SOLVING VEHICLE ROUTING PROBLEM WITH HARD TIME WINDOWS BY GENETIC ALGORITHM

MISS AYE THANT MAY

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF ENGINEERING
(COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2021

Solving Vehicle Routing Problem with Hard Time Windows by Genetic Algorithm

Miss Aye Thant May B.C.Sc. (Software Engineering)

A Thesis Submitted in Partial Fulfillment
of the Requirements for
The Degree of Master of Engineering (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2021

Thesis Committee

.... Chairman of Thesis Committee
(Assoc. Prof. Charnchai Pluempitiwiriyawej, Ph.D.)

.... Member and Thesis Advisor
(Asst. Prof. Jumpol Polvichai, Ph.D.)

.... Member
(Assoc. Prof. Poj Tangamchit, Ph.D.)

.... Member
(Asst. Prof. Nuttanart Facundes, Ph.D.)

.... Member
(Assoc. Prof. Natasha Dejdumrong, Ph.D.)

Thesis Title	Solving Vehicle Routing Problem with Hard Time Windows by Genetic Algorithm
Thesis Credits	12
Candidate	Miss Aye Thant May
Thesis Advisor	Asst. Prof. Dr. Jumpol Polvichai
Program	Master of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2021

Abstract

In the logistic management, the cost reduction for delivering the goods to customers is crucial to save the budget of the company. Since decades ago, various Vehicle Routing Problems (VRPs) have been emerged enormously to improve the productivity and to reduce the logistic cost of the industry. Among them, Vehicle Routing Problem with Time Windows (VRPTW) is one of the most fundamental VRP variants and one of the most applicable variants in the real-world case studies. In this study, VRPTW with hard time windows is solved by developing a special Genetic Algorithm (GA), composed of a problem-specific crossover operator and seven different mutation operators. The proposed GA has better results with the heuristic mutation among seven operators while exploring the new and better features in large search space. The results of the algorithm are tested on the popular Solomon benchmark 100-customer datasets. The results show that the proposed GA is quite comparable with the best-known solutions on the C set of Solomon benchmark and even better than the best-known solutions on the R and RC sets. The motivation behind this research is to find a new problem-specific configurations of GA for VRPTW domain which can have the desirable outputs that can be applied in most of the real-world use cases.

Keywords: Artificial Intelligence / Combinatorial Optimization / Vehicle Routing Problem / Hard Time Windows / Meta-heuristic / Genetic Algorithm

ACKNOWLEDGEMENTS

I would like to heartfully thank King Mongkut's University of Technology Thonburi (KMUTT) for granting the Multi-intellectual Scholarship to pursue master's degree. My sincere acknowledgement goes to the Department of Computer Engineering (CPE) from KMUTT for fully supporting me to conduct this thesis in partial fulfillment of the requirements of the degree of Master of Science (Computer Engineering), under the unpredictable circumstances of the global pandemic. As an international student, I acknowledge the International Affairs Office members for helping me throughout my stay in Thailand.

I would like to show the great gratitude to my thesis advisor, Asst. Prof. Dr. Jumpol Polvichai from the Department of Computer Engineering, KMUTT for his generous support by grounding with his patience, courage, and vast knowledge. The blossom of my research was achieved all because of the continuous support of him in every step of the experiment journey.

Besides my advisor, I would like to thank my proposal and thesis committee members: Assoc. Prof. Dr. Charnchai Pluempittiwiriyawej from the Department of Electrical Engineering, Chulalongkorn University, Assoc. Prof. Dr. Poj Tangamchit from the Department of Control and Instrumentation Engineering, KMUTT, Assoc. Prof. Dr. Natasha Dejdumrong, Asst. Prof. Dr. Nuttanart Facundes, and Lect. Dr. Jaturon Harnsomburana from the Department of Computer Engineering, KMUTT for their encouragement, insightful comments, and discussions.

Moreover, three members of the iNENG Lab from CPE are sincerely acknowledged for their invaluable suggestions, comments, and cheerful courage. Finally, I would like to thank my family and friends from CPE who continuously encouraged me to accomplish this postgraduate milestone.

CONTENTS

CONTENTS (Cont.)

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

LIST OF TECHNICAL VOCABULARY AND ABBREVIATIONS

VRP	=	Vehicle Routing Problem
TSP	=	Travelling Salesman Problem
VRPP	=	Vehicle Routing Problem with Profits
TOP	=	Team Orienteering Problem
CTOP	=	Capacitated Team Orienteering Problem
TOPTW	=	Team Orienteering Problem with Time Windows
VRPDP	=	Vehicle Routing Problem with Delivery and Pickup
VRPLIFO	=	Vehicle Routing Problem with LIFO (stack fashion)
CVRP	=	Capacitated Vehicle Routing Problem
VRPTW	=	Vehicle Routing Problem with Time Windows
VRPHTW	=	Vehicle Routing Problem with Hard Time Windows
VRPSTW	=	Vehicle Routing Problem with Soft Time Windows
VRPMT	=	Vehicle Routing Problem with Multiple Trips
OVRP	=	Open Vehicle Routing Problem
IRP	=	Inventory Routing Problem
MDVRP	=	Multi-Depot Vehicle Routing Problem
MDVRPTW	=	Multiple Depot Vehicle Routing Problem with Time Windows
HFVRP	=	Heterogeneous Fleet Vehicle Routing Problem
FSMVRP	=	Fleet Size Mix Vehicle Routing Problem
VRPSDP	=	Vehicle Routing Problem with Simultaneous Delivery and Pickup
VRPSDPSTW	=	Vehicle Routing Problem with Simultaneous Delivery and Pickup with Soft Time Windows
VRPB	=	Vehicle Routing Problem with Backhauling
VRPMDP	=	Vehicle Routing Problem with Mixed Delivery and Pickup
AI	=	Artificial Intelligence
SI	=	Swarm Intelligence
GA	=	Genetic Algorithm

ACO	=	Ant Colony Optimization
PSO	=	Particle Swarm Optimization
RL	=	Reinforcement Learning
TD	=	Temporal Difference

CHAPTER 1 INTRODUCTION

This chapter completely explains about the research by describing the problem statement, mathematical model of the problem, objective, motivation, and hypothesis to conduct this research. Furthermore, overview of research methodology of the work is also introduced.

1.1 Problem Background

Since from the earlier days of 1950s, the decision makers in large corporations had tried to achieve an incredible improvement and growth in the organizational complexity related managements together with the rapid industrial revolution. To manage the organizational workflow efficiently and effectively, one related problem has risen. A problem of increasing the complexities in organization workflow, that is, the difficulty to allocate the available resources to the work processes efficiently and to plan or accomplish the work activities as well as operations of the organization effectively for gaining more benefits. Beginning from that moment, Operations Research (OR) studies were gradually carried out as a solution for the above-mentioned problem.

It has been widely applied in different organizations in these areas such as government and military, educational institutions, transportation, manufacturing, constructions, financial institutions, health care units and public services. In general, conducting the operations research includes the following steps:

- Observing the problem
- Formulating or modeling the problem
- Solving the problem
- Evaluating the solutions

Combinatorial optimization is a sub-area of mathematical optimization which is connecting with operation research. This topic has many different applications in real-world such as logistics and supply chains, resource allocations and distribution networks. optimization since from the decades ago. Those are the optimization problems where the possible solutions can be driven using combinatorics such as sets, subsets, combinations, or permutations and the graph theory such as vertices and edges. The term "combinatorial" refers to item orderings. Popular examples include Traveling Salesman Problem (TSP) which finds an optimal ordering of the cities that a salesman should visit but the total distance travelled of the route should be minimum. Several computer science related fields such as Artificial Intelligence (AI), machine learning and evolutionary computing which are the subfields of AI, applied mathematics which gives the exact solutions for the specified problem have been active research fields for the combinatorial optimization.

Vehicle Routing Problem (VRP) is the type of combinatorial optimization. The problem asks, "What is the set of optimal ordering of the customers that a set of vehicles should visit in order to deliver the goods, but the total travel cost of the routes should be minimum?" VRP generalizes the traditional TSP as it has many traveling salesmen which are vehicles in this context and more sophisticated constraints for choosing the optimal route for each vehicle. The Vehicle Routing Problem was firstly introduced in [2].

There are many variants of VRP models which are used for different problems depending on the specific constraints. Some of the well-known problems from the previous works include:

- **VRP** – Vehicle Routing Problem. It only has vehicle capacity constraint.
- **VRPTW** – Vehicle Routing Problem with Time Windows. It has both vehicle capacity and time constraints.
- **HVRP** – Heterogeneous Vehicle Routing Problem. The vehicles used in this problem have different capacities and types.
- **MDVRP** – Multi-Depot Vehicle Routing Problem. This problem has multiple depots (warehouses). It has vehicle capacity constraint.
- **MDVRPTW** – Multi-Depot Vehicle Routing Problem with Time Windows. It is MDVRP with time constraint.
- **VRPDP** – Vehicle Routing Problem with Delivery and Pickup. This problem has both the delivery of goods from the depot to the customers and pickup of the goods from the customers to the depot. It has vehicle capacity constraint.
- **VRPDPTW** – Vehicle Routing Problem with Delivery and Pickup with Time Windows. It is VRPDP with time constraint.

- **VRPSDP** – Vehicle Routing Problem with Simultaneous Delivery and Pickup. It is an extension of VRPDP but the delivery and pickup happen at the same time at each customer. It has vehicle capacity constraint.
- **VRPSDPTW** – Vehicle Routing Problem with Simultaneous Delivery and Pickup with Time Windows. It is VRPSDP with time constraint.
- **VRPSDPSTW** – Vehicle Routing Problem with Simultaneous Delivery and Pickup with Soft Time Windows. It is VRPSDPTW with adjustable time constraint.

Throughout the decades, different solution approaches have been proposed for the VRP variants by various researchers. These methods can be categorized into four big groups:

1) Exact algorithms
2) Heuristics algorithms
3) Meta-heuristics algorithms
4) Hybrid algorithms

In this study, a well-known meta-heuristic Genetic Algorithm (GA) is proposed to solve the VRPTW variant. In-depth of VRPTW is discussed in the section 1.2 and chapter 2. The proposed GA is explained in chapter 3.

1.2 Mathematical Formulation of the Problem

VPRTW can be formulated into a mathematical model as follows:

Decision Variable:

x(ijk) = 1 if the vehicle k travels from customer i to customer j and 0 otherwise.
$i \neq j$; $i,j \in \{0,1,2,\ldots,N\}$; 0 refers to depot.

