scaling methods in the GA is linear scaling which adjusts the fitness values of all chromosomes such that the best chromosome gets a fixed number of expected offspring and thus prevents it from reproducing too many [Gen and Cheng, 2000].

2.2.5 Crossover

An important genetic operator is crossover, which simulates a reproduction by parents. It works on a pair of solutions and recombines them in a certain way generating one or more offspring. The offspring share some of the characteristics from the parents through the crossover. In that way, the good characteristics are passed on to the following generations.

Many different crossover operators have been introduced in the literature. The functionality of the crossover depends on the representation, and the performance depends on how well it is adjusted to the problem. Commonly used crossover methods for VRPs are as follows [Gen and Cheng, 2000]:

- Point crossover (One-cut-point, Two-cut-point, Multi-cut-point)
- Partial-mapped crossover
- Order crossover
- Uniform crossover
- Position-based crossover
- Order-based crossover
- Cycle crossover

Among *point crossovers*, one-cut-point crossover is a simple method which selects one cut-point randomly in a chromosome as shown in Figure 2.2. The selected point is indicated by an arrow. P1 represents the first parent and P2 represents the second parent. In Figure 2.2, chromosomes consist of 9 genes. The one-cut-point crossover takes the pre-cut section of the first parent as a proto-child and fills up the offspring by taking in order each legitimate gene from the second parent to generate an offspring as shown in Figure 2.2.

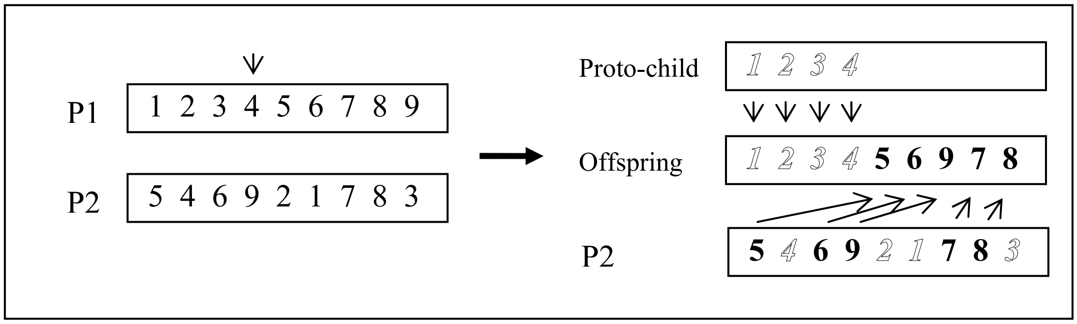

Figure 2.2 The one-cut-point crossover

The more advanced methods over one-cut-point crossover are two-cut-point and multi-cut-point crossovers. Two-cut-point crossover is illustrated in Figure 2.3, where two points are randomly selected in P1 and genes between two chosen points are inherited to the offspring. Then it takes each legitimate gene in the order shown up in P2. In multi-cut-point crossover method, the number of the cut-points at each step is randomly chosen, and then the cut-points are selected according to the chosen number. In Figure 2.4, the chosen number of cuts in 4 and cut sections are alternatively selected from P2. Then it takes each legitimate gene in the order shown up in P2.

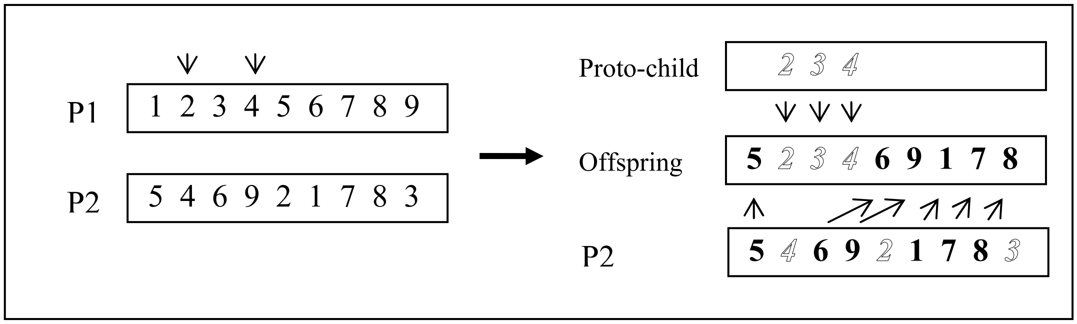

Figure 2.3 The two-cut-point crossover

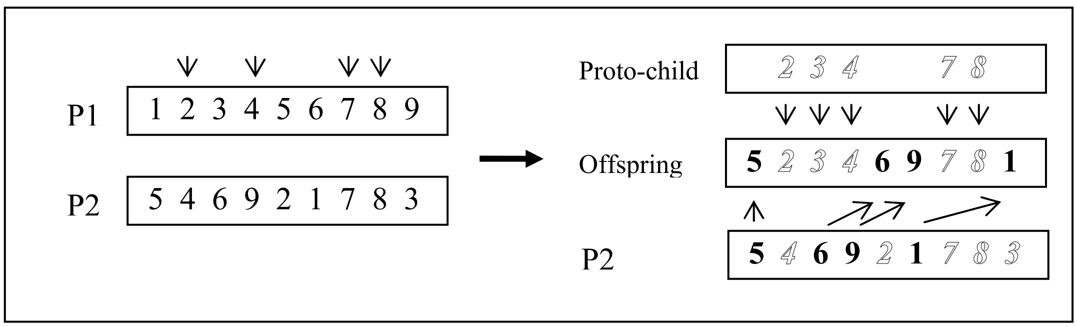

Figure 2.4 The multi-cut-point crossover

Partial-mapped crossover (PMX) is an extension of two-cut-point crossover to binary string representation for the permutation. The PMX uses a special repair procedure to resolve the illegitimacy caused by the two-cut-point crossover. The essentials of the PMX are a two-point crossover a repair procedure. The steps of PMX is detailed in Table 2.1 and illustrated in Figure 2.5.

Table 2.1 Steps of the partial-mapped crossover

Step 1.	Select two positions along the genes from both parents at random. The sub-genes defined by the two positions are called the mapping sections.
Step 2.	Copy the mapping section from P1 to proto-child at the same positions.
Step 3.	Determine the mapping relationship between two mapping sections.
Step 4.	Map the remaining genes from P2 using the mapping relationships. Genes without the mapping relationships is simply copied in the proto-child.

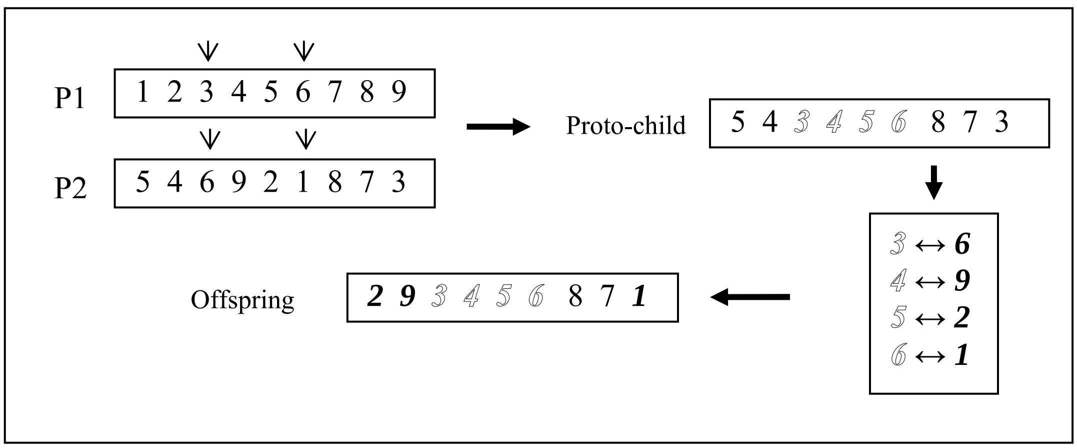

Figure 2.5 The partial-mapped crossover

Order crossover (OX) could be viewed as a variation of the PMX with a different repair procedure as detailed in Table 2.2 and illustrated in Figure 2.6.

Table 2.2 Steps of the order crossover

Step 1.	Select sub-genes from one parent by choosing two points randomly.
Step 2.	Produce a proto-child by copying the sub-genes into the corresponding positions of each gene.
Step 3.	Delete the corresponding genes from P2. The remaining genes from P2 contain the genes that the proto-child needs.
Step 4.	Place the remaining genes into the proto-child from left to right in the order of the sequence of genes in P2.

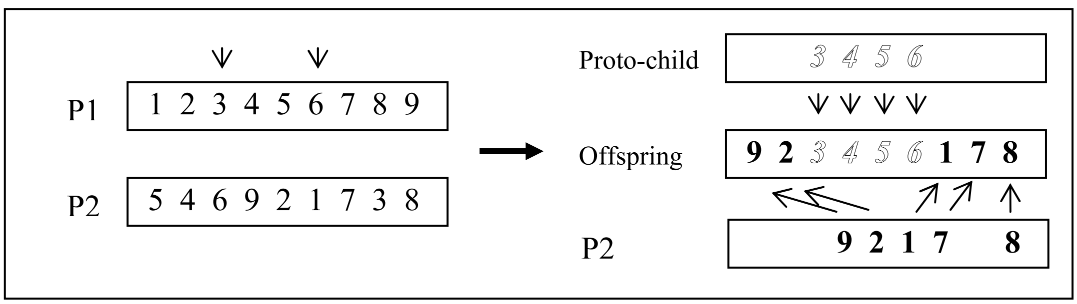

Figure 2.6 The order crossover

Uniform crossover is accomplished by selecting two parent solutions and randomly taking each gene from one parent to form the corresponding position of the child, as detailed in Table 2.3 and illustrated in Figure 2.7.

Table 2.3 Steps of the uniform crossover

Step 1.	Randomly take a gene from parents to form the corresponding gene of the offspring.
Step 2.	Repeat step 2 until the genes of the offspring fill up perfectly.

