T.C.

MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES IN

PURE AND APPLIED SCIENCES

A GENETIC ALGORITHM BASED SOLUTION APPROACH FOR TDVRP

Mehmet Selçuk KORKMAZ

THESIS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
ENGINEERING MANAGEMENT

SUPERVISOR

Asst. Prof. Dr. Serol BULKAN

Assoc. Prof. Dr. Ekrem DUMAN

Istanbul 2013

T.C.

MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES IN

PURE AND APPLIED SCIENCES

A GENETIC ALGORITHM BASED SOLUTION APPROACH FOR TDVRP

Mehmet Selçuk KORKMAZ

(524890981)

THESIS

FOR THE DEGREE OF MASTER OF SCIENCE

IN

ENGINEERING MANAGEMENT

SUPERVISOR

Asst. Prof. Dr. Serol BULKAN

Assoc. Prof. Dr. Ekrem DUMAN

Istanbul 2013

T.C.

MARMARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KABUL ve ONAY BELGESİ

Mehmet Selçuk Korkmaz'ın **"A GENETIC ALGORITHM BASED SOLUTION APPROACH FOR TDVRP"** başlıklı Lisansüstü tez çalışması, M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 09.09.11 tarih ve 201/09 sayılı kararı ile oluşturulan jüri tarafından Mühendislik Yönetimi Anabilim Dalı Mühendislik Yönetimi Programında YÜKSEK LİSANS Tezi olarak Kabul edilmiştir.

Danışman	: Yrd. Doç. Dr. Serol Bulkan (Marmara Üniversitesi)	serol bulkan
1. Üye	: Prof. Dr. Selim Akyokuş (Doğuş Üniversitesi)	
2. Üye	: Doç. Dr. Bahar Sennaroğlu (Marmara Üniversitesi)	

Tezin Savunulduğu Tarih : 12.09.2011

ONAY

M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 13.09.2011 tarih ve 2011/26-09 sayılı kararı ile **Mehmet Selçuk Korkmaz**'ın Mühendislik Yönetimi Anabilim Dalı Mühendislik Yönetimi Programında Y.Lisans (MSc.) derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

Prof. Dr. Abdülkerim KAR

ACKNOWLEDGEMENT

I would like to thank for their valuable contributions and suggestions to my thesis to:

Professor Serol BULKAN

Professor Ekrem DUMAN

I would also like to thank to my friends Betül ÖZDEMİR and Alper TÜRKYILMAZ and my son Alp for their help and support.

September, 2013 Mehmet Selçuk KORKMAZ

TABLE OF CONTENTS

ÖZET

TDVRP İÇİN GENETİK ALGORİTMA BAZLI BİR ÇÖZÜM YAKLAŞIMI

VRP (Araç Rotalama Problemi), zamana bağlı olarak değişen seyahat süreleri kullanıldığında gerçek uygulamalara daha iyi çözümler üretmekte fakat problem daha karmaşık hale gelmektedir. Bu çalışmada VRP probleminin bu varyasyonu olan TDVRP ele alınacaktır. Çalışma boyunca bu problem için matematiksel model oluşturulacaktır. Oluşturulan MIP (Karma Tam Sayılı Doğrusal Programlama) modeli derinlemesine ele alınacak ve bu model için genetik algoritma uygulanacaktır

Çalışma boyunca sürekli bir zamana dayalı fonksiyon kullanılacaktır. Alternatif olarak basamak fonksiyonu kullanılabilir. Bu çalışmada diğer kısıtların etkisini daha yakından incelemek amacıyla sınırlı sayıda eşdeğer araç kullanılacaktır ve modelin karmaşıklığı önlenecektir. Çalışmanın amacı TDVRP için rekabetçi bir çözüm yaklaşımı oluşturmaktır.

Anahtar Sözcükler: Araç Rotalama Problemi, Genetik Algoritma, Zamana Bağlı Araç Rotalama Problemi

ABSTRACT

A GENETIC ALGORITHM BASED SOLUTION APPROACH FOR TDVRP

VRP (Vehicle Routing Problem) generates more realistic and good solutions when time dependent travel times are used, but besides the problem becomes more complicated. In this study, this version of VRP will be examined which is TDVRP. During the study the mathematical model for the problem will be generated. Generated MIP (Mixed Integer Programming) model will be examined in depth and genetic algorithm will be generated for this model.

Within the study a continuous time dependent function will be used. As an alternative a step function can be used. The study will assume that there is a limited number of identical vehicles, in order to examine closely the effects of the other constraints, which will also avoid the complexity of the model. The purpose of the study is to generate a competitive solution approach for TDVRP.

Key Words: Vehicle Routing Problem, Genetic Algorithm, Time Dependent Vehicle Routing Problem

ABBREVIATIONS

CVRP	**:** Capacitated Vehicle Routing Problem
EA	**:** Evolutionary Algorithm
E-child	**:** Exploratory child
FIFO	**:** First In First Out
GA	**:** Genetic Algorithm
GAP	**:** Generalised Assignment Problem
GIS	**:** Geoographical Information System
MIP	**:** Mixed Integer Programming
MDTDVRP	**:** Multi Depot Time Dependent Vehicle Routing Problem
O-child	**:** Optimum child
OCGA	**:** Optimum Child Genetic Algorithm
RT-TDVRPTW	**:** Real Time Time Dependent Vehicle Routing Problem with Time Window
TD	: Time Dependency
TDTSP	**:** Time Dependent Traveling Salesman Problem
TDVRP	**:** Time Dependent Vehicle Routing Problem
VRP	**:** Vehicle Routing Problem
VRPLIB	**:** Vehicle Routing Problem Library
VRPTW	**:** Vehicle Routing Problem with Time Windows

CHAPTER I

INTRODUCTION and AIM

The Time Dependent Vehicle Routing Problem, TDVRP, can be defined as optimally routing a fleet of vehicles of certain capacities when the traveling times between nodes are depending on the time of the day that the trip on that arc was initiated. The delivery to a customer must satisfy the customer demand. The optimization process is finding the solution that minimizes the number of tours (the number of vehicles used), the total distance travelled and the total traveling time. The traveling time is calculated with the knowledge of departing time from a certain customer and the speed of the vehicle depending on the time of the day, along the travelled arc, which is supposed to be known at the beginning of the optimization.
This version of the VRP is motivated by the fact that realistic circumstances such as traffic conditions, especially during rush hours, can not be ignored in order to carry out a realistic optimization.

Because of the interest in its application in logistics and supply chain management, vehicle routing problem has been largely studied. Many different versions of this problem have been formulated to take into account the many possible aspects. One of the most hard ones is to take into account the added complexity deriving from traffic conditions in the road network, not just in the context of an optimization problem, but also when the assumption of constant traveling times can not be considered reliable, like when timely deliveries are to be done and costs of operation are on the edge of a feasible and effective solution to be found. Not taking into account these factors might greatly affect the layout of planned operations. Besides, optimization objectives might not be any longer satisfied.

Throughout this study, TDVRP will be solved by using Genetic Algorithm, which is one of the evolutionary approaches.

The Genetic Algorithm is a popular meta-heuristic, which mimics the natural biological evolution of species (Darwin, 1859), in order to solve combinatorial optimization problems. It was first introduced by Holland (1975). A genetic

algorithm deals with a population of solutions subjected to reproduction process. This process occurs under the direct impact of each individual's relative fitness. The fitness of each individual of a population is defined by the quality of its solution with respect to the objective function under consideration. Such a setup contributes to increasing the average fitness of a population in each subsequent generation. After a certain number of generations, it is likely to get a solution of the quality that is necessary.

However, genetic algorithms have obtained very limited success in the area of Vehicle Routing, due to their high computational cost. The complexity of the TDVRP can be an advantage for genetic algorithms, since it will not put extra computational cost, compared to other meta-heuristic algorithms.

The rest of this thesis is as follows: in Chapter II a brief literature survey is done, in Chapter III the study is explained in detail, in Chapter IV and Chapter V results of the experiments are mentioned and conclusions are stated.

CHAPTER II.

LITERATURE SURVEY

VRP of realistic size can be handled using heuristics. There have been many contributions to the subject, including various extensions to the basic problem described by Laporte. Laporte (1992) gives a survey, and an extensive bibliography has been compiled by Laporte and Osman (1995). The tabu search implementations of Taillard (1993) and Rochat and Taillard (1995) have obtained the best known results to benchmark VRPs. Various authors, throughout their researches have reported similar results, obtained using tabu search (Osman, 1993), (Gendreau et al., 1994), or simulated annealing (Osman, 1993), (Hiquebran et al., 1994). However, it has been observed by Renaud et al. (1996) 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 (Bullnheimer et al., 1999), (Gambardella et al., 1999). With a 2-optimal heuristic incorporated to improve individual routes produced by artificial ants, this approach also has given results which are only slightly inferior to those from tabu search.

The distribution problem in which vehicles based at a central facility (depot) are required to visit geographically dispersed customers in order to fulfill known customer requirements are referred to as the VRP (Christofides, 1985). The main objective of the VRP is to minimize the distribution costs for individual carriers, and can be described as the problem of assigning optimal delivery or collection routes from a depot to a number of geographically distributed customers; subject to constraints. The most basic version of the VRP have also been called vehicle scheduling or delivery problem. A number of different formulations appear in the authoritative work of Christofides (1985). The basic problem can be defined with $G = (V, A)$ being a directed graph.

The basic VRP is to route the vehicles one route per vehicle, each starting and finishing at the depot, so that all customers are supplied with their

demands and the total travel cost is minimized.

The basic VRP makes a number of assumptions, including utilizing a homogeneous fleet, a single depot, one route per vehicle, etc. These assumptions can be eliminated by introducing additional constraints to the problem. This implies increasing the complexity of the problem and by restriction, classifies the extended problem as an *np*-hard problem. It should be noted that most of these additional constraints are often implemented in isolation, without integration, due to the increased complexity of solving such problems.