Parameters:

N = number of customers

K = number of vehicles

D(ij) = distance travelled from customer i to customer j

d(i) = delivery demand of customer i

q(k) = capacity of vehicle k

T(i) = arrival time at customer i

e(i) = earliest arrival time at customer i

l(i) = latest arrival time at customer i

s(i) = service time at customer i

Objective function:

$$\boldsymbol{minimize}\left(\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{j\neq i,k=1}^{K} D(ij).x(ijk)\right) \tag{1.1}$$

Constraints:

$$\sum_{j=0}^{N} x(ijk) = 1 \tag{1.2}$$

i = 0 and ∀k ∈ K

$$\sum_{j=0}^{N} x(ijk) \leq K \tag{1.3}$$

i = 0

$$\sum_{k=1}^{K}\sum_{j=0,j\neq i}^{N} x(ijk) = 1 \tag{1.4}$$

$$\sum_{k=1}^{K}\sum_{i=0,i\neq j}^{N} x(ijk) = 1$$

∀i ∈ N

∀j ∈ N

$$\sum_{i=1}^{N} x(ijk) - \sum_{i=1}^{N} x(jik) = 0 \tag{1.5}$$

∀j ∈ N

∀k ∈ K

$$\sum_{i=1}^{N}\sum_{j=0,j\neq i}^{N} d(i).x(ijk) \leq q(k) \tag{1.6}$$

∀k ∈ K

$$e(i) \leq T(i) + s(i) \leq l(i) \tag{1.7}$$

Equation 1.1 is the objective function used to minimize the total distance traveled by all vehicles without exceeding each vehicle's capacity within customer preferred time windows. Equation 1.2 specifies each route must start from a depot and ends at that depot. Equation 1.3 represents the number of vehicles at the depot which indirectly means the number of routes at that depot. Equation 1.4 defines that each customer can be only visited once by one of the vehicles from the depot. Equation 1.5 constrains the same vehicle must arrive and departs from that customer. Equation 1.6 shows that the demand at each customer in each route of the vehicle must be less than or equal to the vehicle's capacity. Equation 1.7 constrains that the vehicle cannot arrive before the earliest arrival time and cannot be later than the latest arrival time.

1.4 Overview of the Research Methodology

Meta-heuristics algorithms have significantly gained interest from many researchers for solving VRP. Among them, GA based meta-heuristics are popular for having the good quality results. Because of the nature of GA's performance in global search space, it is picked as the solution for most of the VRP variants. Here, new problem-specific crossover operator and seven different mutations are introduced to solve a VRPTW variant on the well-known Solomon benchmarks.

1.5 Objectives of the Research

There was much research published on VRPTW. Researchers solve the problem in different approaches over decades. This research has the following objectives to fulfill:

- Contribute to the active GA research area specifically for the VRPTW.
- Reduce the travel cost of all routes that cover all customers.
- Increase the vehicle utilization.

1.6 Motivation to Conduct the Research

VRP is a popular problem from combinatorial optimization field. Among them, VRPTW is the most fundamental variant for all variants as it has vehicle capacity constraint and time constraint, mostly fit into the real-world problems. Therefore, the motivation behind this research is to find a new simple but problem-specific configurations of GA for VRPTW domain which can have the desirable outputs that can be applied in most of the real-world use cases.

1.7 Research Hypothesis

Hypothesis of this research is to state that the new proposed genetic operators of the GA can have good results for the VRPTW problem especially on the customers with random (not clustered) locations which have many similar use cases in the real-world.

1.8 Technical Requirements for Implementation

For implementation of this research, Python 3.7.2, Visual Studio Code editor and Google Colab runtime environment are used. The proposed GA is implemented from the scratch in Python.

1.9 Organization of this Research

This research is organized as follows: Chapter 1 discusses the background and statement of the problem, objectives and motivation of the research, hypothesis of the research, technical requirements to implement the problem. Chapter 2 provides the detailed review over the previous works on problem set, solution methods and datasets. Chapter 3 is about the methodology that is used to implement this research and finally Chapter 4 is concluded with the results and conclusion for this report.

CHAPTER 2 LITERATURE REVIEW

This chapter is comprised of the previous related works, surveys and articles contributed by other researchers on variants of VRP, VRPTW, benchmark datasets, solution approaches for the variants, genetic algorithm and reinforcement learning based approaches.

2.1 VRP Variants

In the set of combinatorial optimization problems which can be solved by linear integer programming models, VRP is a kind of the problem which considers the optimal cost for each of the route that is planned for each vehicle used to deliver goods to the set of customers. The problem was first described in [2]. The paper solved the problem between the fleet of trucks for the delivery of gasoline, the terminal and a large set of gasoline stations. They proposed the problem of VRP as the generalization of traditional TSP. According to the paper, the traditional TSP has only one salesman while the complex VRP has multiple salesmen and additional constraints.

There are several variants of VRP depending on the constraints of the problem. As it is described in [1], VRPP, TOP, CTOP, TOPTW, VRPDP, VRPLIFO, CVRP, CVRPTW, VRPMT, OVRP and IRP are some of the variants of VRP. Moreover, there are also MDVRP, MDVRPTW described in [3]. A comprehensive and organized survey [3] that presented most of the VRP variants which were categorized into respective constraints that came from relative domains. It also summed up the different categories of solution approaches to the variants and a collection of benchmark datasets and real-world case studies as well.

Additionally, VRP for big cities have some distinct natures according to [4]. The paper summarized the constraints, models and solution approaches for VRP in urban areas grouped by different roles of stakeholders, namely, shipper, carrier, resident and administrator. The evolution of the research of VRP was visualized in [4] by organizing the variants of VRP into different groups respective to their research timelines. The paper reviewed on the city logistics and pointed out the six main characteristics of City VRP. Among them, traffic regulations and fast response were the example of characteristics that need to have sufficient studies in the future.

As the authors mentioned in [3], the variations in the type of vehicles used for the problem was also considered as the variants of VRP. There are two variations in terms of vehicles, namely, homogeneous fleet and heterogeneous fleet. For a specific VRP, the former uses the identical type of vehicles which have the same capacities while the latter uses the different type of vehicles which also have different capacities. If the fleet in VRP used is the heterogeneous type, there are two main classifications: HFVRP and FSMVRP in [5]. The additional constraints such as, capacity, time windows, number of depots, number of products, number of trips and symmetry of routes can be added into the problem according to the requirement.

Relating to handling the robust and stochastic approaches to uncertainty in heterogeneous fleet VRPs, [5] discussed the challenges from the academic to business sectors and the potential research areas by stating that VRP variants that is relating with product restriction, zone restriction, customer restriction and vehicle restriction are missing from the current research areas. The two case studies of HFVRP on LPG distribution sectors are discussed in [6] and [7]. The former paper considered CVRP with variable cost for the industry in Turkey and the latter one was on VRPSDP in Northern district of Kerala. [39] proposed a customized Tabu Search model for VRPSDP and [40] discussed about the case study of VRPSDP for cold-chain logistic.

2.2 VRPTW

Since the previous section was reviewed on most of the variants of VRP, this section is aimed to be reviewed on the specific type of VRP which is the problem that is proposed to solve in this research, VRPTW. The purpose of the VRPTW is to deliver goods to each customer within the customer preferred time windows without violating the capacity of the vehicle. According to [43], There are two types of VRPTW, namely, VRPHTW (Vehicle Routing Problem with Hard Time Windows and VRPSTW (Vehicle Routing Problem with Soft Time Windows) [41] and [42]. VRPHTW strictly constrains the vehicle to arrive within the time windows. Violation is not allowed. However, in VRPSTW, the time windows can be violated that means it allows the vehicle to after earlier or later than the customer preferred time. In that case, the penalty value for constraint violation balances the problem. The greater the penalty value is, the harder the constraint becomes. The penalty value of infinity makes VRPSTW to be equivalent with VRPHTW.

Many research have been flourished on VRPTW using both of exact algorithms, heuristics and meta-heuristics. Among them, popular approaches were with GA and Tabu Search. [34], [35] and [36] proposed hybrid GA approach for VRPTW on Solomon benchmarks while [37] proposed GA with local post-optimization procedures. The survey [8] concluded that different GA algorithms which yielded the good solutions are in the hybridization with route construction heuristics or local search.

2.3 Benchmark Datasets and Case-study Datasets

Datasets are essential to evaluate the results which are driven from the solution method that is implemented. [3] classified the types of datasets for VRP into two groups: datasets with symmetric routes and asymmetric routes. In symmetric routes dataset, the back and forth distances between two locations are the same. That distance can be computed as Euclidean distance. Compared to symmetric routes, distances between two locations can be different in each opposite direction or some routes may not even exist in one-way lanes. In general, most of the symmetric datasets are benchmark datasets while the asymmetric datasets are the real-world case study datasets.

In real-world case study studies datasets, there are different constraints in different case studies. Moreover, the datasets cannot be shared for future studies because some data are confidential in the perspectives of industries. Therefore, there is no common or standard dataset for assuring which solution approach is better. That is the reason why researchers choose benchmark datasets to prove their methods are eligible in the comparison with other methods as the benchmarks are the small sized, sharable, test datasets with BKS (best known solution). BKS value is the standard value for them to prove the effectiveness and efficiency of their implemented solution methods. According to many researchers, the benchmark datasets should be the first work in VRP for the proposed solution methods [3]. After achieving the satisfied results, those methods can then be applied to real-world case study datasets.

Benchmark datasets for different variants of VRP can be obtained from various websites. Among them, http://www.vrp-rep.org/ which is a collaborative open-data platform for sharing vehicle routing problem benchmark instances and solutions [23]. Also from https://neo.lcc.uma.es/vrp/ , the NEO Research group works at the Department of LCC from the University of Malaga (Spain) [24].

2.4 Solomon Benchmark for VRPTW variant

For using Solomon benchmark [38], Solomon assumed that distance travelled and travel time between two customers are the same. The author generated six problem sets which are described as C1, R1, RC1, C2, R2 and RC2. The geographical data in C sets are generated into clusters and the R sets are randomly generated while the RC sets are the combination of C and R sets. Type 1 and 2 describes the difference in width of time windows of the problems and also the vehicle capacity for the vehicles used in each problem. Type 1 has narrow time windows and small vehicle capacity while Type 2 has wide time windows and large vehicle capacity. Therefore, solutions of Type 1 have small number of customers per route approximately (5 to 10) so that there are more vehicles to be used. Solutions of Type2 have larger number of customers per route approximately (more than 30) according to [38].

2.5 Different Approaches for Solving VRP variants

Since VRP is a kind of operation research, the key objective for conducting the research is to achieve the optimized: minimized or maximized outcomes for the logistics domain. Throughout the decades, many algorithms have been proposed as the solution approaches for solving variants of VRP. Thus, considering the appropriate solution method for specific problem of VRP is important to achieve the desired results. The algorithms used as solution approaches for variants of VRP can be grouped into four categories; exact algorithms, heuristics algorithms, meta-heuristics algorithms and hybrid algorithms. The two comprehensive surveys [3] and [8] gathered the previous works on each category of solution approach and reviewed the strengths and weaknesses of approaches.

Exact algorithms are mathematical solutions for solving VRP. They are capable of providing the exact optimal solution to the problem. According to [3] and [8], most of the well-known exact algorithms used to solve for VRP are: Brand-And-Bound, Branch-And-Cut, Branch-And-Price [12] and Set Partitioning [13]. However, VRP is NP-hard combinatorial optimization problem whose computation time increases intolerably with the size of problem. If we have n customers to deliver the goods, the total number of possible combinations of routes to plan is (n-1)!. Therefore, using exact algorithms has limitations to solve this kind of problem as they are very time-consuming and not suitable for larger VRP instances as described in [8], [14].

Because of limitations in exact algorithms for large problems, the later research focuses were directed to heuristics algorithms. Neighborhood search, Integer Programming models, Scatter Search algorithms, Sweep Algorithm that were described in [3] and [8] are examples of heuristics algorithms. They are approximate methods which are not the same as exact solutions that can guarantee the optimality. The solution from the heuristics algorithms cannot guarantee the optimality but they are acceptable and close to optimality in the reasonable amount of computation time. As every method has vulnerability [3], the heuristics algorithms are only capable to search the acceptable solutions in small search space.