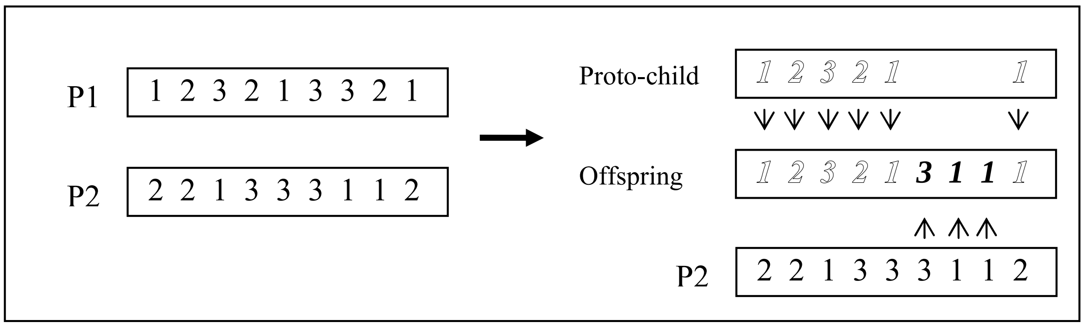

Figure 2.7 The uniform crossover

Position-based crossover is a variation of uniform crossover for permutation representation together with a repair procedure, which can also be viewed as a variation of the OX, where the genes are copied inconsecutively as detailed in Table 2.4 and illustrated in Figure 2.8.

Table 2.4 Steps of the position-based crossover

Step 1.	Select a set of positions from one parent at random.
Step 2.	Produce a proto-child by copying the genes on these positions into the corresponding positions of each gene.
Step 3.	Delete the corresponding genes from P2. The remaining genes from P2 contain the genes that the proto-child needs.
Step 4.	Place the remaining genes into the proto-child from left to right in the order of the sequence of genes in P2.

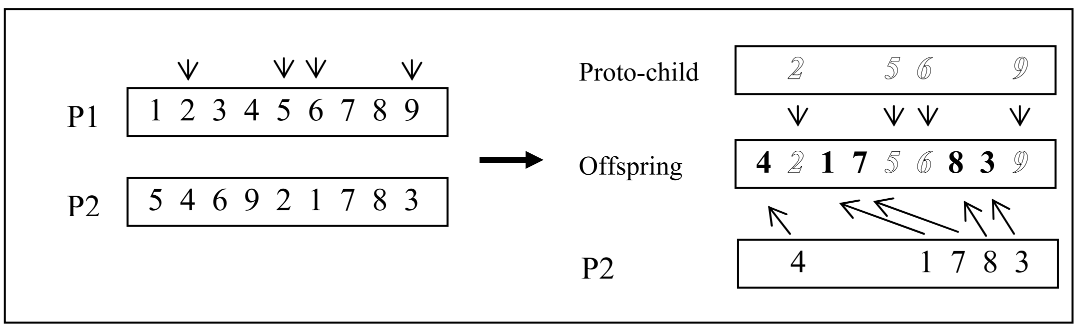

Figure 2.8 The position-based crossover

Order-based crossover is a slight variation of position-based crossover in which the order of genes at the selected position of one parent is imposed on the corresponding genes of the other parent as shown in Figure 2.9.

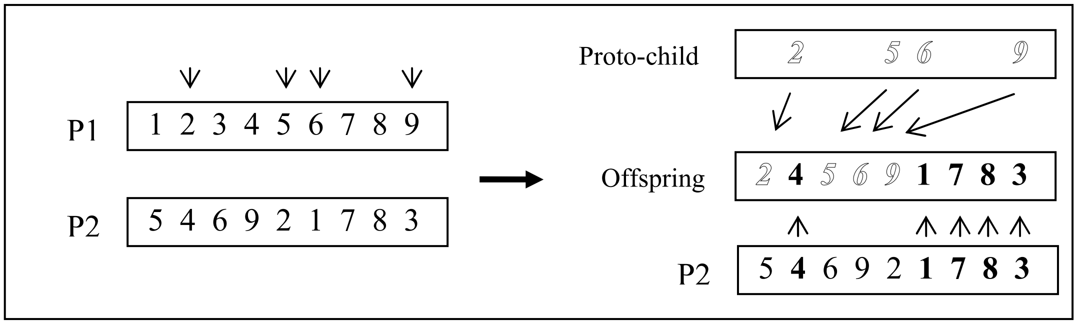

Figure 2.9 The order-based crossover

Cycle crossover (CX), as in the position-based crossover, takes some genes from one parent and selects the remaining genes from the other parent. The difference is that the genes from the first parent are not selected randomly, and only those genes which create a cycle according to the corresponding positions between parents must be selected as detailed in Table 2.5 and illustrated in Figure 2.10.

Table 2.5 Steps of the cycle crossover

Step 1.	Find the cycle which is defined by the corresponding positions of genes between parents.
Step 2.	Copy the genes in the cycle to offspring with the corresponding positions of second parent.
Step 3.	Determine the remaining genes in P2 by deleting those genes which are already in the cycle.
Step 4.	Fulfill the offspring with the remaining genes in P2.

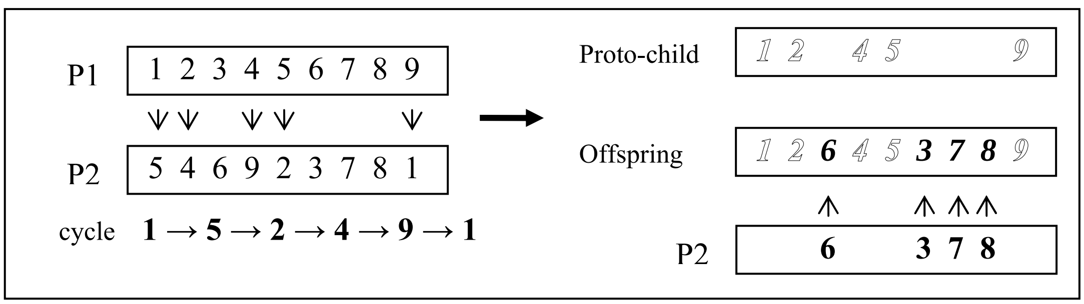

Figure 2.10 The cycle crossover

2.2.6 Mutation

To explore different solutions and avoid local optima, a mutation procedure needs to be implemented. In the GA, mutation plays an important role of either replacing the chromosomes lost from the population during the selection process so that they can be tried in a new context, or providing the chromosomes that were not present in the initial population.

Commonly used mutation methods are as follows [Gen and Cheng, 2000]:

- Inversion mutation
- Insertion mutation
- Reciprocal exchange mutation
- Point mutation

Inversion mutation selects two positions within a chromosome at random and then inverts the sub-genes between these two positions as illustrated in Figure 2.11.

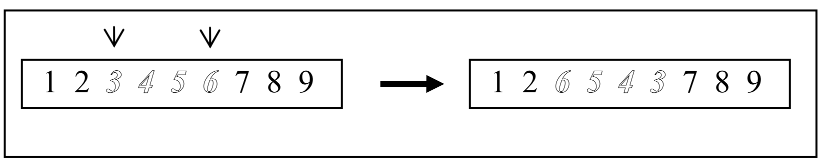

Figure 2.11 The inversion mutation

Insertion mutation selects a gene at random and inserts it in a random position as illustrated in Figure 2.12.

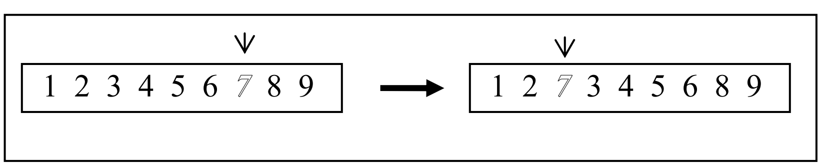

Figure 2.12 The insertion mutation

Reciprocal exchange mutation selects two positions at random and then swaps the genes on these positions as illustrated in Figure 2.13.

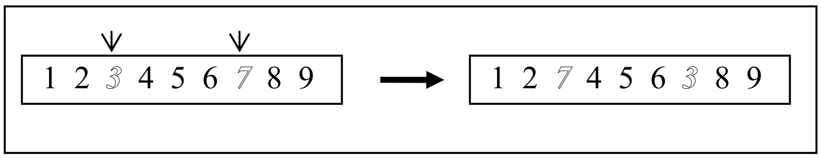

Figure 2.13 The reciprocal exchange mutation

Point mutation selects a position at random and changes the gene in the position to certain gene as illustrated in Figure 2.14.

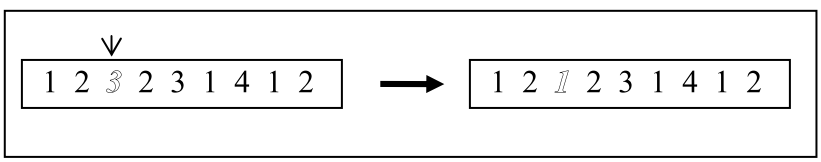

Figure 2.14 The point mutation

2.2.7 Termination condition

The GA continues to select parents, perform the crossovers, and execute the mutations until a termination criterion is met. The most frequently used stopping criterion is a maximum number of generations [Gen and Chang, 2000]. Another notable termination strategy involves population convergence criteria. The GA forces much of the entire population to converge to a single solution. When the sum of the deviations among individuals becomes smaller than a specified threshold, the algorithm is terminated. The algorithm can also be terminated due to a lack of improvement in the best solution over a specified number of generations.

For each criterion, a threshold needs to be carefully selected. Several strategies can be used in conjunction with each other.

2.3 Taguchi method

To fine-tune the performance of algorithms or processes, many parameters must be set carefully. The technique of investigating all possible combinations in experiment conditions involving multiple factors is known as *Design of Experiment*. The method of experimental designs constitutes the preset values of parameters to obtain the optimized output as it allows the designer to determine the significant parameters over the others. Taguchi method has been introduced to search effectively for the optimal parameters.

Taguchi method for parameter designs is an important tool for robust design. Robust design is an engineering methodology for optimizing the product and process conditions which are minimally sensitive to the causes of variation, and which produce high-quality products with low development and manufacturing costs. The orthogonal array and the signal-to-noise ratio (SNR) are two major tools used in the Taguchi method. Additional details can be found in the books presented by Taguchi et al. [2000] and Wu [2000].

Many designed experiments use matrices called orthogonal arrays for determining which combinations of factor levels to use for each experimental run. An orthogonal array is a fractional factorial matrix, which assures a balanced comparison of levels of any factor. It is a matrix of numbers arranged in rows and columns where each row represents the level of the factors in each run, and each column represents a specific factor that can be changed from each run. The symbol of three-level orthogonal arrays is $L_n(3^k)$, where n is the number of experimental runs, 3 is the number of levels for each factor, and k is the number of factors. The letter L comes from Latin, since the orthogonal arrays were associated with Latin square designs from the outset.

