MELTEM YAKTUBAY

MSc THESIS

DEPARTMENT OF INDUSTRIAL ENGINEERING

REPUBLIC OF TURKEY
ADANA SCIENCE AND TECHNOLOGY UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF INDUSTRIAL ENGINEERING

A GENETIC ALGORITHM BASED SOLUTION APPROACH FOR VEHICLE ROUTING PROBLEM

MELTEM YAKTUBAY

MSc THESIS

Adana 2018

REPUBLIC OF TURKEY
ADANA SCIENCE AND TECHNOLOGY UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF INDUSTRIAL ENGINEERING

A GENETIC ALGORITHM BASED SOLUTION APPROACH FOR VEHICLE ROUTING PROBLEM

MELTEM YAKTUBAY

MSc THESIS

SUPERVISOR
ASSOC. PROF. DR. TOLUNAY GÖÇKEN

Adana 2018

Approval of the Graduate School of Natural and Applied Sciences

Director

Assoc. Prof. Dr. Osman SİVRİKAYA

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

Chairman of the Department

Assoc. Prof. Dr. Tolunay GÖÇKEN

tolunay

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

Supervisor

Assoc. Prof. Dr. Tolunay GÖÇKEN

Adana Science and Technology University

tolunay

Examining Committee Members

Prof. Dr. Rızvan EROL

Çukurova University

Assoc. Prof. Dr. Mustafa GÖÇKEN

Adana Science and Technology University

ETHICAL DECLARATION

I hereby declare that presented materials and results in this document are original and I have strictly abided by the academic and ethical rules while preparing this thesis. I affirm that I have prepared this work by rules of the Thesis Writing Guideline of Graduate School of Natural and Applied Sciences. I also declare that except for the information known in general, I have properly submitted knowledge in this thesis by necessary citations.

Name and Surname: Meltem YAKTUBAY

Signature:

A GENETIC ALGORITHM BASED SOLUTION APPROACH FOR VEHICLE ROUTING PROBLEM

YAKTUBAY, Meltem

Master of Science, Department of Industrial Engineering

2018, 94 pages

ABSTRACT

Vehicle Routing Problem with Time Windows (VRPTW) which is a type of classical Vehicle Routing Problem (VRP) handles a transportation issue that is comprised in the logistics management which is a substantial component of the supply chain management. VRPTW searches optimum routes for a fleet of vehicles making delivery from a depot to the customers in a specified time interval. Route optimization has a significant importance in logistics management owing to the effect on the customer satisfaction by fast delivery and lower cost. According to the literature, heuristic or metaheuristic methods are generally preferred for the solution since VRPTW is a combinatorial optimization problem. In this thesis, a multi objective genetic algorithm (GA) approach is offered to solve VRPTW. The objectives are determined as the minimization of the total distance and waiting time of the vehicles. NSGA-II, which is one of the multi objective optimization techniques is used in the evaluation, ranking, and selection of the individuals at GA steps. The influence of the quality of the initial population for an algorithm has been mentioned in different studies. In this study, three different methods are used to analyze this influence in the generation of the initial population step in multi objective GA. The initial populations are generated first randomly, second by a nearest neighbor based algorithm, and third by a sweep based algorithm. The formed three algorithms are tested on Solomon's benchmark problems. The GA with the initial population generated by sweep based algorithm has provided more effective results. The purpose of the study is to reveal the effect of initial population on the solutions obtained from GA and present a comparative approach for VRPTW solution.

Keywords: *Genetic algorithm, initial population generation, multi-objective optimization, NSGA-II, sweep algorithm, vehicle routing*

ARAÇ ROTALAMA PROBLEMINDE GENETIK ALGORITMA TABANLI ÇÖZÜM YAKLAŞIMI

YAKTUBAY, Meltem

Yüksek Lisans, Endüstri Mühendisliği Bölümü

2018, 94 sayfa

ÖZET

Klasik Araç Rotalama Probleminin (ARP) bir türü olan Zaman Pencereli Araç Rotalama Problemi (ZPARP), tedarik zinciri yönetiminin önemli bir parçası olan lojistik yönetiminin içerdiği bir taşımacılık sorununu ele alır. ZPARP, bir depodan müşterilere belirli bir zaman aralığında teslimat yapan araç filosu için optimum rotaları araştırır. Rota optimizasyonu, hızlı teslimat ve daha düşük maliyetle müşteri memnuniyetine olan etkisinden dolayı lojistik yönetiminde önemli bir yere sahiptir. Literatüre göre, ZPARP bir kombinatoryal optimizasyon problemi olduğundan çözüm için genellikle sezgisel veya metasezgisel yöntemler tercih edilir. Bu tezde ZPARP'yi çözmek için çok amaçlı bir genetik algoritma (GA) yaklaşımı önerilmiştir. Amaçlar, araçların toplam mesafesinin ve bekleme süresinin minimizasyonu olarak belirlenmiştir. GA adımlarında bireylerin değerlendirilmesi, sıralanması ve seçilmesinde çok amaçlı optimizasyon tekniklerinden biri olan NSGA-II kullanılmıştır. Literatürde, başlangıç popülasyonunun kalitesinin algoritmalar üzerindeki etkisinden bahsedilmiştir. Bu çalışmada, başlangıç popülasyonunun etkisini analiz etmek için çok amaçlı GA'da başlangıç popülasyonu üretimi aşamasında üç farklı yöntem kullanılmıştır. Başlangıç popülasyonları ilk olarak rasgele, ikinci olarak en yakın komşu tabanlı bir algoritma ile ve üçüncü olarak da süpürme tabanlı bir algoritma ile oluşturulmuştur. Oluşturulan üç algoritma, Solomon'un karşılaştırma problemleri üzerinde test edilmiştir. Başlangıç popülasyonu süpürme tabanlı algoritma ile oluşturulan GA ile daha etkili sonuçlara ulaşıldığı görülmüştür. Bu çalışmanın amacı, GA ile elde edilen sonuçlarda başlangıç popülasyonunun etkisini ortaya koymak ve ZPARP çözümü için karşılaştırmalı bir yaklaşım sunmaktır.

Anahtar Kelimeler: *Araç rotalama, başlangıç popülasyonu oluşturma, çok-amaçlı optimizasyon, genetik algoritma, NSGA-II, süpürme algoritması*

ACKNOWLEDGEMENTS

I would like to express my sincere gratefulness to my supervisor, Assoc. Prof. Dr. Tolunay GÖÇKEN for all her understanding, guidance, support and suggestions during the study.

I am very grateful to Prof. Dr. Rızvan EROL and Assoc. Prof. Dr. Mustafa GÖÇKEN for accepting to read and review this thesis and for their valuable suggestions and remarks.

I would like to thank Assist. Prof. Dr. Fatih KILIÇ and Assist. Prof. Dr. Ayşe Tuğba DOSDOĞRU for their worthwhile assistance and supports.

Lastly, I am very grateful to my family and dearest friends for their enormous support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

CLM	Council of Logistics Management
CSCMP	Council of Supply Chain Management Professionals
CVRP	Capacitated Vehicle Routing Problem
DCVRP	Distance-Limited CVRP
DVRP	Distance-Constrained VRP
GA	Genetic Algorithm
HFVRP	Heterogeneous Fleet VRP
MANOVA	Multivariate Analysis of Variance
MDVRP	Multi-Depot VRP
MOO	Multi Objective Optimization
mTSP	Multi Traveling Salesman Problem
NP-Hard	Nondeterministic Polynomial-Time Hard
NSGA-II	Non-dominated Sorting Genetic Algorithm II
OVRP	Open VRP
PVRP	Periodic VRP
SDVRP	Split Delivery VRP
SVRP	Stochastic VRP
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem
VRPB	VRP with Backhauls
VRPBTW	VRP with Backhauls and Time Windows
VRPPD	VRP with Pickup and Delivery
VRPPDTW	VRP with Pickup and Deliveries and Time Windows
VRPSPD	VRP with Simultaneous Pickup and Delivery
VRPTW	VRP with Time Window

CHAPTER 1. INTRODUCTION

1.1 Supply Chain Management and Logistics Management

For every business transaction, there are a supplier, a customer and some resources and activities connect them. Supply chain management conducts the balance of this connection in order to provide service to the customer at minimum cost and effort. Purchasing, demand forecasting, inventory management, capacity management, scheduling and quality management are the basic functions of a business and of any supply chain. The main goal of supply chain management is to deliver best value to the customer by measuring, planning and managing all the connections in the chain. Supply Chain Management focuses on two main subjects; to meet customer requirements and to keep costs at minimum. Supply chains can be encountered in schools, banks, hospitals, entertainment centers, factories and even homes, everywhere.

In a typical supply chain, raw materials are procured, and items are produced at one or more factories, transported to depots for intermediate storage and then transported to retailers or customers. People at different stages of supply chain can make different definitions for the term supply chain. Each definition is related the processes they do. For some, supply chain is relevant to purchasing and procurement, to others it is warehousing, distribution and transportation. Yet for others it would be sources of capital and labor (Basu and Wright, 2010).

A useful definition of supply chain management is provided by Simchi-Levi et al. (2003):

> *"Supply chain management is a set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses and stores, so that products are produced and distributed at the right quantities, to the right places, and at the right time, in order to minimize system-wide costs while satisfying service level requirements."*

The Council of Logistics Management (CLM) has changed its own name to Council of Supply Chain Management Professionals (CSCMP) in 2004. It shows the approval of that the supply chain management has a wider meaning than the logistics management. The council made the definition of the logistics management as:

> *"Logistics management is that part of supply chain management that plans, implements, and controls the efficient, effective forward and reverse flow and storage of goods, services, and related information between the point of origin and*

the point of consumption in order to meet customers' requirements. Logistics management activities typically include inbound and outbound transportation management, fleet management, warehousing, materials handling, order fulfillment, logistics network design, inventory management, supply/demand planning, and management of third party logistics services providers. To varying degrees, the logistics function also includes sourcing and procurement, production planning and scheduling, packaging and assembly, and customer service. It is involved in all levels of planning and execution—strategic, operational, and tactical. Logistics management is an integrating function which coordinates and optimizes all logistics activities, as well as integrates logistics activities with other functions, including marketing, sales, manufacturing, finance, and information technology." (CSCMP Glossary, 2013)

Supply Chain Management is defined as:

"Supply Chain Management encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third-party service providers, and customers. In essence, supply chain management integrates supply and demand management within and across companies. Supply Chain Management is an integrating function with primary responsibility for linking major business functions and business processes within and across companies into a cohesive and high-performing business model. It includes all of the logistics management activities noted above, as well as manufacturing operations, and it drives coordination of processes and activities with and across marketing, sales, product design, finance and information technology." (CSCMP Glossary, 2013)

The components of supply chain management are not recently developed. The fact is that, the supply chain parts (e.g. buying, planning, scheduling, stock control, warehousing, logistics, distribution, etc.) have been managed for years without perceiving the importance of the whole chain concept. In the meanwhile, various procurement and distribution elements cost has long been known. In 1927, Ralph Borsodi emphasized that: In 50 years between 1870 and 1920 the cost of distribution of consumed needs and luxuries has increased almost threefold, while the cost of production has decreased by one-fifth. It means that the savings in production are lost at the cost of distribution.

Being conscious about transportation economics and pricing is important for successful logistics management. The main factors of transportation costs are distance, volume, density, stacking, transportation, responsibility and market factors. These factors identify the transportation prices offered to customers as the ratios for specific services. (Bowersox et al., 2002)

Organizations extend logistics coverage from the management of raw materials to the delivery of final products (Christopher, 2011). Logistics network consists of resource centers, manufacturing centers, factories, depots, distribution centers and retail outlets. The duration from obtaining raw materials through selling finished products requires transportation, packaging, storage and handling processes. These are controlled by logistics management. Figure 1.1 shows the logistics management process.

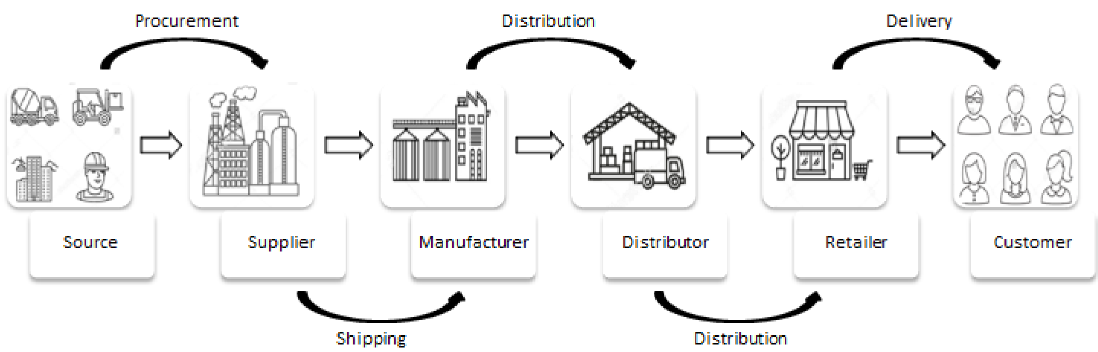

Figure 1.1. Logistics management process.

Increased competition in business reveals new difficulties that have considerable effect on supply chain management and logistics systems. The companies must adapt to that competitive environment and focus on customer satisfaction. In the aim of increasing the customer satisfaction, the company should adopt the minimum selling price. Besides the minimum selling price, the companies' strategy must comprise making faster delivery and having more customer reachability. These goals might be achieved with the improvement in the supply chain and logistics system. Total traveling distance of the vehicles make delivery and the transportation cost directly related to each other. And, the transportation cost is a significant part of the selling price of the product or service. In that case, decreasing the transportation cost as much as possible is critical for the companies.

Customer satisfaction is critical for logistics companies whose intention is having competitive advantage. Because they are aware of that if they do not meet the requirements of the customers, customer preferences will change towards other companies whose activities are more focused on customer expectations. As competition in the sectors is continually increasing, the ability of companies to understand the customers and maintain their satisfaction with the services received is becoming more important. High service quality enhances the company's competitive advantage, consumer loyalty, and reduces the number of competitors (Meidutė-Kavaliauskienė et al., 2014). The companies should satisfy the customers' expectations and requirements. For these reasons, logistics management has big importance for companies.

An essential part of achieving transportation efficiency is the determination of the routes. Routes, the geographical path of the vehicles, show travel way to complete transportation requirements. That is the subject of Vehicle Routing Problem. Vehicle Routing Problem (VRP) which builds the paths of the vehicles while controlling some constraints and minimizing the global distances is a significant management problem in the logistics field.

1.2. Application Areas of VRP

The VRPs are generally related to the distribution of goods and services between specific points in a network. There are many practical variations of the problem in the real life. Some of them are follows:

- Motion of industrial goods along the supply chain
- Public transportation
- School bus routing
- Passenger and cargo transport with airline companies
- Courier services
- Delivery of online purchases
- Mail delivery
- Emergency services (including firefighting and ambulance services)
- Preventive maintenance inspection tours
- Appliance repair services
- Gasoline delivery trucks
- Urban solid waste collection
- Street cleaning
- House call tours by a doctor

1.3. Motivation of The Study

In recent years, globalization increases competition between companies rapidly. Therewithal, customer satisfaction has become one of the most significant factors of the competition. Faster delivery and more customer reachability increase customer satisfaction. Because of this reason, in supply chain, logistics management has become an area that companies pay more attention. Beside this, the companies try to decrease their logistic costs by building better routes to survive in today's competitive world. The purpose of these companies is to maintain glorified and rapid service and minimize the costs as well.

The VRP is one of the classical research areas in operations research with quite economic importance. There are many real-life applications in the transportation sector in particular. For industrial problems, the methods that can produce high-quality solutions in limited time, even for several hundreds of customers, are particularly important. It is expected that this research will make a significant contribution to the VRP area in terms of cost and time saving by determining more appropriate distribution routes. The purpose of the study is to present a multi objective and hybrid solution approach for a VRP variant.

CHAPTER 2. VEHICLE ROUTING PROBLEM

VRPs constitute a significant problems family encountered in the logistics field. In general, VRP is the problem of forming the suitable paths under a set of constraints, to a fleet of vehicles which will serve to a group of customers starting and terminating at a central depot.

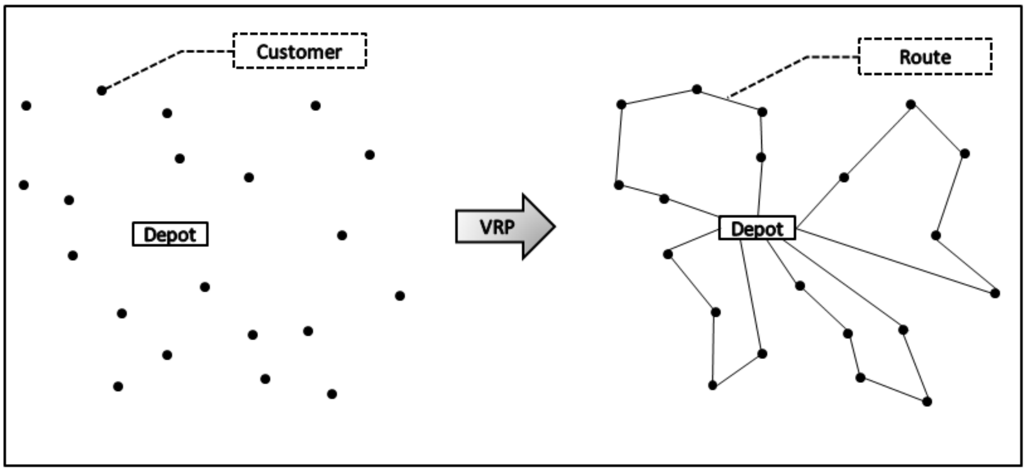

Figure 2.1. The vehicle routing problem example.

VRP is the establishing of the optimal set of routes for a fleet of vehicles to serve a given set of customers. It is one of the most important and studied combinatorial optimization problems. Optimization is composed of finding one or more best (optimal) solutions from all feasible solutions. Optimization problems can be separated into two classes according to whether the variables are continuous or discrete. The second case is recognized as a combinatorial optimization problem. It can be tough to figure out these problems. The difficulty stems from the fact that the feasible solutions are of limited but high degree of stability. In order to find a global optimum, it is necessary to prove that a specific solution is better than all feasible points. A class of optimization problems called easy if a solution algorithm to figure out each instance of the problem class can be developed in polynomial time. That so, a polynomial-time algorithm is considered as an efficient algorithm. Notwithstanding the best attempts of thousands of researchers from all over the world, no effective algorithms were found for combinatorial optimization problems. The NP-completeness theory was formed as a result of these unsuccessful efforts. It is believed that these problems cannot be solved efficiently. If an algorithm is usually required to search the solution space and most often, if it cannot find and prove the optimality in polynomial time; the problems are said to be “NP-hard” (nondeterministic polynomial-time hard) problems. In many cases, combinatorial optimization problems are

NP-hard. Traveling Salesman Problem (TSP), which is a special case of VRP when there is only one vehicle with an unlimited capacity, is also a well-known combinatorial optimization problem that is hard to solve. It is in NP-hard problems category. In practice, solving VRP is harder than solving a TSP of the same size. Different techniques proposed to find optimum solutions. However, a detailed search is usually not possible in enumerative techniques (exact algorithms) due to the time required. In case of having less data, the exact algorithms may be reasonable, but as the data grows using these algorithms is useless because of increase in the required computational time of the computer exponentially. Therefore, the researchers have been trying to develop heuristic and metaheuristic methods for real-world problems. Although metaheuristics cannot prove the optimality of the solutions they find, they usually reach high complexity in the acceptable time. (Ahuja et al., 1993; Laporte, 2007)

2.1. Characteristics of VRPs

VRP is a type of problem that has a broad variety. By regarding the components, constraints and probable objectives, the characteristics of these problems are going to be described.

2.1.1. Components

A VRP constituted of four components:

- the road network;
- the customers;
- the fleet of vehicles;
- the depot(s).

The road network, used for the transportation of the merchandise, is usually designate with a direct or undirect graph and is made up of nodes and links. The nodes represent the road junctions, the depot and customers. The links indicate the pathways between the nodes. The pathways are associated with a cost, which generally represents its length and/or travel time, which is probably dependent on the vehicle type or on the time interval during which the pathway is passed.

Typical characteristics that customers may have can be as follows:

- nodes of the road network in which the customers are placed;

- demand that quantity of the merchandise which must be delivered or picked up at the customer;
- service times that times required to deliver or pick up the commodities at the customer location;
- time intervals of the day (time windows) when the customer can be served (e.g. due to the specific periods during which the customer is available or the location can be reached, due to traffic limitations); and
- required vehicle type that can be utilized to serve the customer (e.g. existing possible access limitations or loading and unloading requirements).

A depot is a node on the road network where the vehicles are loaded, unloaded or parked. The vehicles start their route from the depot and generally return there at the end of the route. Each depot is characterized by the number and types of the vehicles associated with it and by the total supplied or stored quantity of merchandise.

Transportation of the merchandise is carried out using a fleet of vehicles. Typical characteristics of the vehicles may have:

- capacity limit that the maximum weight, or volume, or the number of product can be load to the vehicle;
- probable different compartments of the vehicle that differ according to the capacity limit and/or types of the carried products;
- available devices for the loading and unloading processes;
- subset of pathways which can be travelled by the vehicle and
- costs associated with usage of the vehicle (per distance unit, per time unit, per route, etc.).

Composition and size of the vehicles can be fixed or can be decided according to the needs of the customers. They can be in different types, for example trucks, trains, aircraft, and boats and even pedestrian, laser beams or robot arms. Generally, a driver/operator is performed the task in the vehicle. The vehicle and, the driver/operator if there is, are considered as a whole. (Toth and Vigo, 2002; Labadie et al., 2016)

2.1.2. Objectives

The standard objective in the VRP is the minimization of the total cost, which is dependent on the global travelled distance (or on the global travel time). The costs may emerge owing to the road network characteristics, customer requirements, facility resources, delivery conditions and usage of vehicles. The generalized cost refers to the penalty of various negative effects during the vehicle transportation and delivery tasks Transportation cost can be separated into fixed cost and variable cost. Fixed costs include vehicle-use costs (e.g., purchase cost and depreciation) and the driver salary, while variable costs are relevant to the scheduling of the routing (e.g., routing distance, time, fuel consumption, and loading/unloading time). Additional variants also include the fixed costs of using depots or inventory costs. Furthermore, penalty costs could be generated if delays occur or customer requirements cannot be satisfied. These costs must be minimized.

Sometimes, the minimization of the number of vehicles that need for serving all the customers is selected as one of the objectives of the problem.

To obtain a fair timeline between the vehicles, it may be essential to balance the routes in terms of travel time and / or vehicle load.

Sometimes unusual objectives may arise. For example, in humanitarian logistics, it is required to bring help to victims as soon as possible. Instead of minimizing the total time, it is aimed to minimize the sum of the arrival time that is equivalent to the mean arrival time at each customer.

The potential objectives for VRP might be as below:

- Minimization of the total cost,
- Minimization of the total traveled distance,
- Minimization of the total traveling time,
- Minimization of the number of the vehicles,
- Minimization of the space utility of the vehicles,
- Minimization of the penalties,
- Minimization of the variability in the travel times of the vehicles,
- Minimization of the variability in the traveled distance by the vehicles,
- Minimization of the total waiting time of the vehicles,
- Minimization of the total arrival time of customers.

The one or several of these objectives may be selected. Any weighted combination of these objectives can be determined, or the problem can be turned into the multi objective optimization (MOO) problem. Particularly, when the objectives are conflicting, MOO may be more suitable.