In order to fill up the gap in heuristics algorithms, meta-heuristics algorithms were developed to find the solution of the problem in larger search space. We can also say that meta-heuristics algorithms are the ones which can guide heuristics algorithms in the larger search space. Meta-heuristics algorithms can avoid from being stuck at local optimum. There are two types of meta-heuristics algorithms: local search-based and population-based as in [3]. Simulated Annealing [15], [16], Tabu Search of [17] are local-based meta-heuristic and Genetic Algorithm [26], [27], Evolutionary Algorithm of [20] is a population-based meta-heuristic examples. Nature-inspired algorithms are also in the group of population-based. Swarm Intelligence (SI) algorithms including Particle Swarm Optimization of [21], Ant Colony

Optimization [18], [19] Bat Algorithm, Firefly Algorithm are kind of nature-inspired algorithms which have high efficiency and give better results to solve VRP.

Hybrid algorithms were introduced due to the limitations in each above category of approaches. Rather than solving with one algorithm and having ineffectiveness in the method, applying the combination of two or more exact, heuristics or meta-heuristics algorithms to produce more reliable solution became the modern research focus for researchers. During the recent past years, the implementation of GA improved by hybridization with other algorithms has given the better solutions for VRPTW [8]. According to [3], the recent research potential was on hybridization of two meta-heuristics: non-nature-inspired and nature-inspired, for example, GA with ACO in [22] or GA with PSO in [45], [46] and [47].

2.6 Genetic Algorithm based Approaches

GA (Genetic Algorithm) is a meta-heuristic method that is in the category of population-based search having the objective of applying the principles of biological evolution, natural selection, recombination of the genes which inspires the survival of the fittest theory from Charles Darwin to integrate this theory into the artificial systems. In 1975, it has been introduced by [43]. It has gained high interest from a number of researchers in different fields. Because of its adaptation to the environment in global search space, the solutions that can be driven from the GA can have better results than traditional approaches. In this section, pure and hybrid GA methods that were applied to the VRP variants are represented.

While reviewing the previous related works, pure or hybrid GA applied applications were categorized into two groups:

- In some multiple depot VRP, the customers were assigned to the nearest or optimal depot first [26], [27]. And then assign the appropriate group of customers to each vehicle at the depot and later GA was used to find the best combination of the route among the possible total routes of the assigned vehicle, which covered the all customers to deliver [25], [30]. This type of configuration is suitable when the number of vehicles are limited and not seek the cost saving benefit from reducing the number of vehicles.

- Applying GA directly to the problem set by allocating the depots and the vehicles to the feasible customers who were ordered in the gene representation of each individual, according to the constraints of the problem [26], [27], [28], [29]. This type of configuration is suitable when the number of vehicles are unlimited. This can have the vehicle insufficient problem if it is applied to the limited vehicle problem. Because all customers to deliver in the plan may not be covered by the predefined number of vehicles. In this case, the similar concept as the penalty function for time windows approach which was mentioned in section 2.3 will be required.

The GA has its own operators to achieve the improvement process for the solutions over time. Based on the different set ups of operators which were generally used in GA for VRP, a comprehensive review was discussed in [11]. The following list provides the summary from the review of [11].

- **Chromosome Representation** – permutation representation, direct representation, path representation and binary representation.
- **Initial Population** – randomly generating, using heuristics, using both heuristics and randomly generating.
- **Fitness Function** – equation for the total travel distance that is evaluated to achieve the smaller value at times.
- **Selection** – roulette wheel selection, rank-based selection, tournament selection and uniform selection, elitism.
- **Crossover** – one-point crossover (1PX), two-point crossover (2PX), order crossover (OX), partially mapped crossover (PMX), cyclic crossover (CX), route based crossover (RBX), sequence base crossover (SBX), single parent crossover (SPO), genetic vehicle representation crossover (GVR), best cost best route crossover (BCBRC).
- **Mutation** – swap mutation (SWM), inversion mutation (INVM), insertion mutation (INSM), reallocation mutation (RAM), exchange mutation (EXM), reposition mutation (RPM).

- **Termination Condition** – maximum number of generation where max generation equals number of explored chromosomes divided by population size, other stopping conditions like no improvement over generations.
- **Parameter Selection** – population size > 10, crossover rate as a continuous number between 0 and 1, mutation rate as a continuous number between 0 and 1.

From the initial population set up from that review, using heuristics to generate the initial population of GA becomes the part of hybrid GA. Hybrid algorithms for VRP normally have two parts: route construction and route improvement as described in [31]. Route construction methods starts with a customer who is chosen randomly and subsequent customer which has minimum travel cost from the first one is chosen continuously until all the customers are covered and the route is completed. Route improvement methods starts with a given complete route and then modify the route by recombining or introducing different link to obtain the minimum travel cost.

Some researchers [31], [32] who have been conducting the research of hybrid GA on VRP used one heuristic as a route construction method and fed the initial routes that was obtained from the route construction method into the initial population of GA, which has been used as a route improvement method to achieve the better routes that have minimum travel cost.

2.7 Reinforcement Learning based Approaches

RL (Reinforcement Learning) is the highly active research area in machine learning with its mind-blown achievements in decision making systems like robotics and gaming. Since from recent few years, it has gained the interest of researchers from combinatorial optimization. However, compared to other field, application of reinforcement learning in combinatorial optimization is still few due to some of its limitations. Therefore, it becomes the potential research area of VRP. [9] and [10] applied deep reinforcement learning which was the deep Q-network (DQN), a combination of deep learning and Q-learning for TSP and VRP respectively. Furthermore, RL was used as a part of other algorithms as a hybrid solution as in [31], [32] and [33].

2.8 State-of-the-Art reviews on VRP variants and solutions

This following section provides the detailed state-of-the-art reviews on both of VRP variants and solution approaches to address the variants.

2.8.1 Review of Meta-heuristics

The survey of [56] reviewed 299 articles published between 2009 and 2017 on the VRP variants, as shown in Figure 2.2 and meta-heuristics approaches, as shown in Figure 2.1. The meta-heuristics were classified into two main families, namely, single-solution based approaches and population-based approaches. Table 2.1 shows the number of occurrences of the VRP variants in reviewed articles throughout the years. Table 2.2 and Table 2.3 shows the different types of meta-heuristics approaches used throughout the years. According to the figures and the tables described in [56], it shows that VRPTW and the GA are the most popular variant and solution approach along the review period.

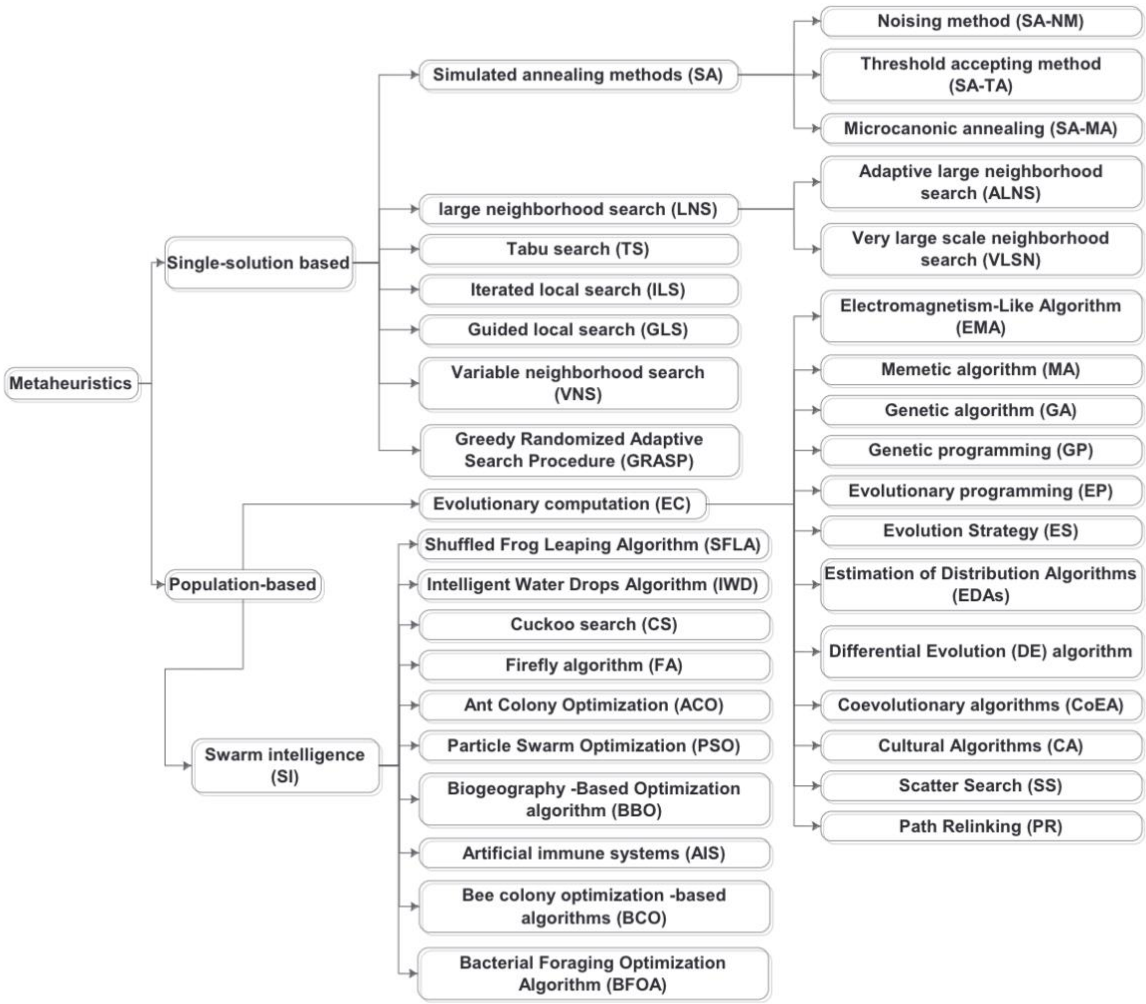

Figure 2.1 Classification Tree of Metaheuristic Algorithms described in [56].

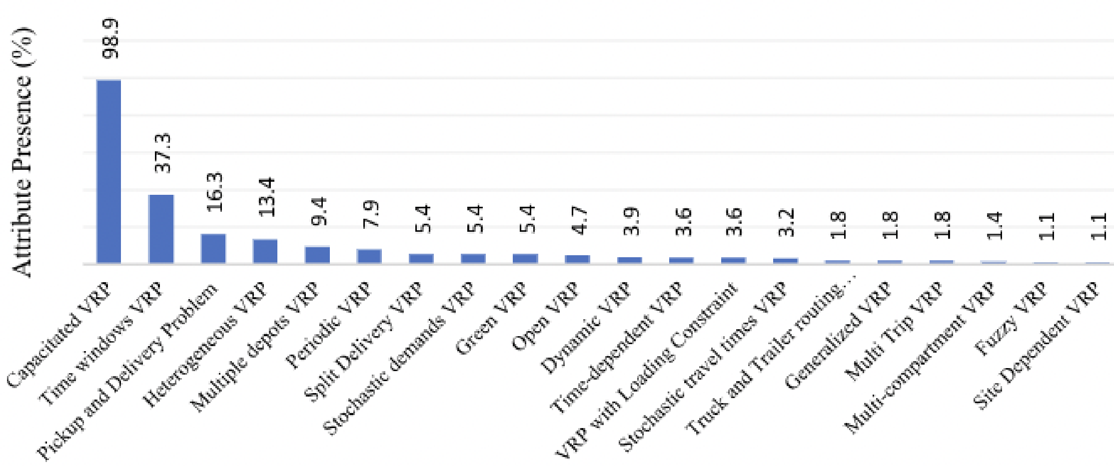

Figure 2.2 Overall percentage of VRP attributes in the reviewed articles. described in [56].