SNR is the ratio of the signal over the noise, which measures the strength of signal with the existence of noises. The higher SNR means that the process or design is more robust. There are several SNRs available depending on the type of characteristic: nominal-is-best, smaller-the-better or larger-the-better. Further details can be found in Taguchi et al. [2000] and Wu [2000]. In the case of smaller-the-better characteristic, suppose that we have a set of experiment runs

$x_1, x_2, ..., x_n$. Since the value of the SNR is large for favorable situations, the following formulation for the smaller-the-better characteristic is used:

$$\text{SNR} = -10\log\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right).$$

Since the objective of VRPs is to minimize the total traveled distance, the smaller-the-better is an appropriate measure in this thesis. The proposed GA with different process parameters shows different performances.

Chapter 3. Methodology

This chapter describes the methodology followed to develop the proposed GA. In Section 3.1, the mathematical model of the problem is developed. Section 3.2 discusses the motivation to develop a GA and describes the details about the components of the proposed GA.

3.1 Mathematical model

The MIP model for the VRP with heterogeneous vehicles from multiple visits, allowing multiple depots has been developed and solved by CPLEX version 7.1. The modeling for CPLEX has been completed using OPL-Studio version 3.5.

3.1.1 Model assumptions

The VRP can be represented as a network, where nodes are customers or depots and the links are the roads. In the network there are N customers with known demand D_i ($i = 1,\ldots, N$), and M depots, each of which has T_m ($m = N+1,\ldots, N+M$) vehicles. Each vehicle might have heterogeneous capacities. The complete assumptions are detailed in the following.

- Each vehicle must start and end its route at a depot.

• Demand of each customer must be satisfied by vehicles in an allowed number of visits.

• The capacity of each vehicle is known.

• The sum of the unloaded amounts at the customers by a vehicle must not exceed the capacity of the vehicle.

• The location of all customers and depots are known.

• The distances between all pairs of two locations are known.

3.1.2 Notations

S_N	A set of customer indices
S_M	A set of depot indices
S	A set of indices for all customers and depots; $S = \{S_N \cup S_M\}$
S_{T_m}	A set of all vehicle indices at depot m
N	Number of customers
M	Number of depots
L	Number of customers and depots ($L = N+M$)
T_m	Number of vehicles at depot m
D_i	Demand of customer i ($1 \leq i \leq N$)
d_{ij}	Distance between nodes i and j ($1 \leq i, j \leq N+M$)
V	Number of visits allowed to each customer

U_{jmt} Unloaded amount by vehicle t from depot m at customer j, where $1 \le j \le N$.

($U_{jmt} = 0$ for $N+1 \le j \le N+M$, $U_{jmt} = 0$ for $t \notin S_{T_m}$)

C_{mt} Capacity of vehicle t from depot m for $t \in S_{T_m}$

B A large number

$$x_{ijmt} \begin{cases} 1, \text{ if the vehicle } t \text{ from depot } m \text{ travels from node } i \text{ to } j, \text{ where } t \in S_{T_m} \\ 0, \text{ otherwise} \end{cases}$$

3.1.3 MIP model

The VRP with heterogeneous vehicles from multiple visits, allowing multiple depots is formulated as a MIP model. Thus the problem can be modeled as:

$$\text{Minimize} \qquad Z = \sum_{i=1}^{L} \sum_{j=1}^{L} \sum_{m=1}^{M} \sum_{t=1}^{T_m} d_{ij} x_{ijmt}$$

Subject to:

$$x_{ijmt} = 0 \qquad \text{if } i \ne j, \text{ for } \forall i \in S_M, \forall j \in S_M, \forall m \in S_M, \forall t \in S_{T_m} \qquad (1)$$

$$\sum_{i=1}^{L} x_{ijmt} = \sum_{i=1}^{L} x_{jimt} \qquad \text{for } \forall j \in S, \forall m \in S_M, \forall t \in S_{T_m} \qquad (2)$$

$$\sum_{t=1}^{T_m}\sum_{m=1}^{M}\sum_{i=1}^{L} x_{ijmt} \le V \qquad \text{for } \forall j \in S_N \tag{3}$$

$$\sum_{i=1}^{L} x_{iimt} = 0 \qquad \text{for } \forall m \in S_M, \forall t \in S_{T_m} \tag{4}$$

$$B\sum_{i=1}^{L} x_{ijmt} \ge U_{jmt} \qquad \text{for } \forall j \in S_N, \forall m \in S_M, \forall t \in S_{T_m} \tag{5}$$

$$x_{ijmt} \le U_{jmt} \qquad \text{for } \forall i \in S, \forall j \in S_N, \forall m \in S_M, \forall t \in S_{T_m} \tag{6}$$

$$\sum_{t=1}^{T_m}\sum_{m=1}^{M} U_{jmt} = D_j \qquad \text{for } \forall j \in S_N, \forall m \in S_M, \forall t \in S_{T_m} \tag{7}$$

$$\sum_{j=1}^{N} U_{jmt} \le C_{mt} \qquad \text{for } \forall m \in S_M, \forall t \in S_{T_m} \tag{8}$$

$$y_{imt} - y_{jmt} + Lx_{ijmt} \le L - 1$$

$$\text{if } i \ne j, \text{ for } \forall i \in S_N, \forall j \in S_N, \forall m \in S_M, \forall t \in S_{T_m} \tag{9}$$

$$x_{ijmt} = \{0, 1\} \qquad \text{for } \forall i \in S, \forall j \in S, \forall m \in S_M, \forall t \in S_{T_m} \tag{10}$$

The objective function of the MIP model is to minimize the total traveled distance by all vehicles to satisfy all customers' demand. Constraint (1) ensures that each vehicle t starts from its origin depot m and terminates its route at the same depot. In other words, each vehicle cannot visit the depots other than its origin depot. Constraint (2) ensures that all vehicles visiting a node must leave that node. At each node, the number of visits must be the same as the number of departures for each vehicle. It ensures the continuous flow of vehicles in the network. Constraint (3) ensures that each customer node can have up to V visits by all vehicles in order to satisfy the customer demand. Constraint (4) prevents the looping of any vehicle at a node. Constraints (5) and (6) ensure that if vehicle t from depot m travels from node i to node j, the vehicle should unload U_{jmt} at the node j. Constraint (7) ensures that the sum of the unloaded amounts at a customer node j should be the same as the demand of customer j. Constraint (8) ensures that the total unloaded amounts of each vehicle over its route cannot exceed the vehicle capacity. Constraint (9) presents the sub-tour elimination constraint typically used in the VRPs.

3.2 The proposed GA

The VRP is known as a NP-hard combinatorial problem. It is difficult to solve even small problems optimally in a reasonable amount of time. The GA has been applied successfully in many combinatorial optimization problems. The GA does not guarantee the optimality because of its stochastic nature, but it finds a good near-optimal solution in significantly less time. In the following subsections, the proposed GA implemented in this thesis is described in detail.

3.2.1 Representation of feasible solutions

Encoding a solution of the problem into a chromosome has a high impact on the GA. Integer encoding has been identified as the most suitable method for the problem. The developed representation is a 2-dimentional matrix ((V+1)×N), where the columns represent customers, the first row contains randomly-generated sequences of visiting order and the other rows contain the vehicles visiting each customer. A representation of a chromosome is illustrated in Figure 3.1, where V = 3 and N = 5.

C_1	C_2	C_3	C_4	C_5
3	1	4	5	2
V_{11}	V_{12}	V_{21}	V_{11}	V_{12}
0	V_{21}	V_{11}	0	V_{31}
V_{12}	0	V_{13}	0	0

Figure 3.1 Representation of a chromosome in the VRP, where 5 customers and 3 vehicles in depot 1 (V_{11}, V_{12} and V_{13}), 1 vehicle in depot 2 (V_{21}) and 1 vehicle in depot 3 (V_{31}), allowing 3 visits to a customer

The way to interpret the representation in Figure 3.1 is in the following. There are 3 vehicles in depot 1 (V_{11}, V_{12} and V_{13}), 1 vehicle in depot 2 (V_{21}) and 1 vehicle in depot 3 (V_{31}). The vehicles visit the five customers, C_1, C_2, C_3, C_4 and C_5. The V_{11} will visit C_1, C_3 and C_4, respectively. The visiting order of the V_{11} depends on the values in cells of the first row. Since the corresponding values for C_1, C_3 and C_4 are 3, 4 and 5, respectively, the route of V_{11} is $[D_1 - C_1 - C_3 - C_4 - D_1]$ where D_1 represent the depot 1. In the same manner, the route of V_{12} is $[D_1 - C_2 - C_5 - C_1 - D_1]$, the route of V_{13} is $[D_1 - C_3 - D_1]$, the route of V_{21} is $[D_2 - C_2 - C_3 - D_2]$ and the route of V_{31} is $[D_3 - C_5 - D_3]$ where D_2 and D_3 represent depots 2 and 3, respectively.

3.2.2 Population initialization and evaluation function

The initial population of the predetermined size is randomly generated. However, a way to obtain the good initial population is desired since it impacts on the performance of the proposed GA.

The objective of the proposed GA is to minimize the overall distance traveled by all vehicles. The fitness, which is the survival chance of a feasible solution satisfying all customer demands, is calculated as follows:

$$\text{fitness} = \frac{1}{\text{sum of the total distance traveled by all vehicles}}$$

The fitness is an inverse of sum of the total distance traveled by all vehicles.

3.2.3 Selection

In the GA, an appropriate method to select chromosomes for the crossover must be employed to give more chance to those chromosomes in a population that are most fit. With too much chance, genetic search will terminate prematurely; with too little chance, evolutionary progress will be slower than necessary. Typically, a lower selection pressure is desirable at the start of the genetic search in favor of a wide exploration of the search space, while a higher selection pressure is recommended at the end to converge efficiently. The roulette wheel selection method and linear scaling method has been used during the selection process in the proposed GA. The roulette wheel selection method is known as the best selection method [Gen and Cheng, 2000]. Linear scaling method is commonly used in GAs as mentioned in Chapter 2. Based on the several runs of the proposed GA, the linear scaling function in the proposed GA has been chosen as $f'_i = 0.1 \times f_i + 1$, where f'_i is the scaled fitness and f_i is the raw fitness for chromosome i.