2.1.3. Constraints

The characteristics of the VRPs components (i.e. customers, vehicles, depots and road network), and additional regulations (such as working periods of the operators during the day, number and duration of breaks, maximum time interval of driving periods, etc.) force the solution model to comply with some operational and regulatory constraints. The routes must satisfy these constraints. Some of them may be as follows:

- Capacity limit may exist in the depots and on the vehicles; the capacity limit of the vehicles should not be exceeded by the customers placed on the same route during the whole travel and capacity limit of the depot should not be exceeded by all customers;
- the requirement of a customer might be a delivery, a picking up or both;
- vehicles may have travelled distance limit;
- a route may finish at a depot where is the beginning node of the same route;
- there may be more than one depot;
- customers may be serviced only within their specified time intervals and the working periods of the operators who use the vehicles serving them;
- vehicles may be similar or different;
- the order in which the customers are routed may change according to the priority constraints;
- synchronization may be required when a customer needs to be visited at least twice at the same time;
- the consideration of the stochastic or time-dependent dynamic versions of the problem is required when the data may not be perfectly known in advance.

Eksioglu et al. (2009) have done a taxonomic study of VRP. They have classified the problem according to the aspects of type of study, scenario characteristics, problem physical characteristics, information characteristics, and data characteristics. They have considered all potential situations and real-world constraints while investigating disparate VRP articles.

1. Type of Study
   - 1.1. Theory
   - 1.2. Applied methods
     - 1.2.1. Exact methods
     - 1.2.2. Heuristics
     - 1.2.3. Simulation
     - 1.2.4. Real time solution methods
   - 1.3. Implementation documented
   - 1.4. Survey, review or meta-research
2. Scenario Characteristics
   - 2.1. Number of stops on route
     - 2.1.1. Known (deterministic)
     - 2.1.2. Partially known, partially probabilistic
   - 2.2. Load splitting constraint
     - 2.2.1. Splitting allowed
     - 2.2.2. Splitting not allowed
   - 2.3. Customer service demand quantity
     - 2.3.1. Deterministic
     - 2.3.2. Stochastic
     - 2.3.3. Unknown[1]
   - 2.4. Request times of new customers
     - 2.4.1. Deterministic
     - 2.4.2. Stochastic
     - 2.4.3. Unknown
   - 2.5. On site service/waiting times
     - 2.5.1. Deterministic
     - 2.5.2. Time dependent
     - 2.5.3. Vehicle type dependent
     - 2.5.4. Stochastic
     - 2.5.5. Unknown
   - 2.6. Time window structure
     - 2.6.1. Soft time windows
     - 2.6.2. Strict time windows
     - 2.6.3. Mix of both
   - 2.7. Time horizon
     - 2.7.1. Single period
     - 2.7.2. Multi period
   - 2.8. Backhauls
     - 2.8.1. Nodes request simultaneous pick ups and deliveries
     - 2.8.2. Nodes request either linehaul or backhaul service, but not both
   - 2.9. Node/Arc covering constraints
     - 2.9.1. Precedence and coupling constraints
     - 2.9.2. Subset covering constraints
     - 2.9.3. Re course allowed
3. Problem Physical Characteristics
   - 3.1. Transportation network design
     - 3.1.1. Directed network
     - 3.1.2. Undirected network
   - 3.2. Location of addresses (customers)
     - 3.2.1. Customers on nodes
     - 3.2.2. Arc routing instances
   - 3.3. Geographical location of customers
     - 3.3.1. Urban (scattered with a pattern)
     - 3.3.2. Rural (randomly scattered)
     - 3.3.3. Mixed
   - 3.4. Number of points of origin
     - 3.4.1. Single origin
     - 3.4.2. Multiple origins
   - 3.5. Number of points of loading/unloading facilities (depot)
     - 3.5.1. Single depot
     - 3.5.2. Multiple depots
   - 3.6. Time window type
     - 3.6.1. Restriction on customers
     - 3.6.2. Restriction on roads
     - 3.6.3. Restriction on depot/hubs
     - 3.6.4. Restriction on drivers/vehicle
   - 3.7. Number of vehicles
     - 3.7.1. Exactly n vehicles (*TSP in this segment*)
     - 3.7.2. Up to n vehicles
     - 3.7.3. Unlimited number of vehicles
   - 3.8. Capacity consideration
     - 3.8.1. Capacitated vehicles
     - 3.8.2. Uncapacitated vehicles
   - 3.9. Vehicle homogeneity (Capacity)
     - 3.9.1. Similar vehicles
     - 3.9.2. Load-specific vehicles[2]
     - 3.9.3. Heterogeneous vehicles
     - 3.9.4. Customer-specific vehicles[3]
   - 3.10. Travel time
     - 3.10.1. Deterministic
     - 3.10.2. Function dependent (a function of current time)
     - 3.10.3. Stochastic
     - 3.10.4. Unknown
   - 3.11. Transportation cost
     - 3.11.1. Travel time dependent
     - 3.11.2. Distance dependent
     - 3.11.3. Vehicle dependent[4]
     - 3.11.4. Operation dependent
     - 3.11.5. Function of lateness
     - 3.11.6. Implied hazard/risk related
4. Information Characteristics
   - 4.1. Evolution of information
     - 4.1.1. Static
     - 4.1.2. Partially dynamic
   - 4.2. Quality of information
     - 4.2.1. Known (Deterministic)
     - 4.2.2. Stochastic
     - 4.2.3. Forecast
     - 4.2.4. Unknown (Real-time)
   - 4.3. Availability of information
     - 4.3.1. Local
     - 4.3.2. Global
   - 4.4. Processing of information
     - 4.4.1. Centralized
     - 4.4.2. Decentralized
5. Data Characteristics
   - 5.1. Data Used
     - 5.1.1. Real world data
     - 5.1.2. Synthetic data
     - 5.1.3. Both real and synthetic data
   - 5.2. No data used

[1] Unknown refers to the case in which information is revealed in real-time (i.e., dynamic and fuzzy studies fall under this category)
[2] Each vehicle can be used to handle specific types of loads
[3] A customer must be visited by a specific type of vehicle
[4] Cost of operating a vehicle is not negligible

Figure 2.2. Taxonomy of the VRP literature (Eksioglu et al., 2009).

2.2. Variants of VRP

The VRP is a generalized version of the well-known TSP, but its solution is much more difficult in practice. In TSP, which is the simplest form of the routing problems, there is only one traveling salesman with unlimited carrying capacity which must visit all the customers, meet their demands and then go back to the depot in a single route. In case of more than one salesman is allowed, the problem turns into the multi traveling salesman problem (mTSP) which is a generalization of the TSP. Additionally, the mTSP is a relaxation of the VRP. If vehicle capacity in the VRP is large enough, the problem is the same as the mTSP. If capacity constraint for the vehicles is considered, then this problem turns into the Capacitated Vehicle Routing Problem (CVRP).

In real life, there are many other restrictions for this problem caused by a lot of variations. The basic variants of the VRP are summarized in the Figure 2.3.

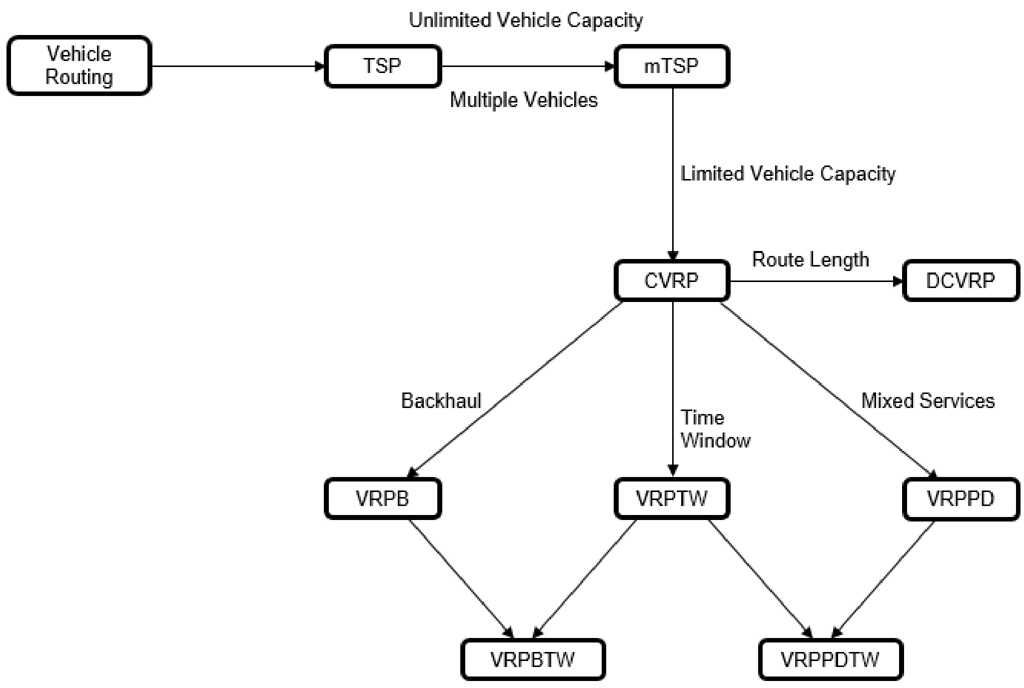

Figure 2.3. The basic variants of the VRP (Adapted from Sandhya, 2013; Toth and Vigo, 2002).

2.2.1. Capacitated and Distance Constrained VRP

The basic form of the VRP is the Capacitated VRP (CVRP). In CVRP, all the customers are known, the demands and locations of the customers are deterministic, and the vehicles are exactly alike with the same capacity constraints. Deliveries cannot be split on different vehicles and the total demand of the assigned customers to each route cannot be in excess of the vehicle capacity. All vehicles depart from the depot and return to the depot again at the end of the route. The objective is to minimize the total cost of the vehicles on the routes that serve all the customers.

In CVRP, there are more than one vehicle and multiple routes at the same time. Excess of the capacity limit is not allowed. In CVRP, total traveling distance of the vehicles is not restricted. If the capacity constraint for each route is replaced by a maximum length (or time) constraint, Distance-Constrained VRP (DVRP) is considered. Distance-Limited CVRP (DCVRP) occurs if both the vehicle's capacity constraints and the maximum distance constraints exist in the problem.

2.2.2. VRP with Time Windows

The VRP with Time Windows (VRPTW) is an extension of the CVRP with an additional restriction of a term time window which is a time interval (e_i, l_i) between the earliest arrival time and latest arrival time for each customer i. This is the consideration of the time limitations on the demand delivery for customers. The customer should be supplied in between this time interval. That means the vehicle must start the service between time interval of e_i and l_i.

In this problem, the service is made by a homogeneous fleet of vehicles with same features and capacity constraints. Each vehicle stays at the customer during loading/unloading task throughout service time. The objectives may be the minimization of the number of tours or routes, and then for the same number of tours, the minimization of the total traveled distance.

2.2.3. VRP with Backhauls

The VRP with Backhauls (VRPB) is another extension of the CVRP in which the customers are divided into two subgroups, namely linehaul and backhaul. Linehaul customers require a certain amount of goods to be delivered. At the backhaul customers, a certain amount of inbound goods must be picked up. A route can contain two subgroups members. In the routes, linehaul customers have precedence order than the backhaul ones. That means all the linehaul customers must be served before any backhaul customer may be served. As a result, the total demand of the subgroups in a route do not exceed the capacity constraint separately. (Toth and Vigo, 2002)

The VRPB state in which the time windows exist is called the VRP with Backhauls and Time Windows (VRPBTW).

2.2.4. VRP with Pickup and Delivery

In the basic version of the VRP with Pickup and Delivery (VRPPD), each customer is associated with two quantities d and p, indicating the demand of products to be delivered and picked up at the customer, respectively. Delivery is done before the picking up at each customer. Hence, the vehicle load is calculated by the initial load minus delivered quantity d plus picked up p quantity. It must be always positive or zero and smaller than or equal to the vehicle capacity.

In VRPPD, the beginning points, where the delivery demand is supplied, and ending points, where the pick up demand is left, of the routes may be different or same. If they are the same, the problem is named as the VRP with Simultaneous Pickup and Delivery (VRPSPD). The case of VRPPD in which time windows exist has been studied in the literature and is named as the VRP with Pickup and Deliveries and Time Windows (VRPPDTW). (Toth and Vigo, 2002)

2.2.5. Other Additional Variants

VRP variations can be extended according to the real-world constraints due to the changing conditions of the problem. Some of these variants are classified in the literature as follows:

- **Multi-Depot VRP (MDVRP):** There might be more than one depot in the VRPs. The vehicles can be originated on any of the depots, but all the vehicles must return to the depot where they are originated.
- **Split Delivery VRP (SDVRP):** If the size of the customer demand is greater than the capacity of the vehicle, it is allowed to split the deliveries and the customer is served by more than one vehicle. A solution is feasible if a customer may be served by two or more vehicles besides conforming all constraints of the VRP.
- **Heterogeneous Fleet VRP (HFVRP):** In case of the vehicles are not identical and have different capacity limits and properties, customers may be served by a heterogeneous fleet.
- **Periodic VRP (PVRP):** The planning of delivery period might be a specified number of days. In that case, it is not compulsory to be serviced all the customers on every day in this period and the vehicles may not return to the depot in the same day they leave. Delivery days must be allocated to each customer and vehicle routes must be determined for each day of the period, so the total cost is minimized.
- **Stochastic VRP (SVRP):** One or more elements of the problem, like customer number, demand quantity, travel time, service time etc., might have random behavior. In that situation, it is assumed, generally, these data follow a probability distribution and the missing ones are estimated for satisfy some constraints.
- **Open VRP (OVRP):** The vehicle may or may not return to the depot after finishing the services of the customers in the route they are assigned. The

vehicles either are not needed to go back to the beginning point, or they have to return by revisiting the customers assigned to them in the reverse order. For this reason, the routes are not closed paths but open ones (Sariklis and Powell, 2000).

CHAPTER 3. SOLUTION METHODS

Many approaches that have utilized exact, heuristic and metaheuristic algorithms have been developed heretofore to solve VRPTW. If the size of the customer set is small, the exact algorithms can be utilizable; else if the set is getting larger it is not viable to use these algorithms due to the high solution duration. For this reason, the solution approaches of the problem via heuristic and metaheuristic strategies which are proposed optimal or approximate solutions are growing in the literature recently (Çolak and Güler, 2009). In this study heuristic algorithms are considered.

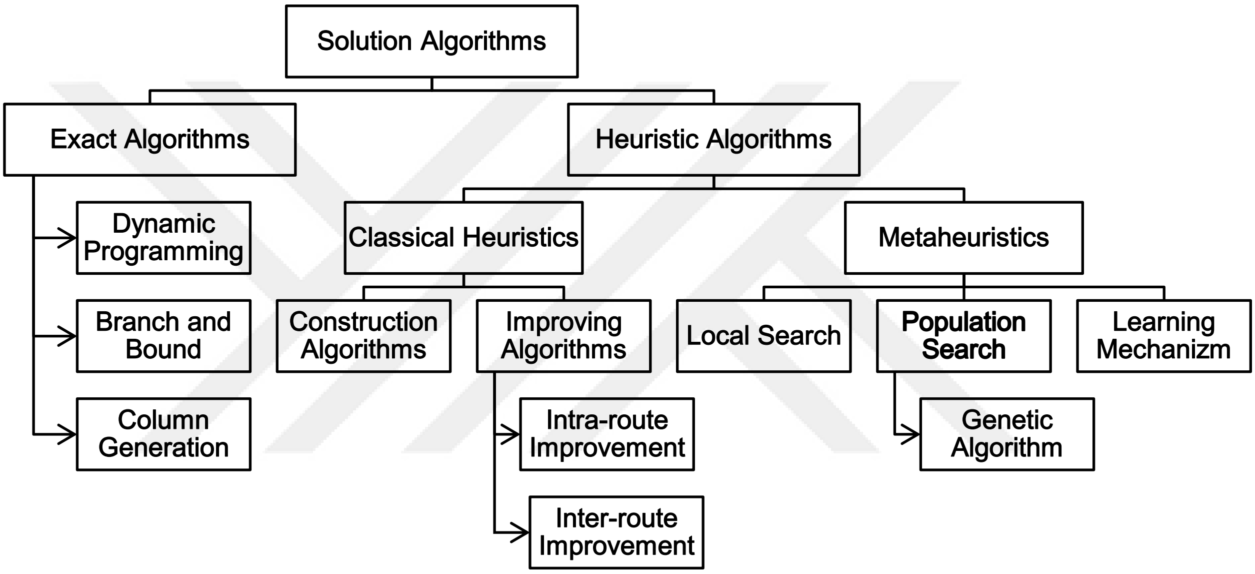

Figure 3.1. Solution algorithms for VRP and its variant.

3.1. Heuristic Solution Methods

Heuristic algorithms that are typically called route construction heuristics construct a set of routes from scratch. On the other hand, route improving heuristics tries to generate an improved solution based on an already feasible solution (Laporte, 2009).

3.1.1. Construction Heuristics

3.1.1.1. Sweep Algorithm

The planar instances of the VRP are applied in the sweep algorithm (Gillett and Miller, 1974). Firstly, feasible clusters are created by rotating a ray centered at the depot.

Then, a TSP is solved to obtain a vehicle route for each cluster. A brief explanation of this method is given as follows.

- **Step 1** (polar coordinate computation). The polar coordinates of each customer are computed with respect to the depot. Customers are sorted in increasing polar angle.
- **Step 2** (customer clustering). Algorithm is started from the non-routed vertex that has the smallest angle. The vertices are assigned to a vehicle as long as the maximal route length or the capacity constraint is not violated. If non-routed vertices remain, continue with the next vehicle.
- **Step 3** (route construction). Each vehicle route is optimized separately by solving the corresponding TSP (exactly or approximately).

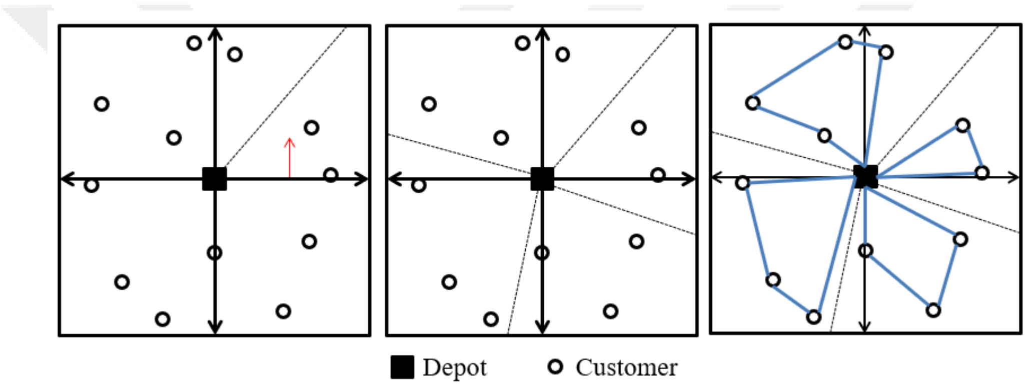

Figure 3.2. Sweep algorithm.

3.1.1.2. Saving Algorithm

The Clarke and Wright algorithm is widely known heuristic for the VRP. It is based on the notion of savings (Clarke and Wright, 1964). When two routes (0, ..., i, 0) and (0, j, ..., 0) can feasibly be merged into a single route (0, ..., i, j, ..., 0), a distance saving $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ is generated. The algorithm has a parallel and a sequential version. The working principle of the algorithm can be summarized as follows.

- **Step 1** (savings computation). The savings $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ are computed for i, j = 1, ..., n and i ≠ j. n vehicle routes (0, i, 0) is created for i = 1, ..., n. The savings are ordered in a decreasing scheme.

Parallel version

- **Step 2** (best feasible merge). Algorithm is started from the top of the savings list. The saving s_{ij} is given. It is determined whether there exist two routes. One route contains arc or edge (0, j). The other route contains arc or edge (i, 0). If so, combine these two routes by deleting (0, j) and (i, 0) and introducing (i, j).

Sequential version

- **Step 2** (route extension). Each route (0, i, …, j, 0) is considered. The first saving s_{ki} or s_{jl} is determined. It can feasibly be utilized to combine the current route with another route including arc or edge (k, 0) or including arc or edge (0, l). The merge is implemented, and this operation is repeated in the current route. If no feasible merge exists, the next route is considered, and the same operations are reapplied. The algorithm stops when no route merge is feasible.

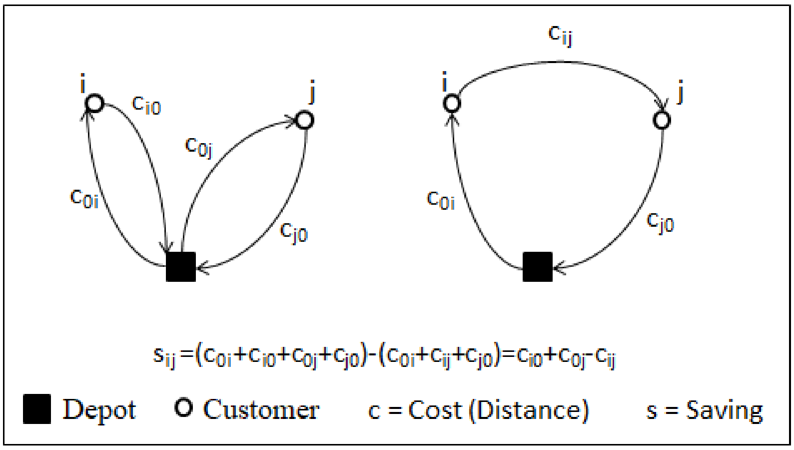

Figure 3.3. Saving algorithm.

3.1.1.3. Nearest Neighbor Algorithm

The nearest neighbor algorithm starts every route by finding the non-routed customer closest in terms of distance to the depot. It adds another non-routed customer that is closest to the last customer who is added to a route. At every subsequent iteration, the algorithm looks for the customer nearest to the last customer added to the route. This search is applied among all the customers who can feasibly be added to the end of the emerging route. A new route is started any time the search fails, unless there are no more customers to schedule. The algorithm has a parallel and a sequential version, like saving algorithm. Sequential heuristic builds one route at a time. On the other hand, parallel one builds the routes simultaneously (Van Breedam, 2002).

- **Step 1** (route initialization). An unused vehicle is chosen.
- **Step 2** (route construction). Algorithm is started from the closest non-routed vertex to the depot. Vertices are assigned nearest the last vertices added to the route to vehicle k as long as the maximal route length or its capacity is not exceeded. Algorithm is returned to Step 1 when non-routed vertices remain.
- **Step 3** (route optimization). Each vehicle route is optimized separately.

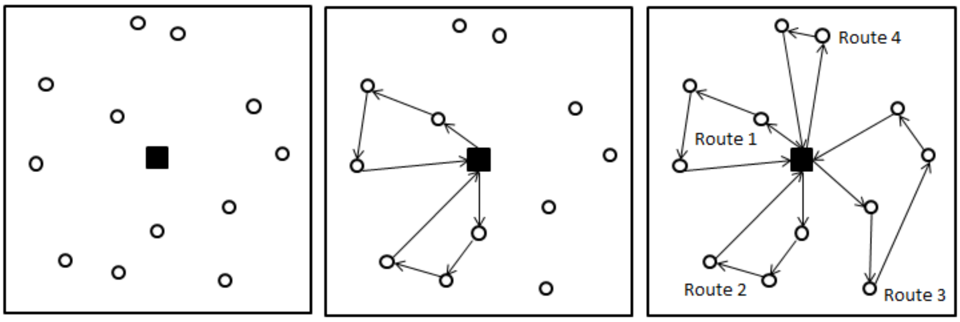

Figure 3.4. Nearest neighbor algorithm.

3.1.2. Improving Heuristics

Every route-improving heuristic is generally based on the notion of a neighborhood. Controlling some of or all the solutions in a neighborhood can improve the solutions with respect to the objective. This is called local search.

For the VRP, improving heuristics work on each vehicle route taken separately or on several routes at a time. The first situation is appropriate for the TSP. The second situation is valid for VRP owing to have a multi-route structure.