II.1. THE CONCEPT OF TIME WINDOWS

A time window can be described as a window of opportunity for deliveries. It is an extension of the VRP that has been researched extensively (Ibaraki et al., 2005, Taillard, 1999, Taillard et al., 1997, Tan et al., 200lc). A time window is the period of time during which deliveries can be made to a specific customer i, and has three main characteristics:

- Earliest allowed arrival time, e_i, also referred to as the *opening time*
- Latest allowed arrival time, l_i, also referre to as the *closing time*
- Whether the time window is considered *soft* or *hard*

II.2. TIME-DEPENDENT TRAVEL TIME

Although unpredictable events such as accidents and vehicle breakdowns render travel times as stochastic, the candidate postulates that the subtle, yet partially predictable event of congestion during peak hours of the day requires more attention. The assumption is made that by addressing the time-dependent nature of travel times, a modeling approach that is a stronger approximation of the actual real-world conditions of vehicle routing and scheduling than by catering for stochastic travel times, will be achieved.

Hill and Benton (1992) review the two main approaches in estimating travel distance between two nodes *i* and *j*, denoted by d_{ij}, namely Minkowski distance and Pythagorean distance. The former is presented below:

$$d_{ij} = \left[\left| x_i - x_j \right|^w + \left| y_i - y_j \right|^w \right]^{\frac{1}{w}}$$

When *w* is 2, the Minkowski distance, denoted by d_{ij}, is the Pythagorean

distance. When w is 1, the Minowski distance is the *city block* right-angled distance. The former mentioned above the coordinate pair (x_i, y_i) of each node i is required. A similar approach can be followed if only latitude and longitude data is available, i.e. from a Geographical Information System (GIS) database. The problem, however, is that researchers often reduce vehicle travel speed to an approximate speed, denoted by r_c, and simply apply the scalar transformation of distance in the formula below to find the travel time between the two nodes,

$$t_{ij} = \frac{d_{ij}}{r_c}$$

without cognisance of an acceleration stage to get onto the road, the cruising stage, and the deceleration stage at the destination node (Assad, 1988). If the three stages were to be acknowledged, d_c denotes the distance required for the vehicle to reach its cruising speed, and α denotes the acceleration, a more appropriate way of calculating the travel time is given in most metropolitan areas, travel times are much longer during the start and end of workday rush hours. If one were to inflate all route times equally during peak periods, one would be able to route and schedule vehicles without taking time-dependent travel times into consideration, and not compromise optimality of routes. However, road networks are unevenly congested, i.e. traveling from A to B during the morning rush hour traffic might be more congested than when traveling from B to A at the same time.

Malandraki and Daskin (1992) state that the travel time is not only a function of the distance, but should take the time of day into account as well. Ichoua et al. (2003) state that research on time-dependent problems started towards the end of the 1950s with references to the time-dependent shortest path problem, the time-dependent path choice problem, and the Time Dependent Traveling Salesman Problem (TDTSP). Of the earliest research found on the Time Dependent Vehicle Routing Problem (TDVRP) is Hill et al. (1988), followed by Hill and Benton (1992). In their papers customer nodes were assigned time-dependent piecewise constant speeds — these speeds reflect the traveling speed surrounding the nodes. The edge travel time between two nodes were derived as the average speed of the two nodes concerned. At the time Hill

and Benton (1992) attribute the lack of time-dependent travel time research to:

- Immense efforts to estimate travel time parameters
- Prohibitive data storage requirements
- Inefficient solution algorithms

Malandraki and Daskin (1992) formulate an elegant variant of the Vehicle Routing Problem with Time Windows (VRPTW) with the introduction of piecewise constant travel times on the edges. Approaches to accommodate time-dependent travel times mentioned so far all allow *passing*: the event where one vehicle my pass another vehicle on the same edge although it started later than the vehicle it passed, but in a different time period with shorter traveling time.

II.3. GENETIC ALGORITHMS IN VRP PROBLEMS

Genetic algorithms (GAs) have seen widespread use amongst modern meta-heuristics, and several applications to VRPs incorporating time windows have been reported (Thangiah et al., 1991), (Schmitt, 1994), (Potvin and Bengio, 1996). Applications of GAs have also been reported for the VRP with backhauls (Potvin et al., 1996), for a multi-depot routing problem (Salhi et al., 1998), and a school bus routing problem (Thangiah et al., 1992).

In some problems a hybrid approach can be used. There are many studies where GAs are used together with other heuristic techniques, where good solution qualities are reported, compared to the ones, GAs used stand alone.

A hybrid approach to vehicle routing using neural networks and GAs has also been reported (Potvin et al., 1996). However, GAs do not appear to have made a great impact so far on the basic VRP. The aim of this study is to put forward a conceptually straight forward GA for the basic VRP and also TDVRP, which is competitive with other modern heuristics in terms of computing time and solution quality. A hybrid heuristic which incorporates neighbourhood search is not considered for the moment.

II.4. BASIS FOR A GENETIC ALGORITHM

The principles of GAs are well known. A population of solutions is maintained and a reproductive process allows parent solutions to be selected from the population. Offspring solutions are produced which exhibit some of the

characteristics of each parent. The fitness of each solution can be related to the objective function value, in this case the total distance travelled, and the level of any constraint violation. 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 it evolves. More details are given by Reeves (1993), for example. The starting point for any GA is in the representation of each solution or population member. Typically, this will be in the form of a string or chromosome. Individual positions within each chromosome are referred to as genes. Although binary strings have been favoured by many GA researchers, some successful implementations use non-binary representations. For example, Chu and Beasley (1997) use a non-binary representation in their approach to the Generalised Assignment Problem (GAP), which requires jobs to be assigned to agents at minimum total cost, subject to resource limitations for each agent. In their representation, each chromosome consists of a string of decimal numbers, in index order of the jobs, specifying the agent number to which each job is assigned. Experiments were carried out with the initial population generated randomly, with an initial population of structured solutions, and with a mixed population containing both random and structured solutions. The expectation is that an initial population of reasonably structured solutions will evolve to high-quality solutions in a relatively small number of generations of the GA. However, a possible drawback is that such a population will lack the diversity needed to obtain near-optimal solutions, comparable in quality to those obtained using tabu search.

In preliminary experiments, the use of randomly generated individuals in the population only slowed the convergence of the GA with no improvement in the quality of the best solution obtained, and so randomly generated individuals were not included in the final versions of the GA.

II.4.1. The Reproductive Process

Two parent solutions are selected from the population 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. Offspring are produced from the two parent solutions using a standard crossover procedure. Best results were obtained using the

2-point crossover, in which two points in the chromosome are chosen randomly. One offspring consists of the gene values from parent one which are to the left of the first point and to the right of the second point, along with the gene values from parent two which are between the two chosen points in the chromosome. A second offspring is produced by swapping round the parents and then using the same procedure. Offspring which duplicate existing members of the population are discarded.

II.4.2. Replacement Scheme

The GA, which they have developed, uses the steady-state approach, in which eligible offspring enter the population as soon as they are produced, with inferior individuals being removed at the same time, so that the size of the population remains constant. The ranking replacement method is used, as described by Beasley and Chu (1998). For each individual, the total travel distance and any constraint violation are recorded as separate values. The travel distance is referred to as the fitness of the individual. Any excess weight on a vehicle is expressed as a proportion of the vehicle capacity, and any excess distance is expressed as a proportion of the maximum vehicle distance. Then the total excess across all vehicles is referred to as the unfitness of this individual. In the ranking replacement method, the population is partitioned into four subsets with respect to the child. Representing the population by P and denoting the fitness and unfitness values of a population member $p \in P$ by f (p) and u(p), respectively, and the child by c, the four subsets are defined as:

$S_1 = \{p \in P: f(p) \geq f(c); u(p) \geq u(p)\}$;

$S_2 = \{p \in P: f(p) < f(c); u(p) \geq u(p)\}$;

$S_3 = \{p \in P: f(p) \geq f(c); u(p) < u(p)\}$;

$S_4 = \{p \in P: f(p) < f(c); u(p) < u(p)\}$:

Then the population member to be replaced by the child is chosen from the first non-empty set in the order S_1; S_2; S_3. In their implementation, if all three of these sets are empty, then the child does not enter the population. In any subset, the member selected for replacement is that with the worst unfitness, with ties broken by worst fitness. In this way, the evolution of the population should achieve feasible solutions, but without being so strongly biased towards achieving feasibility that

good quality solutions are never achieved, or vice versa. Their experiments confirmed that the GA performs better with ranking replacement than with either worst unfitness or worst fitness replacement.

Thus, the steps of the pure GA can be summarised as follows:

Generate an initial population of structured solutions

Evaluate fitness and unfitness of each individual in the population

Repeat

Select two parents from the population using two binary tournaments

Produce two offspring from the parents

Mutate offspring

Evaluate fitness and unfitness of offspring

Choose favoured offspring

If entry criteria are satisfied by chosen offspring

Choose population member to be replaced

Offspring enters population and the chosen member is removed

End if

Until stopping criterion is satisfied

In choosing the favoured offspring, feasible solutions are favoured over infeasible solutions. If both child solutions are infeasible, the one with smaller unfitness value is favoured. The stopping criterion can be defined in terms of elapsed computer time, number of generations, or number of generations with no improvement in the best solution found. Computational results for the pure GA are given later, alongside those for their hybrid algorithm and well-known published results.