3.2.4 Crossover

Through the selection method described in Subsection 3.2.1, two chromosomes from the current population are selected for the mating by means of crossover rate which is a probability of crossover. If a randomly generated number between 0 and 1 is smaller than crossover rate, these chromosomes reproduce to form new members to be included in the next generation. Otherwise, the crossover does not take place. The newly generated members are called offspring. An appropriate method to crossover two selected chromosomes to improve the fitness of offspring has been developed for the VRP under consideration.

In the previous studies for VRPs, various crossover operators have been used. In this thesis, the *position-based crossover* method is applied to the first row. The cells in the first row represent the visiting order of vehicles, thus they cannot have the same gene. *Uniform crossover*, which has been shown to be superior to traditional crossover strategies for combinatorial problems [Syswerda, 1989], is applied to the other rows for visiting vehicles. These crossovers are described in detail in Table 3.1.

Table 3.1 Steps of the crossover in the proposed GA

Step 1.	Select a set of cells from Parent 1 randomly, which are shaded in Figure 3.2.
Step 2.	Produce a proto-child by copying the shaded cells from parent 1 into the corresponding cells. Delete the corresponding cells from Parent 2 (see Figure 3.3).
Step 3.	For the first row, copy the remaining cells from Parent 2 into the empty cells of the proto-child from left to right in the order shown up the first row of Parent 2. Finally, copy the remaining cells from Parent 2 into the empty cells of the proto-child at the other rows until all cells of the proto-child fill up (see Figure 3.4).

Parent 1

1	4	5	2	3
V_{21}	V_{22}	V_{11}	V_{11}	V_{12}
V_{12}	V_{13}	0	0	V_{13}
0	V_{12}	V_{22}	0	V_{21}

Parent 2

3	5	4	1	2
V_{11}	V_{22}	V_{12}	V_{11}	V_{13}
0	V_{12}	0	V_{22}	0
0	V_{21}	V_{21}	0	V_{11}

Figure 3.2 Step 1 of the crossover in the proposed GA

Proto-child

	4		2	
V_{21}	V_{22}			V_{12}
	V_{13}	0	0	
0		V_{22}		V_{21}

Parent 2

3	5		1	
		V_{12}	V_{11}	
0				0
	V_{21}		0	

Figure 3.3 Step 2 of the crossover in the proposed GA

Offspring

3	4	5	2	1
V_{21}	V_{22}	V_{12}	V_{11}	V_{12}
0	V_{13}	0	0	0
0	V_{21}	V_{22}	0	V_{21}

Figure 3.4 Step 3 of the crossover in the proposed GA

During the crossover procedure, except the first row, some offspring might have identical vehicles in the same columns after being generated. However, these offspring has been eliminated since they are infeasible solutions.

3.2.5 Mutation

Mutation is another important operator in GA implementations and is applied to a chromosome with a mutation rate which is a probability of mutation of each chromosome in a population. Mutation operator brings random changes onto a single chromosome. If a randomly generated number between 0 and 1 is smaller than mutation rate, these chromosomes reproduce to form new members to be included in the next generation. Otherwise, the mutation does not take place. These random changes prevent the premature local convergence.

It is relatively simple to implement a mutation procedure. All chromosomes in the population except the good ones (thanks to elitism) are subject to mutation at the mutation rate. The elite rate is set to 10%, which is the percentage of good solutions immune to the mutation. The mutation rate has been dynamically adapted, based on the status of population. It starts with 0.05 and it increases up to 0.75 by 0.1 whenever no improvement observed in the best chromosome over a certain number of generations (50 ~ 100 generations). If the best solution improves, the mutation rate drops to the original mutation rate, i.e., 0.05. While most GA implementation uses the static mutation rate, the proposed GA introduces the dynamic self-adapting mutation rate. It is expected to increase the capability to escape from the premature local optima and to search for better solutions from the diverse directions. This policy has been proved effective in solving the various VRPs. See Section 4.3. The higher mutation rate introduces more changes of chromosomes in the population. In this thesis, the *inversion mutation* is implemented. Table 3.2 explains steps of the mutation. Each step is illustrated in Figures 3.5 and 3.6.

Table 3.2 Steps of the mutation in the proposed GA

Step 1.	Select two columns in a chromosome at random as indicated by two arrows in Figure 3.5.
Step 2.	Invert the corresponding cells between these two columns except the cells on the first row as shaded in Figure 3.6.

3	4	5	2	1
V_{21}	V_{22}	V_{12}	V_{11}	V_{11}
0	V_{13}	0	0	0
0	V_{21}	V_{22}	0	V_{21}

Figure 3.5 Step 1 of the mutation in the proposed GA

3	4	5	2	1
V_{21}	V_{22}	V_{11}	V_{12}	V_{11}
0	V_{13}	0	0	0
0	V_{21}	0	V_{22}	V_{21}

Figure 3.6 Step 2 of the mutation in the proposed GA

3.2.6 Termination condition

In this thesis, two strategies have been used as the termination criteria. One is that the proposed GA completes a specified maximum number of generations, which is 5,000 in this implementation. The other is that the proposed GA can also be terminated due to no improvement of the best solution over a specified number of generations, which is between 50 and 100 while the mutation rate can be varied between 0.05 and 0.75. This number of generations is called *improvement interval*. After the proposed GA terminates, the chromosome with the highest fitness is interpreted as a best known solution for that execution.

3.2.7 GA process parameters

Many GA process parameters have been defined to effectively solve the VRP by using the proposed GA. The values of those process parameters need to be carefully selected. Good process parameter setting is important for the GA to obtain a good final solution. Usually, it is difficult to determine a good set of process parameters because the relationships among them can be rather complicated and unclear. The values of the process parameters in the previous works were referred initially to set the values for the proposed GA. See Table 3.3. The proposed GA has been tested under different values of process parameters to solve the VRPs in Sections 4.1 and 4.2.

Table 3.3 Values of parameters used for the proposed GA

Parameter	Values
Population size	100, 150, 200
Improvement interval	50, 75, 100
Crossover rate	0.6, 0.7, 0.8
Mutation rate	0.05 ~ 0.75
Elitist rate	0.1

Three different population sizes, three different crossover rate, three different numbers of generations for dynamic mutation rate have been tested. Assuming that the number of generations for dynamic mutation rate is 50, the proposed GA terminates when there is no improvement over 400 generations with mutation rate changing from 0.05 to 0.75 every 50 generation. Whenever the best solution in a population is found, the mutation rate returns to the initial mutation rate, 0.05. Elitist rate is used to retain 10% of the best chromosomes at each generation. Such chromosomes can not be lost since they survive automatically being immune to mutation.

Chapter 4. Computational results

In this chapter computational results of the MIP model and the genetic algorithm proposed in Chapter 3 are presented. All computational experiments are carried out on a Dell PC with 3.4 GHz CPU and 2.0 GB RAM. The MIP model is solved with the optimization software, CPLEX version 7.1, which is branch-and-bound MIP solver. The program for the proposed GA is implemented in C++ programming language using the Microsoft Visual Studio.NET Framework 1.1 version.

4.1 Effectiveness of the proposed GA

4.1.1 Test problem

Since the VRP with heterogeneous vehicles from multiple depots allowing multiple visits has been solved for the first time, the hypothetical problem shown in Figure 4.1 has been created to demonstrate the effectiveness of the proposed GA. The problem has 6 customers with known demand at the specified locations and two depots with 3 and 2 vehicles, respectively, of heterogeneous capacity at the specified locations. The coordinates and demand of the customers are shown in Table 4.1. Table 4.2 shows the depots' coordinates and vehicles with their capacities.

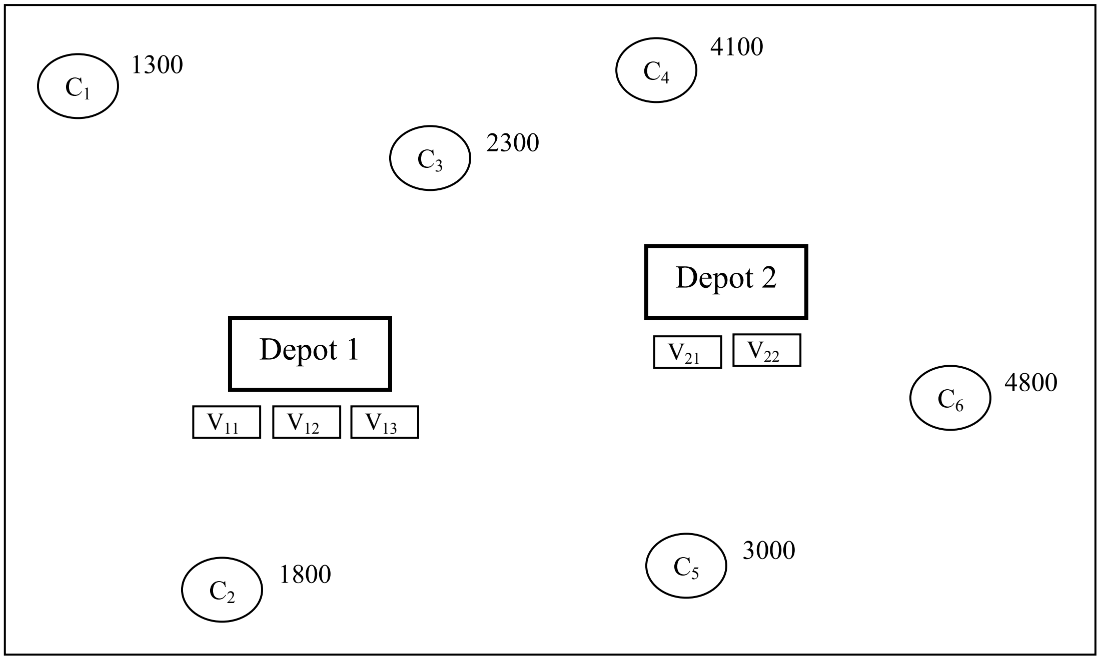

Figure 4.1 Illustration of the hypothetical problem

Table 4.1 Coordinates and demand of the customers

Node	Coordinates	Demand	Node	Coordinates	Demand
Customer 1 (C_1)	(9,94)	1300	Customer 4 (C_4)	(31,144)	4100
Customer 2 (C_2)	(19,62)	1800	Customer 5 (C_5)	(35,65)	3000
Customer 3 (C_3)	(26,126)	2300	Customer 6 (C_6)	(44,88)	4800

Table 4.2 Coordinates and vehicles with their capacities in the depots

	Depot 1			Depot 2	
Coordinates	(21, 86)			(36, 97)	
Vehicle	V_{11}	V_{12}	V_{13}	V_{21}	V_{22}
Vehicle capacity	1500	4800	8000	2200	2500

The customers and the depots with vehicles are illustrated in Figure 4.1. In the figure, the circled C_i stands for customer i. The numbers on the shoulder of the customers stand for the customers' demands. Three vehicles V_{11}, V_{12}, and V_{13} are housed in depot 1 and two vehicles V_{21} and V_{22} in depot 2.