3.1.2.1. Intra-Route Improvement Methods

For the TSP, most improvement methods can be defined in terms of Lin's (1965) λ-optimality (simply λ-opt) concept. Here, λ edges are extracted from the route, and the λ remaining links are reconnected in all possible ways. If any gainful reconnection is detected, it is applied. The method stops at a local minimum when it is impossible to obtain a route with smaller cost by replacing any λ of its links by any other set of λ links. Then, the route is said to be λ-optimal.

3.1.2.2. Inter-Route Improvement Methods

Inter-route methods typically include removing one or several customers from a number of routes and relocating them. In relocate operator, one customer is moved from one route to another. In exchange operator, two customers are interchanged between two routes. In 2-Opt* operator, one link of a route is changed with another link from another route. Details about the review of neighborhood-operator for inter-route improvements can be found in El-Sherbeny (2010). In addition, Labadie et al. (2016) presented 2-Opt and λ-interchange moves as given in Figure 3.5.

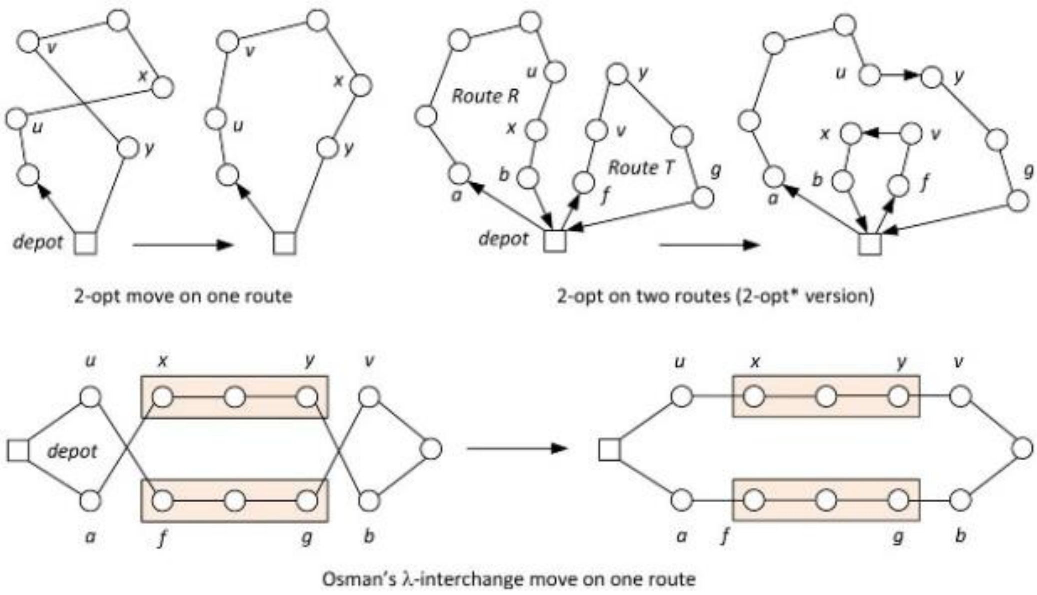

Figure 3.5. 2-Opt and λ-interchange moves (Labadie et al., 2016).

3.2. Metaheuristics

Metaheuristics are optimization techniques that organize an interaction between higher level strategies and basic local improvement heuristics (Griffis et al., 2012). These techniques are used to deal with complex optimization problems where other optimization techniques fail. They intend to avoid local optimums in the solution space. As metaheuristics conduct a more thorough search, they have rapidly become the preferred methods for generating solutions to many complex combinatorial real-world problems that cannot be solved by exact methods (Glover and Kochenberger 2003). A good metaheuristic implementation cannot guarantee the optimal solution as like exact methods, but it can provide at least the near-optimal solutions at reasonable computation times.

Metaheuristics are differentiated from classical heuristics by allowing inferior and even infeasible intermediate solutions in the searching process.

The fundamental properties of metaheuristics are summarized as follows (Blum and Roli, 2003):

- They guide the search process.
- The main purpose is effectively search the solution space to find (near-) optimal solutions.
- They include not only simple local search procedures but also complex learning processes.
- They are approximate and generally non-deterministic.
- They may consolidate the techniques to escape getting trapped in restricted areas of the search space.
- They allow an abstract level description.
- They are not specific to problems.
- They can make use of domain-specific knowledge in the form of heuristics that are audited by the upper level strategy.

Metaheuristic have proved to be especially successful to solve various kind of complex problems in any fields of science (Labadie et al., 2016). In metaheuristic, practical advantage is both their effectiveness and general applicability (Ólafsson, 2006). The search structure has many general elements across numerous metaheuristic techniques. In metaheuristic techniques, an initial solution or an initial set of solutions is firstly obtained. Then, an improving search is initiated using certain principles.

The general metaheuristic framework can be defined in Table 3.1.

Table 3.1. The general metaheuristic framework (Ólafsson, 2006).

Obtain **an initial solution (set) θ_0 and set $k = 0$.**
Repeat
Identify the neighborhood $N(\theta_k)$ of the current solution(s).
Select candidate solution(s) $\{\theta^c\} \subset N(\theta_k)$ from the neighborhood.
Accept the candidate(s) and set $\theta_{k+1} = \theta^c$ or reject it and set $\theta_{k+1} = \theta_k$.
Increment $k = k + 1$.
Until **stopping criterion is satisfied.**

Metaheuristic algorithms can be classified in several ways according to the characteristics that differentiate them. Blum and Roli (2003) have been summarized the classification criterions of the algorithms as: inspiration source, searching area, objective function type, neighborhood structure and memory usage. Details can be found in Blum and Roli (2003).

Laporte (2009) has classified the metaheuristics into local search, population search and learning mechanisms. Local search methods explore the solution space by moving at each iteration from the current solution to another solution in its neighborhood. The main components of a local search are the rules used to describe the neighborhood of a solution and the mechanism put forward to discover it. Simulated annealing, deterministic annealing, adaptive large neighborhood search, iterated local search, variable neighborhood search, and tabu search, are classical examples of local search heuristics. Population search deals with a population of solutions rather than a single solution. Population-based algorithms evolve a set of solutions and produce new solutions by either combining selected ones in the hope of generating better ones. They generally inspire from nature concepts like the evolution of species and the behavior of social insects foraging. The distinguished examples of population-based algorithms are GA, scatter search, ant colony optimization, path relinking, and particle swarm optimization. Learning mechanisms are able to learn from experience and have different memory structures. Neural networks and ant algorithms are derived from learning paradigm. In this paper, GA that is a population-based and bio-inspired metaheuristic is considered.

3.2.1. Genetic Algorithm

Darwin's principle of evolution based on "survival of the fittest" has motivated and has instigated the studies on evolutionary computation. Evolutionary computing and evolution strategies have been appeared firstly in 1960s. They have been started with the idea of adapting the theory of biological evolution to computer science. Evolutionary computation techniques integrate evolutionary principles into the algorithms which can be utilized to find optimal solutions to a problem. In the subject of evolutionary computation; GA, genetic programming, evolutionary strategies and evolutionary programming are the fundamental paradigms (Sivanandam and Deepa, 2008). They are all grouped under the name of evolutionary algorithms. The main differences between them lie in the nature of the representation schemes, the reproduction (crossover and mutation) operators and selection methods.

GA was developed by J. Holland in 1975 and was presented in the book named "Adaptation in natural and artificial systems". GA mimics the natural selection process based on "survival of the fittest" principle. It is a useful tool to produce good solutions to optimization and search problems.

GA iteratively improves a set of solutions by mimicking the biological evolution in natural selection. In each iteration, it chooses the individuals as parents from the current population and produces the children by using them for the next generation (Aggarwal et al., 2014). Due to the fact that GA concepts are derived directly from natural evolution, the terminologies of GA and natural evolution are related. Some terminologies to be used in GA are briefly explained below:

Population: It is a valid set of alternative solutions. It consists of multiple individuals, called chromosomes, which reside in search space.

Chromosome: A chromosome is an individual of solution which is formed of a set of genes. Each of it is an alternative candidate solution. All genetic information is stored on chromosomes. They permit to perform of genetic operators.

Gene: A gene is a sub-unit of a chromosome which is formed of a set of alleles. Genes are joined into a string. This string is analogous to the chromosome. Genes code the characteristics of an individual.

Allele: It is the smallest information unit of a chromosome. The possible states of the genes for one feature are named allele and a gene may receive different alleles.

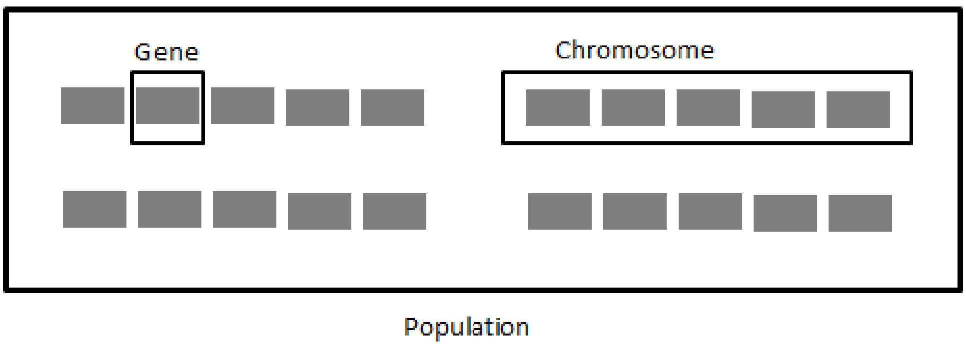

Figure 3.6. Representation of a population with four chromosomes.

Table 3.2. Relations between natural evolution and GA terminology and GA for VRP

Natural evolution	Genetic algorithm	Genetic algorithm for VRP
Population	Solution pool	Whole routing solutions
Chromosome	String (One individual)	All vehicle routes for one solution
Gene	Feature or character	One vehicle route
Allele	Feature value	Customer number
Genotype	Coded string	Order of customer
Phenotype	Decoded structure/Fitness value	Routes/Total distance

Genotype and Phenotype: Genotype is the structure of the solution produced by the computing system. In other words, it is the representation of the individual as a coded string. Phenotype specifies the properties encoded by the genotype of the individual. The properties mean the outward aspects of the individuals in the actual real world. GA works on genotype and phenotype spaces. The genetic operations like crossover and mutation are implemented to the genotype of the individuals whereas evaluation and selection processes are performed on the phenotype.

Encoding and Decoding: Coding means the representation or mapping of the problem. The mapping between genotype and phenotype is necessary to convert solution sets from the model into a form that the GA can work with, and for converting new individuals from the GA into a form that the model can evaluate. Encoding is the way to represent individual genes and chromosomes (Sivanandam and Deepa, 2008). While encoding represents a transformation process of a solution from the phenotype to the genotype space, decoding is a transformation process from the genotype to the phenotype space. Fitness value calculation is an example of decoding.

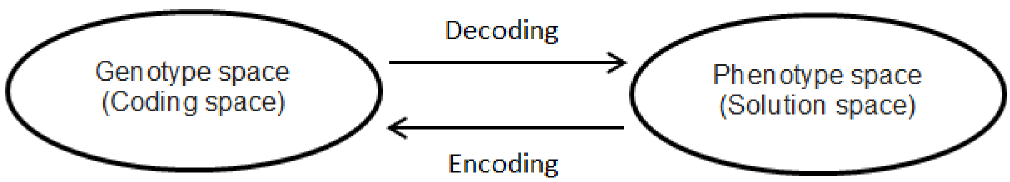

Figure 3.7. Transformation between spaces.

Search space: Search space or state space denotes the space of all feasible solutions which is the set of solutions among which the desired solution resides.

Fitness: It is the evaluation of an individual in accordance with the objective(s) of the problem. Fitness function is used to compute the fitness value for each solution. The objective function and the fitness function can be same depending on the problem. Fitness values show how good the solution is and close the solution to the optimum.

Genetic operators: Selection, crossover and mutation operators are used in the evolution process of GA. Based on a population of solutions denoted as chromosomes, two parents are chosen considering some criteria in selection step. Then, crossover operator is used to produce offspring solutions. Finally, mutation operator is utilized to ensure the diversity of the population (Labadie et al., 2016)

Search Termination: It is the stopping status of the algorithm. It can be based on some condition or criteria. A specified maximum number of generation or a specified elapsed time can be determined for ending the searching process. Or, the process can be terminated if there is no change to the population's best fitness value for a specified number of generations.

3.2.1.1. Genetic Algorithm Steps

The basic GA steps are as follow:

- **Step 1** (initial population generation). The initial population of solutions is generated for the problem.
- **Step 2** (fitness evaluation). The fitness value of each chromosome is evaluated in the population.
- **Step 3** (generation evolution). A new population is created by repeating following steps. The structure of the chromosomes is changed via genetic operators until the new population is completed.
    - **Step 4** (selection). The parent chromosomes are selected for the reproduction from the population according to some criteria generally related to the fitness values.
    - **Step 5** (crossover). The parents are recombined with a crossover probability, to generate the offspring (children).
    - **Step 6** (mutation). With a mutation probability, alter each offspring by a mutation operator.
    - **Step 7** (replacement). The fitness value of each offspring is evaluated, and a new population is formed by replacing the parent chromosomes by new chromosomes according to some criteria.

- **Step 8** (test). If the termination criterion is met, stop, and take the best solution in the current population. If it is not satisfied, go to step 3.

Figure 3.8 represents the main steps in GA. The GA cycle and the flowchart of a basic GA are as shown in Figure 3.9 and Figure 3.10.

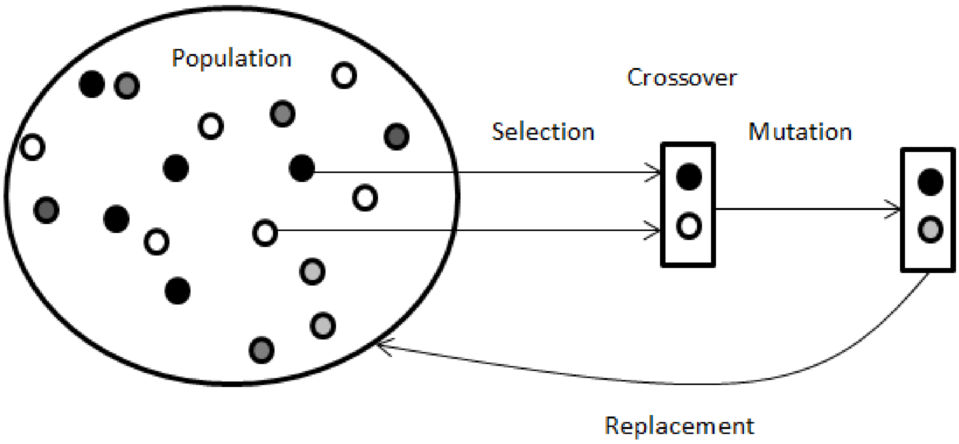

Figure 3.8. Main steps in GA.

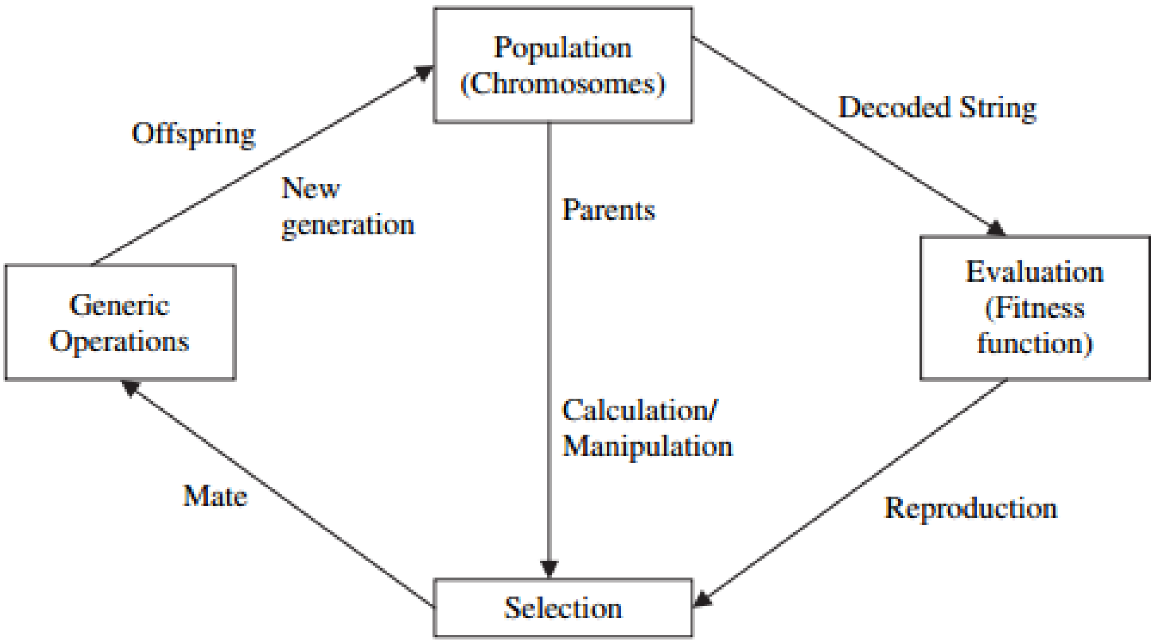

Figure 3.9. GA cycle (Sivanandam and Deepa, 2008).

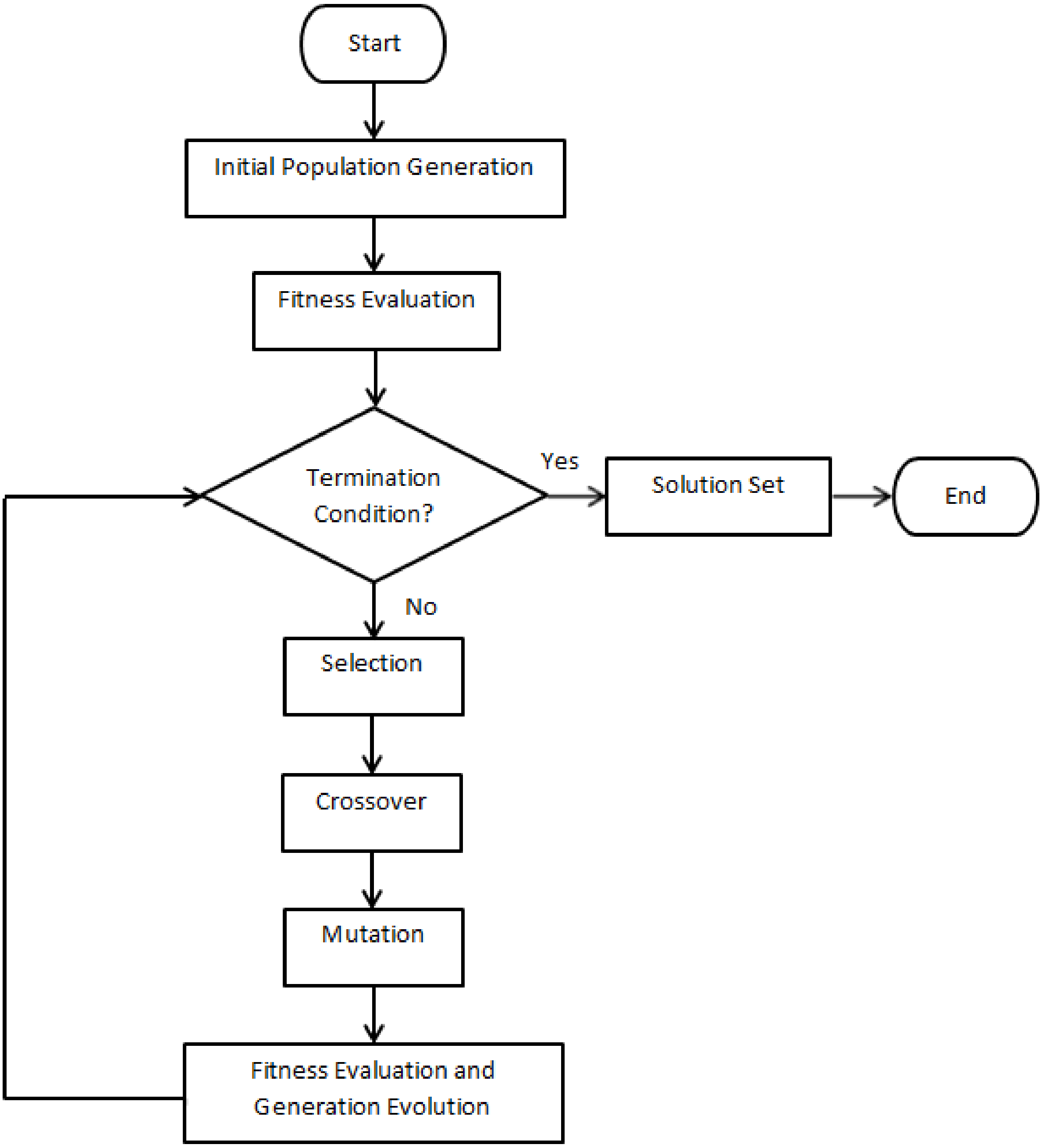

Figure 3.10. Flowchart of a basic GA.

3.2.1.2. Encoding

Encoding is a process of representing a chromosome. For searching and learning methods, the way the candidate solutions are encoded is quite effective on solving the problem. Besides, the problem has an impact on determining the encoding way of solutions. In general, fixed length strings have been used in many GA applications. However, in recent years, different encoding kinds are taking into account.

The representation of GA can be made by means of bits, numbers, trees, arrays, lists or any other objects (Sivanandam and Deepa, 2008).

- Binary encoding uses binary (bits) strings to encode the chromosomes with 0s and 1s. It is the most common encoding way.
- Octal encoding utilizes string that contains octal numbers (0–7).
- Hexadecimal encoding employs hexadecimal numbers (0–9, A–F) to form a string.
- Permutation encoding is used to encode the chromosomes with integer/real values in a sequence.
- Value encoding can use numbers, real numbers or characters (letters or words) to represent the individual solutions.
- Tree encoding is generally utilized to evolve program expressions for genetic programming. Each chromosome is a tree of some objects.

In this study, permutation encoding is considered. The solutions of a VRPTW are composed of the routes of the vehicles. The routes are formed of sequenced customers who sorted by the order of visits. Thus, the integer values represent the customer numbers, and the sequence shows the visiting order. Figure 3.11 represents a permutation encoding example of a chromosome which is a VRPTW solution consists of 9 customers and 3 vehicles.

3 5 7	2 1 6	4 8 9

Figure 3.11. Permutation encoding example.

3.2.1.3. Selection

Selection is the pairing process of the individuals as parents from the population for the reproduction. After determining the encoding schema, selection mechanism should be determined i.e. how to select individuals in the population that will generate offspring for the next generation and how many offspring each will produce. With the selection operator, the diversity of the individual in the algorithm will be increased, so that different regions can be searched in the solution space.

The selection operator mainly depends on the fitness values of the individuals. The individuals are matched considering their fitness values in the hope of producing offspring with better fitness values for the next generation. Generally, the chromosomes with higher fitness values have a greater chance to be selected as parents.

Typically, two types of selection mechanism can be distinguished, proportionate-based selection and ordinal-based selection (Sivanandam and Deepa, 2008). First one chooses individuals based upon their fitness values relative to the fitness of the other individuals in the population. Second one picks out individuals not upon their raw fitness, but upon their rank within the population. This is solely based upon the relative ordering (ranking) of the population.