II.5. TIME DEPENDENT VRP

The Time Dependent VRP was first formulated by Malandraki and Daskin (1989), (1992) using a mixed integer linear programming formulation. Malandraki and Daskin treated travel time as a function of both distance and the time of the day resulted in a piecewise constant distribution of the travel time. Although they only incorporated the temporal component of traffic-density variability, they acknowledged its importance. They developed two algorithms for solving the time-dependent vehicle-routing problem. The first algorithm was a greedy nearest-

neighbor algorithm (three variants of the algorithm were introduced), and the second was a branch and bound-based algorithm that provided better solutions, but was suitable only for small problems. Hill and Benton (1992) considered a time dependent VRP (without time windows) and proposed a model based on time dependent traveling speeds that alleviates both the data collection and data storage problems inherent in time-dependent travel speed vehicle scheduling models. They also discussed the issue of developing algorithms to find nearoptimal vehicle schedules with time-dependent travel speeds. Computational results for one vehicle and five customers were reported. Ahn and Shin (1991) discussed modifications to the savings, insertion, and local improvement algorithms to better deal with TDVRP. In randomly generated instances, they reported computation time reductions as a percentage of "unmodified" savings, insertion, and local improvement algorithms. Malandraki and Dial (1996) proposed a "restricted" dynamic programming algorithm for the time dependent traveling salesman problem, i.e. for a fleet of just one vehicle. A nearest-neighbor type heuristic was used to solve randomly generated problems. Although it is argued that many different types of travel time functions can be handled by this algorithm, results are only reported for step functions. An important property for time dependent problems is the First In - First Out (FIFO) principal (Ichoua, 2003); (Ahn, 1993). A model with a FIFO property guarantees that if two vehicles left the same location for the same destination (and traveled along the same path), the one that left first would never arrive later than the other. This is an intuitive and desirable property though it is not present in all models. Earlier formulations and solutions methods (Malandraki, 1989), (Malandraki et al., 1992), (Hill et al., 1992), (Malandraki et al., 1996) do not guarantee the FIFO property. Ichoua et al. (2003) introduced a model that guarantees the FIFO principle. This model is satisfied by working with step-like speed distributions and adjusting the travel speed whenever a vehicle crosses the boundary between two consecutive time periods.

The algorithms that they developed, which were based on tabu-search meta-heuristics, provided better solutions for most test scenarios. Fleischmann et al. (2004) utilized route construction methods already proposed in the literature, savings and insertion, to solve uncapacitated time dependent VRP with and without time windows. Fleischmann et al. assume travel times to be known between all pairs of interesting locations and constant within given time slots. Neighbor slots with similar

travel times are joined to reduce memory requirements, and the transitions between slots are smoothed to ensure a FIFO property on travel times. Fleischmann et al. tested their algorithms in instances created from Berlin travel time data. Jung and Haghani (2001), (2005) proposed a genetic algorithm to solve time dependent problems.

Formulating the problem as a mixed integer linear programming problem, they obtain lower bounds by relaxing most of the integer requirements. The lower bounds are compared with the primal solutions from the genetic algorithm to evaluate the quality of the solutions. Using randomly generated test problems, the performance of the genetic algorithm was evaluated by comparing its results with exact solutions.

II.6. OPTIMISED CROSSOVER APPROACH

Habibeh N. and Lee L.S. (2012) introduced optimised crossover approach. The idea behind a GA is to model the natural evolution by using genetic inheritance together with Darwin's theory. A population of individuals representing tentative solutions is maintained over many generations. New individuals are produced by combining members of the population via crossover and mutation operators, and these replace existing individuals with some policies. Most of the current methods of crossover determine a child by using a stochastic approach and without reference to the objective function. Optimised crossover was proposed by Aggarwal et al. (1997) for the independent set problem. They applied optimised crossover within GA, which takes into account the objective function in a straight forward way, and produced two new children: the O-child (Optimum child) and E-child (Exploratory child). The O-child is constructed in such a way that has the best objective function value from a feasible set of children, while the E-child is constructed so as to maintain the diversity of the search space. Balas and Niehaus (1998) developed this approach to produce a superior GA.

In the remainder of this section, they describe the proposed optimised crossover and discuss how various steps of the GA are implemented for the proposed algorithm. Briefly, the OCGA selects two parents from the population and uses an optimised crossover mechanism designed by a complete undirected bipartite graph to generate two children. The OCGA uses a swap node operator to produce children when the optimised crossover is not applied to the parents. To maintain the diversity

within the population, two different mutations are randomly applied to each child. Moreover an elitism replacement scheme with filtration strategy is used to preserve good solutions and to avoid premature convergence. The general framework of OCGA can be shown as follows:

Algorithm. OCGA

begin

Initialise Population (randomly generated);

Fitness Evaluation;

repeat

Selection (probabilistic binary tournament selection);

Optimised Crossover;

Swap Node (if the optimised crossover is not applied);

Mutation (inversion and swap sequence);

Fitness Evaluation;

Elitism replacement with **Filtration**;

until the end condition is satisfied;

return the fittest solution found;

end

In this paper, they studied the capacitated vehicle routing problem, which is mainly characterized by using vehicles of the same capacity. They proposed a genetic algorithm using an optimised crossover operator for solving the CVRP. The computational experiments showed that the proposed algorithm is competitive in terms of the quality of the solutions found. As for future work, it may be interesting to test OCGA with other variants of VRPs.

II.7. MUTATION OPERATOR IN GA

Baker, B.M. and Ayechew, M.A. (2003) mentioned about mutation operator. The performance of their GA was found to be improved by applying a simple mutation to new offspring, in which two genes are selected at random and their values are exchanged. Thus, two randomly chosen customers are switched between vehicles, except in cases where the two customers happen to be on the same vehicle. Other types of mutation such as shifting one or more randomly chosen customers to a neighbouring vehicle were less effective.

II.8. SPEED MODEL IN TDVRP

Kok, A.L. et al. (2010), set forth the Speed Model. To investigate the impact of the different congestion avoidance strategies in a realistic setting, they proposed a speed model for real road networks that represents peak hour traffic congestion. The speed model defines for each arc in the road network a speed step function. This function defines speeds for different time intervals, such that in each time interval the speed is constant. They used five different time intervals: the morning and evening peak periods, and the periods before, in between, and after the peak periods. By defining speeds for each single road, the model is more refined than the majority of the papers dealing with the TDVRP, in which speeds are usually defined for customer links. Note that the speed model still contains the unrealistic effect that the speed suddenly changes when entering a new time interval. This effect can be reduced by considering more time intervals, but to keep the analysis simple, they chose to use five time intervals. The speed model reflects the key elements of peak hour traffic congestion, as observed by the Dutch motorists' organization ANWB (2010) and the English Highways Agency of the Department for Transport (2010). These key elements are: large delays in urban areas, large delays on road lanes toward urban areas during the morning peak and in the opposite direction during the evening peak, and large delays on roads with a high speed limit (high- ways). The common observation by ANWB and the Highways Agency indicate that the key elements hold in general. They developed the speed model for the road network data used in this paper. However, the methodology can be applied to other road network data. Note that they based the speed model on the key elements of peak hour traffic congestion; they didn't base it on real (historical) travel time data. Therefore, for practical use, the speed model should be tailored to the road networks under consideration. This tailoring is beyond the scope of this paper: the objective of this paper is to get a good estimation of the performance of different congestion avoidance strategies in a broad and realistic setting.

II.9. THE REAL-TIME TIME-DEPENDENT VEHICLE ROUTING PROBLEM

Chen, H.K. et al., (2003) presented RT-TDVRPTW. In the RT-TDVRPTW, a planner may not be aware of all information for route planning at the time the routing process begins. Moreover, information, including demands and time-dependent travel times, may change after the initial routes are constructed, and such information cannot be known in advance. When a new customer appears, the main task of the dispatching center is to include the new customer into the current routing plan. Yet, if time dependent travel times have changed due to an unexpected incident(s), in order to fulfill time window constraints and achieve a lower travel cost objective, scheduled incoming customers must be re-scheduled based on the position and loading of en-route vehicles. The dispatching center needs a very quick answer in order to respond in real-time to both the real-time demands and time-dependent travel times.

In order to capture the dynamic characteristics of the RT-TDVRPTW, they employed the concept of time rolling horizon and hence adopt a series of mixed integer programming submodels to formulate the problem. Each submodel represents a special VRPTW at a particular point in time when travel times and/or demands have changed. Each special VRPTW involves vehicles that may depart from current customers (called critical nodes, not the actual depot) and return to the actual depot, because those vehicles are already on their routes when the travel times and/or demands have changed. The goal of the RT-TDVRPTW at any particular time is to find a set of minimum cost vehicle routes, which originate from the critical nodes or the depot, $N_c(\tau) \cup \{0\}$, visit each unserviced node, $N_u(\tau)$, and terminate at the depot. The cost of the vehicle routes consists of travel times and waiting times, and each waiting time resulting from early arrival before a time window and/or from delaying departure after a service has finished.

Time-dependent travel time and critical node are two important characteristics of the RT-TDVRPTW that need to be discussed in more detail, as follows:

(a) Time-dependent travel times are characterized by a step-wise function, which represents predictive travel times at different time intervals. Once an unexpected incident happens, the predictive travel times need to be updated in real time. As a result, the scheduled vehicle routes need to be updated as

well. Note that the number of time intervals and the travel time in each time interval may change if non-recurrent congestion occurs.

(b) Once a customer has been serviced or the vehicle has left, that customer will be removed from the planned route thereafter. Only customers who have not been serviced will be considered in the planning of vehicle routes. Critical node plays an important role in distinguishing between a serviced customer and an unserviced customer. A critical node is defined as a customer who is currently using a vehicle or to whom a vehicle is heading. Critical nodes need to be identified instantly when a real-time demand or travel time has changed so that the route can be reconstructed.

II.10. SUGGESTED GA PARAMETERS

Talbi (2009) suggested following parameters for GA in Metaheuristics From Design to Implementation.

Mutation probability p_m**:** A large mutation probability will disrupt a given individual and the search is more likely random. Generally, small values are recommended for the mutation probability ($p_m \in [0.001, 0.01]$). Usually, the mutation probability is initialized to 1/k where k is the number of decision variables. Hence, on average, only one variable is mutated.

Crossover probability p_c**:** The crossover probability is generally set from medium to large values (e.g., [0.3, 0.9]).

Population size: The larger is the size of the population; the better is the convergence toward "good" solutions. Sampling errors are more important in smaller populations. However, the time complexity of EAs grows linearly with the size of the population. A compromise must be found between the quality of the obtained solutions and the search time of the algorithms. In practice, the population size is usually between 20 and 100. Some theoretical results indicate that the size of the population should grow exponentially with the size of individuals, which lead to very large populations to be practical.

CHAPTER III.

THE STUDY

In this chapter first of all the mathematical model for TDVRP is explained. In the following sections the solution approach is mentiones in detail.