4.1.2 Comparison of the results from the MIP model and the proposed GA

The test results from the MIP model presented in Chapter 3 are shown in Table 4.3. The numbers in the parenthesis are the unloaded amounts for the customer's demand by the corresponding vehicle in Tables 4.3 and 4.4. With multiple visits to a customer the total distances has been reduced as shown in Tables 4.3 and 4.4.

Table 4.3 Results from the MIP model

		Allowed visits by each vehicle		
		One	Two	Three
Total distance		358.77	300.67	263.68
Routes for the vehicles	V_{11}	$D_1 \rightarrow C_1$ (1300) $\rightarrow D_1$	Unused	Unused
	V_{12}	$D_1 \rightarrow C_4$ (4100) $\rightarrow D_1$	$D_1 \rightarrow C_2$ (1800) $\rightarrow C_5$ (3000) $\rightarrow D_1$	$D_1 \rightarrow C_2$ (1800) $\rightarrow C_5$ (3000) $\rightarrow D_1$
	V_{13}	$D_1 \rightarrow C_5$ (3000) $\rightarrow C_6$ (4800) $\rightarrow D_1$	$D_1 \rightarrow C_6$ (2600) $\rightarrow C_4$ (4100) $\rightarrow C_1$ (1300) $\rightarrow D_1$	**$D_1 \rightarrow C_1$ (1300) $\rightarrow C_3$ (2300) $\rightarrow C_4$ (4100) $\rightarrow C_6$ (300) $\rightarrow D_1$**
	V_{21}	$D_2 \rightarrow C_2$ (1800) $\rightarrow D_2$	$D_2 \rightarrow C_6$ (2200) $\rightarrow D_2$	$D_2 \rightarrow C_6$ (2200) $\rightarrow D_2$
	V_{22}	$D_2 \rightarrow C_3$ (2300) $\rightarrow D_2$	$D_2 \rightarrow C_3$ (2300) $\rightarrow D_2$	**$D_2 \rightarrow C_6$ (2300) $\rightarrow D_2$**

The test results from the proposed GA are shown in Table 4.4. It is shown that the total distances of the MIP model and the proposed GA are identical, which indicates that the proposed GA achieves the optimality for the test problem in Subsection 4.1.1. Note that the routes in bold show the different amounts of demands satisfied by different vehicles from the MIP model and the proposed GA, even if the vehicle routes are identical. These are alternative optimal solutions. The CPLEX terminates the branch-and-bounds when it finds the first optimal solution.

Table 4.4 Results from the proposed GA

		Allowed visits by each vehicle		
		One	Two	Three
Total distance		358.77	300.67	263.68
Routes for the vehicles	V_{11}	$D_1 \rightarrow C_1$ (1300) $\rightarrow D_1$	Unused	Unused
	V_{12}	$D_1 \rightarrow C_4$ (4100) $\rightarrow D_1$	$D_1 \rightarrow C_2$ (1800) $\rightarrow C_5$ (3000) $\rightarrow D_1$	$D_1 \rightarrow C_2$ (1800) $\rightarrow C_5$ (3000) $\rightarrow D_1$
	V_{13}	$D_1 \rightarrow C_5$ (3000) $\rightarrow C_6$ (4800) $\rightarrow D_1$	$D_1 \rightarrow C_6$ (2600) $\rightarrow C_4$ (4100) $\rightarrow C_1$ (1300) $\rightarrow D_1$	**$D_1 \rightarrow C_1$ (1300) $\rightarrow C_3$ (2300) $\rightarrow C_4$ (4100) $\rightarrow C_6$ (100) $\rightarrow D_1$**
	V_{21}	$D_2 \rightarrow C_2$ (1800) $\rightarrow D_2$	$D_2 \rightarrow C_6$ (2200) $\rightarrow D_2$	$D_2 \rightarrow C_6$ (2200) $\rightarrow D_2$
	V_{22}	$D_2 \rightarrow C_3$ (2300) $\rightarrow D_2$	$D_2 \rightarrow C_3$ (2300) $\rightarrow D_2$	**$D_2 \rightarrow C_6$ (2500) $\rightarrow D_2$**

From Tables 4.3 and 4.4, the proposed GA is effective in solving the VRP with heterogeneous vehicles from multiple depots, allowing multiple visits while the MIP model can only solve the small problem. CPLEX cannot solve the problem with 7 customers and 2 depots due to the memory limitation. Note that the increase in the number of visits leads to the shorter travel distance. In other words, to minimize the total distance by all heterogeneous vehicles from multiple depots, allowing multiple visits to customers is a good way to reduce the delivery cost in the case of the generalized VRPs. The vehicle routes with different allowed visits from the MIP model are illustrated in Figures 4.2, 4.3 and 4.4.