The most commonly used selection methods are clearly summarized as follows:

- Roulette wheel selection which is one of the traditional selection methods is a proportionate-based selection scheme according to the relative fitness value of the chromosomes. So, the probability of the selection of a chromosome as a parent is directly proportional to its fitness value and the chromosomes with higher fitness values will have a greater probability to be selected to generate the next generation.
- Random selection method randomly selects a parent from the population.
- Tournament selection strategy first randomly selects t individuals from the population and holds a tournament competition among these individuals. The individual with the highest fitness value is the winner of the tournament. The tournament competition is iterated until the mating pool for generating new offspring is filled. The winners would be the parents of the offspring. t is the tournament size that is the parameter for the selection, and takes values ranging 2 to N (population size).
- Rank selection strategy ranks (sorts) the individuals in the population according to the fitness values first and then each individual takes a new fitness value defined by this ranking. The worst has the fitness 1 and the best has the fitness N (number of chromosomes in population). Assigning the new individual fitness depends only on its position in the individual rank and not on the actual fitness value. The parent selection process performs according to this new fitness. This strategy prevents very fit individuals from predominating over the less fit ones.
- Elitist selection operators guarantee that the selection of the best solutions of each generation as parents.

3.2.1.4. Crossover

After selection process of the parents for reproduction, crossover process starts. Crossover is known as the process of producing offspring by taking the parents in the hope to attain better solutions. A transfer of information between the chromosomes is at issue in the crossover process. The selected parent chromosomes are recombined via the crossover operator to generate a new population.

The crossover has a basic parameter. It is the crossover probability (P_c). It indicates how often the crossover is performed. If the crossover probability is 100%, the crossover operator has been applied to all parent pairs to produce the offspring. If crossover operator is not used, the offspring are exact copies of the parents.

The crossover operator changes the genetic information of one parent with the corresponding genes of the other. In other words, a parent pair is recombined in a certain way to produce one or more offspring. Due to this gene exchange, the offspring carry some of the characteristics of their parents and these characteristics are inherited on to the next generations. There are various crossover operators which can be found in the literature as follows; single point (one-point) crossover, two-point crossover, ordered crossover, precedence preservative crossover, and etc.

- Single point crossover operator selects a point as a cut point at first. The cut point (crossover position) can be chosen randomly. Then the genes after that point are exchanged between the chromosomes. Figure 3.12 presents a single point crossover.

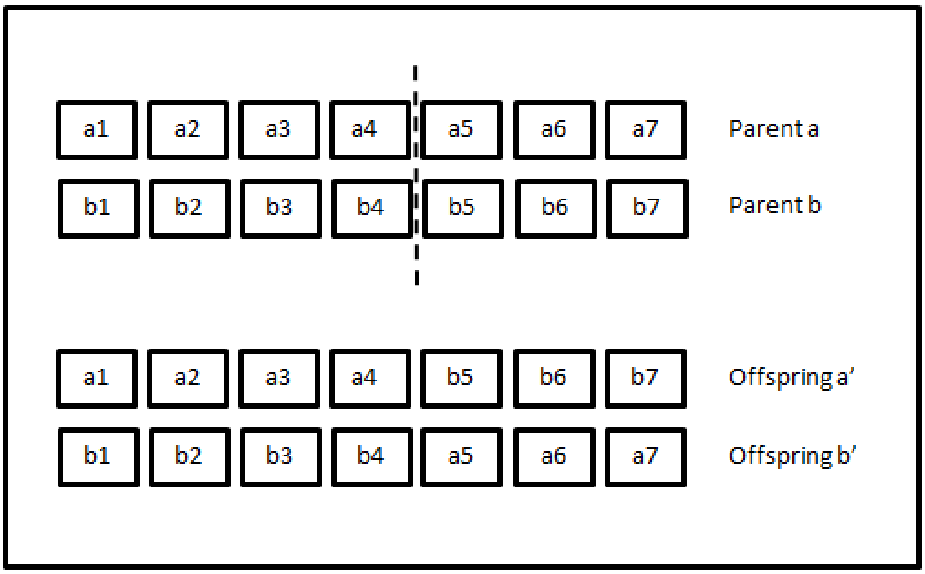

Figure 3.12. Single point crossover.

- Two-point crossover determines two cut points for two parents. Then the contents between these points are changed between the parent pair to generate new children for mating in the next generation. Figure 3.13 illustrates a two-point crossover.

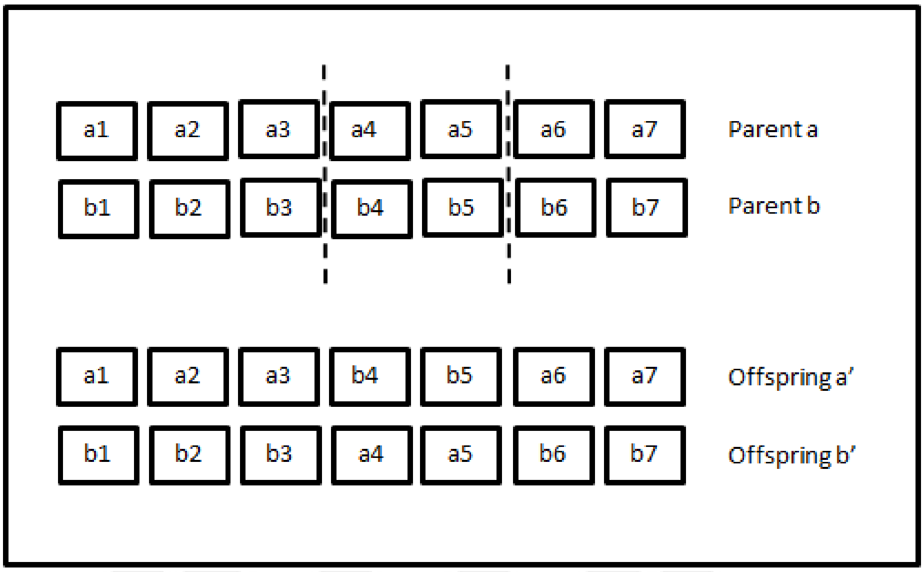

Figure 3.13 Two-point crossover.

3.2.1.5. Mutation

After crossover process, offspring are subjected to the mutation process. Mutation is used to preserve the genetic diversity. It prevents the GA to be trapped in a local minimum. The aim of mutation is to explore the search space which perhaps cannot be reached by crossover alone.

Mutation provides minor changes in the chromosomes with a low probability. The mutation probability (P_m) is the parameter of the mutation process. It shows how often the mutation is performed and the chromosomes are mutated. Mutation should not use very often, because then GA will in fact transformed to random search. Its probability is generally set to be inversely proportional to the number of variables. It is generally taken about 1/L, where L denotes the length of the chromosome (Sivanandam and Deepa, 2008). The greater the length of the chromosome is, the lower the probability is.

Mutation operator generally alters one or more genes in a chromosome. There are several mutation forms for the various kinds of representation. It can exchange a string position or alter a value of the string. Some of them are as follows:

- Insertion mutation operator selects one allele at random. Then removes from and inserts back into the chromosome in a different location.
- Inversion mutation selects two alleles at random and then reverses the alleles between them.
- Scramble mutation selects two alleles at random and then shuffles the alleles between them.
- Displacement mutation selects two alleles at random and the alleles between them are regarded as a group. Then the group is removed from and inserted back into the chromosome.
- Reciprocal exchange (swap) mutation chooses two alleles at random and then interchanges their positions.

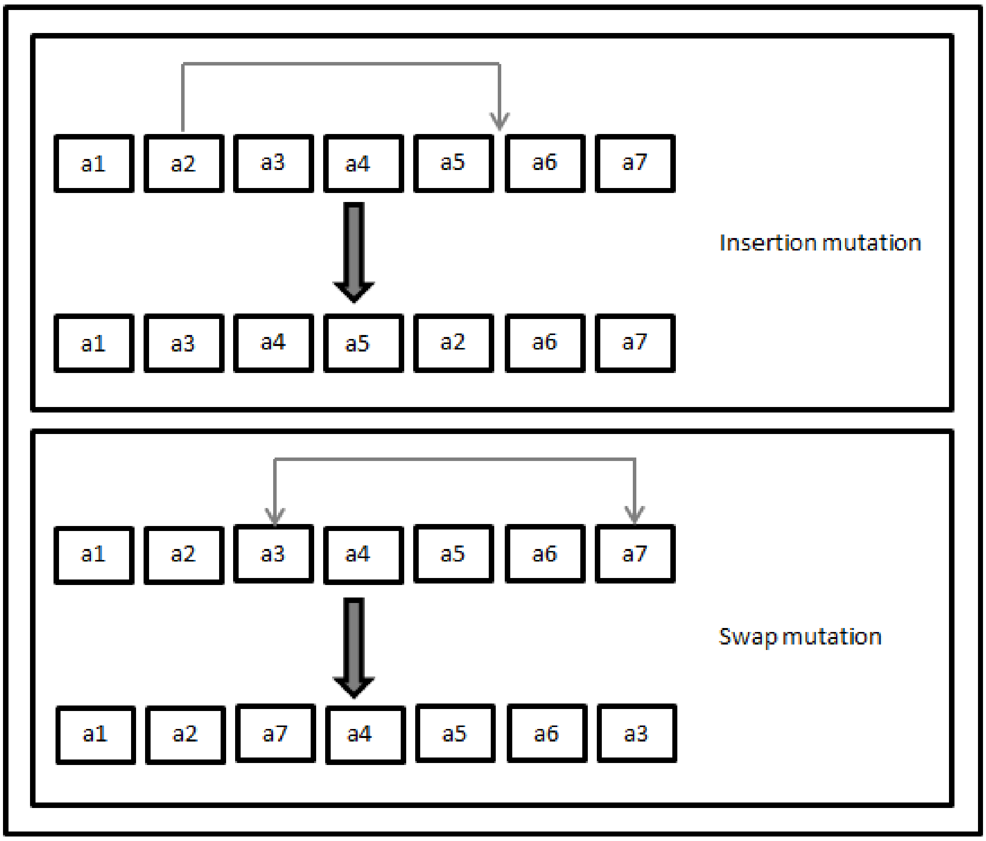

Figure 3.14. Mutation examples.

3.2.1.6. Replacement

A GA operates on a fixed size population. After generating the offspring by reproduction processes, it must be decided which of these newly generated offspring would move forward to the next generation and would replace which chromosomes of the current generation. The process of composing the next generation of individuals by replacing or removing some offspring or parent individuals is done by replacement operator. This process in evolution is known as replacement scheme. It defines the survival principle of the evolution. The individuals transferred to the new generation will be the parents of that next generation.

Replacement is the last stage in a GA cycle. Fitness evaluation of the new chromosomes is made, and generation evolution takes place in this stage according to a replacement strategy. Basically, there are two types of replacement strategies:

- Generational replacement strategy replaces all chromosomes of the population with the newly generated offspring. Full replacement of the parent population with new population of children is at issue. There are derived forms of the generational replacement. One of them evaluates combined chromosome set of parent and offspring ones considering their fitness values and transfers the chromosomes as many as the population size to the next generation. The other form generates offspring more than the population size and replaces the best ones as many as the population size.
- Steady state replacement strategy inserts the offspring in the population when they are produced, as opposed to the generational replacement where a complete new generation is generated at each time step. The insertion of a new individual generally requires the replacement of another population member. The individual to be removed may be selected as the worst one, or as the best one, or as the most similar one of the population. Or it can be chosen randomly or by a tournament method.

Elitism, or elitist selection, is a property of selection techniques. Elitist selection strategy keeps the best individual(s) of the population to the next generation. Every time a new population is generated, there is a probability of destroying the chromosome with the best fitness value because of the crossover and mutation processes. Retaining the best one or the few individuals in a generation unchanged in the next generation is elitism, and it guarantees the survival of the best individual(s). The number of elite individuals should be low; otherwise it causes to premature convergence, and degeneration of the population. Elitism significantly improves the GA's performance.

3.3. Hybrid Algorithms

Studies in metaheuristics for combinatorial optimization problems, so for VRPs, tend towards hybridization. The fundamental motivation of this tendency is to utilize the strengths of a number of algorithms to have a stronger approach. A first hybridization concept emerged by bringing together different metaheuristics in one main frame with the aim of completing each other and reaching more effective solutions when they are operated one by one (Labadie et al., 2016). Hybrid algorithms utilize a combination of exact, heuristic and metaheuristic methods to resolve the problems.

The main hybridization generally used to consolidate the metaheuristics can be classified in four forms. The first form comprises of including specific characteristic of a metaheuristic into another one; e.g. the application of metaheuristics for each individual in the use of population-based metaheuristics. The second scheme of hybrids forms by replacing a component from one metaheuristic into another one. Trajectory methods are better in exploring promising areas in the search space while population based methods are better in identifying promising areas in the search space. Thus, metaheuristic hybrids are generally successful since they combine the advantage of population based methods with the strength of trajectory methods (Blum and Roli, 2003). The third scheme takes place of running two or more metaheuristics sequentially, which signifies that the output solutions of the first metaheuristic are given as inputs to the next one to reach more effective results. The last scheme is based on a decomposition of a main complex problem into sub-problems. These problems are figured out by various metaheuristics that cooperate and exchange information within an upper-level method to reach high-quality complete solutions for the overall problem. (Labadie et al., 2016)

Laporte et al. (2014) have specified some important hybridization families. Population-based and local search methods complementation, meta-meta hybridizations, hybridizations with large neighborhoods, hybridizations exact algorithms, parallel algorithms, decompositions or coarsening phases, diversification vs. intensification are the family segments. The hybrid methods which combine metaheuristics with mathematical programming solvers or other exact algorithms are called matheuristics.

A taxonomic study of hybrid metaheuristics presented by Talbi (2002) can be examined for further information. According to that study, a high percentage of metaheuristics hybridizing population-based metaheuristics with local search heuristics has been created for various optimization problems.

Because GA is a versatile and effective approach, it is prone to the hybridization. When a GA is incorporated with other techniques, a hybrid GA is formed. It aims to increase the probability of getting the best solution of an optimization problem and to decrease the time of searching. Hybrid GAs have been drawing attention in recent years and are being widely used to figure out optimization problems.

3.4. Multi Objective Optimization Techniques

MOO takes into account more than one objectives to be optimized simultaneously. Generally, in MOO, each objective function considers a different feature of a desired result. Most of the time, these objectives are in a conflict where there is no single result that simultaneously optimizes all functions. That means further improvement of one of the objectives may cause to deterioration of the other. Therefore, a set of better solutions are at issue. These solutions are not superior to each other. This set is called Pareto optimal set in Pareto optimality concept. Each solution in this set is an alternative solution of the optimization.

3.4.1. Pareto Optimality

There is no single global optimum solution in MOO problems as it is in single objective optimization problems. In MOO problems, there is a set of best solutions called Pareto optimal solutions instead of one single best solution. A solution which is good according to an objective in the solution set of MOO problems can be bad according to the other objective. Therefore, the main purpose is to find or approximate to the Pareto front and Pareto optimal set, and to provide alternative choices for the decision.

In MOO problems, the objective functions can aim minimization or maximization. For this reason, there are three possible situations: all the objective functions are minimized; all the objective functions are maximized; and some are minimized, and others are maximized. So, a MOO problem with two objective functions has four different scenarios for the Pareto front set. There is an illustration of the Pareto fronts of a feasible objective space in Figure 3.15 for the scenarios.

For finding the Pareto optimal solutions, some dominance terms are used. For a solution, if any of the objective functions cannot be improved without deteriorating the other objective function, then the solution is Pareto optimal. That means a Pareto optimal solution set is the set of non-dominated solutions. They are not dominated by any other solution in all objective function values. These solutions cannot simply be compared with each other. The solutions which are non-Pareto optimal ones compose the dominated set of solutions. Figure 3.16 demonstrates the dominated and non-dominated solutions with Pareto front.

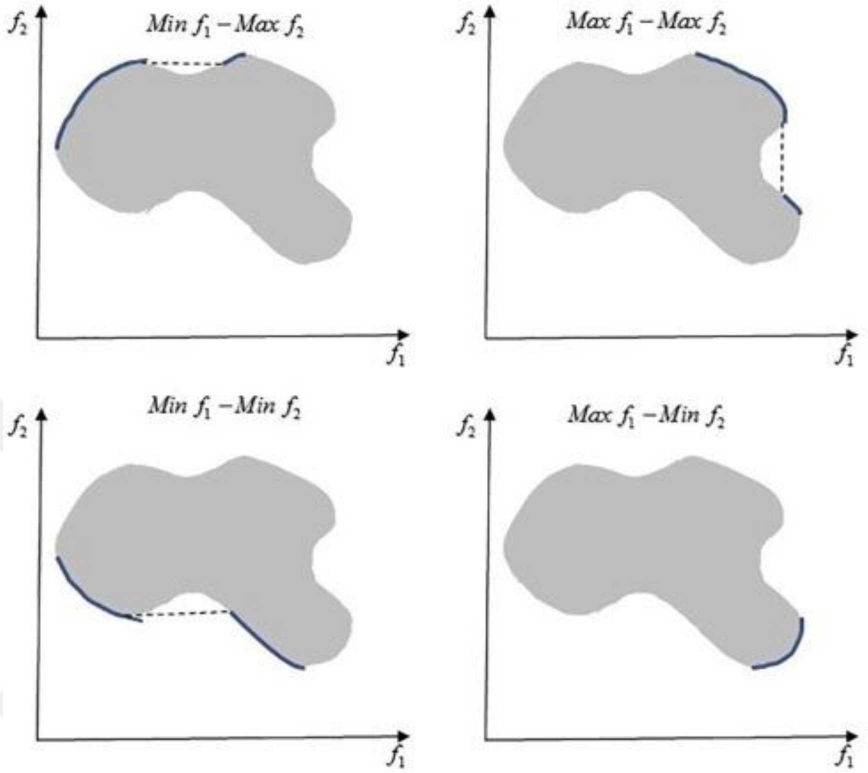

Figure 3.15. Pareto-Front set for four different scenarios with two objective functions (Correia et al., 2017).

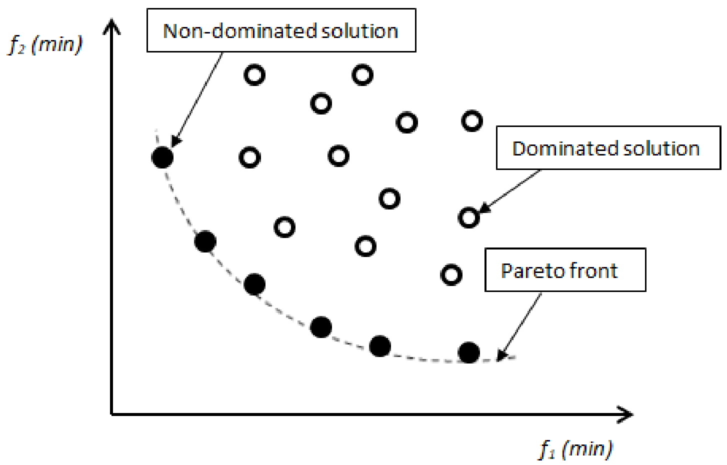

Figure 3.16. Dominated and non-dominated solutions with Pareto front.

3.4.2. NSGA-II

NSGA-II (Non-dominated Sorting Genetic Algorithm II) is a fast and elitist MOO solution method (Deb et al., 2002). NSGA-II can be used to determine whole points of the Pareto fronts. Also, all the solutions (i.e. points or individuals) can be classified considering their dominance over the other solutions. When a solution dominates another solution, the situation occurred is as follows: Both values of the objective functions are not worse and at least one objective function value is better (Seshadri, 2006). For example, solution A dominates solution B in a minimization problem in the following cases:

$$A \leq B \; (A \; dominates \; B) \Leftrightarrow \; \forall_i : A_i \leq B_i \,, \exists_i : A_{i_0} < B_{i_0} \qquad (1)$$

Binary comparisons are made for all solutions in the population considering each objective function. In accordance with this comparison for each solution i, the domination count (n_i) that is the number of solutions which dominate i is computed, and a set of solutions (S_i) that the solution i dominates is created.

The solutions whose n value is zero are offered in the first non-dominated front which is Pareto front. The rank of the solutions in the first front is 1. To detect the other fronts, the n values of the solutions in the S sets of the solutions that are in the first front are decreased by 1. If any n values of a solution that becomes zero, that solution is a member of the second non-dominated front and has a rank 2. These steps are repeated until all fronts are determined. Figure 3.17 illustrates the ranking scheme. The low rank is more preferred, and the solutions are classified from low to high ranks.

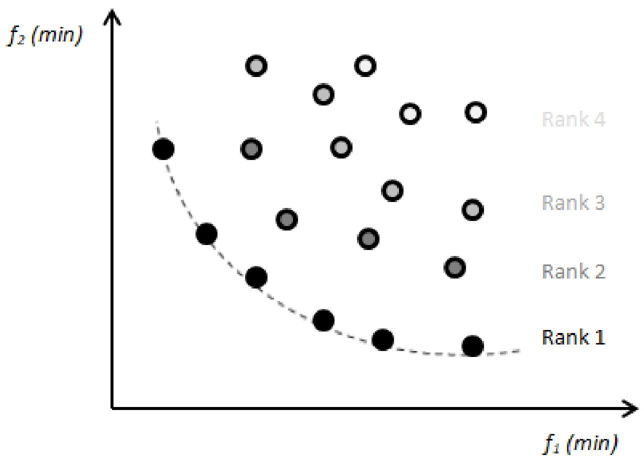

Figure 3.17. Ranking scheme of the solutions.

Crowding Distance:

For sorting the solutions which are on the same front, a second measure is required. Therefore, crowding distance is utilized. The crowding distance can be defined as the sum of the Euclidean distance of the solution i to its neighbors for each objective. It is computed by Eq. 2.

$$CD_i = \sum_{m=1}^{n} \frac{|f_{i+1}^m - f_{i-1}^m|}{f_{max}^m - f_{min}^m} \tag{2}$$

Where, CD denotes crowding distance, n represents total objective numbers, and f denotes objective function value. f_{max}^m and f_{min}^m are the biggest and the smallest values of the m'th objective function on the front. The biggest and the smallest values of the objective function belong to boundary solutions of the concerned front and it is considered that they have infinite crowding distance value. Crowding distances of the solutions between boundary individuals are computed like in the Figure 3.18. Solutions with small crowding distances are closer to other individuals. The order of dominance of the individuals on the same front is done towards to those who have the high distances. In a comparison, the individuals who are away from the density (i.e. who have high crowding distance) are chosen.

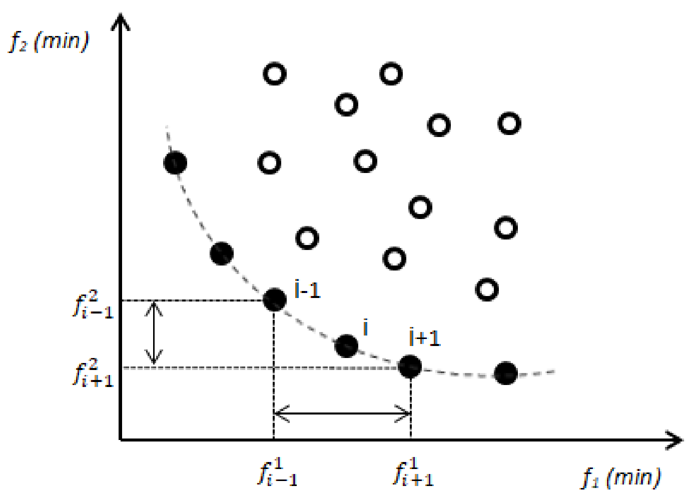

Figure 3.18. Crowding distance calculation for the solutions on the same non-dominated front.

Elitist Selection:

After the genetic operators that are crossover and mutation, new population is formed. As given in the Figure 3.19, for the selection of the next population, the population of that generation and the produced offspring population are combined. Therefore, the conservation of the best solutions in the current population and the transference to the next generation, that is, elitism is ensured. Each population has N solutions. 2N solutions are sorted considering their dominance with the non-dominated sorting for choosing N solutions. The selection process is initiated by transferring the solutions of the rank 1 to the new generation. By moving in the first, second, third and fourth fronts, the solutions are received to the population in turn. When the total number of solutions exceeds the population size, the crowding distances of the last front solutions are checked. For this reason, the crowding distance values of the solutions in the concerned front are classified in descending order and the solutions as much to complete the new population is chosen as the new population. Consequently, the generation of the next population is completed.