III.1. TDVRP MATHEMATICAL MODEL

$G = (V, A)$ be a graph where $A = \{(v_i, v_j) : i \neq j \wedge i, j \in V\}$

$V(v_{0,\ldots,} v_{n+1})$ is the vertex set

$C(v_{1,\ldots\ldots} v_n)$ specifies a set of n customers

q_i and g_i associated demand and service times for vertices

d_{ij} is distance for arc v(i,j)

a_i and b_i are arrival and departure times for node i

$t_{ij}(b_i)$ is a function of departure time from node i

$|e_i, l_i|$ is the time window for node i

x_{ij}^k binary decision variable if the vehicle k travels from i to j

y_i^k service start time for customer i served by vehicle k

Primary Objective

Minimize $\sum_{k \in K} \sum_{j \in C} x_{0_j}^k$ (1)

Secondary Objective

Minimize $C_d \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^k + C_t \sum_{k \in K} \sum_{j \in C} (y_{n+1}^k - y_0^k) x_{0_j}^k$, (2)

Subject to $\sum_{i \in C} q_i \sum_{j \in V} x_{ij}^k \leq q_{\max}, \forall k \in K$ (3)

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \ \forall i \in C \quad (4)$$

$$\sum_{i \in V} x_{il}^k - \sum_{i \in V} x_{lj}^k = 0, \ \forall l \in C, \forall k \in K \quad (5)$$

$$x_{i0}^k = 0, x_{n+1,i}^k = 0, \ \forall i \in V, \forall k \in K \quad (6)$$

$$\sum_{j \in V} x_{0j}^{k} = 1, \forall k \in K \quad (7)$$

$$\sum_{j \in V} x_{j,n+1}^{k} = 1, \forall k \in K \quad (8)$$

$$e_i \sum_{j \in V} x_{ij}^{k} \leq y_i^{k}, \forall i \in V, \forall k \in K \quad (9)$$

$$l_i \sum_{j \in V} x_{ij}^{k} \geq y_i^{k}, \forall i \in V, \forall k \in K \quad (10)$$

$$x_{i,j}^{k}(y_i^{k} + g_i + t_{i,j}(y_i^{k} + g_i)) \leq y_j^{k}, \forall (i,j) \in A, \forall k \in K \quad (11)$$

$$x_{ij}^{k} \in \{0,1\}, \forall (i,j) \in A, \forall k \in K \quad (12)$$

$$y_i^{k} \in R, \forall i \in V, \forall k \in K \quad (13)$$

The primary and secondary objectives are defined by (1) and (2) respectively. The constraints are defined as follows: vehicle capacity cannot be exceeded (3); all customers must be served (4); if a vehicle arrives at a customer it must also depart from that customer (5); routes must start and end at the depot (6); each vehicle leaves from and returns to the depot exactly once (7) and (8) respectively; service times must satisfy time window start (9) and ending (10) times; and service start time must allow for travel time between customers (11). Decision variables type and domain are indicated in (12) and (13).

III.2. ENCODING OF SOLUTIONS

The first step in developing a genetic algorithm for solving the problem is to demonstrate all the solutions in a form of bits sequences having the same dimension. Each of the sequence symbolizes the random node in the possible solution space of the problem (Yeniay, 2001:38). Encoding the parameters enables the problem-specific data to convert into a form used by genetic algorithm (Jang, 1997: 176).

In this study, each solution is represented by the chromosome composed of two gene set. The first gene set represents the nodes to be visited, the second gene set represents the assigned vehicles; in other words the vehicles which meet the demands. The gene structure is shown in Figure 3.1 below:

1st Gene	N1	N2	……	N(n-1)	N(n)
2nd Gene	N1	N2	……	N(n-1)	N(n)

Figure 3.1 Gene Structure

An Example of this representation is given in Figure 3.2. Within this solution, an 8-customer-problem with 3 vehicles is represented. Vehicle 0 visits customers 5 and 2, Vehicle 1 visits customers 8, 1, 7, 3 and 4, and Vehicle 2 visits customer 6 just in the order given in the solution. This representation gives a rich manipulation power to crossover and mutation operators. As it will be explained in the next sections, either vehicle gene set can be changed without changing the customer sequence or changing the customer sequence yields to different solutions.

5	**2**	**8**	**1**	**7**	**3**	**4**	**6**
0	**0**	**1**	**1**	**1**	**1**	**1**	**2**

Figure 3.2 An Example of Gene Structure

III.3. CREATING THE INITIAL POPULATION

A solution group is created where the possible solutions are encoded. The solution group is called population and the solution code is called chromosome. Random number generators may be used for creating first population when demonstrating chromosomes using binary alphabet. Random number generator is called and if the value is less than 0,5 the position is set to 0, if more, the position is set to 1 (Yeo ve Agyel, 1998: 269). The position value can be determined with heads or tails method in problems where the length of the individual number and the chromosomes are shorter.

Except the binary encoding methods in genetic algorithms, different encoding methods are used depending on the problem to be solved (Goldberg, 1989: 9).

In this study integer encoding is used. While creating the initial population, beside the random solutions, solutions coming from 4 simple heuristic algorithms are also created. Although they improve the quality of the solution, it is limited in order to avoid an early convergence.

The four moderate structured solutions in the initial population are as follows:

Structured Solution 1

This is a nearest neighbour solution applied to one vehicle at a time. When the next customer demand violates the capacity constraint of the vehicle, that vehicle returns back to depot and the next vehicle starts from the depot to serve the nearest unvisited customer.

1. Current vehicle = vehicle 1
2. Current position = depot
3. For current vehicle find nearest neighbour from current position which is not served
4. If capacity of vehicle is available
   - fulfill demand
   - label customer as served
   - current position = customer
   - go to 3

   else

   vehicle = next vehicle
   - go to 2
5. Fulfill the remaining demand with the unused capacities

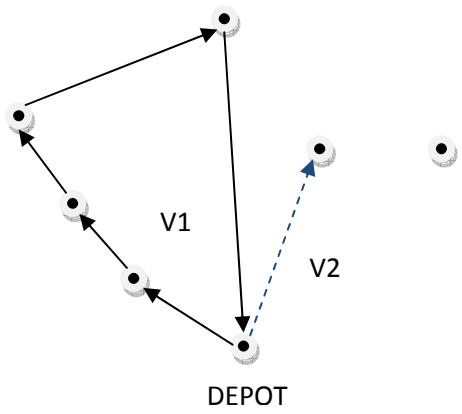

Figure 3.3 Structured Solution 1

Structured Solution 2

This solution is similar to solution 1 except here, vehicles start one after another. In other words, vehicle 2 does not wait for vehicle 1 to select all customers. Next customer for each vehicle is selected at once. When a vehicle uses its capacity it returns to depot and the others continue visiting the unserved customers.

1. For each vehicle Current position = depot
2. For each available vehicle find nearest neighbour from current position which is not served
3. For each available vehicle

   If capacity of vehicle is available

   - fulfill demand
   - label customer as served
   - Find new current position
   - go to 2

   else

   vehicle is not available
4. Fulfill the remaining demand with the unused capacities

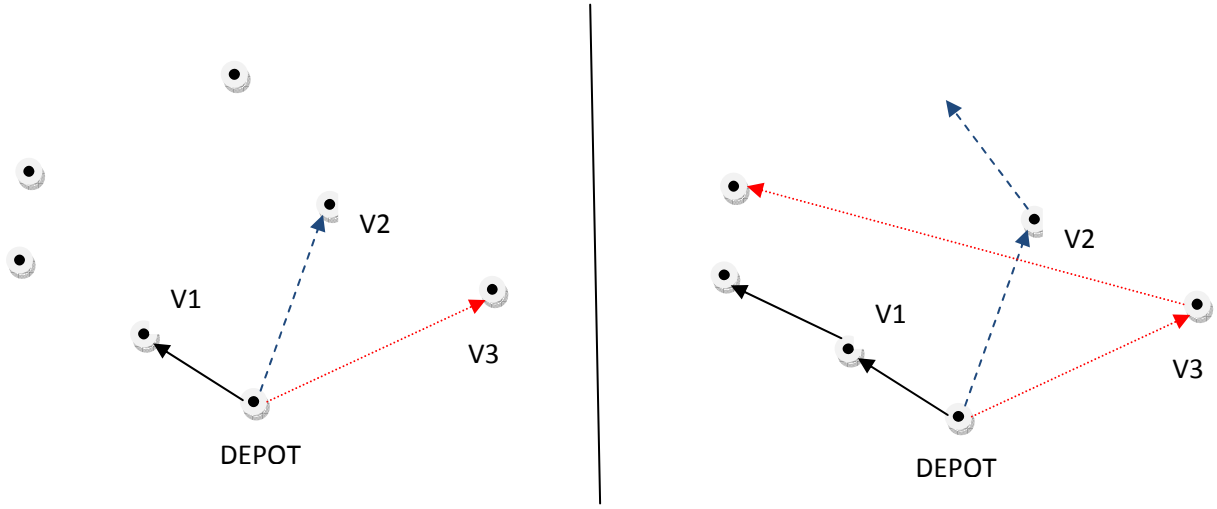

Figure 3.4 Structured Solution 2

Structured Solution 3

This solution again a nearest neighbour solution applied to one vehicle at a time. The difference is that, it is built using travels from start and from end of the vehicle's tour. At the beginning two customers are selected that are nearest to the depot. One of them can be assumed the first customer to serve while the other one as the last customer to serve. Then forward and backward assignments take place in turn. When the next customer demand violates the capacity constraint of the vehicle, the current forward and backward subtours are linked and the whole tour for the vehicle is built.

1. Current vehicle = vehicle 1
2. Current positions = depot
3. Current positiont = depot
4. For current vehicle find nearest neighbour from current positions and current positiont which is not served
5. If capacity of vehicle is available
   - fulfill demand
   - label customer as served
   - update **current positions** and **current positiont**
   - go to 4

   else

   vehicle = next vehicle
   - go to 2
6. Fulfill the remaining demand with the unused capacities

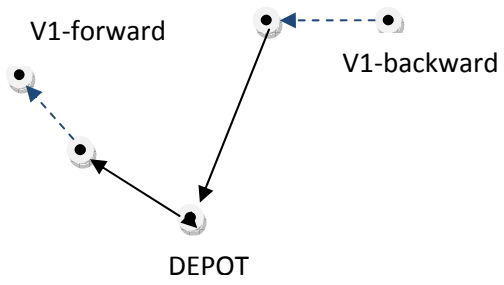

Figure 3.5 Structured Solution 3

Structured Solution 4

This solution is similar to solution 3. Instead of handling one vehicle at a time all vehicles start their forward and backward subtours at the same time.