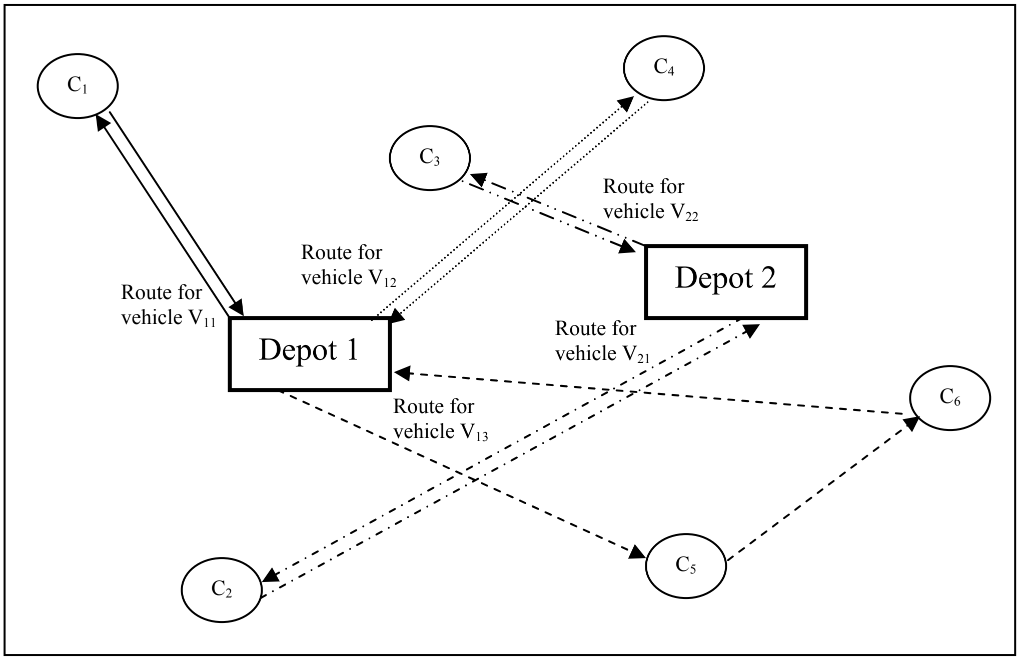

Figure 4.2 Vehicle routes for one allowed visit

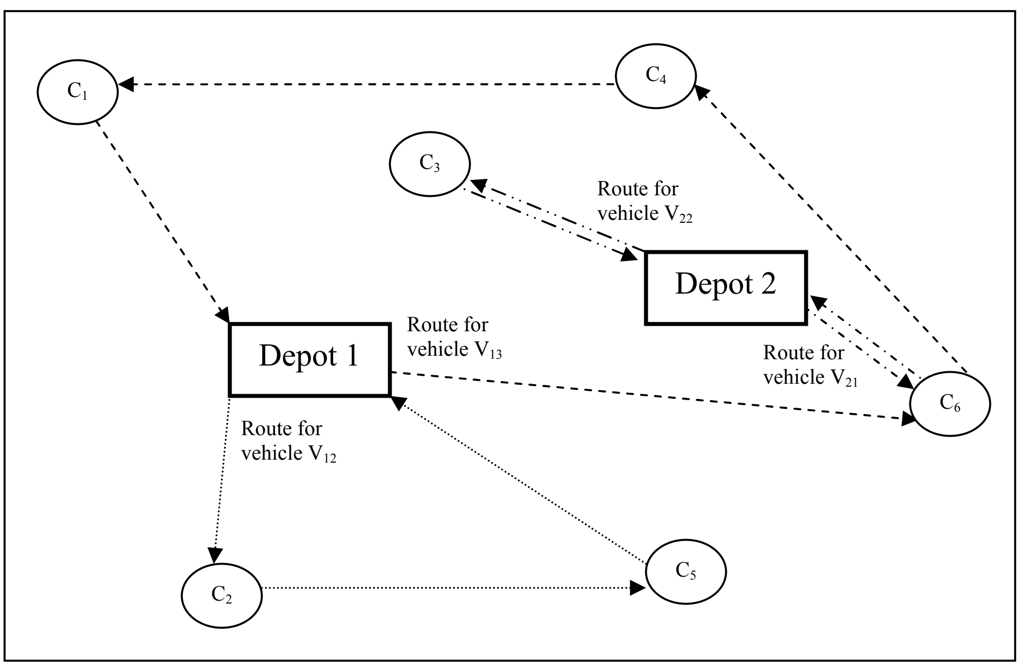

Figure 4.3 Vehicle routes for two allowed visits

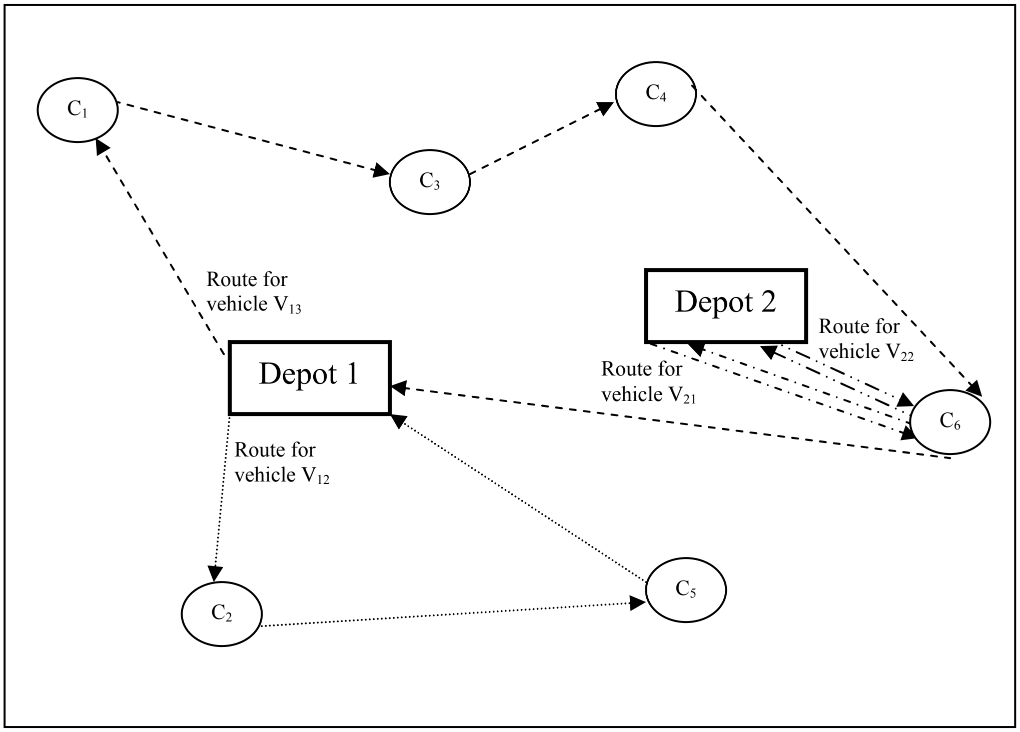

Figure 4.4 Vehicle routes for three allowed visits

In the figures, routes for different vehicles are represented by different arrows. For example, vehicles V_{11}, V_{12}, V_{21} and V_{22} only visit C_1, C_4, C_2 and C_3, respectively, shown in Figure 4.2. Vehicle V_{13} visits C_5 and C_6 in order.

4.2 Performance comparisons with prior works

In this subsection, the proposed GA has been applied to benchmark VRPs available at the VRPLIB repository on the web (http://www.or.deis.unibo.it /research_pages/ORinstances/VRPLIB/VRPLIB.html). These problems have been widely used as benchmarks and they are derived from Eilon et al. [1971].

4.2.1 The benchmark problems

The VRPs examined have only a single depot and a number of vehicles with identical capacity. Table 4.5 shows the number of customers, the number of vehicles at a single depot, the vehicle capacity and the best known solution in the VRPLIB repository. Customers and a single depot are in a network, and all distances are calculated using the Euclidean coordinates.

Table 4.5 Best known solutions of the benchmark problems

Problem	Customers	Vehicles	Vehicle capacity	Best known solution
*E-n22-k4	21	4	6000	375
E-n23-k3	22	3	4500	569
E-n30-k4	29	3	4500	534
E-n33-k4	32	4	8000	839

*E-n22-k4 stands for Elion et al. problem which has 22 nodes (21 customers, 1 depot) and 4 vehicles at the depot.

4.2.2 The results of the VRP examples

Table 4.6 compares the performance of the best solutions obtained by the proposed GA in comparison with best known solutions. Note that the improvement is calculated as: Improvement = (-1) $\times$ [(The best solution from the proposed GA – Best known value)/Best known value] $\times$ 100.

Table 4.6 Results of the benchmark problems from the proposed GA

Problem	Best known solution	The best solution from the proposed GA	Improvement
E-n22-k4	375	375	0
E-n23-k3	569	569	0
E-n30-k3	534	534	0
E-n33-k4	839	837	0.24

From Improvement in Table 4.6, it is known that the proposed GA is effective in solving VRPs especially for the problem with a large number of customers. The proposed GA achieves or outperforms the best known solutions. For E-n33-k4 with 32 customers and 4 vehicles, the total traveled distance of the proposed GA is 0.24% shorter than the best known solution. The proposed GA does solve not only the VRP with heterogeneous vehicles from multiple depots, allowing multiple visits to customers, but also the classical VRP with a single depot. Route of all vehicles in the benchmark problems are reported in Table 4.7 for the archival purpose.

Table 4.7 Vehicle routes of the benchmark problems from the proposed GA

Problem	Vehicle routes
E-n22-k4	*D - *C_{17} (1000) - C_{20} (1800) - C_{18} (900) - C_{15} (900) - C_{12} (1300) – D D - C_6 (400) - C_1 (1100) - C_2 (700) - C_5 (2100) - C_7 (800) - C_9 (500) - D D - C_{10} (600) - C_8 (100) - C_3 (800) - C_4 (1400) - C_{11} (1200) - C_{13} (1300) - D D - C_{14} (300) - C_{21} (700) - C_{19} (2500) - C_{16} (2100) - D
E-n23-k3	D - C_{12} (300) - C_{11} (225) - C_6 (175) - C_1 (125) - C_2 (84) - C_3 (60) - C_{16} (100) - C_{15} (150) - C_{14} (500) - C_{17} (250) - C_{22} (75) - C_{20} (500) - C_{19} (600) - C_{18} (120) - D D - C_{10} (4100) - C_{13} (250) - D D - C_7 (350) - C_9 (1100) - C_8 (150) - C_5 (300) - C_4 (500) - C_{21} (175) - D
E-n30-k3	D - C_{19} (400) - C_{15} (550) - C_{16} (150) - C_{13} (150) - C_7 (150) - C_{17} (100) - C_9 (300) - C_{14} (150) - C_8 (450) - C_{12} (125) - C_{11} (950) - C_{10} (100) - C_{23} (300) - C_{18} (150) - D D - C_{21} (1500) - C_6 (150) - C_{24} (500) - C_{25} (800) - C_{29} (1000) - C_{27} (100) - C_{28} (150) - C_{26} (300) - D D - C_{22} (100) - C_2 (3100) - C_5 (200) - C_1 (300) - C_4 (100) - C_3 (125) - C_{20} (300) -D
E-n33-k4	D - C_{13} (250) - C_{17} (550) - C_{25} (1400) - C_{24} (750) - C_{23} (700) - C_{20} (400) - C_{22} (1300) - C_{21} (300) - C_{19} (200) -C_{18} (650) - C_{10} (750) - C_3 (400) -D D - C_2 (400) - C_{12} (150) - C_{11} (1500) - C_{32} (1100) - C_8 (900) - C_9 (600) - C_7 (2000) - C_6 (80) - C_5 (40) - C_4 (1200) - D D - C_{30} (2500) - C_{14} (1600) - C_{31} (1700) - D D - C_1 (700) - C_{15} (450) - C_{26} (4000) - C_{27} (600) - C_{28} (1000) - C_{16} (700) - C_{29} (500) - D

*D stands for the depot in the problem.
*C_{17} (1000) stands for customer 17 and its demand, 1000.

4.3 Optimization of process parameters in the proposed GA using Taguchi method

Based on the preliminary tests in Sections 4.1 and 4.2, four process parameters are identified as important design factors for the performance of the proposed GA; population size, crossover rate, improvement interval, and mutation policy. For each process parameter design factor, based on the related research, three possible levels of the process parameters are considered: (100, 150, 200) for the population size, (0.6, 0.7, 0.8) for the crossover rate, (50, 75, 100) for the improvement interval, and (MP 1, MP 2, MP 3) for the mutation policy. In MP 1, mutation rate is fixed at 0.05. In MP 2, mutation rate linearly increases of the mutation self-adapting as described in Subsections 3.2.5, 3.2.6 and 3.2.7. MP 3 uses logarithmic increase of the mutation rate. The logarithmic increase of MP 3 instead of linear increase in MP 2 (possible mutation rates are 0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65 and 0.75) can be calculated as

$$A_n = 0.05 + \ln(1+0.15n),$$

where n= {1, 2, …, 8} (possible mutation rate can be 0.05, 0.19, 0.31, 0.42, 0.52, 0.61, 0.69 and 0.77). Therefore, the minimum and the maximum of the mutation rate in MP 3 are 0.05 and 0.77 which are close to ones of the MP 2. Figure 4.5 shows mutation rates over 8 improvement intervals for MP 1, MP 2 and MP 3.

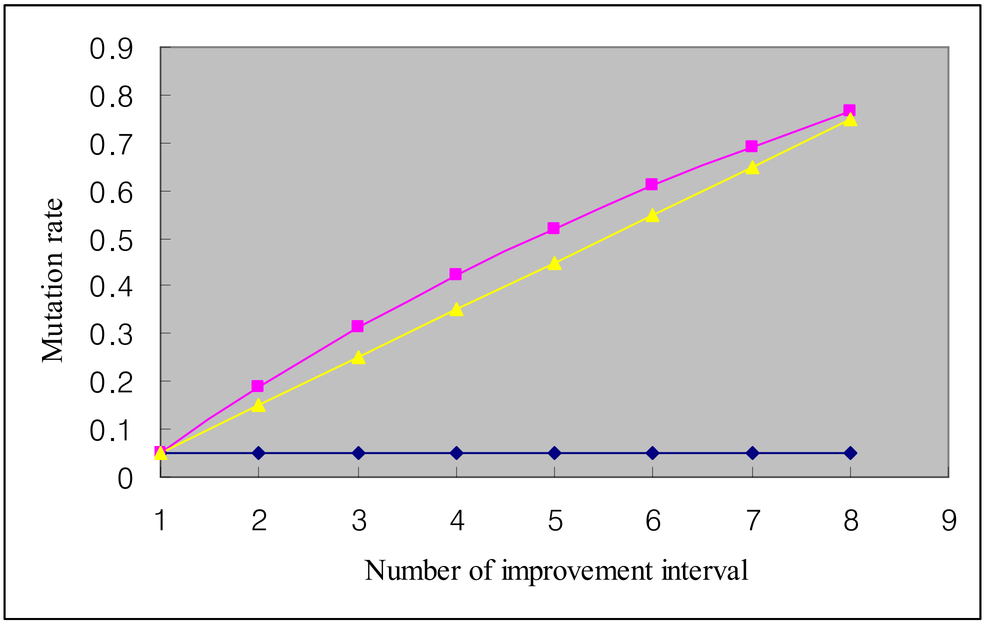

Figure 4.5 The mutation rates over 8 improvement intervals of MP 1, MP 2 and MP 3

Table 4.8 presents four process parameters with three levels in the proposed GA. To conduct the full factorial experiment with all factors, 3^4 (or 81) experiments are necessary to determine the optimal process parameters. However, Taguchi method only requires 9 runs to optimize the process parameters when L_9 orthogonal array is used. The following paragraphs explain how to optimize the process parameters for the proposed GA using Taguchi method.