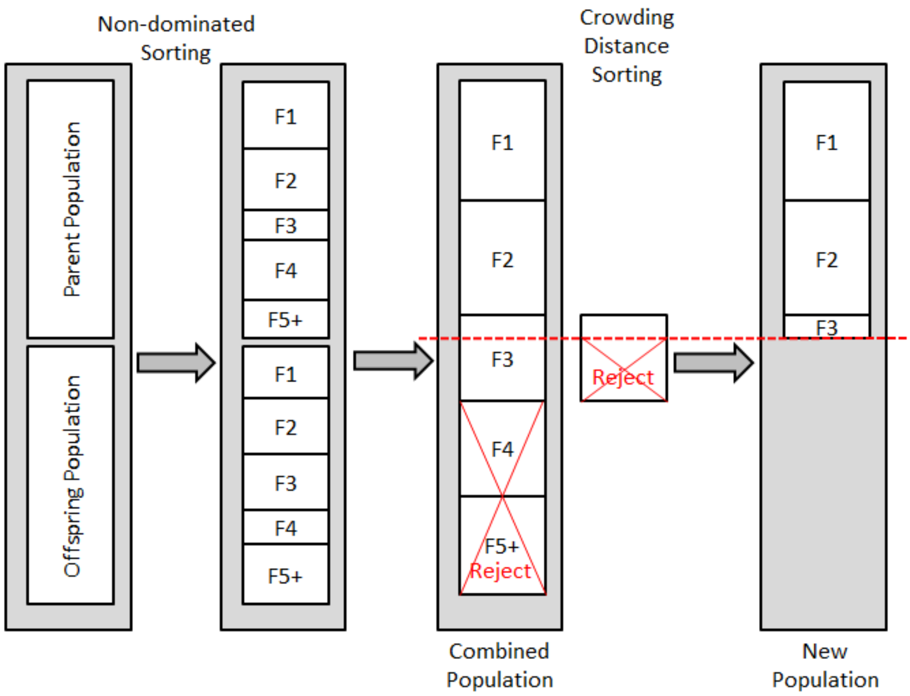

Figure 3.19. Elitist selection procedure of NSGA-II.

CHAPTER 4. LITERATURE REVIEW

First paper published on the VRP is written by Dantzig and Ramser under the title “The Truck Dispatching Problem” in 1959. It is intended to find optimum routes for a fleet of gasoline delivery trucks between a bulk terminal and many service stations supplied by the terminal. They proposed a mathematical programming model and algorithmic approach for the solution. In 1964, Clarke and Wright developed a construction heuristic that advanced on the Dantzig-Ramser method. Following these two influential studies, a lot of models and algorithms have been proposed to solve the VRP and its varieties.

Many approaches that have utilized exact, heuristic and metaheuristic algorithms have been developed heretofore for solving VRPTW. If the size of the customer set is small, the exact algorithms can be utilizable; else if the set is getting larger it is not viable to use these algorithms due to the high solution duration. For this reason, the solution approaches of the problem via heuristic and metaheuristic strategies which are proposed optimal or approximate solutions are growing in the literature recently (Çolak and Güler, 2009). Saving algorithm, sweep algorithm, and petal algorithms are some of the instances of the classical heuristics; genetic, simulated annealing, taboo search, ant colony, particle swarm and local search algorithms are some of the examples of the metaheuristic techniques (Şahin and Eroğlu, 2014).

Heuristic methods can produce solutions, or they can provide the iterative development of feasible solutions. Their solution capabilities nonetheless are restricted by problem size and complexity. Besides, they may end the search at a ‘‘local’’ optimal solution, disregarding better solutions in different regions of the solution space (Griffis et al., 2012). Therefore, there is a tendency of using heuristic methods with metaheuristics together as mentioned in the hybrid algorithms section. It aims to utilize the strengths of a number of algorithms to have a stronger approach.

GA is one of the metaheuristic strategies which are utilized very often in dealing with the VRP and VRPTW. GA has a capability of building a hybrid algorithm with classical heuristics or other metaheuristics. Good initial solutions which are generated by some construction heuristics generally result in better final solutions after the application of improving heuristics. Thus, producing good initial solutions is substantial in any solution technique (Na et al., 2011). According to Baker and Ayechew (2003), an initial population of feasible solutions will evolve to effective solutions in a relatively small number of generations of the GA.

Baker and Ayechew (2003) have been offered a hybrid approach for the solution of a CVRP. They have been hybridized the GA by using neighborhood search methods. Sweep and generalized assignment approaches in the initial population generation step of GA have been used.

Karagul and Gungor (2014) have been composed the initial solution space with saving algorithm, sweep algorithm and random permutation alignment for their proposed GA. Then, standard GA and random search algorithms that are two well-known solution techniques have been used for evolving the initial solutions.

Mester and Bräysy (2007) have been employed a competitive approach that composed of guided local search and evolution strategies metaheuristics into an iterative two-stage procedure for CVRP.

Berger and Barkaoui (2004) have been applied the sequential insertion heuristic together with GA for the solution of a VRPTW. Their proposed parallel hybrid GA has dialed with the simultaneous evolution of two populations with different objectives. While one of the populations was aimed at minimizing total travelled distance, the other was aimed at minimizing temporal constraint violation.

Thangiah et al. (1991) have been offered a GA system named as GIDEON which is composed of two distinct modules which the one module is forming the clusters of customers and the other is forming the routes.

Ibrahim et al. (2016) have been applied a hybrid GA which hybridized with the nearest neighbor heuristic method to a real VRPTW circumstance of bottled water delivery from warehouse to retail. Their proposed hybrid GA results have been compared with the company actual route according to total distance, total time, total cost and total penalties. With the proposed algorithm, they have reached better solutions.

Prins (2004) has been proposed an effective evolutionary algorithm for VRP. He has developed a Splitting algorithm to split the routes in a chromosome without using any trip delimiters and applied local search procedure in 9 different rules in the mutation step. The algorithm has obtained competitive results.

Chang and Chen (2007) have adapted Prins's algorithm to a VRPTW. They have considered a single-sided time window (only the earliest arrival time is included, not the latest arrival time) and vehicles with unlimited capacity for simplicity. They have performed 3 different population sizes and 3 different mutation rates for 2 different data sets. They have concluded that as increasing of the mutation rate, not only relative errors are decreasing, but also generally the number of trips is decreasing. Furthermore, while population size and mutation rate increase, the results are improving.

Ombuki et al. (2006) have been developed a multi objective GA approach using the Pareto ranking technique to the VRPTW. The objectives considered were minimizing the number of vehicles and total cost. Through the Pareto fitness evaluation, it has been unrequired to give weights to the objectives for weighted sum method. They have reached quite effective solutions.

Haddadene et al. (2016) have been proposed hybridized NSGA-II for VRP with time windows, synchronization and precedence (VRPTW-SP) constraints on the field of home health care. Their objectives were traveling cost minimization and patients' preferences maximization. They have compared basic NSGA-II and local search based NSGA-II and have concluded that the hybrid NSGA-II is more suitable than the basic one.

Tan et al. (2006) have been developed a hybrid multi objective evolutionary algorithm (HMOEA) that includes different heuristics for local exploitation in the evolutionary search and specializes with genetic operators and variable length chromosome representation. The objectives were minimization of traveling distance and number of vehicles. The HMOEA produced very good results.

Mungwattana et al. (2016) have been devised a method by GA, modified push forward insertion heuristic (MPFIH) and λ-interchange local search descent method (λ-LSD) for VRPTW. Minimizing vehicle number and minimizing total travel time have been determined as objectives for the problem that considered soft time window constrained. The results of their proposed algorithm provide effective solutions in general.

Ghoseiri and Ghannadpour (2010) have been presented a model using goal programming and GA for the solution of VRPTW. They also have considered the problem as multi objective that the objectives were minimizing vehicle fleet size and total traveling distance. In their model various heuristics were incorporated, Pareto ranking scheme was used for Pareto ranks instead of fitness values, and elitism strategy was applied to keep good solutions among the generations. They have reached quite sufficient solutions.

Najera and Bullinaria (2011) have been offered a multi objective evolutionary algorithm for solving the VRPTW. Their proposed algorithm has integrated with Pareto ranking technique and a similarity measurement method. It has achieved highly competitive results.

Göçken et al. (2017) have been presented a hybrid multi objective GA for the solution of a VRPTW. They have integrated NSGA-II to their proposed GA. They also have considered minimizing total distance and waiting time of the vehicles as objective functions. They have tested the difference of the generation initial population techniques and have concluded that sweep algorithm has achieved better solutions.

Additional survey about VRPTW solving via evolutionary algorithms (i.e. GAs and evolution strategies) can be found in the study of Braysy et al. (2004).

Table 4.1. Overview of the reviewed studies.

Reference	Problem Type	Solution Method	Initial Population Generation Method	Objective(s)	Data Set
Baker and Ayechew (2003)	CVRP	GA	Sweep and generalized assignment approaches	• Minimizing total distance travelled	Benchmark on OR library
Karagul and Gungor (2014)	Fleet Size and Mix VRP	Standard GA and random search	Saving algorithm, sweep algorithm and random permutation alignment	• Finding shortest distance by using the minimum cost	Real-life data
Mester and Bräysy (2007)	CVRP	Active-guided evolution strategies	Hybrid cheapest insertion heuristic	• Minimizing total cost route	Benchmarks of Christofides et al., Golden et al., Li et al. and Gehring and Homberger
Berger and Barkaoui (2004)	VRPTW	Parallel hybrid GA	Sequential insertion heuristic	• Minimizing total traveled distance • Minimizing temporal constraint violation	Solomon's benchmark
Thangiah et al. (1991)	VRPTW	GA system	Sweep algorithm	• Minimizing the route cost	Solomon's benchmark
Ibrahim et al. (2016)	VRPTW	Hybrid GA	Random and nearest neighbor heuristic	• Minimizing total distance	Real-life data
Prins (2004)	CVRP	Evolutionary algorithm	Saving, sweep and sequential route building algorithms and random permutation	• Minimizing total cost route	Benchmarks of Christofides et al. and Golden et al.
Chang and Chen (2007)	VRPTW	GA	Random	• Minimizing total cost route	Standard VRPTW instances

Table 4.1. Overview of the reviewed studies (Continued).

Reference	Problem Type	Solution Method	Initial Population Generation Method	Objective(s)	Data Set
Ombuki et al. (2006)	VRPTW	Multi objective GA	Random permutation and greedy procedure	• Minimizing number of vehicles • Minimizing total cost.	Solomon's benchmark
Haddadene et al. (2016)	VRPTW-SP	Hybridized NSGA-II	Parallel randomized constructive heuristic	• Minimizing travel cost • Maximizing patients preferences	Bredström and Rönnqvist's benchmark
Tan et al. (2006)	VRPTW	Hybrid multi objective evolutionary algorithm	Random	• Minimizing traveling distance • Minimizing number of vehicles	Solomon's benchmark
Mungwattana et al. (2016)	VRPTW	Hybrid GA	Modified push forward insertion heuristic	• Minimizing total travel time • Minimizing number of vehicles	Solomon's benchmark
Ghoseiri and Ghannadpour (2010)	VRPTW	GA and goal programming	Random, push forward insertion heuristic and λ-interchange mechanism	• Minimizing total required fleet size • Minimizing total traveling distance	Solomon's benchmark
Najera and Bullinaria (2011)	VRPTW	Multi objective evolutionary algorithm	Random	• Minimizing number of routes • Minimizing travel distance • Minimizing delivery time	Solomon's benchmark
Göçken et al. (2017)	VRPTW	Hybrid multi objective GA	Sweep and nearest neighbor algorithm	• Minimizing total distance • Minimizing total waiting time of the vehicles	Solomon's benchmark

CHAPTER 5. VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

VRP is the problem of constructing optimal routes to vehicles that will serve a customer set. The information of the customers and the depot(s) (e.g. numbers, demand quantities, geographic data) are known before starting the solution of the problem. The vehicles that serve the customers are assumed to compose of a homogeneous fleet with a capacity limitation. The total demand of the customers travelling on the same route should not exceed the capacity limit of a vehicle. Each vehicle begins onto the route from the depot and returns to the depot at the end of the route. The requirements of each customer must be met in one single vehicle at a time.

VRP with only one restriction, i.e. vehicle capacity restriction, refers to the Capacity Vehicle Routing Problem (CVRP). The components which all are found in CVRP, i.e. a homogeneous fleet with certain number of vehicles with the same capacity and characteristics, customers with known demands and locations, a warehouse with a known geographical location, are exist also in VRPTW. The constraint that makes VRPTW different and challenging from other VRP varieties is that there is a specific time interval (e_i, l_i) at which service can be started for each customer. This time interval is the time window constraint of each relevant customer. The time window of the depot signifies the maximum traveling time of the individual routes. Figure 5.1. shows a simple VRPTW example. Making delivery or offering services to each customer takes as long as service duration s_i. At the end of the service duration, the vehicle drives to the next customer or the depot. The service cost is decided according to the necessary number of vehicle for service or delivery to the customers and the total distance that have been travelled by the vehicles (Ho et al., 2001). The minimization of the total cost is the main objective of the problem.

VRPTW is divided into two according to hard or soft situation of the time window constraint.

- In hard time window situation, customers are not allowed to be served outside of the time interval. The vehicle has to wait if it arrives before the ready time of the customer; and also, cannot serve after the due date.
- In soft time window situation, the time window restriction may be violated in return of the penalty cost.

In this thesis, VRP with hard time windows is tried to be solved.

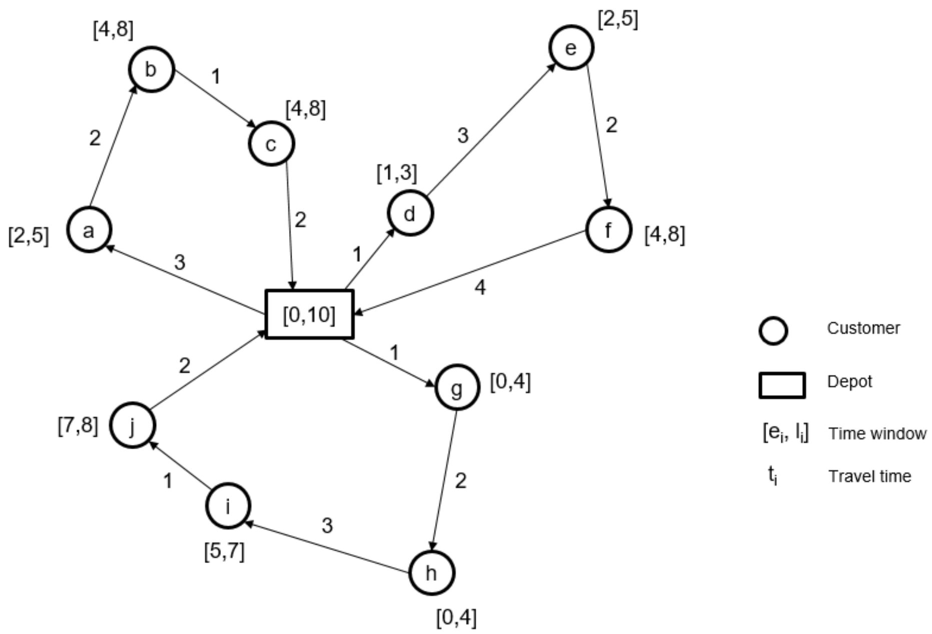

Figure 5.1. VRPTW example.

5.1. VRPTW Assumptions

The assumptions of the problem are listed below:

- The number of customers is stable and known;
- Each customer's geographic location is known;
- The maximum number of vehicle is stable;
- The fleet of the vehicles is homogeneous and have constant capacities;
- The vehicles can be loaded and routed only once;
- All the vehicles depart from the depot at the time t=0;
- When the customer is reached in the time window interval, service is started at that time;
- The depot has sufficient capacity limit to meet the demands of all customers;
- The distance between the nodes is determined by Euclidean distance formula;
- 1 unit of distance is equal to 1 unit of time;
- Transportation costs depend on travel distance.

5.2. VRPTW Constraints

The constraints of the problem are written below:

- The first location and last destination of the routes should be the depot;
- The vehicles must go back to the depot before the maximum travel time is up;
- The demand of each customer must be met in one single vehicle at a time;
- The total demand of the customers travelling on the same route should not be in excess of the total capacity of a vehicle;
- The service of the customers should start in their time window. The vehicles that reach before the ready time have to wait the customer.

5.3. VRPTW Objectives

In this study, the following objectives are considered:

- Minimization of the total distance, and
- Minimization of the total waiting time of the vehicles.

The objectives are implemented for the solution of the problems thus MOO is conceived. To achieve effective results, the selected objectives should be conflict to each other in MOO. The received results are analyzed and compared with respect to the objective functions.

5.4. Mathematical Model of VRPTW

The decision variables, parameters and classes defined in the mathematical model of VRPTW are indicated as follows:

Decision variables:

x_{ijk} $\begin{cases} 1, \text{if vehicle k travels directly from customer i to customer j} \\ 0, \text{otherwise} \end{cases}$

t_i The arrival time to customer i

w_i The waiting time at customer i

Parameters:

d_{ij} The distance between customer i and customer j

s_i Service duration of customer i

e_i Earliest arrival time of customer i

l_i Latest arrival time of customer i

l_0 Latest arrival time of the depot (Maximum travel time one of each vehicle)

m_i Demand of customer i

Q Capacity of identical vehicles

Classes:

C $\{1, 2, \ldots, n\}$ Customer class

N $\{0, 1, 2, \ldots, n\}$ Node class (Node 0 represents the depot.)

V $\{1, 2, \ldots, v\}$ Vehicle class

The VRPTW model can be mathematically formulated as follows:

Minimize

$$\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{v} d_{ij} * x_{ijk} \tag{3}$$

$$\sum_{i=1}^{n} w_i \tag{4}$$

Subject to

$$\sum_{k=1}^{v}\sum_{j=1}^{n} x_{0jk} \leq v \tag{5}$$

$$\sum_{j=1}^{n} x_{0jk} = \sum_{j=1}^{n} x_{j0k} \leq 1, \quad \forall k \in V \tag{6}$$

$$\sum_{k=1}^{v}\sum_{i=0\ i\neq j}^{n} x_{ijk} = 1, \quad \forall j \in C \tag{7}$$

$$\sum_{k=1}^{v}\sum_{j=0, j\neq i}^{n} x_{ijk} = 1, \quad \forall i \in C \tag{8}$$

$$\sum_{i=1}^{n}\left(m_i * \sum_{j=0, j\neq i}^{n} x_{ijk}\right) \leq Q, \qquad \forall k \in V \tag{9}$$

$$\sum_{i=0}^{n}\sum_{j=0, j\neq i}^{n} x_{ijk}(d_{ij} + s_i + w_i) \ \leq \ l_0, \qquad \forall k \in V \tag{10}$$

$$t_0 \ = \ w_0 \ = \ s_0 \ = 0 \tag{11}$$

$$t_i + w_i + s_i + d_{ij} = t_j, \qquad \forall (i,j) \in N, i \neq j, if\ x_{ijk} = 1 \tag{12}$$

$$w_i = max\{e_i - t_i, 0\}, \qquad \forall i \in C \tag{13}$$

$$e_j \leq (t_j + w_j) \leq l_j, \qquad \forall j \in C \tag{14}$$

$$x_{ijk} \ \in \ \{0,1\}, \qquad \forall k \in V, \forall (i,j) \in N \tag{15}$$

$$t_i \ \geq 0, \qquad \forall i \in C \tag{16}$$

$$w_i \ \geq 0, \qquad \forall i \in C \tag{17}$$

Objective functions defines Eq. 3 total distance, Eq. 4 total waiting time of the vehicles should be minimized. Eq. 5 indicates there are maximum v vehicles departing from the depot. There is not a necessity of the usage of all vehicles. Eq. 6 verifies that the depot is the beginning and ending nodes of each route. Eq. 7 and Eq. 8 ensure that each customer can be visited by single vehicle at one time. Eq. 9 avoids exceeding the capacity of the vehicle, with the total demand of the customers that are placed in the same route. Equations 10-14 demonstrate the time windows restriction. Eq. 10 is the maximum travel time constraint. Eq. 11 identifies the decision variables and a parameter of the depot for Eq. 12 in the state of i is 0. And Eq. 12 calculates the arrival time of customer j in case of the vehicle travels from customer i to customer j by adding waiting time at customer i, service time of customer i, and travelling time from customer i to customer j to the arrival time of customer i. Eq. 13 computes the waiting time of a vehicle at customer i in case of arriving the vehicle to the customer before the customer's earliest arrival time. Eq. 14 ensures that the service starts in the time window of the customer. Equations 15-17 detect the sets of the values that the decision variables can take. This mathematical model determines the feasible solutions for VRPTW.

5.5. Application Areas of VRPTW

VRPTW is often preferred in systems where the shelf life is short, or the distribution period is short. Some of the applications of the VRPTW are (El-Sherbeny, 2010):

- Bank and postal deliveries,
- Waste collection,
- Milk delivery,
- National franchise restaurant deliveries,
- School and urban bus routing,
- Ship, train, and aircraft scheduling,
- Security patrol services, and
- Emergency services.

CHAPTER 6. PROPOSED ALGORITHM

In this thesis, the effect of initial population of GA on multi objective problems is investigated. The initial populations are generated first randomly, the second by a nearest neighbor based algorithm, and the third by a sweep based algorithm. Thus, GA becomes hybridized. Our intention is to consolidate the advantage of GA with the strength of the constructive heuristics; sweep algorithm or nearest neighbor algorithm. The outputs of the constructive heuristics are given as inputs to the GA to reach more effective results in less computational time.

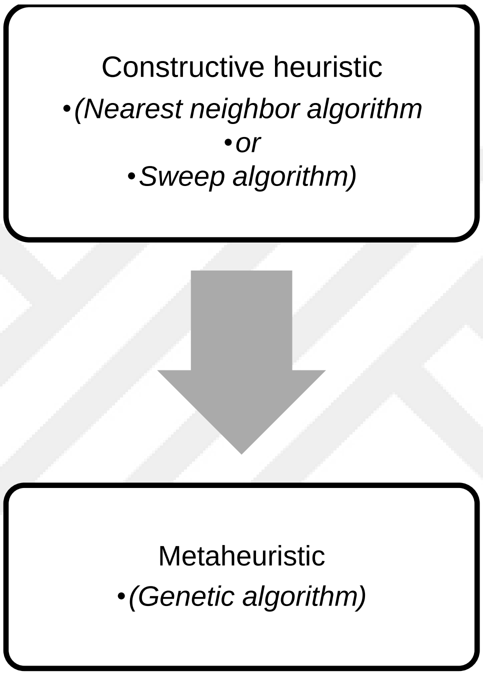

Figure 6.1. Framework of proposed hybrid algorithm.

6.1. Initial Algorithm I (Random)

The GA's primary stage is producing the initial population formed of feasible solutions. The primary algorithm of producing initial population applied in the study is established on randomness. It is going to be observed that how effective the use of constructive heuristics for generating initial population instead of randomly production of solutions. Taking the time window constraint into account, the steps of the Initial Algorithm I are as follows:

- **Step 1.** A customer is selected at random as the first visited location of the first route.

- **Step 2.** Then, a non-routed customer is selected randomly again. If the capacity and time restrictions are not violated, it is placed on the current route after the former customer. If any of the restrictions are violated, a new route is built, and this customer would be the first node of that route.
- **Step 3.** Iterate Step 2 until all customers are routed.