1. For each vehicle Current positions = depot
2. For each vehicle Current positiont = depot
3. For each available vehicle find nearest neighbour from current positions and current position which is not served

4. For each available vehicle

   If capacity of vehicle is available

   - fulfill demands
   - label customers as served
   - Find new current positions and current positiont
   - go to 2

   else

   set vehicle as unavailable
5. Fulfill the remaining demand with the unused capacities

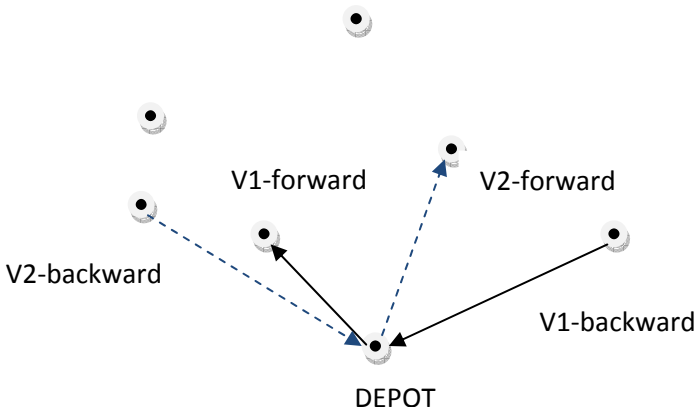

Figure 3.6 Structured Solution 4

III.4. CALCULATION OF FITNESS VALUE

The first step after creating a generation is to calculate the fitness value of each member in the population. E.g. the fitness value f(i) of i. member for maximation problem is usually the value of the objective function at that node (Jang, 1997: 176). A fitness function is available for each problem to be solved. The fitness function for a given particular chromosome gives a numerical fitness value which is propotional to the usage or ability of the solution that the chromosome represents. This data leads to an appropriate selection in each generation. The higher the fitness value, the higher chances of the survival and the reproduction will take place and the rate of representing will be higher in the next generation (Yeniay, 2001: 38-39).

There is a decision variable in this study determining whether the time dependency values will be used or not. If it is to be used the objective function consists of 2 terms, one is for the cost coming out from the travel distances and the other is for the time cost. If constant speeds are used, in other words time dependency is not taken into attention, the only term in the objective function and also fitness function is the cost for travel distances.

III.5. IMPLEMENTATION OF REPRODUCTION PROCESS

The sequences in the reproduction operator are copied according to objective function and individuals are selected that transfers the good genetic charachteristics (inherited attributes) better to the next generation. The reproduction operator is an artificial selection. Copying of the sequences according to the fitness value denotes the sequence with a higher fitness value is likely to contribute to one or more offspring in the next generation. The reproduction consists of following processes; selection of individuals, copying the selected individuals to a match-up pool and grouping the individuals in pairs in the pool (Fığlalı, 2002: 3). After calculating the fitness value, a new population should be created from the current generation. The selection process determines which families should take place in order to generate offspring for the next generation. This is similar to the condition where the most appropriate one survives in the natural selection. The purpose of this method is to provide the opportunity to generate over the value of average fitness. The chance of copying a sequence depends on the fitness value of the sequence that is calculated with fitness function (Jang, 1997: 176).

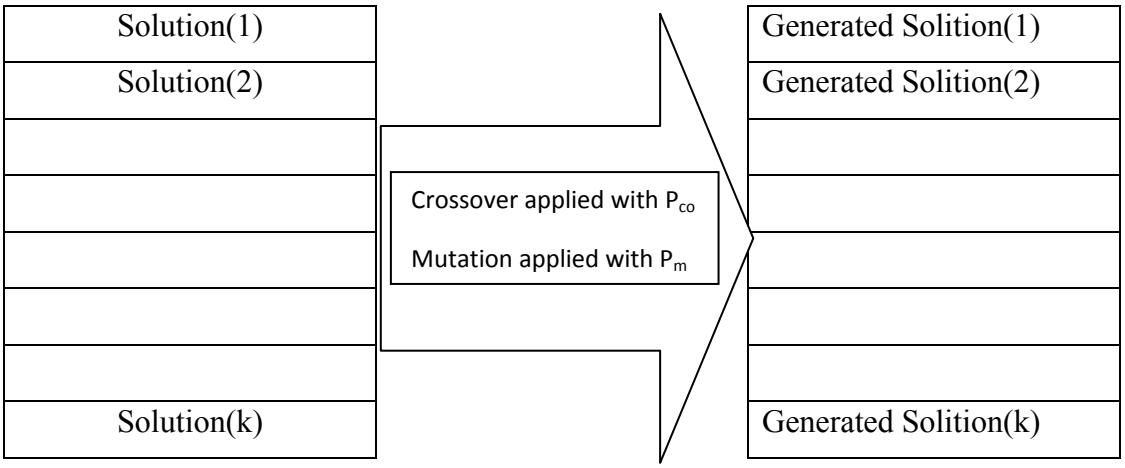

Figure 3.7 Reproduction Process

Generation of the next population consists of two phases. In the first phase using cross-over and mutation operators, potential solutions of the next population are created. In the second phase using a certain rate of elitism, solutions are passed to the next population, from the current population and the potential solutions created. The rest of the solutions are taken randomly from these two sets.

Elitism replacement scheme is applied in this study, as follows: both parent and offspring population are combined into a single population and sorted in a non-increasing order of their associated fitness value. Then, the first half of the combined population is selected as the individuals of the new population for the next generation. We propose the filtration strategy to identify identical individuals from the new population. The identical individuals will be removed and replaced by uniformly randomly generated new individuals to avoid premature convergence and to add diversity to the new population.

During this process no duplications are allowed. Once the next generated population is formed it becomes the current population and the next iteration is ready to start at this point. Reproduction Process is shown in Figure 3.7.

III.6. IMPLEMENTATION OF CROSSOVER

Crossover operator is used to research the potential of the current gene pool in order to create new chromosomes with better quality than the previous generation. Crossover operator, which is one of the most important parameters affecting the performance of the genetic algorithm, corresponds to the crossover in the natural

population. Two chromosomes are selected randomly from the new population obtained from the crossover process and they are subjected to mutually crossover process. The sequence is subjected to crossover according to the integer value. Single-point crossover method is the simplest method. To perform single-point crossover, both chromosomes shall have the same gene lenght. In 2-point; crossover chromosome is cut in two points and the positions are replaced mutually (Fığlalı, 2002: 3).

In this study, a more complicated crossover method is suggested which is expected to pass common features of the parents to childeren more effectively.The algorithm for this method is as follows:

1. Find separetely the sets of nodes which are not assigned from parent chromosomes 1 and 2, which are served by the same vehicle.
   Let $S_{11}, S_{12},\ldots,S_{1n}$ *and* $S_{21}, S_{22},\ldots,S_{2m}$ be the defined sets
2. Regardless of vehicle number, find the intersection of 2 sets from different set groups with maximum cardinality ($\max(|S_{1k} \cap S_{2l}|$ for all k,l) .
3. If there is no set determined at step 2 with at least a cardinality of 2 go to step 5.
4. Assign the genes of chromosome, which are included in the set determined in step 2, to child 1 chromosome at gene positions like exactly they are in parent 1 chromosome. Label these genes as assigned in parents. Go to step 1.
5. Apply all steps switching the order of parents in order to generate child 2 chromosome.

III.7. IMPLEMENTATION OF MUTATION PROCESS

Crossover is used to research the potentials of current gene. If the population doesn't contain all necessary encoded data; the crossover cannot produce satisfactory solution. For this reason, an operator is required capable to produce new chromosomes from the current chromosomes. Mutation realizes this task. In artificial gene systems mutation operator provides a protection against the loss of a good solution that can't be obtained any more (Goldberg, 1989: 14). Mutation, in problems using binary encoding system converts a bit value to the other bit (0 or 1). Different mutation methods are used in the problems where binary encoding system is not

used. Regeradless which method is used, the geneal purpose of the mutation is to provide and maintain the genetic diversity (Braysy, 2001: 54).

In this study, 2 different mutation operators are used with different probabilities.In the first operator, one of the genes in the 2nd gene set (vehicles) is switched while no change is performed in the 1st gene set (nodes) of the represented chromosome as shown in Figure 3.8. The second mutation operator swaps 2 genes in the 1st gene set (nodes) while no change is performed in the 2nd gene set (vehicles), as shown in Figure 3.9.

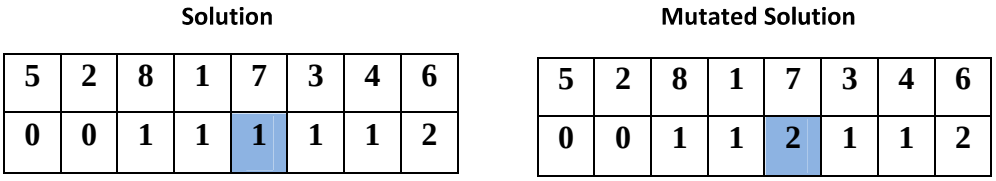

Figure 3.8 Mutation Operator 1

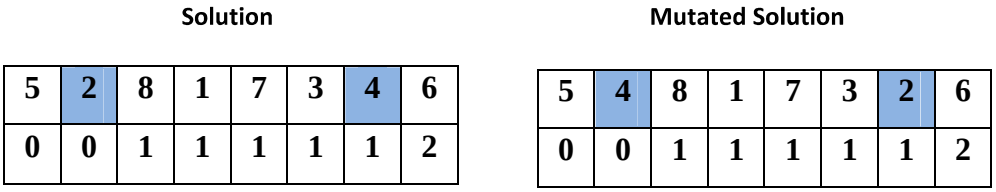

Figure 3.9 Mutation Operator 2

III.8. CREATING NEW GENERATION AND STOPPING CYCLE

New generation is defined after reproducing, crossover and mutation processes and they will be the parents of the next generation. The process continues with the new generation with determined fitness for reproducing. This process continues until predetermined number of generation or until reaching a destination or providing another stopping criterion (Yeo ve Agyel, 1998: 271).