Table 2.1 Relative presences of VRP variants in reviewed articles described in [56].

VRP Variants	Number of models	Relative presence (%)									
		Overall	2009	2010	2011	2012	2013	2014	2015	2016	2017
Capacitated VRP	273	98.91	100.00	96.43	95.45	100.00	100.00	96.77	100.00	100.00	100.00
Time windows VRP	103	37.32	24.14	28.57	31.82	32.14	50.00	45.16	53.66	28.21	31.82
Pickup and Delivery Problem	45	16.30	27.59	14.29	4.55	14.29	19.44	9.68	26.83	15.38	4.55
Heterogeneous VRP	37	13.41	20.69	10.71	9.09	14.29	19.44	12.90	9.76	15.38	4.55
Multiple depots VRP	26	9.42	0.00	0.00	18.18	10.71	13.89	16.13	7.32	12.82	4.55
Periodic VRP	22	7.97	6.90	10.71	9.09	10.71	8.33	9.68	9.76	5.13	0.00
Split Delivery VRP	15	5.43	3.45	7.14	9.09	0.00	13.89	3.23	4.88	5.13	0.00
Stochastic demands VRP	15	5.43	0.00	7.14	4.55	10.71	2.78	9.68	4.88	7.69	0.00
Green VRP	15	5.43	3.45	3.57	0.00	3.57	8.33	9.68	2.44	7.69	9.09
Open VRP	13	4.71	3.45	7.14	9.09	10.71	2.78	6.45	2.44	2.56	0.00
Dynamic VRP	11	3.99	3.45	3.57	4.55	3.57	5.56	0.00	4.88	5.13	4.55
Time-dependent VRP	10	3.62	3.45	3.57	4.55	7.14	0.00	3.23	0.00	7.69	4.55
VRP with Loading Constraint	10	3.62	0.00	3.57	0.00	3.57	2.78	0.00	4.88	10.26	4.55
Stochastic travel times VRP	9	3.26	0.00	3.57	0.00	3.57	5.56	3.23	2.44	5.13	4.55
Truck and Trailer routing problem	5	1.81	3.45	0.00	4.55	0.00	5.56	0.00	0.00	0.00	4.55
Generalized VRP	5	1.81	0.00	0.00	4.55	0.00	2.78	3.23	4.88	0.00	0.00
Multi Trip VRP	5	1.81	0.00	0.00	0.00	0.00	2.78	9.68	0.00	0.00	4.55
Multi-compartment VRP	4	1.45	0.00	3.57	0.00	0.00	0.00	0.00	4.88	0.00	4.55
Fuzzy VRP	3	1.09	0.00	3.57	0.00	0.00	0.00	3.23	0.00	0.00	4.55
Site Dependent VRP	3	1.09	0.00	0.00	0.00	3.57	2.78	0.00	2.44	0.00	0.00

Table 2.2 Relative Presences of single solution-based metaheuristics in the study period described in [56].

Single-solution based metaheuristics	Number of models	Relative presence (%)									
		Overall	2009	2010	2011	2012	2013	2014	2015	2016	2017
TS	74	30.1	36.8	37.0	42.9	35.0	33.3	34.6	25.7	25.0	19.0
VNS	56	22.8	15.8	15.0	14.3	15.0	18.5	19.2	28.6	22.2	38.0
LNS	38	15.4	10.5	3.70	14.3	15.0	14.8	11.5	8.57	25.0	26.0
SA	30	12.2	21.1	7.40	14.3	10.0	14.8	11.5	11.4	16.7	7.10
ILS	24	9.76	0.00	7.40	0.00	10.0	11.1	19.2	17.1	5.56	9.50
GRASP	21	8.54	5.26	26.0	14.3	15.0	7.41	3.85	8.57	5.56	0.00
GLS	3	1.22	10.5	3.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 2.3 Relative presences of population-based metaheuristics in the reviewed articles described in [56].

Population-based metaheuristics		Number of models	Relative presence (%)									
			Overall	2009	2010	2011	2012	2013	2014	2015	2016	2017
Evolutionary computation (58.1%)	GA	46	58.0	62.5	18.0	75.0	73.0	46.7	66.7	66.7	33.3	90.0
	MA	13	16.0	25.0	27.0	0.00	18.0	20.0	16.7	11.1	16.7	0.00
	PR	8	10.0	0.00	18.0	25.0	0.00	20.0	0.00	11.1	16.7	0.00
	SS	6	7.50	12.5	9.10	0.00	9.10	13.3	0.00	0.00	0.00	10.0
	DE	2	2.50	0.00	9.10	0.00	0.00	0.00	16.7	0.00	0.00	0.00
	CoEA	2	2.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	33.3	0.00
	EMA	2	2.50	0.00	9.10	0.00	0.00	0.00	0.00	11.1	0.00	0.00
	ES	1	1.30	0.00	9.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	EP	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	EA	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	GP	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	EDAs	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Swarm intelligence (41.9%)	ACO	28	46.7	60.0	33.3	62.5	25.0	60.0	0.00	42.0	40.0	60.0
	PSO	18	30.0	40.0	66.7	25.0	75.0	40.0	0.00	25.0	20.0	20.0
	ABC	5	8.33	0.00	0.00	12.5	0.00	0.00	33.3	8.30	0.00	20.0
	BBMO	2	3.33	0.00	0.00	0.00	0.00	0.00	33.3	8.30	0.00	0.00
	SFLA	2	3.33	0.00	0.00	0.00	0.00	0.00	33.3	8.30	0.00	0.00
	BBO	1	1.67	0.00	0.00	0.00	0.00	0.00	0.00	8.30	0.00	0.00
	GSO	1	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.0	0.00
	CS	1	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.0	0.00
	IIWD	1	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.0	0.00
	GFA	1	1.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.0	0.00
	BFOA	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

2.8.2 Review of Machine Learning Methods

This section describes the review [57] of machine learning approaches to address the static and dynamic VRP variants. VRPTW in this research falls under the category of static VRP through the timeline of 2015 to 2020. Table 2.3 shows the summary of the survey. Reinforcement learning bloomed in the research field in the late 2018.

Table 2.4 VRP variants and machine learning methods described in [57].

Variant	Methods
Static	• Pointer Network - Vinyals et al. (2015) • Pointer Network with DRL - Nazari et al. (2018) • Graph Neural Network - Prates et al. (2019) • Actor-Critic Model - Vera and Abad (2019) • Multi-Agent Architecture for Metaheuristics - Silva et al. (2019) • GNN with Monte Carlo Tree Search - Xing et al. (2020) • Dynamic Attention Model - Peng et al. (2020) • Multi-agent Attention Model – Zhang et al. (2020)
Dynamic	• UCT, the extension of MCTS method - Świechowski and Mańdziuk (2016) • Simulated Annealing with RL - Waldy and Hoong (2020)

2.8.3 Review of Model Objectives

The survey of [58] reported that there are model objectives which are prioritized while solving the VRP variants. Figure 2.3 shows an objective to consider time window rate which are to be soft or hard stands third place among the others. This survey reviewed the articles from 2019 to 2021.

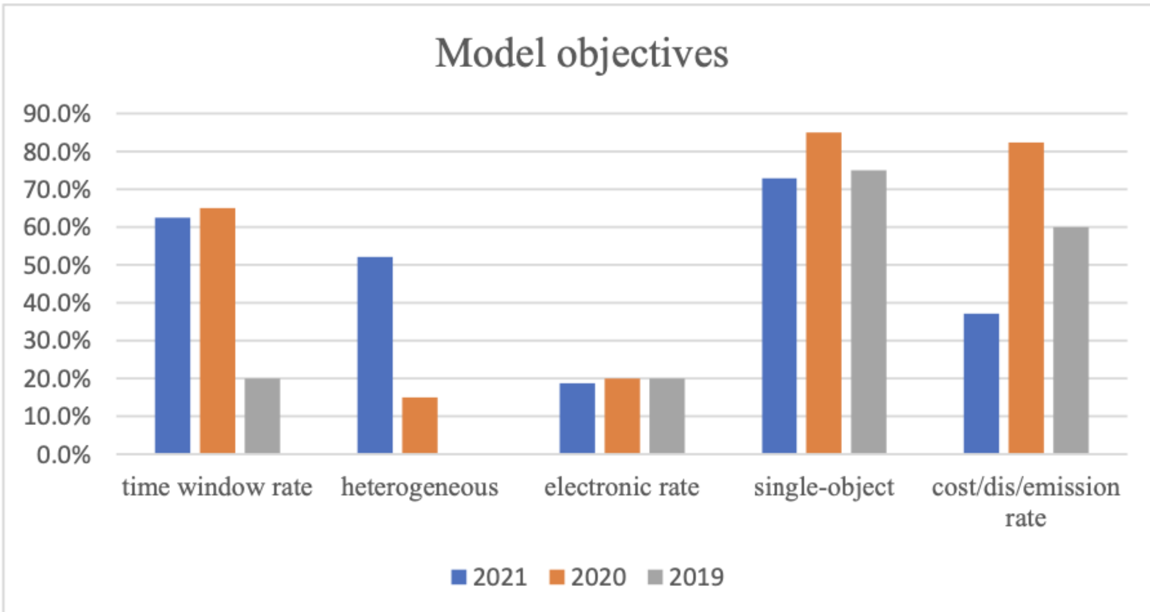

Figure 2.3 Different model objectives in different years described in [58].

CHAPTER 3 METHODOLOGY

This chapter is focused on the details of the improved Genetic Algorithm which has the problem-specific crossover and different mutations to solve the popular VRPTW instances, Solomon Benchmarks.

3.1 Genetic Algorithm

Genetic Algorithm (GA) is in the category of population-based search meta-heuristics which has the adaptation of applying the principles of biological evolution, natural selection, recombination of the genes which inspires the survival of the fittest concept from Charles Darwin's theory of natural evolution. This inspiration to natural evolution allows genetic algorithms to overcome some of the limitations of traditional search and optimization algorithms.