6.2. Initial Algorithm II (Nearest Neighbor Based)

The second algorithm of generating initial population has been developed from basis on the Nearest Neighbor Algorithm. The Nearest Neighbor algorithm picks the other customer who is nearest according to the distance to the previous customer who is located in the route. Nevertheless, because of existing the time window restriction, the determination of the customers to be appointed to the route is not enough for achieving the optimum solution by considering only the distance. The steps of the Initial Algorithm II are as follows:

- **Step 1.** The comparison that the Euclidean distance between the customer and the depot and the ready time of the customer for the service that is present in the customer data is made for choosing the first customer to be assigned to the route that start from the depot. The bigger value of this evaluation is designated as the selection value (c_i) to that customer as seen in Eq. 18. Due to the assumption of that the travelling of 1 unit distance takes 1 unit time, the comparison can be made clearly.

$$c_i = max(d_{0i}, e_i) \tag{18}$$

- **Step 2.** The customer who has the minimum c_i is chosen to the route as the first node. Each chosen customer is discarded from the customer list and the capacity loaded of the vehicle is computed.
- **Step 3.** The distances between the non-routed customers in the customer list and the last customer added to the route are computed for the choice of the next customer. To determine the next node of the route, an assessment is made amongst the probable customers who ensure the time window constraint by computing the service start times with Eq. 19.

$$max(t_i, e_i) + s_i + d_{ij} \leq l_j \tag{19}$$

The choice of the customer that is appointed to the route as the next node is determined a random based selection method. According to this method, the customer who has small c_i has big chance for being appointed.

- **Step 4.** The customers keep going to be assigned to the route as explained in the Step 3, until the loaded quantity exceeds the vehicle capacity. When the capacity of the vehicle is full, the depot is added as the final node of the route. Subsequently, it is initiated a new route to be formed. Finished route solutions are attached to the route set.
- **Step 5.** Improvement process is executed to the single-customer routes when all customers are routed. Customers on single-customer routes are attempted to be attached to the other routes in order to decrease the number of routes.

The pseudocode of the Nearest Neighbor based Initial Algorithm II is given in Appendix A.

6.3. Initial Algorithm III (Sweep Based)

The third algorithm to produce the initial population utilized the sweep algorithm developed by Gillett and Miller in 1974. The steps of the Initial Algorithm III are as follows:

- **Step 1.** In the sweep algorithm, the polar angles of the customers are computed using the Eq. 20. The (x, y) coordinates of the customers are specified in the problem data. The node 0 that is the depot is accepted as the center of the coordinate system.

$$\theta(i) = arctan\left[\frac{(y(i) - y(0))}{(x(i) - x(0))}\right] \tag{20}$$

  The computed polar angles are aligned in increasing scheme.

- **Step 2.** As indicated in the Figure 6.2, the sweeping process is provided by twisting the ray which begins from the origin at an angle in random in the counterclockwise aspect. Customers shown with circular shapes who have crossed over the ray, are gathered the vehicle by sequent.
- **Step 3.** Assignments to a vehicle stop when the load quantity is equal to or more than the capacity of the vehicle. Then, the process continues with another vehicle until the clustering of all the customers is completed.

- **Step 4.** The routing step begins with the inclusion of the depot in the clusters. At first, the depot is added and then the customer which is nearest to the depot in one cluster is assigned to the route.
- **Step 5.** The distances between the non-routed customers and the last customer assigned to the route are computed. The closest one that provides the time constraint is assigned to the route. Thus, the routes are built for the customers at each cluster.
- **Step 6.** If any non-routed customer exists, those who do not violate the limitations are assigned to a route. In the lack of a feasible location, a new route is formed. The practicable solutions are achieved via these improvements.

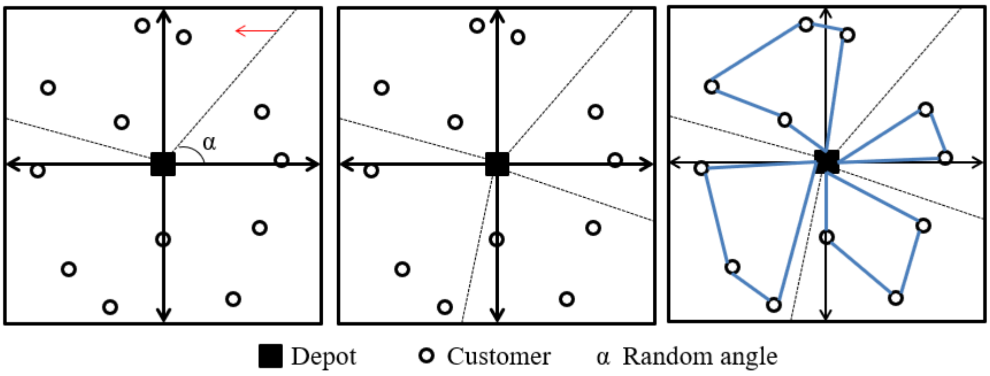

Figure 6.2. Proposed sweep algorithm.

The pseudocode of the Sweep based Initial Algorithm III is given in Appendix B.

6.4. Genetic Algorithm

The fundamental stages of the proposed GA are as follows:

- **Encoding.** Permutation encoding is utilized in the representation of the route solutions. The customers are represented by their numbers and sorted according to the order of visits on the route. In the VRPTW solutions, there are multiple routes. To represent all routes in a single array, the routes are separated by using a separator, -1. An example of a VRPTW solution with 15 customers and 4 vehicle routes is shown in Figure 6.3. For the calculation of the total distance the vehicles traveled, the vertex 0 is placed beginning and ending of the array and before and after the -1's. So, the distances between the depot and the customers would be summed up.

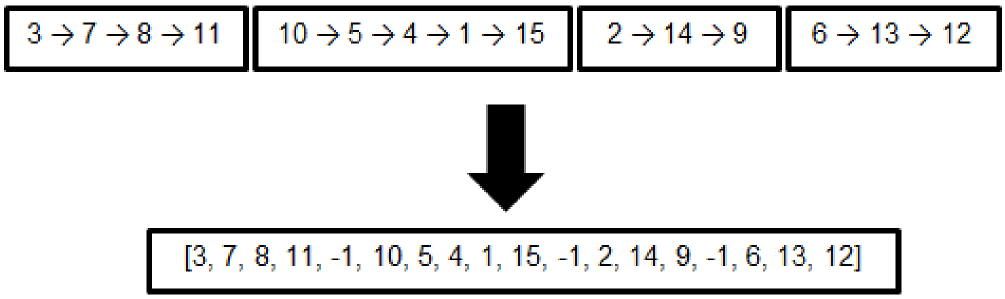

Figure 6.3. Encoding scheme of a VRPTW solution.

- **Initial population generation.** Initial populations are produced running the Initial Algorithms I, II and III explained in the prior sections.
- **Parent selection.** At the parent selection stage for producing the new generation, the dominance of the individuals is taken into account. Because MOO is considered, the order of dominance is determined by the optimal Pareto fronts. The individuals ranked with respect to their dominance and are paired in consecutive form as seen in Figure 6.4. There is no randomness at this stage.

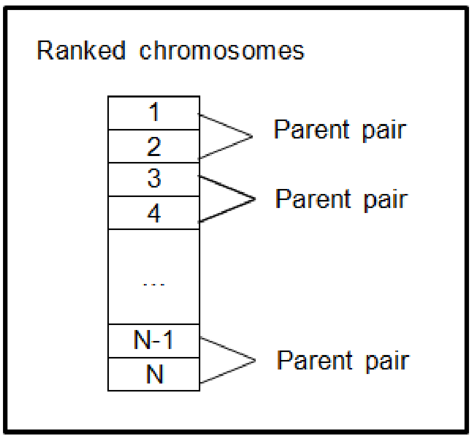

Figure 6.4. Parent selection scheme.

- **Crossover process.** The two-point crossover operator is utilized in the crossover operation. The cut points are determined as places after the separator -1 in the array. The contents between the cut points are exchanged between the parent pair. In the case of the repetition of a customer in the array, the customer is kept in the first place where it has been seen and deleted from where it is repeated. Also, in the case of the remaining non-routed customers, it is added to a place that makes the solution feasible when it is added. If such a place is absent, a new route is generated. Figure 6.5 illustrates a two point crossover example.

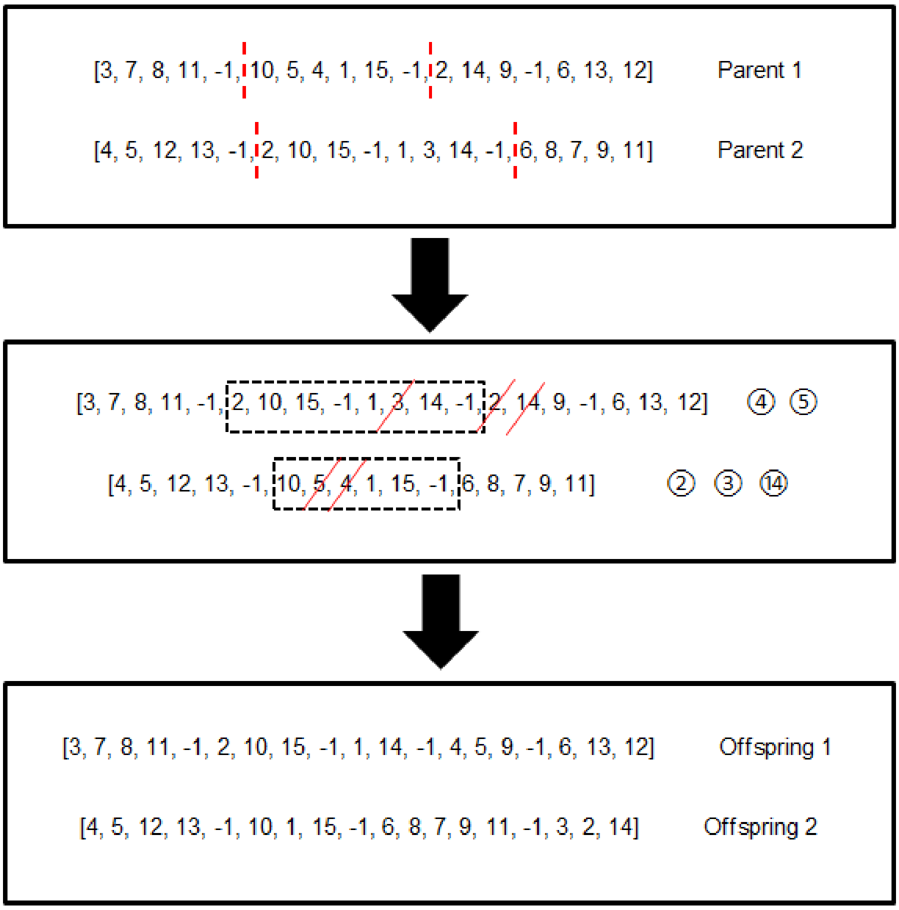

Figure 6.5. Crossover operator scheme.

- **Mutation process.** For the protection of the genetic diversity and preventing to be trapped in the local optimum, mutation operation is applied to discover neighbor solutions. As the mutation operator, insertion method is accomplished as follows. Firstly, a random route is picked from the produced offspring. Secondly, a random customer is chosen from the picked route and is extracted from that route. Lastly, it is entrenched to the best space to improve the objective function value.

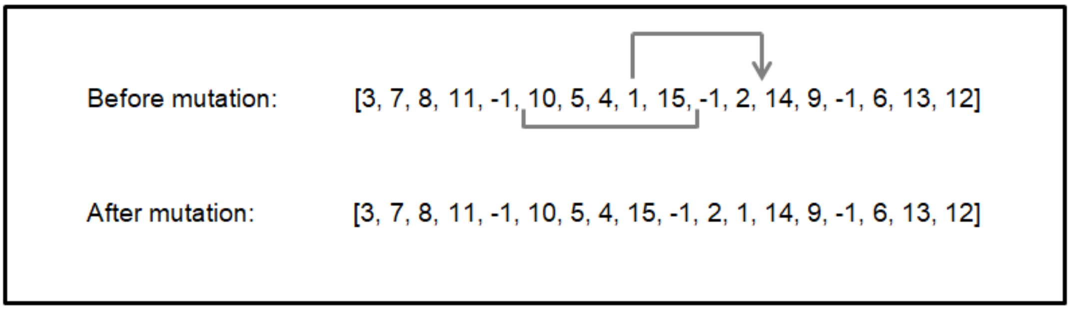

Figure 6.6. Mutation operator scheme.

- **Replacement.** In order to select the individual solutions of the subsequent population NSGA-II is utilized. The individuals are sorted based on their domination rank and crowding distance values. Elitist selection is applied for preserving the best solutions. Next generation is created by selecting N individuals from the combined populations of parent and offspring. DEAP (Distributed Evolutionary Algorithms in Python) Library is employed to perform NSGA-II (Deb, 2001; Fortin et al., 2012).
- **Termination.** The stopping status is based on the specified maximum number of generation. It is set to 100. GA repeats until the termination criterion is satisfied. The algorithm is run 100 times.

The values of parameters of GA are given in Table 6.1. The probability parameters are determined based on the values used in the literature.

The pseudocode of the proposed GA is given in Appendix C.

The flowchart of the proposed GA is shown in Figure 6.7.

Table 6.1. The values of the parameter of GA.

Parameters of GA	Values
Population Size	200
Number of Generation	100
Crossover Probability	0.7
Mutation Probability	0.1

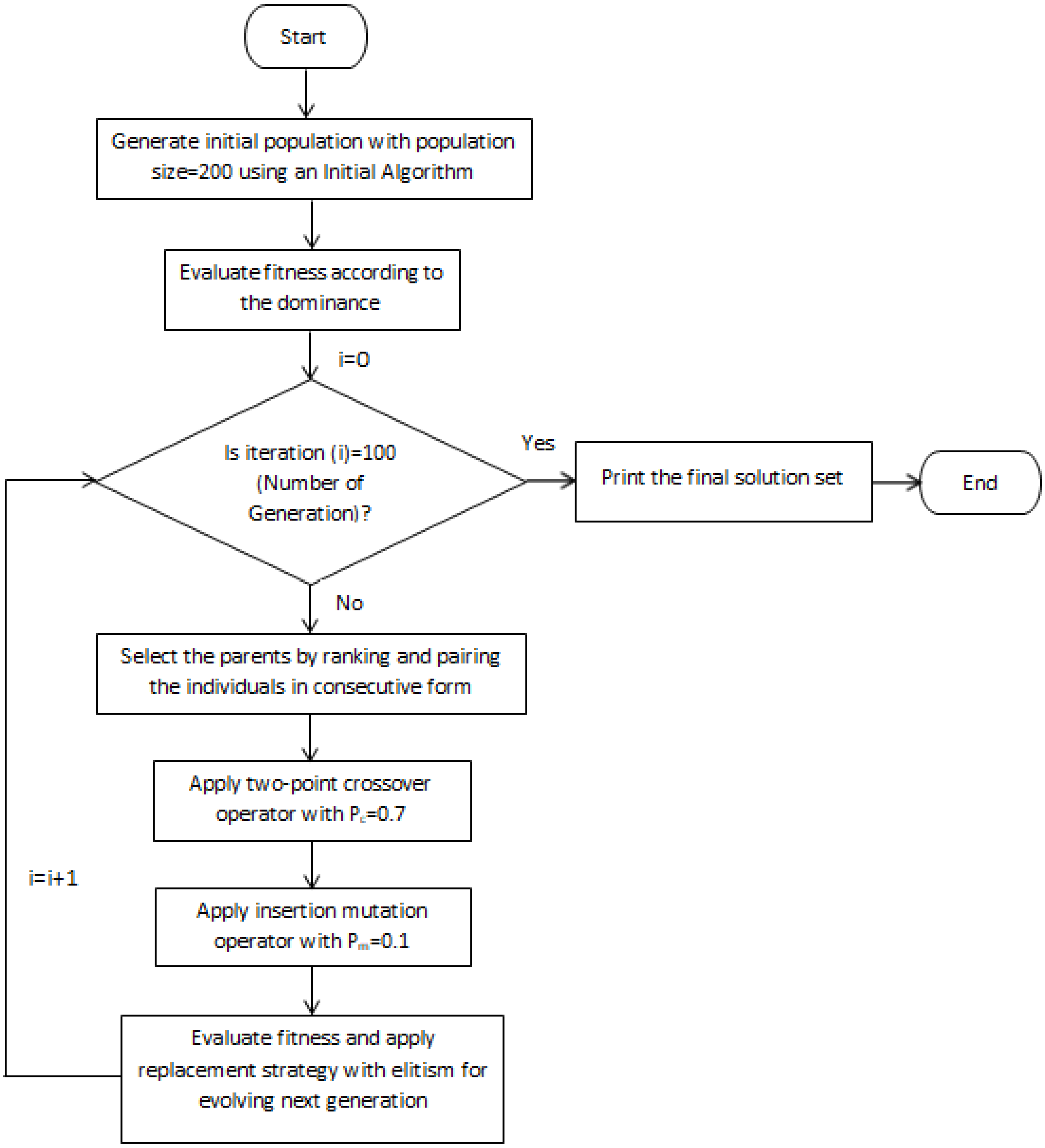

Figure 6.7. Flowchart of the proposed GA.

CHAPTER 7. RESULTS AND DISCUSSION

In the current study a solution approach for VRPTW via GA is considered. The influence of initial population of GA on multi objective problems is investigated. The algorithms tested are illustrated in Table 7.1. The algorithms are coded in the Python language and are run on a computer with i7 processor technology, 2.6 GHz Turbo processor speed and 8 GB RAM capacity.

Table 7.1. The generated algorithms.

Algorithms	Initial Population Generation Method	Proposed Meta Heuristic
Alg.1	Initial Algorithm I (Random)	GA
Alg.2	Initial Algorithm II (Nearest Neighbor Based)	GA
Alg.3	Initial Algorithm III (Sweep Based)	GA

7.1. Problem Data Set

The generated algorithms are tested on the VRPTW benchmark problem instances proposed by Solomon (1987). Solomon's benchmark problems are widely used for comparing algorithms that proposed for vehicle routing.

There are 56 test problems which composed of the information like the geographical data, customer demand and time window characteristics. Each test problem has one warehouse and 100 customers. Geographic positions of the customers and the warehouses are given by (x, y) coordinates. The route length between them is computed by Euclidean distance and the value get is in the unit of distance. The assumption of that the travelling of 1 unit distance takes 1 unit time is made. Customer demand quantities and service durations are available in the data set. The earliest and latest arrival times (ready time and due date), i.e. time windows, of each customer are indicated. The number of customers with time constraints varies 25%, 50%, 75% and 100% of the customers from a data set to another. The vehicles are comprised of a homogeneous fleet and the vehicle capacities for each data set are given.

The data sets are clustered into 6 classes; C1, C2, R1, R2, RC1 and RC2; according to the geographical data feature. In C category classes, the customers are clustered. In R category classes, the customers are distributed randomly and uniform. In RC category classes, the customers are semi clustered, they mixed R and C classes features. Furthermore, the problems in C1, R1 and RC1 classes have a short scheduling horizon; i.e. the time window of the depot is narrow; and low vehicle capacity. So, only a few customers are allowed to be in a route. On the contrary, the problems in C2, R2 and

RC2 classes have a long scheduling horizon; i.e. the time window of the depot is wide; and high vehicle capacity. So, many customers are allowed to be in a route.

7.2. Computational Results and Analyses

The results of the algorithms are presented in Appendices D, E, and F in detail. The best known solutions in the tables are taken from the website of Solomon. It must be aware of that; the best known data have obtained from single objective studies. The other best results according to concerned objective, i.e. minimization of the total distance or minimization of the waiting time of the vehicles, have obtained from the final solution sets of the tested algorithms separately. For each best distance and best waiting time result of the 56 instances; total distance, waiting time of the vehicles and vehicle number data have computed. In MOO, an improvement of one of the objectives may cause to deterioration of the other. Because of the difference of the solution techniques according to the objective numbers, the comparison between the best results and the proposed algorithms' results may not give an accurate conclusion. So, the effectiveness of the algorithm has analyzed with making comparison only between the tested algorithms.

Exclusively, Alg.1 has not reached any solution for R101 and RC101 problem sets. Therefore, class averages are used instead of class totals for the analyses.

Table 7.2 and Table 7.3 present the descriptive statistics of the best results according to the relevant objective.

Table 7.2 shows the class means of the criteria, i.e. travelled distance, waiting time and vehicle number, of the results according to the best travelled distance values. It is obviously seen that Alg.3 has attained better results. For each class, means of travelled distance values are less than the others. In total, it has been able to achieve the solutions with less vehicles. In C classes, the vehicles have been waiting less time at solution of Alg.3. R and RC classes' mean waiting time results are minimum at Alg.1 solutions.

Table 7.3 indicates the class means of the criteria of the results according to the best waiting time values. For each class, means of waiting time values in the Alg.1 results are less than the others. Besides, it has obtained better results on travelled distance criterion in R1, R2 and RC2 classes. It should be noted that the value of R1 class may has affected of missing result. Means of travelled distance values at C1, C2 and RC2 classes are minimum at Alg.3 solutions. Furthermore, Alg.3 has required fewer vehicles than other algorithms in this case too.

Table 7.2. The class means of travelled distance (TD), waiting time (WT) and vehicle number (VN) values of obtained best results via the algorithms according to travelled distance objective (N*=missing value).

Class	Alg.1					Alg.2					Alg.3				
	N	N*	TD	WT	VN	N	N*	TD	WT	VN	N	N*	TD	WT	VN
C1	9	0	1215,7	199,3	11,3	9	0	1118,5	307,6	10,8	9	0	950,3	99,2	10,8
C2	8	0	851,9	1118	4	8	0	843,1	612	4	8	0	659,2	121,7	3,3
R1	11	1	1450,8	104,5	14	12	0	1483,7	296,7	14,7	12	0	1421,7	275,9	14,3
R2	11	0	1177,2	157,9	3,7	11	0	1240,6	579	3,6	11	0	1143,3	592	3,5
RC1	7	1	1614,8	123	13,8	8	0	1626,8	256,7	14,2	8	0	1546,3	245,7	14,1
RC2	8	0	1441,7	352,6	4,6	8	0	1500,6	838	4,2	8	0	1327,6	783	4,2

Table 7.3. The class means of travelled distance (TD), waiting time (WT) and vehicle number (VN) values of obtained best results via the algorithms according to waiting time objective (N*=missing value).

Class	Alg.1					Alg.2					Alg.3				
	N	N*	TD	WT	VN	N	N*	TD	WT	VN	N	N*	TD	WT	VN
C1	9	0	1311,5	20,3	11,3	9	0	1306,2	32,5	11,1	9	0	1025,3	53,7	11
C2	8	0	900,3	6	3,9	8	0	1105,1	20,8	4	8	0	733,6	9,14	3,2
R1	11	1	1490	31,5	13,9	12	0	1869	90,3	15,5	12	0	1682,4	117,6	15,1
R2	11	0	1314,2	23,1	3,5	11	0	1613,9	57,6	3,4	11	0	1395,8	27,9	3,3
RC1	7	1	1841	31,9	15	8	0	2024	49,4	15	8	0	1785,5	70,1	14,1
RC2	8	0	1690	14	4,2	8	0	2102	58,9	4	8	0	1774	46,4	3,7

The individual value plot with the means of the algorithms' best travelled distance results according to the travelled distance objective is indicated in Figure 7.1. Alg.3 with green symbol has appeared to reach better results than Alg.1 and Alg.2.