According to the required sensivity degree, number of iterations can be determined and at that point the cycle can be stopped. Stopping criterion could be number of iteriations or targeted fitness value as well (Fung et al., 2001: 270).

III.9. TIME DEPENDENCY APPROACH

Throughout the readings, it is examined that TDVRP solutions are generated from models using time dependent travel times. Here we will introduce time dependent speeds which will more easily overcome the so called FIFO rule violations, which will also present more realistic problem nature.

The idea is to define a downtown region first, which is assumed to have more traffic congestion than other regions of the city. Distribution of nodes and downtown regions for the two problems studied are shown in Figure 3.10 and Figure 3.11.

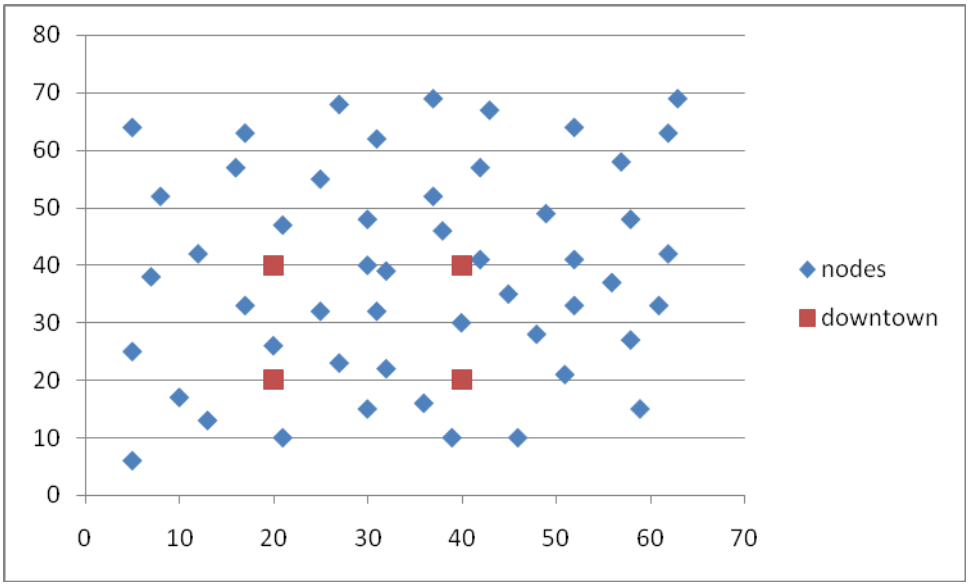

Figure 3.10 Nodes and Downtown Region for E051

There are initial minimum travel speeds assigned to each (i,j) couple at time t=0, which represents the earliest time that vehicles can leave depot. The schedule is assumed to start at 8 a.m. and end at 4 p.m. which indicates a travel time of maximum 8 hours for each vehicle. If a travel takes place from i to j with a certain vehicle, the current time of the vehicle and t(i,j) function will be used to determine arrival time of the vehicle to customer j as shown in Figure 3.12.

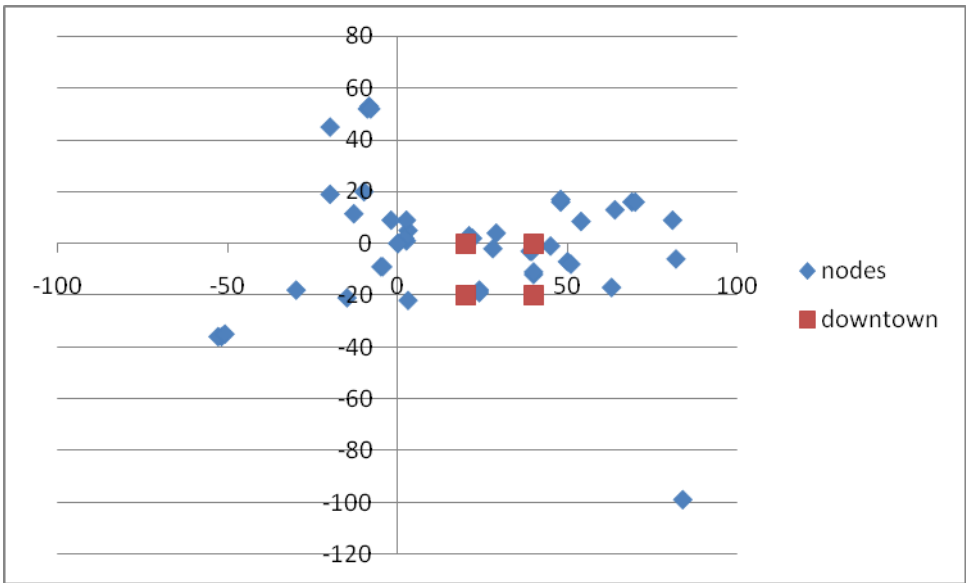

Figure 3.11 Nodes and Downtown Region for E045

The parameters of speed function are defined below. Formulas (3.1) and (3.2) are used to find the travel times.

$minspeed_{ij}$: speed of the vehicle, travelling the arc (i,j), at time t=0

$maxspeed_{ij}$: speed of the vehicle, travelling the arc (i,j), at time t=240

d_{ij} : distance of the arc (i,j)

t_{di} : the time the assigned vehicle departs from node i

t_{aj} : the time the assigned vehicle arrives node j

v_i : starting speed of the vehicle travelling the arc (i,j), at time $t=t_{di}$

v_j : speed of the vehicle travelling the arc (i,j), at time $t=t_{aj}$

$tt_{ij} = t_{aj} - t_{di}$

t_{d0} = 0 for all vehicles at the beginning of the schedule.

t_{aj} values are extracted from the equations (1) and (2) and they are used as the departure time from customer j for the next travel, namely arc (j,k).

tt_{ij} values are added to the time cost of the specified vehicle.

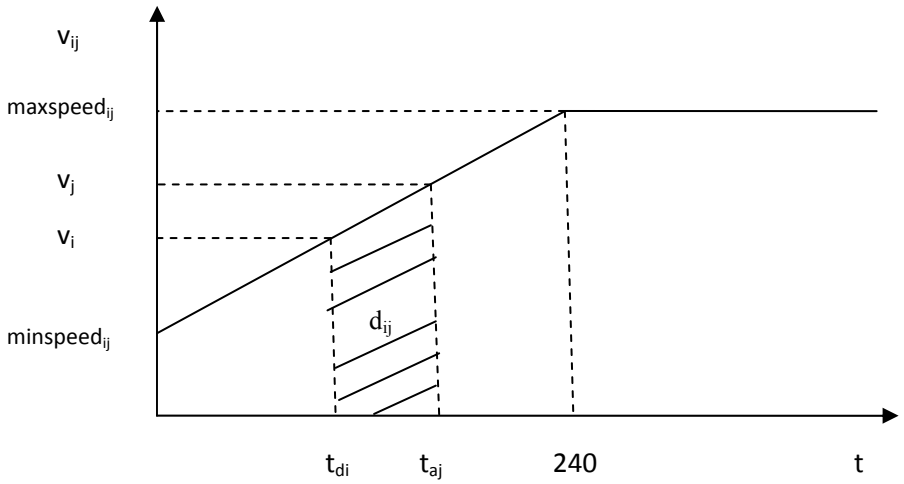

Figure 3.12 Time dependent speed function

$$d_{ij} = \int_{t_{di}}^{t_{aj}} v_{ij}(t)\,dt \tag{3.1}$$

$$v_{ij}(t) = \begin{cases} \text{minspeed}_{ij} + \dfrac{\text{maxspeed}_{ij} - \text{minspeed}_{ij}}{240} * t, & \text{if } t < 240 \\ \text{max speed}_{ij} & \text{otherwise} \end{cases} \tag{3.2}$$

CHAPTER IV.

RESULTS AND DISCUSSION

The GA parameters applied for the problems solved are as follows:

Probability of crossover = 0.75

Probability of mutation = 0.10

Population size = 30

Iteration number = 30 000

GA constructed in this study is applied to problems in VRPLIB library http://www.or.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html. for **45-**node problem in **E045-04f** file, **51-**node problem in **E051-05e** file, **72-**node problem in **E072-04f** file and **101-**node problem in **E101-08e** file are solved and the results are as follows.

These 4 problems are originally CVRP problems and they are solved without TD function. The purpose is to observe the performance of GA generated.

Table 4.1 Results of E045-04f

vehicle0 0-15-1-2-16-10-18-11-12-13-14-17-0 **tour distance = 131.75908174275** **demand fulfilled by vehicle = 1634**
vehicle1 0-37-36-34-31-32-30-41-40-39-42-38-0 **tour distance = 116.830474320565** **demand fulfilled by vehicle = 2004**
vehicle2 0-24-9-4-3-35-7-5-6-27-29-33-28-44-43-8-0 **tour distance = 358.58527707896** **demand fulfilled by vehicle = 2007**
vehicle3 0-21-20-19-22-23-26-25-0 **tour distance = 131.348026324701** **demand fulfilled by vehicle = 1575**
total tour distance = 738.522859466976

Table 4.2 Best known solution for E045-04f

vehicle0 0-24-16-2-1-15-9-0 **tour distance = 24.7238057688087** **demand fulfilled by vehicle = 1612**
vehicle1 0-8-19-22-23-26-25-20-21-0 **tour distance = 147.543835803483** **demand fulfilled by vehicle = 2010**
vehicle2 0-43-44-28-33-29-27-6-5-7-35-3-4-14-13-12-11-18-17-10-0 **tour distance = 437.806666434634** **demand fulfilled by vehicle = 1594**
vehicle3 0-37-38-36-39-40-34-31-32-30-41-42-0 **tour distance = 113.466933302579** **demand fulfilled by vehicle = 2004**
total tour distance = 723.541241309505

For the first problem distance cost found is 738.52 and the best solution cost is 723.54 as shown in Table 4.1 and Table 4.2. For the second problem distance cost found is 589.77 and the best solution cost is 524.61 as shown in Table 4.3 and Table 4.4. These 2 problems are also solved using TD function and downtown regions defined. At this point vehicle speeds are assumed as constant.