[59] stated the concept that, in the bit representations of GA, GA applies implicit sampling to the partitions of the hyperplane in a search space. The concept is on the process of crossover. Individual representation in the population can be increased or decreased by completing the hyperplanes. For tracking the sampling rate of hyperplane H, M(H,t) is the number of strings at current generation t, f(H,t) is the average evaluation of the strings in the population.

$$M(H, t + intermediate) = M(H,t)\frac{f(H,t)}{\bar{f}}.$$

The following concept explains the effect of crossover at the time of new generation formation from current generation. Disruptive effect of crossover makes gains or losses in the representation although the representation of the population remains the same before the crossover is applied.

$$M(H, t+1) = (1 - p_c)M(H,t)\frac{f(H,t)}{\bar{f}} + p_c\left[M(H,t)\frac{f(H,t)}{\bar{f}}(1 - losses) + gains\right]$$

Figure 3.1 describes the five stages to perform a GA such as:

- Initial Population
- Fitness Evaluation
- Selection
- Crossover
- Mutation

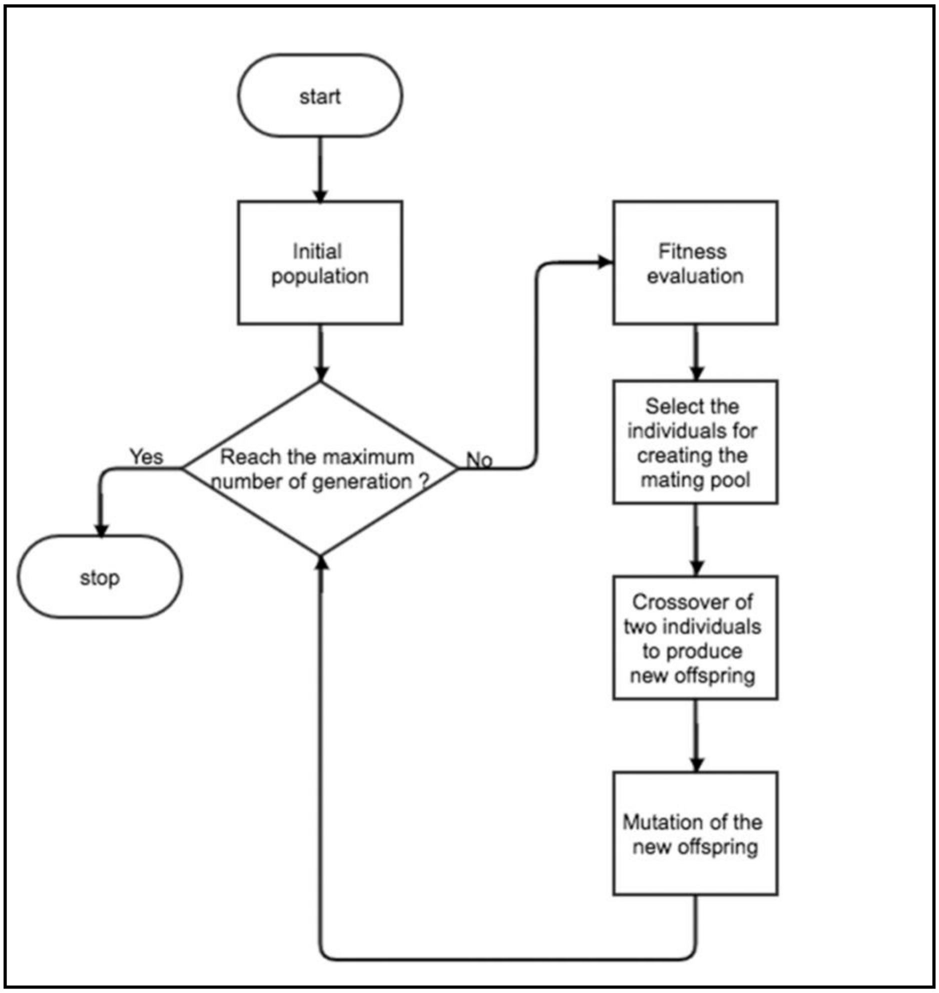

Figure 3.1 Workflow of a Genetic Algorithm

A new design of crossover operator for VRP, Remove-Insert Reverse Crossover and seven different mutation operators are implemented for this study. Each stage of this implemented GA will be explained in next sections.

3.1.1 Initial Population

Population that is used in GA refers to the group of individuals. The individuals are also referred as the chromosomes or possible solutions of the problem that is aimed to solve. Like a chromosome has a set of genes, the individual has a set of contents such as strings. With the help of its genetic operators such as selection, crossover and mutation, GA evolves its population by generations until the predefined condition for the evolution is met. After that the best individual or the best solution is chosen as the optimal solution for the problem.

Each individual has a representation of its contents. Sometimes, it is called as individual representation. It is a representation of genes or the contents of the individual. That means instead of processing on the solutions directly, GA performs on the encoded solutions, known as representation. GA does not consider interpreting what the individual represents. This behavior decouples the search from the original domain of the problem.

In this study, permutation representation alike route representation is chosen as the representation of the individuals which are used for the population of the implemented GA.

Example, 1-6-4-9-7-5-2-8 (a permutation sequence of the customer IDs in the complete route.) Then the sub-routes are:
sub-route 1: 0-1-6-4-0
sub-route 2: 0-9-7-5-2-8-0

Sub-routes are divided according to the constrain satisfaction such as vehicle capacity and time windows. Each sub-route reflects one vehicle. 0 refers to the depot.

In Python programming, the representation is implemented with nested lists. That is: A route contains two sub-routes: [[1, 6, 4], [9, 7, 5, 2, 8]]. This kind of notation with nested lists will be used for the examples described in next sections. Figure 3.2 shows the simple representation of a population that consists of three individuals and their genes.

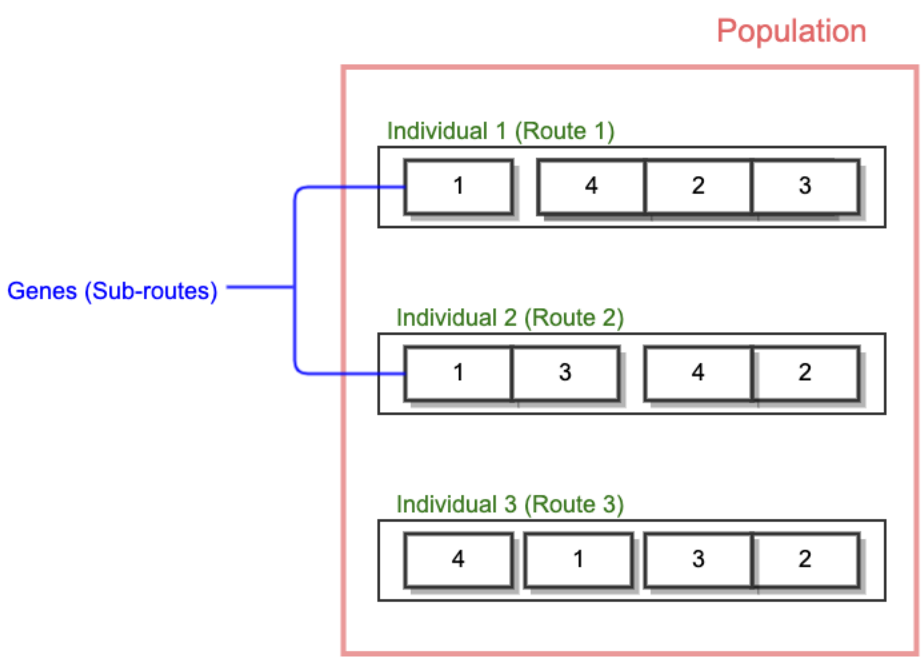

Figure 3.2 A simple example of individual representation

3.1.2 Fitness Evaluation

The objective function of a GA is also called as fitness function which is used to evaluate the fitness of each individual for the selection. In this study, it is a combination of total travel cost and total number of vehicles which are targeted to be minimized in VRPTW. Individuals are chosen according to the evaluation of their fitness to reproduce the new offspring. Individuals with greater fitness values have more chance to be selected as parents to recombine a new offspring. As the fitness function checks the minimum total travel cost and total number of vehicles in VRPTW studies, the reciprocal function in Python helps to change minimum value to have great fitness value to be selected.

VRPTW can be considered as multiple objective optimization because it has to reduce the number of vehicles and the total distance traveled. Some researchers solved this as bi-objective optimization by using the Pareto ranking method as in [50] to fairly consider both dimensions of the objectives. Some researchers used the weighted sum fitness method to solve multiple objectives as a single objective. In this paper, weighted sum fitness that is also used in [51] is applied to evaluate the fitness of individuals because the proposed work is the design of crossover and mutation to generally minimize both objectives rather than considering the bias of one specific objective as in some real- world use cases. The equation is described as follows.

$$\text{Fitness} = \alpha * \text{no. of vehicles} + \beta * \text{total distance travelled} \tag{3.1}$$

Equation 3.1 shows that α and β are the weight values for the number of vehicles and the total distance traveled, respectively. After the experiments with different parameter values, to give more priority to the cost than the number of vehicles, the weight values are adopted from [50] as $\alpha = 100$ and $\beta = 0.001$.

3.1.3 Selection

A selection operator is used to select the individuals with the great fitness values to create the mating pool of parents to reproduce the new offspring. For this algorithm, Tournament selection is used. The workflows of the tournament selection are to pick up the random n individuals from the population and choose one individual from them. And repeat this procedure from the start until meeting the number of individuals needed for performing crossover. These random n individuals are called tournament set size. Here, a tournament set size of 4 is chosen. With tournament selection, the elitism model is also used along. Elitism model means the specified number of elite individuals from the population are directly saved to the next generation without passing through the steps of crossover and mutation. Here, the number of elite individuals that are directly saved to the next generation is 10.

3.1.4 Crossover

Crossover is the recombination of two individuals and produces one or two offspring as an output. It is known as exploitation which is good for local search. Normally, the chance to occur the crossover operation should be high to reproduce the good combinations. Solving VRP like combinatorial optimizations needs to maintain the genes to not destroy the order structure. Other crossover operators like uniform crossover can happen multiple occurrences of genes as well as destroy the order structure. For this case, specifically designed crossover operators are required. The implemented Remove-Insert Reverse Crossover recombines the two parents from the selected population and provides one offspring as an output. To proceed, one or more consecutive sub-routes must be defined at Parent 1. In this paper, the number of sub-routes to remove is chosen randomly between 1 and the number of total sub-routes where 1 is considered inclusively and the number of total sub- routes is considered exclusively. Once the set of sub-routes from the Parent 1 is defined, the removal operation is performed in Parent 2. All the sub-routes in the set are removed in Parent 2 elementwise. After this step has been completed, Parent 2 lacks some customers as they have been removed. To insert the removed customers back to Parent 2, the previous set of sub-routes from the Parent 1 is used. One sub-route from the set is sequentially chosen at a time to insert at the first index of Parent 2. This procedure occurs until the set is empty which means the sub-routes from the set are inserted in reverse order at the head part of Parent 2.

The following example shows the step-by-step crossover operations. Since list notation is used as the data structure for performing the tasks, sub-routes within the complete route are described as lists inside the list. Lists are indexed starting from 0. Elements inside the lists are the customer IDs.

Parent 1: [[8,1,2], [3,4,5], [6,7]]

Parent 2: [[3,2,1], [4,8,5], [7], [6]]

Index 1 and 2 of the Parent 1 are chosen as the set for the element-wise removal in Parent 2. Therefore,

Removal Set: [[3,4,5], [6,7]]

Then the elements from the set are removed in Parent 2. Therefore, Parent 2 is now appeared as follows.

Parent 2: [[2,1], [8]]

After the insertion of sub-routes of the removal set in reverse order at the head part of Parent 2, the final offspring is ready for the new generation.