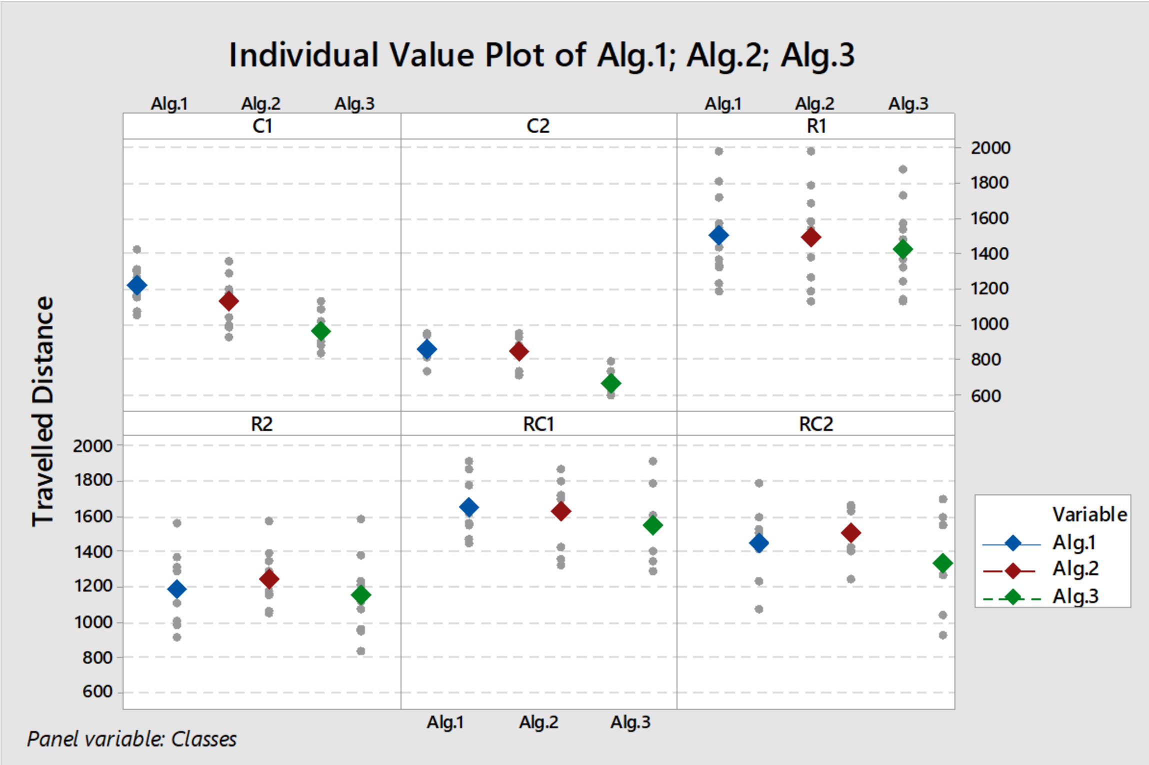

Figure 7.1. Individual value plot of the results of the algorithms according to travelled distance objective.

Figure 7.2 and Figure 7.3 display the individual value plots of the means of the results of the algorithms according to travelled distance and waiting time objectives separately. The values are taken from the Table 7.2 and Table 7.3 and then visualized.

Figure 7.4 and Figure 7.5 show the bar charts of the mean travelled distance and waiting time values of the classes according to travelled distance and waiting time objectives separately. It is seen that decrement in waiting time values cause increment in travelled distance values. This fact can be observed in all data sets and at all algorithms.

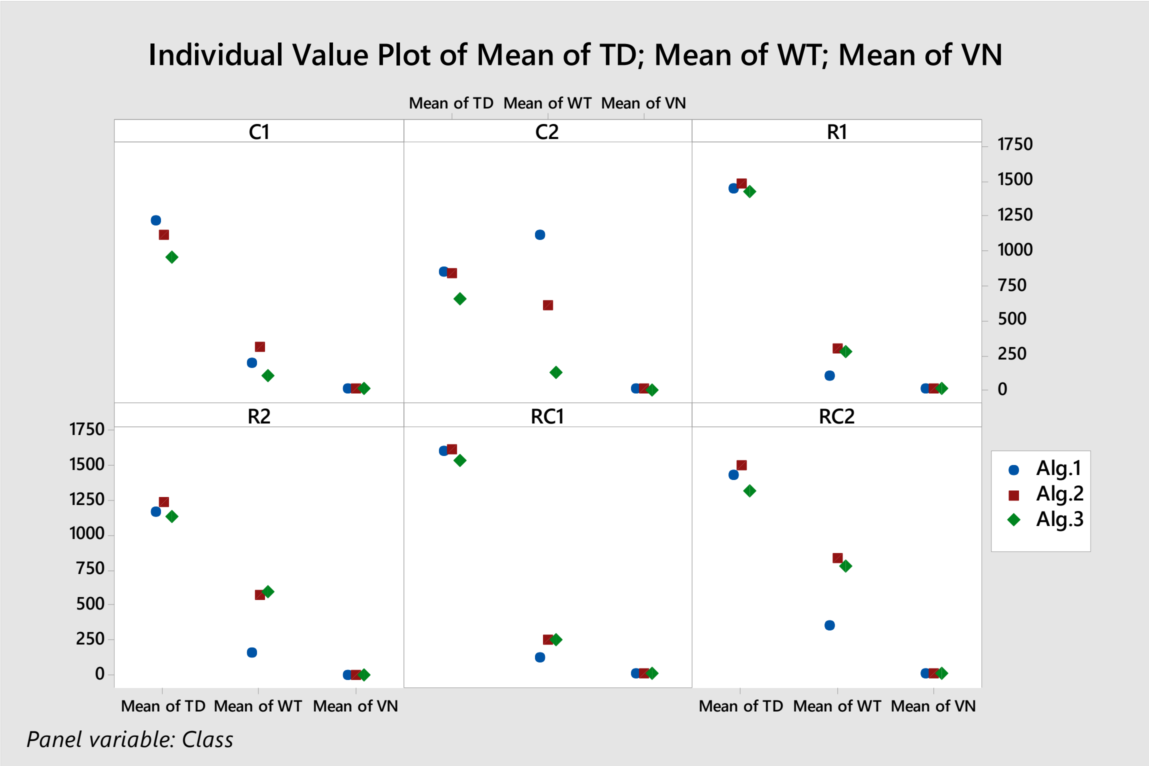

Figure 7.2. Individual value plot of the means of the results of the algorithms according to travelled distance objective.

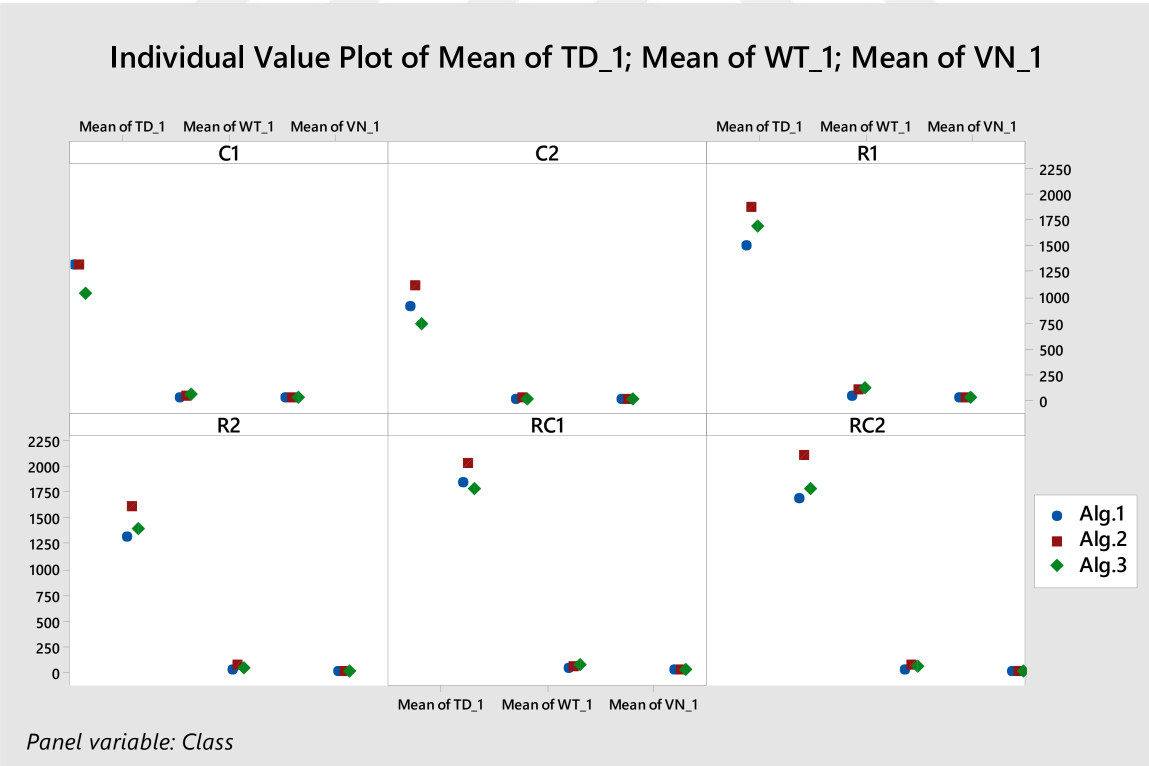

Figure 7.3. Individual value plot of the means of the results of the algorithms according to waiting time objective.

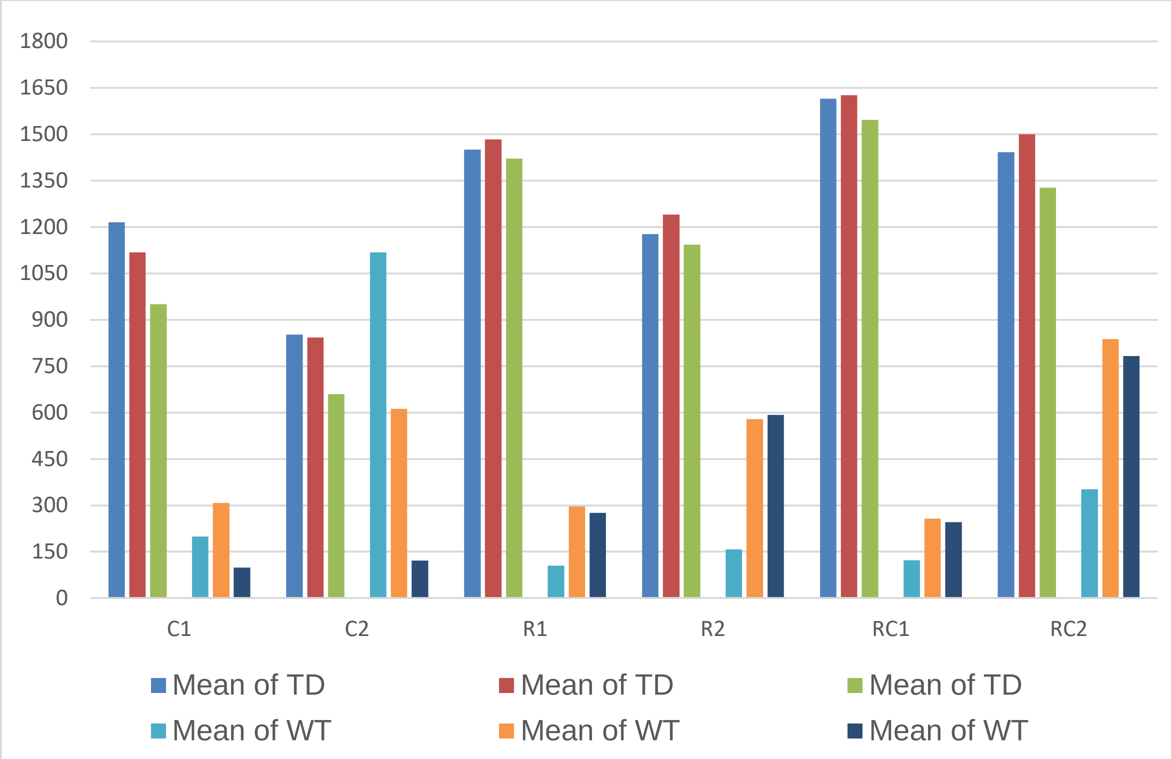

Figure 7.4. Mean travelled distance and waiting time values of the classes according to travelled distance objective.

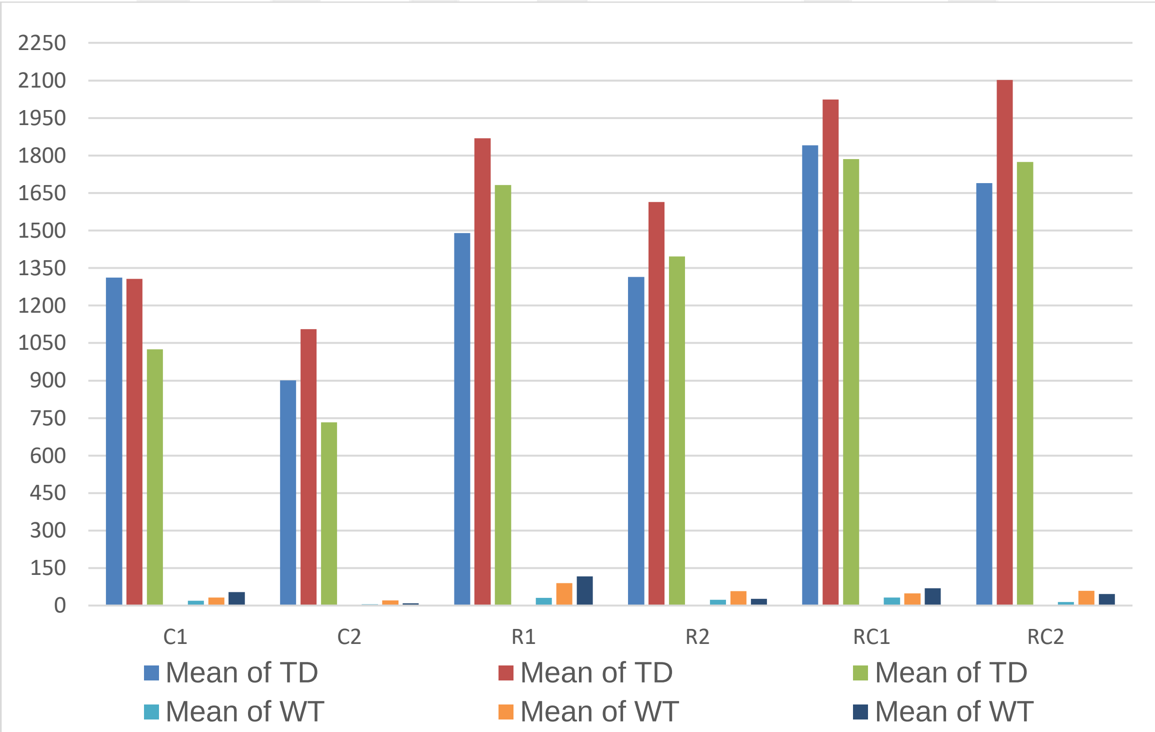

Figure 7.5. Mean travelled distance and waiting time values of the classes according to waiting time objective.

Table 7.4 gives the information of the number of best instances obtained from the comparison of three algorithms according to the related objectives. With respect to the travelled distance objective, Alg.3 with 43 out of 56 instances is much better than the others. It can be concluded that using sweep algorithm at the initial population generation step of the GA for travelled distance minimization objective is more effective than using nearest neighbor algorithm or random generation algorithm. According to the minimization objective of the waiting time, GAs with a construction algorithm have not reached the number of routes of a random GA with 0 wait duration. Nevertheless, Alg.3 has showed better performance than Alg.2.

Table 7.4. Number of best instances obtained from the comparison of the algorithms.

According to	Travelled Distance Obj.	Waiting Time Obj.
Alg.1	7	43 (29 of them are 0.)
Alg.2	6	18 (12 of them are 0.)
Alg.3	43	18 (15 of them are 0.)

Table 7.5 demonstrates the number of instances of the algorithms that reached to the best known solutions. Travelled distance and vehicle number criteria have used. The waiting time values are not available on the Solomon's web site. Alg.3 has reached the best known results at C101 and C201 problem sets with the travelled distance values. Alg.1 and Alg.2 have not attained any best known travelled distance values. Alg.3 has better performance on constructing the routes with fewer vehicles. It has achieved more number of best known vehicle number values than Alg.1 and Alg.2.

Table 7.5. Number of instances that reached to the best known.

According to	Travelled Distance Obj.		Waiting Time Obj.	
Criteria	Travelled Distance	Vehicle Number	Travelled Distance	Vehicle Number
Alg.1	-	-	-	3
Alg.2	-	4	-	7
Alg.3	2	10	2	16

The results of the study have shown that Alg.3 has solved the problem more effectively with respect to the total distance travelled. Also, it has not worse than the others according to the waiting time values. Besides, it has required fewer vehicles for routing. This has indicated that the initial population generation method for GAs affects the performance of the algorithm.

CHAPTER 8. CONCLUSIONS

The effect of the logistic management as an important part of the supply chain management is significant for the competition between the companies, especially in the customer satisfactory subject. Increasing the customer reachability, decreasing the travelling time and distance, and decreasing the transportation cost in this way are related topics for this issue. To reach these aims, the routes that the paths of the vehicles for transportation are established by applying VRP. In VRP, a depot, a set of customers and a homogeneous fleet of vehicles exist. Through the determined routes, which start from and end at the depot, demands of the customers are satisfied in one time and with one vehicle while paying attention to the vehicle capacity constraints.

In this study, VRPTW which is a variant of VRP is considered. An additional constraint is handled here. It is the time window constraint which satisfies that the requirement of the customers should be served in a specified time interval. The vehicles must wait for the customer in the case of arrival before the time window. Minimization of the total distance and waiting time of the vehicles are decided as the objectives. GA that is one of the meta-heuristic methods is used and a multi objective hybrid GA approach for the VRPTW solution is proposed. NSGA-II is used in the evaluation, ranking and selection of the individuals at GA steps for the MOO.

In this thesis, the effect of using different initial populations for GA is investigated. The initial populations are generated first randomly, second by a nearest neighbor based algorithm, and third by a sweep based algorithm. Thus, GA becomes hybridized. The intention of the study is to consolidate the advantage of GA with the strength of the constructive heuristics; sweep algorithm or nearest neighbor algorithm. The outputs of the constructive heuristics are given as inputs to the GA to achieve more efficient results. The formed three algorithms are tested on Solomon's VRPTW benchmark problems.

It can be concluded that according to travelled distance minimization objective, using sweep algorithm at the initial population generation step of the GA is more effective than using nearest neighbor algorithm or random generation algorithm. Besides, it has required fewer vehicles for routing. However, according to the minimization objective of the waiting time, GAs with a construction algorithm have not a significant difference between the results. Nevertheless, the analysis results have indicated that the initial population generation method for GAs affects the performance of the algorithm.

In the future works, different methods can be tried to generate initial population and be presented as alternative results. The crossover and mutation operators utilized in this thesis can be developed, and parameter analysis can be made for examining the influence on the model performance. Alternatively, different objective pairs can be considered, or different problems can be solved.

REFERENCES

Aggarwal S, Garg R, Goswami P. 2014. A review paper on different encoding schemes used in genetic algorithms., Int J Adv Res Comput Sci Softw Eng, 4(1): 596–600.

Ahuja RK, Magnanti TL, Orlin JB. 1993. Network flows: theory, algorithms, and applications. Prentice Hall, Upper Saddle River, NJ, pp. 846.

Baker BM, Ayechew MA. 2003. A genetic algorithm for the vehicle routing problem. Comput Oper Res, 30(5): 787–800.

Basu R, Wright JN. 2010. Total supply chain management. Elsevier, Oxford, pp. 392.

Berger J, Barkaoui M. 2004. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Comput Oper Res, 31(12): 2037–2053.

Blum C, Roli A. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput Surv, 35(3): 268–308.

Borsodi, R. 1927. The distribution age. D. Appleton and Company.

Bowersox DJ, Closs DJ, Cooper MB. 2002. Supply chain logistics management. Vol. 2. McGraw-Hill: New York, pp. 656.

Braysy O, Dullaert W, Gendreau M. 2004. Evolutionary algorithms for the vehicle routing problem with time windows. J Heuristics, 10(6): 587–611.

Chang Y, Chen L. 2007. Solve the vehicle routing problem with time windows via a genetic algorithm. Discrete Contin Dyn Syst, 240–249.

Christopher M. 2011. Logistics, the Supply Chain and Competitive Strategy, fourth ed. Financial Times Series, Dorset, pp. 276.

Clarke G, Wright JW. 1964. A scheduling of vehicles from a central depot to a number of delivery points. Oper Res, 12(4): 568–581.

Correia VMF, Madeira JFA, Araújo AL, Soares CMM. 2017. Multiobjective design optimization of laminated composite plates with piezoelectric layers. Compos Struct, 169: 10–20.

CSCMP, Council of supply chain management professionals. August 2013. Supply chain management terms and glossary. https://cscmp.org/imis0/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms.aspx Access date: 02.06.2018.

Çolak S, Güler H. 2009. Dağıtım rotaları optimizasyonu için meta sezgisel bir yaklaşım. Gazi Univ J Fac Econ Adm Sci, 11(2): 171–190

Dantzig GB, Ramser JH. 1959. The truck dispatching problem. Manag Sci, 6(1): 80–91.

Deb K. 2001. Multi-objective optimization using evolutionary algorithms. Vol. 16. John Wiley & Sons, pp. 497.

Deb K, Pratap A, Agarwal S, Meyarivan T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE T Evolut Comput, 6(2): 182–197.

Eksioglu B, Vural AV, Reisman A. 2009. The vehicle routing problem: A taxonomic review. Comput Ind Eng, 57(4): 1472–1483.

El-Sherbeny NA. 2010. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. J King Saud Univ Sci, 22(3): 123–131.

Fortin FA, Rainville FMD, Gardner MA, Parizeau M, Gagné C. 2012. DEAP: Evolutionary algorithms made easy. J Mach Learn Res, 13: 2171–2175.

Garcia-Najera A, Bullinaria JA. 2011. An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. Computers & Operations Research Comput Oper Res, 38(1): 287–300.

Ghiani G, Laporte G, Musmanno R. 2004. Introduction to logistics systems planning and control. John Wiley & Sons, Chichester, England, pp.352.

Ghoseiri K, Ghannadpour SF. 2010. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. Appl Soft Comput, 10(4): 1096–1107.

Gillett BE, Miller, LR. 1974. A heuristic algorithm for the vehicle-dispatch problem. Oper Res, 22(2): 340–349.

Glover FW, Kochenberger GA. (eds.) 2003. Handbook of metaheuristics. Boston: Kluwer Academic Publishers, xi–xii.

Göçken T, Yaktubay M, Kılıç F. 2017. Improvement of a Genetic Algorithm Approach for the Solution of Vehicle Routing Problem with Time Windows. International Symposium on Artificial Intelligence and Data Processing (IDAP2017), 16-17 September, Malatya, Turkey.

Griffis SE, Bell JE, Closs DJ. 2012. Metaheuristics in logistics and supply chain management. J Bus Logis, 33(2): 90–106.

Haddadene SRA, Labadie N, Prodhon C. 2016. NSGAII enhanced with a local search for the vehicle routing problem with time windows and synchronization constraints. IFAC-PapersOnLine, 49(12): 1198–1203.

Ho WK, Ang JC, Lim A. 2001. A hybrid search algorithm for the vehicle routing problem with time windows. Int. J. Artif. Intell. Tools, 10(3): 431–449.

Holland JH. 1975. Adaptation in natural and artificial systems. Ann Arbor MI: The University of Michigan Press.

Ibrahim MF, Masudin I, Saputro TE. 2016. A hybrid genetic algorithm implementation for vehicle routing problem with time windows. Jurnal Ilmiah Teknik Industri, 14(2): 196–204.

Karagul K, Gungor I. 2014. A case study of heterogeneous fleet vehicle routing problem: Touristic distribution application in Alanya. Int J Opt Control: Theor Appl, 4(2): 67–76.