Four factors with three-levels per factor are summarized in Table 4.8. To obtain the optimal process parameters, L_9 (3^4) orthogonal array has been chosen.

Table 4.8 Four factors with three levels per factor

	Level		
Process parameter	1	2	3
A (Population size)	100	150	200
B (Crossover rate)	0.6	0.7	0.8
C (Improvement interval)	50	75	100
D (Mutation policy)	*MP 1	MP 2	MP 3

*MP: Mutation policy.

Each row in Table 4.9 shows 9 experiments with process parameters A, B, C and D in the corresponding levels. To account for the characteristics of stochastic disturbance in the proposed GA, each experiment has been tested 40 times. The noise factors can be considered various sizes and structures of VRPs. G_i is calculated as

$$G_i = \frac{\text{Experimented solution} - \text{Best known solution}}{\text{Best known solution}} ,$$

,

where i = {1, 2, …, 40}. The value of G_i stands for the relative gap between the experimented solution and the best known solution. For example, if the experimented solution of the total traveled distance in run 1 is 340,

$$G_1 = \frac{340 - 375}{375} = 0.013,$$

where the best known solution of E-n22-k4 problem is 375.

Table 4.9 SNR values of the L_9 experiments

Run	A	B	C	D	E-n22-k4	E-n23-k3	E-n30-k3	E-n33-k4	SNR
1	1	1	1	1	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	26.15
2	1	2	2	2	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	27.55
3	1	3	3	3	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	26.54
4	2	1	2	3	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	27.82
5	2	2	3	1	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	29.64
6	2	3	1	2	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	29.65
7	3	1	3	2	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	32.20
8	3	2	1	3	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	30.21
9	3	3	2	1	$G_1,\ldots,G_{10}$	$G_{11},\ldots,G_{20}$	$G_{21},\ldots,G_{30}$	$G_{31},\ldots,G_{40}$	29.58

Since the smaller G_i is more desirable, the smaller-the-better SNR calculation has been used as

$$\mathrm{SNR} = -10\log\left(\frac{1}{n}\sum_{i=1}^{n} G_i^2\right),$$

where n = 40. The SNR values of process parameter A are calculated as

$$\mathrm{SNR}_{A_1} = \mathrm{SNR}_1 + \mathrm{SNR}_2 + \mathrm{SNR}_3,$$

$$\mathrm{SNR}_{A_2} = \mathrm{SNR}_4 + \mathrm{SNR}_5 + \mathrm{SNR}_6,$$

$$\mathrm{SNR}_{A_3} = \mathrm{SNR}_7 + \mathrm{SNR}_8 + \mathrm{SNR}_9,$$

where SNR_i represents the SNR value of the i^{th} run and A_i denotes the level i of process parameter A.

The optimal levels of process parameter A, B, C and D are the level with the largest SNR value, and the calculated SNR_{A_i}, SNR_{B_i}, SNR_{C_i} and SNR_{D_i} are shown in Table 4.10. From Table 4.10, the optimal level of process parameter A is the third level, the optimal level of process parameter B is the second level, the optimal level of process parameter C is the third level, and the optimal level of process parameter D is the second level. SNR values at optimal level for each process parameters are in bold. According to Table 4.8, the optimal population size is 200, the optimal crossover rate is 0.7, the optimal termination condition is 75, and the optimal mutation policy is MP 2 for the proposed GA in solving the VRPs. From the percentage contribution of each parameter in Table 4.10, process parameter A impacts 77% and process parameter D influences 15% on the performance of the proposed GA. In other words, the large size of the population and the mutation policy of linear increase in mutation rate dominate other process parameters on the performance of the proposed GA.

Table 4.10 SNR values of the process parameters

	Process parameter			
Level	A	B	C	D
1	80.24	86.17	86.01	85.37
2	87.12	**87.40**	84.94	**89.40**
3	**91.98**	85.77	**88.38**	84.56
Percentage contribution	77	2	7	15

Note: Bold typeface represents the optimal values in each column of parameters.

The process parameters optimized by the Taguchi method are robust, so the signal or performance measure always centralizes to the optimal expected values, and are less affected by noise. After the optimal process parameters suggested by the Taguchi method were used in the examples of VRPs, the search ability of the proposed GA has been improved and the solution of the proposed GA has been well achieved.

4.4 A real-life scale VRP

In this section, a real-life scale VRP (RLS_VRP) with heterogeneous vehicles from multiple depots, allowing multiple visits is presented and solved by the proposed GA using optimized process parameters by Taguchi method in Section 4.3. The RLS_VRP has 35 US cities (customers), 3 depots and 9 heterogeneous vehicles (3 vehicles in each depot). This problem demonstrates a didactic illustration of the characteristics and the originality of the proposed GA so that the readers catch its distinctive flavor and understand its potential. Furthermore, the distances among all cities are the real driving distances on road in stead of the Euclidean metric.

4.4.1 Problem description

38 nodes of RLS_VRP are illustrated in Figure 4.6. In Figure 4.6, the circled cities are 35 retailer cities and the boxed ones are 3 warehouse cities; Denver, Chicago and Atlanta. The all distances among the cities are referred from http://www.convertit.com/Go/ConvertIt/Calculators/Geography/Driving_Distance_Calc.ASP, which shows approximation of driving distances. There are three vehicles in each warehouse city. The capacities of vehicles in each warehouse city are given in Table 4.11. The demands of the retailer cities are given in Table 4.12.

Figure 4.6 The problem of 35 US cities and 3 depots

Table 4.11 Vehicle capacities in each warehouse city

Warehouse city	Vehicle	
	Identification	Capacity
Denver	V_{11}	1200
	V_{12}	1800
	V_{13}	2500
Chicago	V_{21}	1200
	V_{22}	1800
	V_{23}	2500
Atlanta	V_{31}	1200
	V_{32}	1800
	V_{33}	2500

Table 4.12 Demands of the retailer cities

No.	City	Demand	No.	City	Demand
1	Boise	140	19	Milwaukee	550
2	Boston	100	20	Minneapolis	720
3	Charlotte	180	21	Nashville	780
4	Columbia	620	22	New Orleans	240
5	Columbus	900	23	New York	150
6	Dallas	310	24	Oklahoma City	180
7	Des Moines	990	25	Philadelphia	650
8	Detroit	110	26	Phoenix	240
9	Hartford	140	27	Portland, OR	310
10	Houston	190	28	Reno	450
11	Indianapolis	980	29	St. Louis	350
12	Jacksonville	210	30	Salt Lake City	160
13	Kansas City	250	31	San Antonio	170
14	Las Vegas	310	32	San Diego	120
15	Los Angeles	310	33	San Francisco	200
16	Memphis	880	34	Seattle	240
17	Louisville	170	35	Washington, D. C.	310
18	Miami	650			

4.4.2 The results of the implementation

The results of the proposed GA for RLS_VRP with one and two allowed visits are shown in Table 4.13. Note that routes with two allowed visits are better than ones with one visit. The vehicle routes with two allowed visits are given in Table 4.14. Two visits to customer 11 in bold by vehicles V_{22} and V_{33} reduce the total traveled distance in the result. All other customer's demand are satisfied only with one visit in Table 4.14.

Table 4.13 Results of one allowed visit and two allowed visits from the proposed GA

	Allowed visits	
	One	Two
Total distance	17950 miles	17710 miles

Table 4.14 Vehicle routes with two allowed visits

Warehouse city	Vehicle routes
Denver	V_{11}: *D - *14 - 15 - 32 - 26 - D V_{12}: D - 20 - 7 - D V_{13}: D - 28 - 33 - 27 - 34 - 1 - 30 - D
Chicago	V_{21}: D - 19 - D V_{22}: D - 22 - 10 - 31 - 6 - 24 - 13 -**11(420)** - D V_{23}: D - 29 - 17 - 21 - 16 - 8 - D
Atlanta	V_{31}: D - 4 - D V_{32}: D - 12 - 18 - 35 - 3 - D V_{33}: D - 25 - 9 - 2 - 23 - 5 - **11(560)** - D

*D stands for each warehouse city of the left column in the table.
*14 stands for the number of the retailer city from Figure 4.12.

The variation of the best solution during 1776 generations by the proposed GA is shown in Figure 4.7. The graph shows the convergence of the best solution over the generation. After 976th generation, the best solution of the problem is finally obtained. After additional 800 generation based on mutation policy 2, the proposed GA is terminated since is concluded no improvement of the best solution.