Offspring: [[6,7], [3,4,5], [2,1], [8]]

The following is the pseudocode to implement the crossover.

```
// pop_size is the size of population
// cx_pb is the probability to happen crossover
SET i = 0
FOR i to pop_size STEP 2
        IF random probability < cx_pb THEN
                SET child1 = CALL BCRC(parent1, parent2)
                    SET child2 = CALL BCRC(parent2, parent1)
        END IF
END FOR
RETURN (child1, child2)
```

```
BCRC(parent1, parent2)
SET cut_length = random integer from 1 to length of parent2 - 1
SET removal_set1 to empty list
COPY parent1 to C1
COPY parent2 to C2
FOR r1 in C2[0 to cut_length]
    APPEND r1 to removal_set1
    REMOVE r1 from C1
END FOR
FOR p in removal_set1
    INSERT p to C1 at index 0
END FOR
COPY C1 to child
RETURN (child)
```

3.1.5 Mutation

Mutation is used to introduce the new feature to the offspring. It is known as exploration. This action explores globally new features to approach the optimal solutions. The chance to occur mutation should be few as to not destroy the good gene representations from the crossover stage. According to the genes representation, the mutation rate to apply the individual is different. For bit representation, mutation is the independent bit flipping rate P_m of each individual. [60] described the mathematical notation of this as follows where the L refers to the length of the population:

$$P[\text{individual is mutated}] = 1 - (1 - p_m)^L$$

However, in order-based representation, which is permutation representation, P_m of each individual determines the current generation would proceed the mutation process or not. It is also known as the frequency for implementing the mutation process.

Seven different mutation operators are considered for this GA. A mutation operator to be used is chosen according to a random number selected from 0 to 6. The seven mutation operators used in this paper are as follows.

Mutation 1

This mutation is a proposed work of [50] and was also used in [51]. It is called Constrained Route Reversal Mutation. The idea of this mutation was proposed by the author by employing the reversal of 2 or 3 customers in a chosen sub-route so that it reduces the route corruption. A reversal of several customers in a sub-route can harm the sequence of route permutation that the crossover operator exploited.

Mutation 2

This mutation is a heuristic-guided operation. At each sub-route, if the distance from the current customer to the next customer is greater than the predefined threshold, cut off the customer points in the sub-route starting from that customer. Append one removed customer as one sub-route at the end of the complete route and stop the mutation. This is called adding additional heuristics by checking the distance threshold at the customer points to guide the algorithm to explore the new efficient features in the large solution space.

Mutation 3

At every customer point within a complete route, under 0.1 random probability, swap two customers at a time.

Mutation 4

At every customer point within a complete route, under 0.02 random probability, move the current customer to the end of the route at a time.

Mutation 5

Shuffle within each sub-route.

Mutation 6

At every customer point within each sub route, under 0.3 random probability, swap the two customers which are two offsets away.

Mutation 7

Under the 0.33 random probability, at the midpoint customer of each sub-route, exchange the first partition and the second partition of the sub-route.

The following is the pseudocode to implement the mutation.

```
// pop_size is the size of population
// mut_pb is the probability to happen mutation
SET i = 0
FOR i to pop_size
        IF random probability < mut_pb THEN
              SET random_mutation from 0 to 6
              IF random_mutation == 0 THEN
                  SET child = CALL Mutation 1(parent)
              END IF
              IF random_mutation == 1 THEN
                  SET child = CALL Mutation 2(parent)
              END IF
              IF random_mutation == 2 THEN
                  SET child = CALL Mutation 3(parent)
              END IF
              IF random_mutation == 3 THEN
                  SET child = CALL Mutation 4(parent)
              END IF
              IF random_mutation == 4 THEN
                  SET child = CALL Mutation 5(parent)
              END IF
              IF random_mutation == 5 THEN
                  SET child = CALL Mutation 6(parent)
              END IF
              IF random_mutation == 6 THEN
                  SET child = CALL Mutation 7(parent)
              END IF
        END IF
END FOR
RETURN (child)
```

```
Mutation 2 (mutated child)
SET cutout to empty list
COPY child to mutated child
FOR sub-route in mutated child
        SET i = 0
        FOR i to length of sub-route – 1
                SET distance = distance from i to next node of i
                IF random float < distance / 100 THEN
                        EXTEND nodes from index i+1 to cutout
                        DELETE nodes from index i+1
                END IF
        END FOR
APPEND cutout to mutated child
RETURN (mutated child)
```

```
Mutation 3 (mutated child)
UNLIST the mutated child route
SET mutated child to empty list
SHUFFLE mutated child with 0.1 probability
STRUCTURE mutated child according to constraints to form the sub-route structure again.
RETURN (mutated child)
```

```
Mutation 4 (mutated child)
UNLIST the mutated child route
FOR i in mutated child
        IF random probability < 0.02 THEN
        REMOVE i from mutated child
        APPEND i to mutated child
END FOR
STRUCTURE mutated child according to constraints to form the sub-route structure again.
RETURN (mutated child)
```

```
Mutation 5 (mutated child)
FOR sub-route in mutated child
    SHUFFLE sub-route
END FOR
RETURN (mutated child)
```

```
Mutation 6 (mutated child)
FOR sub-route in mutated child
    FOR i in sub-route
        IF random probability < 0.3 THEN
            SET Cur to index of i in sub-route
            SET Rs to random integer from -2 to 2
            SET Off = Cur + Rs
            IF Off < 0 THEN
                SET Off = 0
            ELSE IF Off > length or sub-route – 1 THEN
                SET Off = length or sub-route – 1
            END IF
            IF Cur != Off THEN
                M[Cur], M[Off] = M[Off], M[Cur]
            END IF
    END FOR
END FOR
RETURN (mutated child)
```

```
Mutation 7 (mutated child)
FOR sub-route in mutated child
    IF random integer from 0 to 3 == 0 and length of sub-route > 1:
            SET Rs to random integer from - length of sub-route to length of sub-
            route
            sub-route = second half portion of sub-route from index Rs + first half
            portion of sub-route to index Rs
    END IF
END FOR
RETURN (mutated child)
```

3.2 Parameter Settings Used in the Proposed GA

Size of individual = 100
Size of population = 10000
Number of predefined generations = 500
Probability of crossover = 0.8
Probability of mutation = 0.5

The algorithm was run firstly as 500 generations. To optimize the computing power, the number of generations are limited if there is no change in the solution after the current hundredth generation. Figure 3.3, 3.4 and 3.5 shows C1, R1 and RC1 have been found the optimal solution by 300, 400 and 400 generations respectively. Figure 3.6, 3.7 and 3.8 shows C2, R2 and RC2 have been found the optimal solution by 100 generations.

Narrow Time Windows Datasets

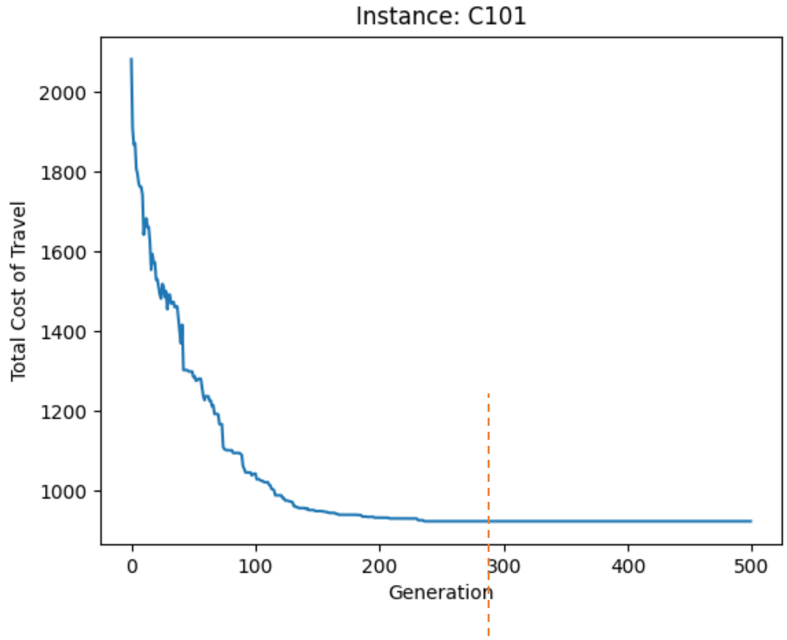

Figure 3.3 Trimming the generation span of C1 at 300.

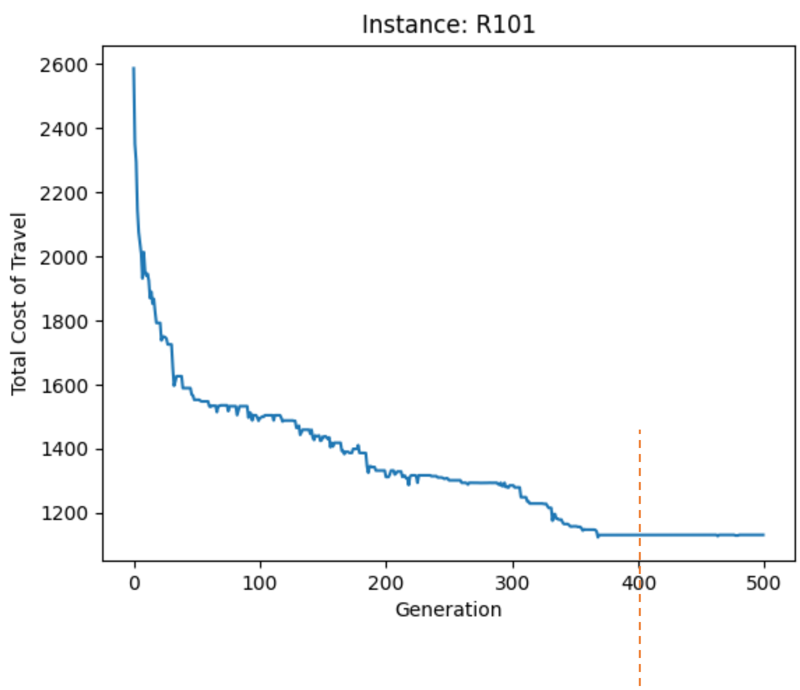

Figure 3.4 Trimming the generation span of R1 at 400.

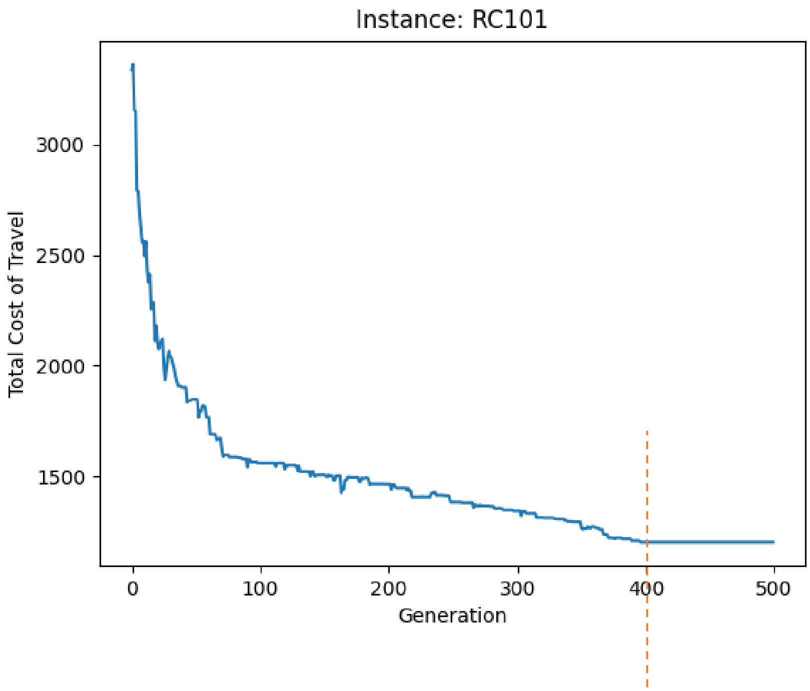

Figure 3.5 Trimming the generation span of RC1 at 400.