Labadie N, Prins C, Prodhon C. 2016. Metaheuristics for vehicle routing problems. John Wiley & Sons, USA, pp. 225.

Laporte G. 2007. What you should know about the vehicle routing problem. Nav Res Log, 54(8): 811–819.

Laporte G. 2009. Fifty years of vehicle routing. Transport Sci, 43(4): 408–416.

Laporte G, Ropke S, Vidal T. 2014. Heuristics for the vehicle routing problem, in: Toth P, Vigo D. (Eds.), Vehicle Routing: Problems, Methods, and Applications, second ed. Society for Industrial and Applied Mathematics, pp. 87–116.

Lin S. 1965. Computer solutions of the traveling salesman problem. Bell Syst Tech J, 44(10): 2245–2269.

Meidutė-Kavaliauskienė I, Aranskis A, Litvinenko M. 2014. Consumer satisfaction with the quality of logistics services. Procedia Soc Behav Sci, 110: 330–340.

Mertler CA, Reinhart RV. 2016. Advanced and multivariate statistical methods: Practical application and interpretation, sixth ed. Taylor and Francis, pp. 374.

Mester D, Bräysy O. 2007. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. Comput Oper Res, 34(10): 2964–2975.

Mungwattana A, Manisri T, Charoenpol K, Janssens GK. (2016). A solution for the bi-objective vehicle routing problem with the windows using local search and genetic algorithms. Int J Traffic Trans Eng, 6(2): 149–158.

Na B, Jun Y, Kim B-I. 2011.Some extensions to the sweep algorithm. Int J Adv Manuf Technol, 56(9-12): 1057–1067.

Ólafsson S. 2006. Metaheuristics, in: Henderson SG, Nelson BL. (Eds.), Handbooks in Operations Research and Management Science. North-Holland, 13, pp. 633-654.

Ombuki B, Ross BJ, Hanshar F. 2006. Multi-objective genetic algorithms for vehicle routing problem with time windows. Appl Intell, 24(1): 17–30.

Prins C. 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Comput Oper Res, 31(12): 1985–2002.

Sandhya VK. 2013. Issues in solving vehicle routing problem with time window and its variants using meta heuristics-A survey. Int J Eng Technol, 3(6): 668–672.

Sariklis D, Powell S. 2000. A heuristic method for the open vehicle routing problem. J Oper Res Soc, 51(5): 564–573.

Seshadri A. 2006. A fast elitist multiobjective genetic algorithm: NSGA-II. MATLAB Central.

Simchi-Levi D, Kaminsky P, Simchi-Levi E. 2003. Designing and managing the supply chain, second ed. McGraw-Hill, London, pp. 321.

Sivanandam SN, Deepa SN. 2008. Introduction to genetic algorithms. Springer Science & Business Media, New York, pp. 442.

Solomon MM. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res, 35(2): 254–265.

Solomon MM. Best known solutions identified by heuristics. http://web.cba.neu.edu/~msolomon/heuristi.htm Access date: 04.06.2018.

Şahin Y, Eroğlu A. 2014. Kapasite kısıtlı araç rotalama problemi için metasezgisel yöntemler: Bilimsel yazın taraması. Suleyman Demirel Univ J Fac Econ Adm Sci, 19(4): 337–355.

Tan KC, Chew YH, Lee LH. 2006. A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. Comput Optim Appl, 34(1): 115–151.

Talbi EG. 2002. A taxonomy of hybrid metaheuristics. J Heuristics, 8(5): 541–564.

Thangiah SR, Nygard KE, Juell PL. 1991. Gideon: A genetic algorithm system for vehicle routing with time windows. 7th IEEE Conference on Artificial Intelligence Applications, 24-28 February, Miami Beach-Florida, pp. 322–328.

Toth P, Vigo D. 2002. The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications: Vol. 9. Society for Industrial and Applied Mathematics, Philadelphia, pp. 367.

Van Breedam A. 2002. A parametric analysis of heuristics for the vehicle routing problem with side-constraints. Eur J Oper Res, 137(2): 348–370.

APPENDICES

APPENDIX A: Nearest neighbor based algorithm pseudocode (Göçken et al., 2017).

Steps of Initial Algorithm II

Input: **CustomerInfo, VehicleCapacity, NumberOfVehicle.**

Output: **RouteSet.**

1: RouteSet ← ∅

2: route ← Add the depot to the route ***// Route solutions begin with the depot.***

3: customerList ← Form the customer list

4: properCustomerList ← CalculateDistance (customerList, Depot, VehicleCapacity)

5: While customerList ≠ ∅ do

6: selectedNode ← PickCustomer (properCustomerList) ***// Probability based selection is applied.***

7: route ← Add the selectedNode to the route

8: customerList.remove(selectedNode)

9: properCustomerList ← CalculateDistance (customerList, selectedNode, VehicleCapacity)

10: If properCustomerList = ∅

11: route ← Add the depot to the route ***// Route solutions finish with the depot.***

12: RouteSet ← Add the route to the solution set

13: route ← Form an empty route and add the depot to the route

14: properCustomerList ← CalculateDistance (customerList, Depot)

15: End if

16: End while

17: RouteSet ← Improve (RouteSet, VehicleCapacity, NumberOfVehicle)

APPENDIX B: Sweep based algorithm pseudocode (Göçken et al., 2017).

Steps of Initial Algorithm III

Input: **CustomerInfo, VehicleCapacity, NumberOfVehicle.**

Output: **RouteSet.**

1: RouteSet ← ∅

2: UnusedCustomer ← ∅

3: customerList ← Calculate and sort polar angles of customers according to the depot

4: randomAngle ← PickRandomNumber (0, 359)

5: rawRouteSet ← ClusterCustomers (customerList, randomAngle, VehicleCapacity)

6: For each customer cluster customerGroup ← rawRouteSet // *An element from the set is drawn.*

7: route ← Add the depot to the route ***// Route solutions begin with the depot.***

8: selectedNode ← Depot

9: properCustomerList ← CalculateDistance (customerGroup, selectedNode)

10: While customerGroup ≠ ∅ do

11: selectedNode ←PickCustomer (properCustomerList, selectedNode) ***//The closest customer is selected.***

12: route ← Add the selectedNode to the route

13: customerGroup.remove(selectedNode)

14: properCustomerList ← CalculateDistance (customerGroup, selectedNode)

15: If properCustomerList = ∅ and customerGroup ≠ ∅

16: UnusedCustomer ← customerGroup

17: customerGroup ← ∅

18: End if

19: End while

20: route ← Add the depot to the route ***// Route solutions finish with the depot.***

21: RouteSet ← Add the route to the solution set

22: End

23: RouteSet ← Improve (RouteSet, UnusedCustomer, VehicleCapacity, NumberOfVehicle)

APPENDIX C: GA pseudocode (Göçken et al., 2017).

Steps of Genetic Algorithm
Input: **NumberOfGeneration, PopulationSize, InitialAlgorithm, $P_{mutation}$, $P_{crossover}$, CustomerInfo, VehicleCapacity, NumberOfVehicle.**
Output: **Population.**
Population ← InitialProcedure (PopulationSize, InitialAlgorithm, DataFile)
Iteration ← 0
While Iteration ≤ NumberOfGeneration do
Offspring ← SelectParents (Population)
CrossoverOperation (Offspring, $P_{crossover}$)
MutationOperation (Offspring, $P_{mutation}$)
Population ← NSGA-II (Population + Offspring, PopulationSize)
Iteration ← Iteration + 1
End while

APPENDIX D: The results of the Alg. 1 (Random based GA).

Data Set	BEST KNOWN Total Distance	BEST KNOWN Vehicle Number	BEST DISTANCE Total Distance	BEST DISTANCE Waiting Time	BEST DISTANCE Vehicle Number	BEST WAITING TIME Total Distance	BEST WAITING TIME Waiting Time	BEST WAITING TIME Vehicle Number
C101	828,94	10	1150,800033	179	11	1177,58631	53	11
C102	828,94	10	1265,005036	221	12	1389,88721	40	12
C103	828,06	10	1417,786651	173	12	1468,33707	8	12
C104	824,78	10	1291,834061	327	11	1438,73447	8	11
C105	828,94	10	1161,287794	463	12	1217,6815	29	11
C106	828,94	10	1060,953459	42	11	1252,82987	29	11
C107	828,94	10	1246,764182	168	11	1432,26758	8	12
C108	828,94	10	1041,003487	89	11	1073,28671	8	11
C109	828,94	10	1305,640372	132	11	1352,52559	0	11
C201	591,56	3	726,41303	1533	4	736,161655	0	4
C202	591,56	3	820,3966756	2244	4	849,275138	0	**3**
C203	591,17	3	941,2466185	0	4	941,246619	0	4
C204	590,6	3	827,6887209	109	4	840,987031	0	4
C205	588,88	3	826,8807377	2301	4	1080,72456	48	4
C206	588,49	3	939,4245898	896	4	969,949286	0	4
C207	588,29	3	932,1534433	1773	4	979,788943	0	4
C208	588,32	3	801,1090447	88	4	804,010839	0	4
R101	1645,79	19	-	-	-	-	-	-
R102	1486,12	17	1804,348499	295	18	1846,33634	143	18
R103	1292,68	13	1566,703202	100	15	1577,6495	41	15
R104	1007,24	9	1176,615145	28	12	1186,88824	0	12
R105	1377,11	14	1715,313565	228	17	1782,11915	131	17
R106	1251,98	12	1484,671257	53	14	1507,08186	8	14
R107	1104,66	10	1328,880885	35	13	1342,13339	0	13
R108	960,88	9	1223,538417	48	11	1234,53014	0	11
R109	1194,73	11	1547,035276	102	15	1603,38432	13	15
R110	1118,59	10	1430,569368	150	14	1491,34213	7	13
R111	1096,72	10	1364,113334	75	13	1434,79216	4	13
R112	982,14	9	1316,904919	35	12	1383,30665	0	12
R201	1252,37	4	1556,644655	281	5	1731,46222	225	5
R202	1191,7	3	1357,478931	270	4	1621,18486	0	4
R203	939,54	3	1173,580015	146	4	1255,79973	0	4
R204	825,52	2	974,2518633	34	3	975,507693	0	3
R205	994,42	3	1310,60979	254	4	1767,25254	1	4
R206	906,14	3	1182,800975	157	4	1289,52802	0	**3**
R207	893,33	2	1002,062181	26	3	1026,10198	0	3
R208	726,75	2	906,4828112	26	3	952,654522	0	3
R209	909,16	3	1286,582926	322	4	1366,0987	28	4
R210	939,34	3	1097,825922	118	4	1319,51866	0	**3**
R211	892,71	2	1100,999778	103	3	1150,74365	0	3
RC101	1696,94	14	-	-	-	-	-	-
RC102	1554,75	12	1776,420363	152	15	2141,42119	46	17
RC103	1261,67	11	1549,462879	62	13	1621,47092	0	13
RC104	1135,48	10	1552,714674	29	12	1698,42849	0	13
RC105	1629,44	13	1905,802637	350	17	2393,67687	136	20
RC106	1424,73	11	1613,587287	106	15	1762,66343	24	15

The results of the Alg. 1 (Random based GA) (Continued).

Data Set	BEST KNOWN		BEST DISTANCE			BEST WAITING TIME		
	Total Distance	**Vehicle Number**	**Total Distance**	**Waiting Time**	**Vehicle Number**	**Total Distance**	**Waiting Time**	**Vehicle Number**
RC107	1230,48	11	1446,054243	74	13	1519,28406	17	13
RC108	1139,82	10	1459,299004	88	12	1750,87094	0	14
RC201	1406,91	4	1778,562905	588	6	2114,35487	53	5
RC202	1367,09	3	1526,664764	595	5	1902,67038	0	4
RC203	1049,62	3	1223,350238	256	4	1331,43731	0	4
RC204	798,41	3	1063,312864	0	4	1063,31286	0	4
RC205	1297,19	4	1586,23701	562	6	1972,45451	59	5
RC206	1146,32	3	1500,288182	263	4	1759,31124	0	4
RC207	1061,14	3	1444,299512	316	4	1783,90751	0	4
RC208	828,14	3	1410,783827	241	4	1589,92137	0	4

* Bold and gray cells show the results that are better than or equal to the best known results.

APPENDIX E: The results of the Alg. 2 (Nearest Neighbor Based GA).

Data Set	BEST KNOWN Total Distance	BEST KNOWN Vehicle Number	BEST DISTANCE Total Distance	BEST DISTANCE Waiting Time	BEST DISTANCE Vehicle Number	BEST WAITING TIME Total Distance	BEST WAITING TIME Waiting Time	BEST WAITING TIME Vehicle Number
C101	828,94	10	913,2806806	23,0576	**10**	913,280681	23,0576	**10**
C102	828,94	10	1167,836212	352,374	12	1428,99939	108,814	12
C103	828,06	10	1280,094012	209,197	11	1504,55665	104,78	11
C104	824,78	10	1348,078388	726,68	11	1706,54347	0	12
C105	828,94	10	981,2677285	29,6044	10	981,267729	29,6044	10
C106	828,94	10	976,9145512	813,834	11	1100,31881	10,0499	11
C107	828,94	10	1035,912032	161,458	11	1178,66262	0	12
C108	828,94	10	1195,62455	253,475	11	1454,99103	14,4016	11
C109	828,94	10	1167,346227	198,817	11	1486,92117	1,71989	11
C201	591,56	3	730,5732674	27,9471	4	764,79791	21,2348	4
C202	591,56	3	817,5863697	241,709	4	1015,4759	0	4
C203	591,17	3	923,0200231	878,199	4	1167,11411	11,1716	4
C204	590,6	3	942,8521962	552,925	4	1526,19078	31,0761	4
C205	588,88	3	867,2810496	1296,3	4	958,104851	24,8589	4
C206	588,49	3	846,9399938	0	4	846,939994	0	4
C207	588,29	3	700,1189966	266,895	4	1065,20118	0	4
C208	588,32	3	916,3522763	1630,2	4	1496,75595	78,064	4
R101	1645,79	19	1971,390445	1022,62	22	2413,85627	553,928	21
R102	1486,12	17	1778,497239	604,298	19	2126,91663	274,82	19
R103	1292,68	13	1534,127855	185,812	15	1761,98665	39,2684	15
R104	1007,24	9	1121,110635	137,465	11	1530,27257	2,07731	12
R105	1377,11	14	1676,416452	465,926	17	2126,80753	134,47	17
R106	1251,98	12	1582,344352	178,553	15	1761,85792	26,0063	15
R107	1104,66	10	1369,651088	211,772	14	2020,07911	2,95378	16
R108	960,88	9	1179,165793	92,9783	11	1243,53924	0	11
R109	1194,73	11	1481,28309	224,689	14	2241,02154	33,1972	17
R110	1118,59	10	1475,849583	109,333	13	1963,36401	12,1564	16
R111	1096,72	10	1377,901216	194,858	14	1813,01467	5,11401	15
R112	982,14	9	1257,187296	132,304	12	1421,76457	0	12
R201	1252,37	4	1565,907249	1144,25	5	1848,99632	228,356	**4**
R202	1191,7	3	1388,828566	759,797	4	1965,89197	210,674	4
R203	939,54	3	1160,027528	1051,39	4	1636,75627	83,5735	**3**
R204	825,52	2	1054,837502	448,588	3	1354,8304	1,19152	3
R205	994,42	3	1342,850499	541,149	4	2112,81935	20,6917	4
R206	906,14	3	1173,032755	227,689	**3**	1324,66671	13,1819	**3**
R207	893,33	2	1219,656544	459,048	3	1598,67198	2,05398	3
R208	726,75	2	1042,079502	63,8987	3	1184,97228	0	3
R209	909,16	3	1281,502899	265,526	**3**	1508,54776	10,6256	**3**
R210	939,34	3	1267,934157	1095,78	5	1838,01701	63,7263	4
R211	892,71	2	1149,941558	311,206	3	1378,8621	0	3
RC101	1696,94	14	1861,696172	471,751	17	2485,48033	181,074	18
RC102	1554,75	12	1689,298893	310,608	15	1917,89169	70,5718	15
RC103	1261,67	11	1715,112601	112,968	14	2307,69982	1,70079	17
RC104	1135,48	10	1356,2666	18,7669	11	1384,65652	0	11
RC105	1629,44	13	1862,905301	506,795	17	2277,74038	70,8731	16
RC106	1424,73	11	1789,157239	284,927	15	1914,74382	56,4388	14

The results of the Alg. 2 (Nearest Neighbor Based GA) (Continued).

Data Set	BEST KNOWN		BEST DISTANCE			BEST WAITING TIME		
	Total Distance	**Vehicle Number**	**Total Distance**	**Waiting Time**	**Vehicle Number**	**Total Distance**	**Waiting Time**	**Vehicle Number**
RC107	1230,48	11	1422,41617	232,298	13	2066,07919	14,3832	15
RC108	1139,82	10	1317,281024	115,259	12	1835,92742	0	14
RC201	1406,91	4	1641,698493	1885,22	6	2074,91242	141,959	**4**
RC202	1367,09	3	1621,036496	757,067	4	2463,8605	92,6475	4
RC203	1049,62	3	1396,817663	762,186	4	1912,38459	70,5379	4
RC204	798,41	3	1238,716185	379,266	**3**	1646,17705	0	**3**
RC205	1297,19	4	1513,01488	1174,8	5	2832,78599	80,1447	5
RC206	1146,32	3	1658,036463	679,406	4	2092,06263	57,2278	4
RC207	1061,14	3	1516,630141	576,344	4	1912,11896	25,581	4
RC208	828,14	3	1419,097994	489,09	4	1879,33852	3,49175	4

* Bold and gray cells show the results that are better than or equal to the best known results.

APPENDIX F: The results of the Alg. 3 (Sweep Based GA).

Data Set	BEST KNOWN Total Distance	BEST KNOWN Vehicle Number	BEST DISTANCE Total Distance	BEST DISTANCE Waiting Time	BEST DISTANCE Vehicle Number	BEST WAITING TIME Total Distance	BEST WAITING TIME Waiting Time	BEST WAITING TIME Vehicle Number
C101	828,94	10	**828,9368669**	0	**10**	**828,936867**	0	**10**
C102	828,94	10	1012,552774	237,737	12	1012,55277	237,737	12
C103	828,06	10	1122,761362	157	12	1122,76136	157	12
C104	824,78	10	1072,936212	61	11	1378,37948	8	12
C105	828,94	10	925,6934216	146	11	1245,01072	29	11
C106	828,94	10	890,2397702	198	11	930,730458	51	11
C107	828,94	10	956,7962672	92,3185	11	966,796267	0	11
C108	828,94	10	874,6283266	0	10	874,628327	0	10
C109	828,94	10	867,9248906	0,85786	10	867,924891	0,85786	10
C201	591,56	3	**591,5565567**	0	**3**	**591,556557**	0	**3**
C202	591,56	3	668,4144578	120,8	4	696,491534	9,55068	**3**
C203	591,17	3	721,139414	305,052	4	1211,72079	59,0246	4
C204	590,6	3	783,3421637	543,761	4	853,634486	4,57841	4
C205	588,88	3	623,7795329	0	**3**	623,779533	0	**3**
C206	588,49	3	639,5584812	1,38447	**3**	641,61486	0	**3**
C207	588,29	3	622,0696023	2,53569	**3**	626,528294	0	**3**
C208	588,32	3	623,7652925	0	**3**	623,765293	0	**3**
R101	1645,79	19	1871,781495	849,681	20	2065,34855	634,255	20
R102	1486,12	17	1720,868604	678,593	19	2025,8661	386,828	19
R103	1292,68	13	1537,270271	304,027	16	2048,94899	58,0207	18
R104	1007,24	9	1231,816725	99,387	12	1563,13775	4,93729	13
R105	1377,11	14	1571,052707	372,127	16	1888,06043	177,506	17
R106	1251,98	12	1476,602014	278,205	15	1679,32401	55,1857	15
R107	1104,66	10	1360,444244	134,518	13	1718,3397	9,49296	14
R108	960,88	9	1130,642638	89,3476	11	1206,18969	0	11
R109	1194,73	11	1364,172445	117,083	13	1485,23661	44,9966	14
R110	1118,59	10	1315,006491	165,305	13	1595,1918	21,1001	14
R111	1096,72	10	1360,969554	139,004	13	1540,68573	18,9949	14
R112	982,14	9	1120,326718	83,7773	11	1372,00028	0	12
R201	1252,37	4	1576,706851	1320,81	5	1985,55965	96,8269	**4**
R202	1191,7	3	1378,527847	670,608	4	1774,87754	110,523	4
R203	939,54	3	1226,112473	808,695	4	1525,90122	75,8093	4
R204	825,52	2	940,5705227	404,137	3	1255,19645	4,93244	3
R205	994,42	3	1202,644682	374,167	**3**	1393,73935	2,59732	**3**
R206	906,14	3	1151,439629	186,234	**3**	1243,80569	0	**3**
R207	893,33	2	1063,278422	386,981	3	1195,8824	0	3
R208	726,75	2	822,5075024	375,337	3	961,307126	0	3
R209	909,16	3	1150,579524	638,644	4	1395,04225	7,42714	**3**
R210	939,34	3	1107,123482	1086,58	4	1387,83024	8,14796	**3**
R211	892,71	2	957,3176019	261,621	3	1234,19273	0,38279	3
RC101	1696,94	14	1785,81427	361,989	16	1999,13977	208,375	16
RC102	1554,75	12	1595,666513	439,711	15	1855,61362	53,0262	15
RC103	1261,67	11	1539,65125	157,929	14	1709,37049	9,8601	14
RC104	1135,48	10	1343,633179	48,0705	12	1356,34865	0	12
RC105	1629,44	13	1906,320457	451,425	17	2197,2778	228,05	17
RC106	1424,73	11	1520,823332	169,711	14	1839,8314	37,2166	14

The results of the Alg. 3 (Sweep Based GA) (Continued).

Data Set	BEST KNOWN		BEST DISTANCE			BEST WAITING TIME		
	Total Distance	Vehicle Number	Total Distance	Waiting Time	Vehicle Number	Total Distance	Waiting Time	Vehicle Number
RC107	1230,48	11	1400,985188	202,81	13	1717,48271	24,1094	13
RC108	1139,82	10	1277,553924	133,989	12	1608,6774	0,24234	12
RC201	1406,91	4	1693,648355	1062,64	5	2508,79611	174,909	5
RC202	1367,09	3	1547,468849	732,705	4	2037,04646	81,525	4
RC203	1049,62	3	1254,455398	956,248	5	1737,47616	0	4
RC204	798,41	3	922,6664121	668,48	**3**	1255,41276	5,05459	**3**
RC205	1297,19	4	1593,750547	1464,11	6	2046,27187	62,8944	**4**
RC206	1146,32	3	1322,467078	541,207	4	1584,90105	1,23799	**3**
RC207	1061,14	3	1258,146058	590,311	4	1745,27775	40,7334	4
RC208	828,14	3	1028,267975	251,255	**3**	1277,7169	4,80024	**3**

* Bold and gray cells show the results that are better than or equal to the best known results.

VITA

PERSONEL INFORMATION

Name Surname	: Meltem YAKTUBAY
Place and Year of Birth	: Adana, 1990
Contact Number	: (0322) 455 00 00 - 2138
Mail	: myaktubay@adanabtu.edu.tr

EDUCATION

Master of Science	: Adana Science and Technology University and Çukurova University (Joint Master of Science Program) (2015-present) *Graduate School of Natural and Applied Science* *Industrial Engineering Department*
Bachelor of Science	: Gaziantep University (2008-2013) *Faculty of Engineering* *Industrial Engineering Department*
High School	: Seyhan Rotary Anatolian High School (2004-2008)

WORK EXPERIENCE

Research Assistant	: Adana Science and Technology University (2015-present) *Faculty of Engineering* *Industrial Engineering Department*