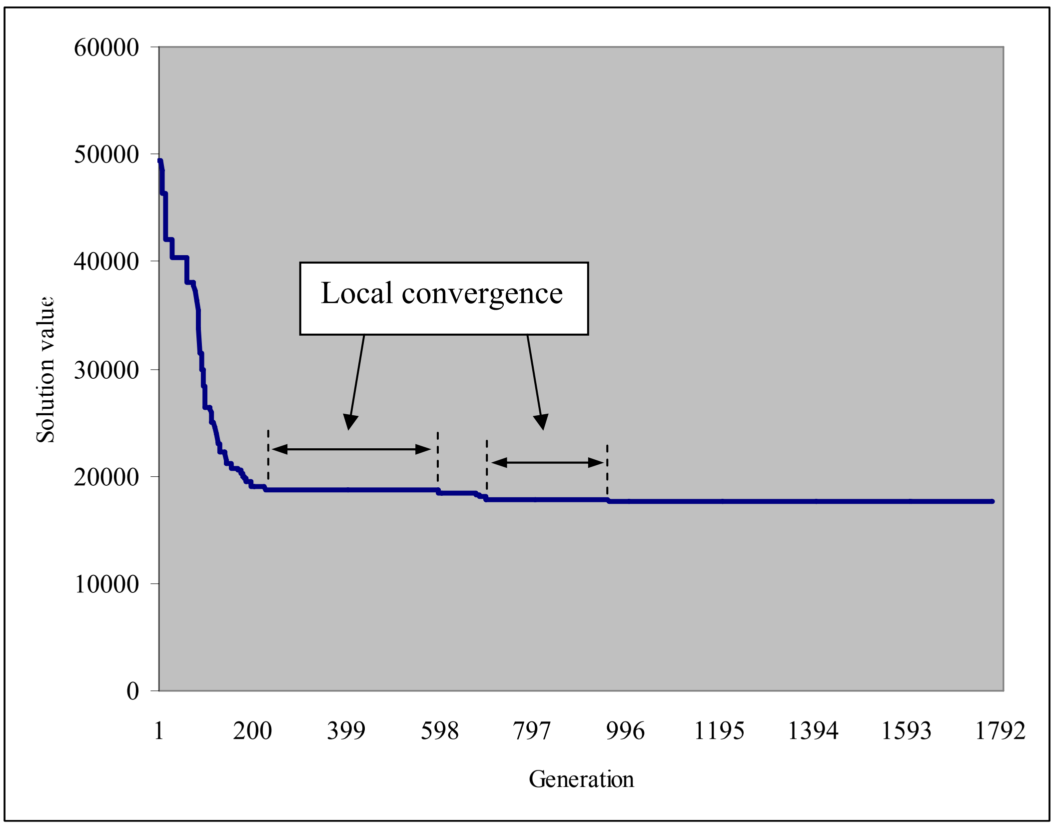

Figure 4.7 The variation of the best solution

Chapter 5. Conclusions and Future research

The objective of this thesis described in Section 1.1 has been successfully achieved. A generalized VRP with heterogeneous vehicles from multiple depots, allowing multiple visits has been identified by relaxing from the classical VRPs the constraints of the number of vehicles, the number of depots, and the number of visits allowed to each customer.

The identified VRP has been modeled into a MIP model and tested to solve a small problem. Due to the memory limitation of branch-and-bound algorithm in CPLEX, only 6-customer 2-depot problem could be optimally solved. As the motivation of this thesis conjectures, it has been identified that the introduction of multiple visits to each customer leads to the reduction of delivery cost.

A GA to effectively and efficiently solve the medium or large VRPs with heterogeneous vehicles from multiple depots, allowing multiple visits has been developed and validated successfully. The proposed algorithm has been produced the solutions which are equal to or better than the best known solutions for the benchmark VRPs. The proposed algorithm has been executed with the restriction of one depot, one allowed visit and heterogeneous vehicles even if it can generate better routes.

Taguchi robust design method has been introduced and applied in optimizing the process parameters of the proposed GA. Using the optimized parameters, the proposed GA shows the robust performance regardless of the size or structure of the problems. The proposed algorithm has effectively solved the test problem with 35 US cities and 3 depots, allowing the multiple visits and heterogeneous vehicles.

A new mutation policy has been developed to the proposed GA. The existing GA implementations use the fixed mutation rate. This thesis proposed the idea of self-adapting mutation rate, which enables dynamic speeds of evolutions in nature. The proposed mutation policy has been proved effective from the Taguchi method analysis and has generated consistently good solutions.

As for future research, it may be useful to investigate the issue where there is restriction on the driving distance of vehicles available at each depot. The problem where the customers have different time windows as their requirements for delivered goods may also be worth considering. Also, future research will be conducted to improve the proposed algorithm. Additional improvements might lie in the combination of various selection and population replacement schemes and new fitness models. Applications of the approach to related problems will be explored as well.

Bibliography

Aarts E., and Lenstra J. K. (1996), “Local Search in Combinatorial Optimization,” *Springer-Verlag.*

Anderson C., Jones K., and Ryan J. (1991), “A two-dimensional algorithm for the ising problem,” *Complex Systems*, **5**, 327-333.

Ball M. O., Magnanti T. L., Monma C. L., and Nemhauser G. L. (1995), “Network Routing,” *Handbooks in Operations Research and Management Science*, **8**, North Holland.

Berger J., and Barkaoui M. (2003), “A new hybrid genetic algorithm for the capacitated vehicle routing problem,” *Journal of the Operational Research society*, **41(2)**, 179-194.

Bodin L. D., Golden B. L., and Bender A. (1983), “The state of the art in the routing and scheduling of vehicles and crews,” *Computers & Operations Research*, **10(2)**, 79-116.

Bodin L. D., Golden B. L., Assad A. A., and Ball M. (1983), “Routing and scheduling of vehicles and crews, the state of the art,” *Computers and Operations Research*, **10(2)**, 63-212.

Chao I. M., Golden B. L., and Wasil E. A. (1993), “A new heuristic for multiple-depot vehicle routing problem that improves upon best-known solutions,” *American Journal of Mathematical and Management Sciences,* **13***,* 371-406.

Christofides N. (1985), “Vehicle routing: *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*,” John Wiley & Sons Ltd., Chichester, UK, 431-448.

Christofides N., Mingozzi A., and Toth P. (1979), “The vehicle routing problem,” *Combinatorial Optimization*, Wiley, Chichester, UK, 315-338.

Cohoon P., and Paris W. (1986), “Genetic placement,” in *Proceedings of the IEEE International Conference on Computer-Aided Design*, 422-425.

Cordeau J-F., Gendreau M., Laporte G., Potvin J-Y., and Semet F. (2002), “A guide to vehicle routing heuristics,” *Journal of the Operational Research Society,* **53**, 512-522.

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

Davis L. (1987), "*Genetic algorithms and simulated annealing*," Pitman Publishing.

Desrochers M., Lenstra J. K., and Savelsbergh M. W. P. (1990), "A classification scheme for vehicle routing and scheduling problem," *Journal of Operational Research Society,* **46**, 322-332.

Desrosiers J., Dumas Y., Solomon M., and Soumis F. (1995), "Time Constrained Routing and Scheduling in: *Network Routing*," *Handbooks in Operations Research and Management Science*, **8**, North Holland.

Eilon. S., Watson-Gandy, C.D.T., and Christofides, N. (1971), "*Distribution Management: Mathematical Modelling and Practical Analysis*," Griffin, London.

Eshelman L., and Schaffer J. (1993), "Real-coded genetic algorithms and interal-schemata," in Whitley L., editor, *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco, **2**, 187-202.

Falkenauer E. (1996), "A hybrid grouping genetic algorithm for bin packing," *Journal of Heuristics*, **2**, 5-30.

Fisher M. L. (1994), "Optimal Solution of Vehicle Routing Problems using Minimum K-Trees," *Operation Research*, **42(4)**, 626-642.

Fisher M. L. (1995), "Vehicle routing in: *Network Routing*," *Handbooks in Operations Research and Management Science*, **8**, North Holland.

Gen M., and Cheng R. (2000), "*Genetic algorithms and engineering optimization*," John Wiley & Sons, Inc.

Golden B. L., and Addad A. (1995), "The vehicle routing problem in Network Routing," *Handbooks in Operations Research and Management Science*, **8**, North Holland.

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

Laporte G. (1992), "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, **23**, 631-640.

Laporte G., Gendreau M., Potvin J-Y., and Semet F. (2000), "Classical and modern heuristics for the vehicle routing problem," *International Transactions in Operational Research*, **7**, 285-300.

Laporte G., Nobert Y., and Arpin D. (1984), "Optimal solutions of capacitated multi-depot vehicle routing problem," *Congressus Numeratium*, **44**, 283-292.

Laporte G., and Nobert Y. (1987), "Exact algorithms for the vehicle routing problem," *Annals of Discrete Mathematics*, **31**, 147-184.

Laporte G., and Semet F. (1999), "Classical heuristics for the vehicle routing problem," *Technical Report G-98-54,* GERAD.

Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., and Shmoys D. B. (1985), "*The Traveling Salesman Problem*," Wiley, Chichester, UK.

Magnanti T. L. (1981), "Combinatorial optimization and vehicle fleet planning: Perspectives and prospects," *Networks*, **11**, 179-214.

Michalewicz Z. (1996), "*Genetic Algorithm + Data Structure = Evolution Programs*," 3rd edition, Springer-Verlag, New York.

Osman I. H. (1993a), "Metastrategy simulated annealing and Tabu search algorithms for the vehicle routing problem," *Operation Research*, **41**, 421-451.

Osman I. H. (1993b), "Vehicle routing and Scheduling: applications, algorithms and developments," *Proceedings of the International Conference on Industrial Logistics*, Rennes, France.

Pereira F. B., Tavares J., Machado P., and Costa E. (2002), "GVR: a new genetic representation for the vehicle routing problem," *Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science*, 95-102.

Rawlins G. J. E. (1991), "*Foundations of genetic algorithms*," Morgan Kaufmann Publishers.

Reeves C. R. (1993), "*Modern heuristic techniques for combinatorial problems*," Oxford: Blackwell Scientific Press.

Renaud J., Boctor F. F., and Laporte G. (1996), "An improved petal heuristic for the vehicle routing problem," *Journal of Operational Research Society*, **47**, 329-336.

Rochat Y., and Taillard E. (1995), "Probabilistic diversification and intensification in local search for vehicle routing," *Journal of Heuristics*, **1**, 147-167.

Shin H. W., and Kang M. K. (1991), "A Heristic for Vehicle Routing Problem Allowing Multiple Visits," *The Society of Korea Industrial and Systems Engineering*, **14(24)**, 141-147.

Syswerda G. (1989), "Uniform crossover in genetic algorithms," in Schaffer J., editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, CA, 2-9.

Taguchi G., Chowdhury S., and Taguchi S. (2000), "*Robust Engineering*," McGraw-Hill, New York.

Tillman F. A., and Hering R. W. (1971), "A study of look-ahead procedure for solving the multi-terminal delivery problem ," *Transportation Research,* **5**, 225-229.

Toth P., and Vigo D. (1998), "Exact algorithms for vehicle routing," *Fleet Management and Logistics*, Kluwer, Boston, MA, 1-31.

Toth P., and Vigo D. (2002), "The vehicle routing problem," *SIAM Monographs on Discrete Mathematics and Applications,* Philadelphia, PA, SIAM.

Walters G. A., and Smith D. K. (1995), "Evolutionary design algorithm for optimal layout of tree networks," *Engineering Optimization*, **24**, 261-281.

Wu Y. (2000), "*Taguchi Methods for Robust Design*", New York: ASME.