Wide Time Windows Datasets

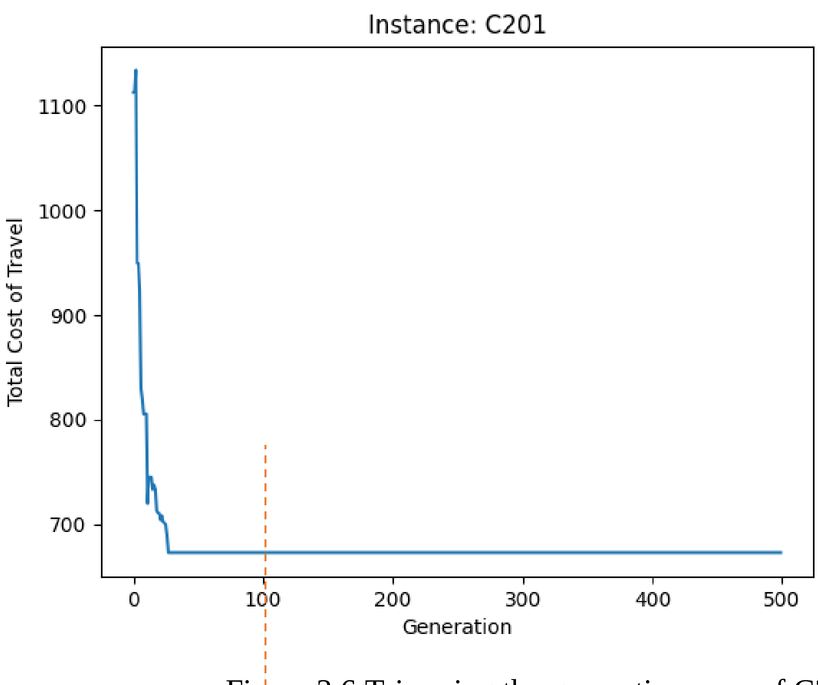

Figure 3.6 Trimming the generation span of C2 at 100.

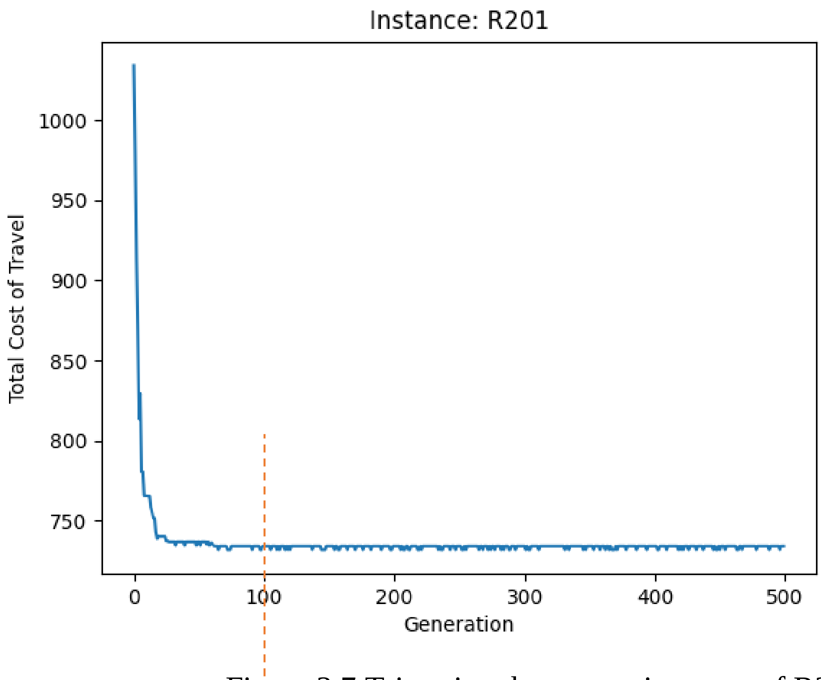

Figure 3.7 Trimming the generation span of R2 at 100.

Figure 3.8 Trimming the generation span of R2 at 100.

CHAPTER 4 RESULTS

This chapter describes the experiment results of the proposed GA on the Solomon benchmark datasets of narrow time window instances such as C1, R1 and RC1, as well as on the wide time window instances such as C2, R2, and RC2. Section 4.1 shows the performance comparison of the proposed GA and the best-known solutions. Section 4.2 is about the performance comparison of the seven different mutations. And section 4.3 concludes with the route structure visualizations of the results.

4.1 Comparison of the Proposed GA with Best-known Solutions

Results obtained from the proposed algorithm are compared with the best-known results that came from the previous best published studies [50] and [51] on VRPTW as shown in Table 4.1 and Table 4.2. The results of the proposed GA are averaged in 3 runs. It is shown that the proposed algorithm has better performance on R sets in which the customer locations are randomly generated and RC sets where the mixture of R and C. Although the performance on C sets is still approaching to the optimum, it has the competitive results with the other algorithms. The bolder results show the best performance in minimization of the number of vehicles, or the total distance traveled.

Table 4.1 Comparison of the algorithm performance on narrow time windows instances of Solomon Benchmark

Instance	Best-known optimal results from previous studies		Ref.	Proposed GA	
	Vehicle	**Distance**		**Vehicle**	**Distance**
C101	10	829	[51]	10	882
C102	10	829	[50]	10	878
C103	10	829	[51]	10	876
C104	10	829	[50]	10	872
C105	10	829	[50]	10	878
C106	10	829	[50]	10	875
C107	10	829	[50]	10	901
C108	10	829	[50]	10	888
C109	10	829	[50]	10	889
R101	19	1650	[51]	**11**	**1141**
R102	17	1426	[51]	**11**	**1234**
R103	13	1237	[51]	**11**	**1232**
R104	10	1010	[50]	10	1234
R105	15	1472	[50]	**11**	**1317**
R106	12	1274	[50]	**12**	**1268**
R107	12	1273	[50]	**10**	**1273**
R108	10	960	[50]	10	1231
R109	12	1212	[50]	**11**	**1185**
R110	11	1146	[50]	**10**	1270
R111	11	1133	[50]	13	1212
R112	10	989	[50]	12	1176
RC101	15	1676	[50]	**12**	**1358**
RC102	13	1536	[50]	**11**	**1291**
RC103	12	1230	[50]	**11**	1315

Table 4.1 Comparison of the algorithm performance on narrow time windows instances of Solomon Benchmark

Instance	Best-known optimal results from previous studies		Ref.	Proposed GA	
	Vehicle	**Distance**		**Vehicle**	**Distance**
RC104	10	1154	[50]	10	1242
RC105	14	1623	[50]	**11**	**1313**
RC106	12	1441	[50]	**11**	**1275**
RC107	11	1272	[50]	**11**	**1250**
RC108	10	1142	[50]	10	1292

Table 4.2 Comparison of the algorithm performance on wide time windows instances of Solomon Benchmark

Instance	Best-known optimal results from previous studies		Ref.	Proposed GA	
	Vehicle	Distance		Vehicle	Distance
C201	3	591	[50]	3	672
C202	3	591	[50]	3	671
C203	3	591	[50]	3	672
C204	3	596	[50]	3	672
C205	3	588	[50]	3	670
C206	3	588	[50]	3	671
C207	3	588	[50]	3	671
C208	3	588	[50]	3	672
R201	4	1276	[50]	**2**	**730**
R202	4	1087	[50]	**2**	**743**
R203	3	952	[50]	**2**	**722**
R204	3	761	[51]	**2**	**738**
R205	3	1036	[50]	**2**	**731**
R206	3	921	[50]	**2**	**729**
R207	3	821	[50]	**2**	**736**
R208	3	738	[50]	**2**	**732**
R209	3	929	[50]	**2**	**734**
R210	3	984	[50]	**2**	**731**
R211	3	786	[50]	**2**	**731**
RC201	4	1438	[50]	**2**	**712**
RC202	4	1182	[50]	**2**	**714**
RC203	3	1078	[50]	**2**	**707**
RC204	3	810	[50]	**2**	**721**
RC205	4	1334	[50]	**2**	**718**

Table 4.2 Comparison of the algorithm performance on wide time windows instances of Solomon Benchmark

Instance	Best-known optimal results from previous studies		Ref.	Proposed GA	
	Vehicle	Distance		Vehicle	Distance
RC206	3	1203	[50]	**2**	**712**
RC207	3	1093	[50]	**2**	**712**
RC208	3	913	[50]	**2**	**714**

Figure 4.1 shows the performance comparison of the proposed GA and BKS on each of C sets. As shown in the figure, BKS has slightly better solutions than the proposed GA on C sets. On the instances of C1, BKS has the constant value while the proposed GA has some fluctuations. The behavior shows the proposed GA explores the global optimum while the BKS has found the one. For the case of C2 instances, the proposed GA stands still as the BKS although the values are a bit greater than the latter. Per section 4.2, this happens because while the BKS used the mutation 1 alone with its specific crossover, the proposed GA randomly chooses one mutation from the seven different mutations for a generalization.

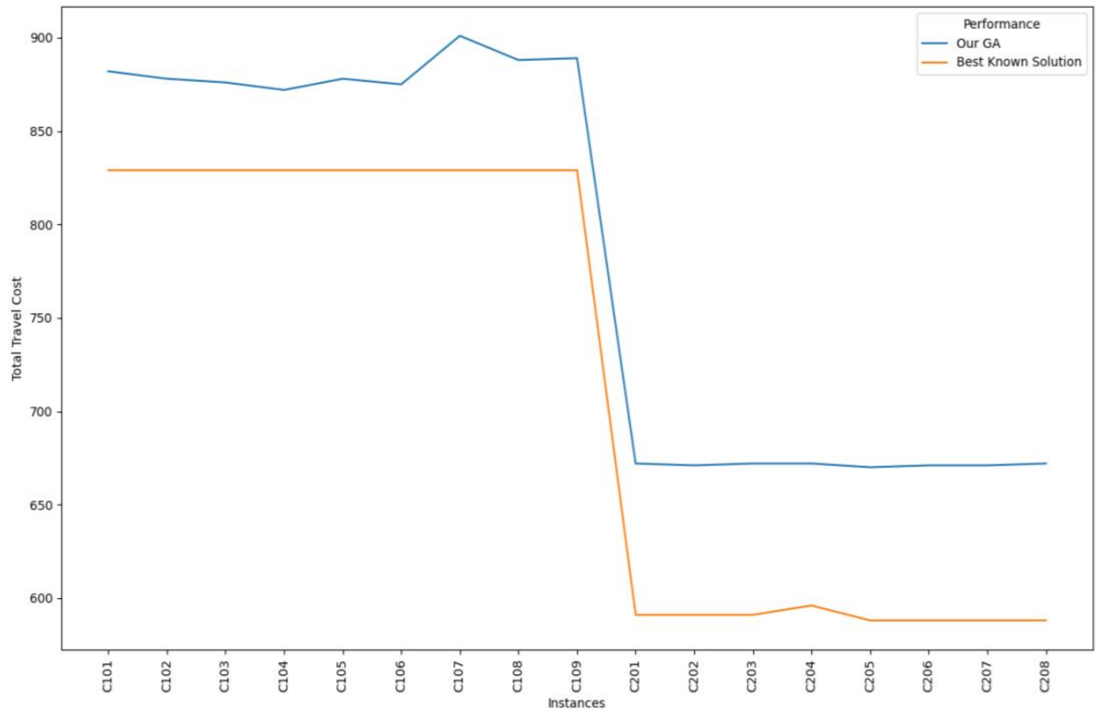

Figure 4.1 Performance Comparison on C sets

Figure 4.2 shows the performance comparison of the proposed GA and BKS on each of R sets. As shown in the figure, the proposed GA have some better solutions than BKS in R1 sets. For R2 sets, all the solutions that the proposed GA found are better than the solutions from BKS. Per section 4.2, this happens because of the nature of randomly choosing one mutation from seven different mutations. Although this nature has few vulnerabilities for C sets as mentioned above, it has outstanding scores on R sets because it maximizes the diverse exploration of GA on the randomly generated customers. There is also an effect of mutation 2 which is the heuristic-guided mutation, useful when the customer points are random and need some heuristic information for the precise exploration.