There are 2 other problems solved and obtained results and compared with best solutions in this section, namely **E101-08e** and **E072-04f** **.** These problems are solved in order to test the performance of the algorithm in larger size problems and it is examined that solution quality is not as good as smaller size problems. One way may be to increment the population size and number of iterations which will yield to very long computation times. The other way is to increment the number of structured solutions in the initial population. Meanwhile using better quality structured slutions may also help GA to converge to better solutions in earlier iterations compared to previous approach.

Table 4.3 Results of E051-05e

vehicle0 0-37-15-45-44-4-18-47-12-46-0 **tour distance = 88.1769422915354** **demand fulfilled by vehicle = 154**
vehicle1 0-17-42-19-40-41-13-25-14-24-6-0 **tour distance = 127.928217046455** **demand fulfilled by vehicle = 156**
vehicle2 0-27-1-22-28-31-8-26-7-43-23-48-0 **tour distance = 117.229675544674** **demand fulfilled by vehicle = 148**
vehicle3 0-11-2-29-21-16-50-34-30-9-38-0 **tour distance = 99.3330656854988** **demand fulfilled by vehicle = 159**
vehicle4 0-5-49-10-33-39-20-35-36-3-32-0 **tour distance = 157.102425549412** **demand fulfilled by vehicle = 160**
total tour distance = 589.770326117575

Table 4.4 Best known solution for E051-05e

vehicle0 0-38-9-30-34-50-16-21-29-2-11-0 **tour distance = 99.3330656854988** **demand fulfilled by vehicle = 159**
vehicle1 0-32-1-22-20-35-36-3-28-31-26-8-0 **tour distance = 118.519085054731** **demand fulfilled by vehicle = 149**
vehicle2 0-27-48-23-7-43-24-25-14-6-0 **tour distance = 98.4517211873084** **demand fulfilled by vehicle = 152**
vehicle3 0-18-13-41-40-19-42-17-4-47-0 **tour distance = 109.056029802896** **demand fulfilled by vehicle = 157**
vehicle4 0-12-37-44-15-45-33-39-10-49-5-46-0 **tour distance = 99.251244912073** **demand fulfilled by vehicle = 160**
total tour distance = 524.611146642507

Table 4.5 Results of E101-08e

vehicle0 0-26-4-72-74-22-41-15-43-38-86-82-7-52-0 **tour distance = 153.489033634226** **demand fulfilled by vehicle = 188**
vehicle1 0-27-69-1-50-76-33-81-9-51-20-30-70-0 **tour distance = 95.6136267009963** **demand fulfilled by vehicle = 156**
vehicle2 0-28-80-68-77-3-79-78-34-35-71-65-66-0 **tour distance = 126.71564504208** **demand fulfilled by vehicle = 193**
vehicle3 0-18-83-60-5-84-8-48-62-88-31-29-24-12-0 **tour distance = 152.293368577852** **demand fulfilled by vehicle = 190**
vehicle4 0-58-2-57-87-42-14-44-61-19-11-63-90-32-10-0 **tour distance = 159.283310516373** **demand fulfilled by vehicle = 195**
vehicle5 0-89-6-59-98-85-93-99-96-94-0 **tour distance = 52.0124678059376** **demand fulfilled by vehicle = 166**
vehicle6 0-13-95-97-92-37-100-91-16-17-45-46-47-36-49-64-0 **tour distance = 151.774825655317** **demand fulfilled by vehicle = 192**
vehicle7 0-53-40-21-73-75-56-23-67-39-25-55-54-0 **tour distance = 106.854967770691** **demand fulfilled by vehicle = 178**
total tour distance = 998.037245703473 **total tour distance for known best solution = 826.136**

Table 4.6 Results of E072-04f

vehicle0 0-35-11-18-71-0 **tour distance = 33.5366474419152** **demand fulfilled by vehicle = 29789**
vehicle1 0-36-31-32-34-54-55-57-56-60-61-58-59-63-62-64-65-66-67-69-37-38-40-68-39-41-0 **tour distance = 72.1266027498241** **demand fulfilled by vehicle = 27222**
vehicle2 0-20-29-23-24-25-26-30-21-22-28-27-42-44-43-46-53-45-52-48-47-50-70-51-49-0 **tour distance = 71.5279781670932** **demand fulfilled by vehicle = 28236**
vehicle3 0-33-14-19-15-2-17-16-12-13-1-6-10-5-3-8-4-7-9-0 **tour distance = 82.171571532883** **demand fulfilled by vehicle = 29593**
total tour distance = 259.362799891716 **total tour distance for known best solution = 241.973**

In this part for the first 2 problems that are already solved without TD, speed functions and downtown areas are defined and the problems are solved using these parameters. Besides distance costs time costs are generated for each solution and GA used the total cost throughout its selection and reproduction processes. It is observed that best known solutions without taking TD parameters into account did not perform well and the results that come out from their schedules generated higher costs than GA used in this study.

Although the distance costs of problems **E045-04f** and **E051-05e** are lower than the ones obtained here, time costs of them are higher than the ones obtained in the study. For the problem **E045-04f** even the total cost is higher than the solution obtained using GA. For **E045-04f** and **E051-05e** problems downtown is defined as the rectangular area identified by points (20,-20), (20,0), (40,-20) and (40,0) ; (20,20), (20,40), (40,20) and (40,40) respectively. The solutions are shown in Table 4.7 and Table 4.8 for problem **E045-04f** and in Table 4.9 and Table 4.10 for problem **E051-05e.** Nodes in brackets represent the customers located in the downtown region.

Table 4.7 Results of E045-04f as TDVRP

vehicle0 0-36-34-31-32-30-41-40-39-42-38-37-0 **tour distance = 117.366270436874** **demand fulfilled by vehicle = 2004**
vehicle1 0-24-9-4-3-35-7-5-6-27-29-33-28-44-43-8-0 **tour distance = 358.58527707896** **demand fulfilled by vehicle = 2007**
vehicle 2 0-15-1-2-16-10-18-11-12-[13]-[14]-[17]-0 **tour distance = 131.75908174275** **demand fulfilled by vehicle = 1634**
vehicle 3 0-21-20-19-22-23-[26]-[25]-0 **tour distance = 131.348026324701** **demand fulfilled by vehicle = 1575**
total tour distance = 739.058655583285 **total tour time = 813.067299444172** **total tour cost = 1552.12595502746**

Table 4.8 Best known solution for E045-04f as TDVRP

vehicle 0 0-24-16-2-1-15-9-0 **tour distance = 24.7238057688087** **demand fulfilled by vehicle = 1612**
vehicle 1 0-8-19-22-23-26-25-20-21-0 **tour distance = 147.543835803483** **demand fulfilled by vehicle = 2010**
vehicle2 0-43-44-28-33-29-27-6-5-7-35-3-4-14-13-12-11-18-17-10-0 **tour distance = 437.806666434634** **demand fulfilled by vehicle = 1594**
vehicle 3 0-37-38-36-39-40-[34]-[31]-32-[30]-[41]-[42]-0 **tour distance = 113.466933302579** **demand fulfilled by vehicle = 2004**
total tour distance = 723.541241309505 **total tour time = 932.567091919754** **total tour cost = 1656.10833322926**

Table 4.9 Results of E051-05e as TDVRP

vehicle0 0-23-7-26-8-31-28-3-36-35-20-0 **tour distance = 129.214691067339** **demand fulfilled by vehicle = 157**
vehicle1 0-[6]-48-32-11-38-5-12-47-46-0 **tour distance = 83.6949178150345** **demand fulfilled by vehicle = 158**
vehicle2 0-16-50-9-[30]-34-21-29-2-22-1-27-0 **tour distance = 107.615253551369** **demand fulfilled by vehicle = 155**
vehicle3 0-49-10-39-33-45-15-[44]-37-17-4-18-0 **tour distance = 115.343934173464** **demand fulfilled by vehicle = 158**
vehicle4 0-42-19-40-41-13-[25]-[14]-24-43-0 **tour distance = 149.363260259283** **demand fulfilled by vehicle = 149**
total tour distance = 585.23205686649 **total tour time = 695.320067834617** **total tour cost = 1280.55212470111**

Table 4.10 Best known solution for E051-05e as TDVRP

vehicle0 0-38-9-30-34-50-16-21-29-2-11-0 **tour distance = 99.3330656854988** **demand fulfilled by vehicle = 159**
vehicle1 0-32-[1]-22-20-35-36-[3]-28-31-26-8-0 **tour distance = 118.519085054731** **demand fulfilled by vehicle = 149**
vehicle2 0-27-48-23-7-43-24-25-14-6-0 **tour distance = 98.4517211873084** **demand fulfilled by vehicle = 152**
vehicle3 0-18-13-41-40-19-42-[17]-4-47-0 **tour distance = 109.056029802896** **demand fulfilled by vehicle = 157**
vehicle4 0-12-37-44-15-45-33-[39]-[10]-49-5-46-0 **tour distance = 99.251244912073** **demand fulfilled by vehicle = 160**
total tour distance = 524.611146642507 **total tour time = 690.633025094999** **total tour cost = 1215.24417173751**

CHAPTER V.

CONCLUDING REMARKS AND RECOMMENDATIONS

There is a limited numbers of researches in the TDVRP area, and therefore significant contribution for this problem can be performed.

Future studies can contribute to this problem in two directions. One is to study TD (Time Dependency) in depth. The model suggested in this thesis can be improved and it can evolve to a more realistic model, which will represent the real world cases. Throughout the studies surveyed it is examined that several TD models are suggested but still it needs time to have a generally accepted model and a benchmark data related to this model.

The other direction could be to develope new heuristic approaches, since TDVRP differentiate a lot from the classical VRP. Future studies will show which classes of algorithms will perform good for TDVRP. In this thesis GA is suggested as one of the powerfull algorithms for TDVRP. Hybriding GA with other heuristic and metaheuristic algorithms can have much better performances in the future studies.

In real world applications there are areas, other than logistics, with high operational costs where TDVRP models and solutions can be applied and benefited, such as energy & telecommunication.

In this study, compared to previous studies a more realistic TD model is suggested, however it is realized that it is hard to collect data related to TD speed functions of internodes. A realtime speed tracking system may be used to feed the model, which will improve the accuracy of speed functions. A project in the industry can show how efficiently this model can be implemented due to the fact that most of the schedules must be accomplished in a limited time.

Future studies related to TDVRP will be about the following topics

- More than one high congestion regions
- Location of depot or depots if MDTDVRP is studied

- Behaviour of the algorithm throughout different types of TD speed functions.
- Adding time window constraints to the TDVRP model.

REFERENCES

Aggarwal C.C., J.B. Orlin, R.P. Tai, Optimized crossover for the independent set problem, Operations Research 45 (**1997**) 226–234.

Ahn, B.H. and J.Y. Shin, *"Vehicle-routeing with time windows and time-varying congestion. The journal of the Operational Research Society", **1991**. 42(5): p. 393-400.*

ANWB Reisinformatie. Dagelijks drukke trajecten. World Wide Web, last checked on January 14th **2010** /http://www.anwb.nl/verkeer/nederland/verkeersinformatie,/verkeersverwachting/Dagelijkse-drukke-trajecten.htmlS [in Dutch].

Assad, A.A. (**1988**). Modeling and implementation issues in routing. In Golden, B.L. and Assad, A.A., editors, Vehicle Routing: Methods and Studies, volume 16 of Studies in Management Science and Systems, chapter 2, pages 7-45. North-Holland, Amsterdam.

B.M. Baker, M.A. Ayechew / Computers & Operations Research 30 (**2003**) 787–800

Balas E., W. Niehaus, Optimized crossover-based genetic algorithms for the maximum cardinality and maximum weight clique problems, Journal of Heuristics 4 (**1998**) 107–122.

Beasley JE, Chu PC. Constraint handling in genetic algorithms: the set partitioning problem. Journal of Heuristics **1998**;4:323–57.

Blanton JL, Wainwright RL. Multiple vehicle routing with time and capacity constraints using genetic algorithms. Proceedings of the Fifth International Conference on Genetic Algorithms **1993**:452–459. Los Altas, CA.

Braysy O. (**2001**), Local Search and Variable Neighborhood Search Algorithms for The Vehicle Routing Problem With Time Windows, PH D Thesis.

Bullnheimer B, Hartl RF, Strauss C. An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research **1999**;89:319–28.

Chang, M.S., Chen, S.R., Hsueh, C.F., (**2003)**. Real-time vehicle routing problem with time windows and simultaneous delivery/pickup demands. Journal of the Eastern Asia Society for Transportation Studies 5, 2273–2286

Chu PC, Beasley JE. Agenetic algorithm for the generalised assignment problem. Computers & Operations Research **1997**;24:17–23

Clarke G., Wright J. W. (**1964**), *"Scheduling of Vehicles from a Central Depot to a Number of Delivery Points"*, Operations Research, c. 12, sf. 568-581

Dantzig G., Fulkerson R., Johnson S. (**1954**), *"Solution of a Large-Scale Travelling Salesman Problem"*, Journal of Operations Research Society, c.2, sf. 393-410

Darwin C. On the Origin of Species by means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. Online Book (**1859**) pp. 247.

English Highways Agency of the Department for Transport. Traffic Forecastes. World Wide Web, last checked on October 6th **2010** /http://www.highways. gov.uk/traffic/forecast.aspxS.

Fığlalı A. and Engin O. (**2002**), *"Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu"*, İTÜ Dergisi, s. 1-6.

Fung R.Y.K., Tang J. and Wang D. (**2001**), *"Extension Of A Hybrid Genetic Algorithm For Nonlinear Programming Problems With Equality And Inequality Constraints"*, Computers & Operations Research, Volume: 29, Issue: 3, s. 261-274.

Gambardella LM, Taillard R E, Agazzi G. A multiple ant colony system for vehicle routing problems with time windows. In: Corne D. Dorigo M, Glover F, editors. New ideas in optimization. New York: McGraw-Hill, **1999**. p. 63–76 [Chapter 5].

Gendreau M, Hertz A, Laporte G. A Tabu search heuristic for the vehicle routing problem. Management Science **1994**;40:1276–90.

Goldberg D.E. (**1989**), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, USA.

Hill, A., Mabert, V., and Montgomery, D. (**1988**). A decision support system for the courier vehicle scheduling problem. Omega, International Journal of Management Science, 16(4):333-345.

Hill, A.V. and W.C. Benton, *Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. The journal of the Operational Research Society, **1992**. 43(4): p. 343-351.*

Hiquebran DT, Alfa AS, Shapiro A, Gittoes DH. A revised simulated annealing and cluster-first route-second algorithm applied to the vehicle routing problem. Engineering Optimization **1994**;22:77–107.

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

Ichoua, S., Gendreau, M., and Potvin, J.-Y. (**2003**). Vehicle dispatching with time dependent travel times. European Journal of Operational Research, 144(2):379-396.

Jang J. S. R. (**1997**), Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence, Chapter 7: Derivative-Free Optimization, Prentice-Hall, USA, s. 173-196.

Laporte G, Osman I.H. Routing problems: a bibliography. Annals of Operations Research 1995;61:227–62

Kok AL, Hans EW, Schutten JMJ. Vehicle routing under time-dependent travel times: the impact of congestion avoidance; details of the computational tests, **2010** http://www.utwente.nl/mb/ompl/staff/Schutten/ICA/tests.pdf

Malandraki, C. and M.S. Daskin, *Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms. Transportation Science,**1992**. 26(3): p. 185-200.*

Malandraki, C. and R.B. Dial, *A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. European Journal of Operational Research, **1996**. 90(1): p. 45-55.*

Malandraki, C., *Time dependent vehicle routing problems: Formulations, solution algorithms and computational experiments. **1989**, Northwestern University.*

Nazif, H., L.S. Lee / Applied Mathematical Modelling 36 (**2012**) 2110–2117

Osman IH. Metastrategy simulated annealing and Tabu search algorithms for the vehicle routing problem. Operations Research **1993**;41:421–51.

Potvin JY, Bengio S. The vehicle routing problem with time windows—Part II: genetic search. INFORMS. Journal on Computing **1996**;8:165–72.

Potvin JY, DubRe D, Robillard C. Ahybrid approach to vehicle routing using neural networks and genetic algorithms. Applied Intelligence **1996**;6:241–52.

Potvin JY, Duhamel C, Guertin F. Agenetic algorithm for vehicle routing with backhauling. Applied Intelligence **1996**;6:345–55.

Reeves CR, editor. Modern heuristic techniques for combinatorial problems. Oxford: Blackwell Scientific Press, **1993**.

Renaud J, Boctor FF, Laporte G. An improved petal heuristic for the vehicle routing problem. Journal of the Operational Research Society **1996**;47:329–36.

Rochat Y, Taillard R E. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics **1995**;1:147–67.

Salhi S, Thangiah SR, Rahman F. Agenetic clustering method for the multi-depot vehicle routing problem. In: Smith GD, Steel NC, Albrecht RF, editors. ICANNGA'97, Vienna. New York: Springer, **1998**. p. 234–7.

Schmitt LJ. An empirical computational study of genetic algorithms to solve order-based problems: an emphasis on the TSP and the VRP with time constraints. Ph.D. dissertation, Fogelman College of Business and Economics, University of Memphis, **1994**.

Talbi E.G., **(2009)**, "Metaheuristics from Design to Implementation" p.221

Taillard R E. Parallel iterative search methods for vehicle routing problems. Networks **1993**;23:661–73

Taillard, E. D, (**1999**). A heuristic column generation method for the heterogeneous VRP. Operation Research – Recherche opérationelle 33, 1-14

Taillard, E. D, Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y. (**1997**). A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science 31, 170-186

Tan, K. C., Lee, L. H., Zhu, Q. L., and Ou, K. (**2001c**). Heuristic methods for vehicle routing problem with time windows. Artificial Intelligence in Engineering, 15(3):281–295.

Thangiah SR, Gubbi AV. Effect of genetic sectoring on vehicle routing problems with time windows. In: Proceedings of the IEEE International Conference on Managing Intelligent System Projects, Washington, DC, **1993**. p. 146–53.

Thangiah SR, Nygard PL, Juell PL. GIDEON: a genetic algorithm system for vehicle routing with time windows. In: Proceedings of the Seventh IEEE

Conference on Artificial Intelligence Applications, Miami, FL, **1991**. p. 322–8.

Thangiah SR, Nygard PL. School bus routing using genetic algorithms. In: Proceedings of the SPIE Conference on Applications of Arti2cial Intelligence X: Knowledge Based Systems, Orlando, FL, **1992**. p. 387–97.

Thangiah SR, Vinayagamoorthy R, Gubbi AV. Vehicle routing with time deadlines using genetic and local algorithms. In: Proceedings of the Fifth International Conference on Genetic Algorithms. Los Altas, CA: Morgan Kaufmann, **1993**. p. 506–15.

Thangiah SR. An adaptive clustering method using a geometric shape for vehicle routing problems with time windows. In: Proceedings of the Sixth International Conference on Genetic Algorithms. Los Altas, CA: Morgan Kaufmann, **1995**. p. 536–43.

Thangiah SR. Vehicle routing with time windows using genetic algorithms. In: Chambers L, editor. Application handbook of genetic algorithms: new frontiers, vol. II. Boca Raton, FL: CRC Press, **1995**. p. 253–77.

T. Ibaraki, S. Imahori, M. Kudo, T. Masuda, T. Uno and M. Yagiura, "Effective Local Search Algorithms for Routing and Scheduling Problems with General Time Window Constraints", Transportation Science 39, 206-232 (**2005**).

Yeniay Ö. (**2001**), "An Overview of Genetic Algorithms", Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, Cilt: 2, Sayı: 1, s. 37-49.

Yeo M. F. and Agyel E. O. (**1996**), "Optimising Engineering Problems Using Genetic Algorithms", Engineering Computations, Volume: 15, Number: 2, p. 268-280.