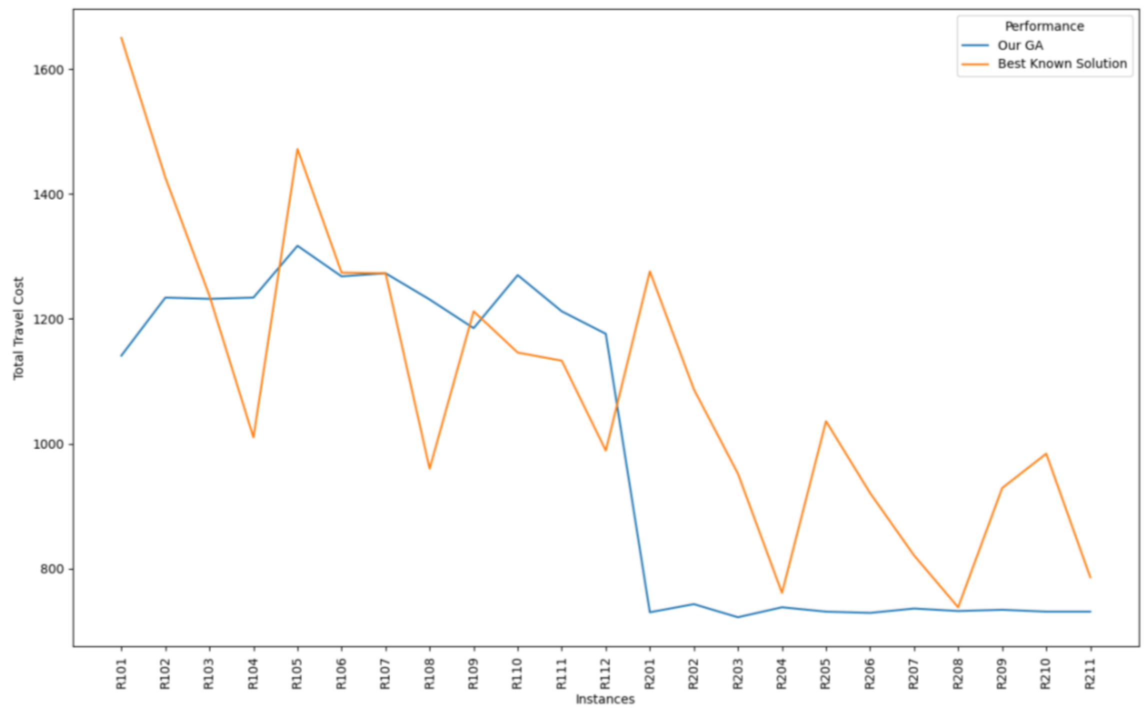

Figure 4.2 Performance Comparison on R sets

Figure 4.3 shows the performance comparison of the proposed GA and BKS on each of RC sets. As shown in the figure, on the instances of RC1, the proposed GA has overall better solutions except very few points have the better solutions of BKS. On the RC2 sets, all the instances show that the proposed GA has better solutions than BKS. This overall behavior shows the nature of randomly choosing one mutation from seven different mutations. RC is a set that is a mixture of R and C. This nature has few weaknesses in C but strong on R, so the result on RC shows it is a mixture of strength and weakness of the proposed GA.

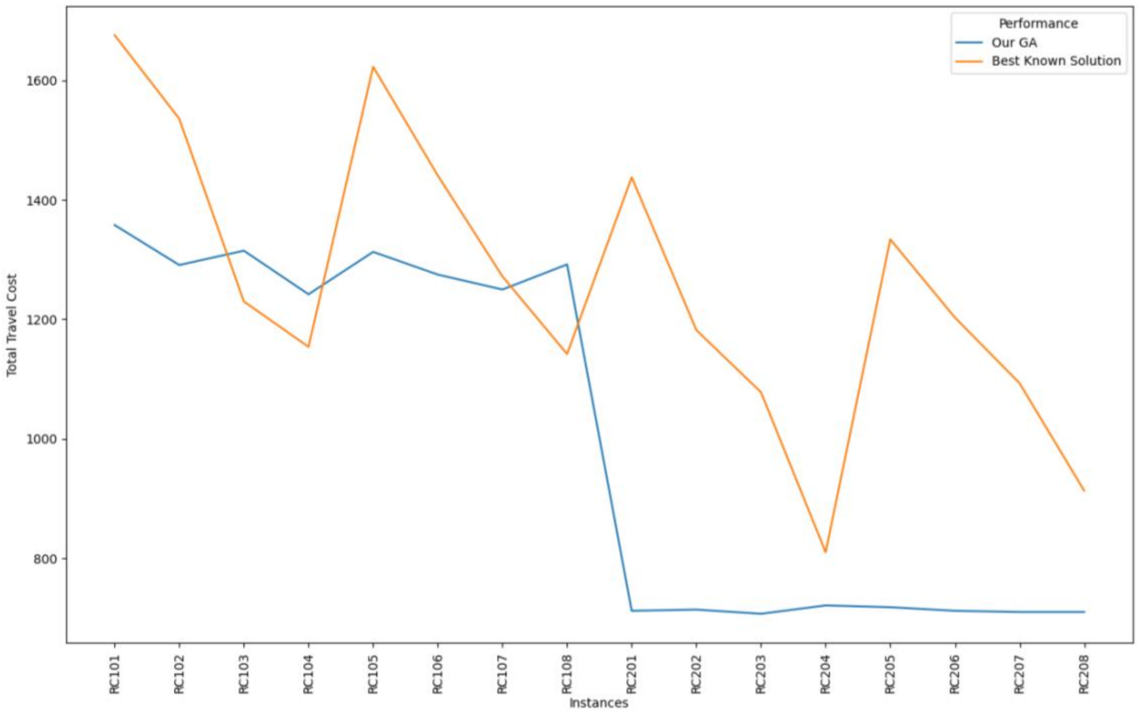

Figure 4.3 Performance Comparison on RC sets

In this GA, Remove-Insert Reverse Crossover and the seven types of mutations are only used. On the other hand, BKS used the route repair scheme with their crossover before mutation. Finally, the overall performance comparison of the proposed GA and BKS on each instance of Solomon Benchmark 100 customers is described in Figure 4.4 as follow:

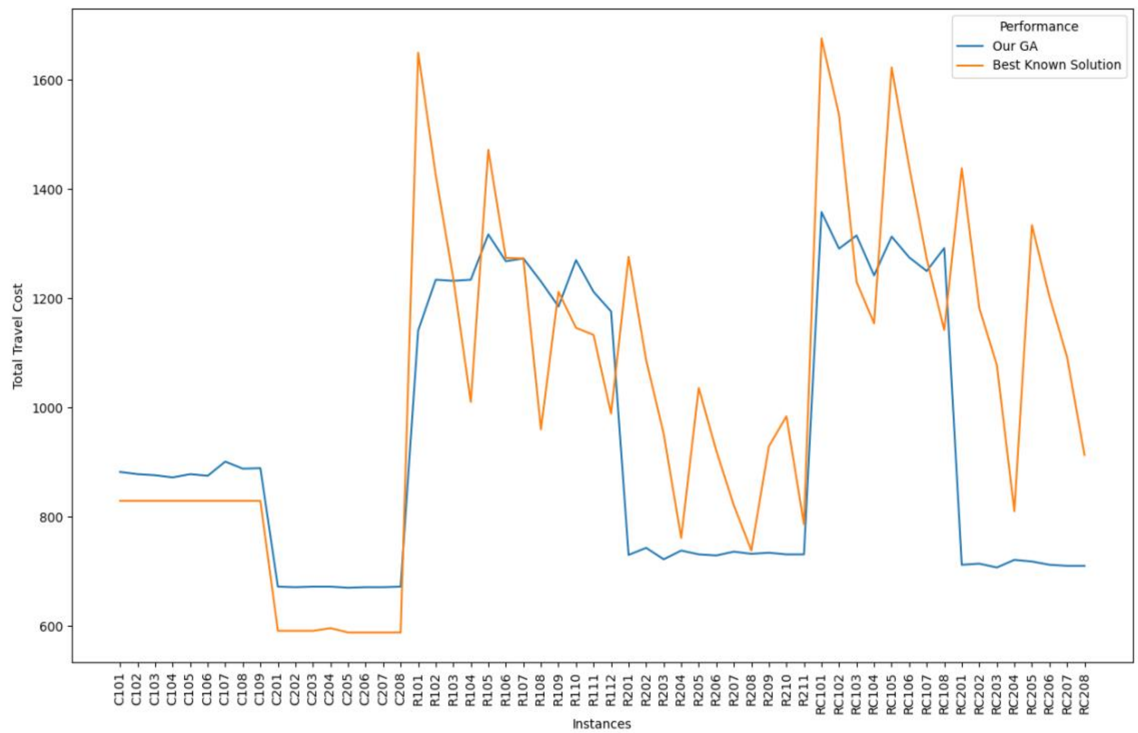

Figure 4.4 Overall Performance Comparison

The gap between the results of the proposed GA and Best-Known Solutions (BKS) is shown in Table 4.3. Results of C1 represents the average of all C1 instances. Same applies to C2, R1, R2, RC1, and RC2. It describes that BKS is slightly better than our proposed GA in C sets. BKS solutions are 6.41% and 13.77% better than the proposed GA on C1 and C2 respectively. For R1 and R2, the proposed GA is 0.06% and 21.71% better than the BKS. Also, for RC1 and RC2, the proposed GA is 6.66% and 36.98% better than the BKS. The percentage values described in the Table 4.3 are calculated as Percent Error = [(|BKS value – Proposed GA value|) / BKS value] * 100. Again, average performance comparison on C1, C2, R1, R2, RC1, and RC2 are shown in a bar chart in Figure 4.5.

Table 4.3 Percentage of Error Between Proposed GA and BKS

	Our GA	**BKS**	**Percent Error**
C1	882.11	829	6.41%
C2	671.38	590.13	13.77%
R1	1231.08	1231.83	−0.06%
R2	732.45	935.55	−21.71%
RC1	1292	1384.25	−6.66%
RC2	713	1131.38	−36.98%

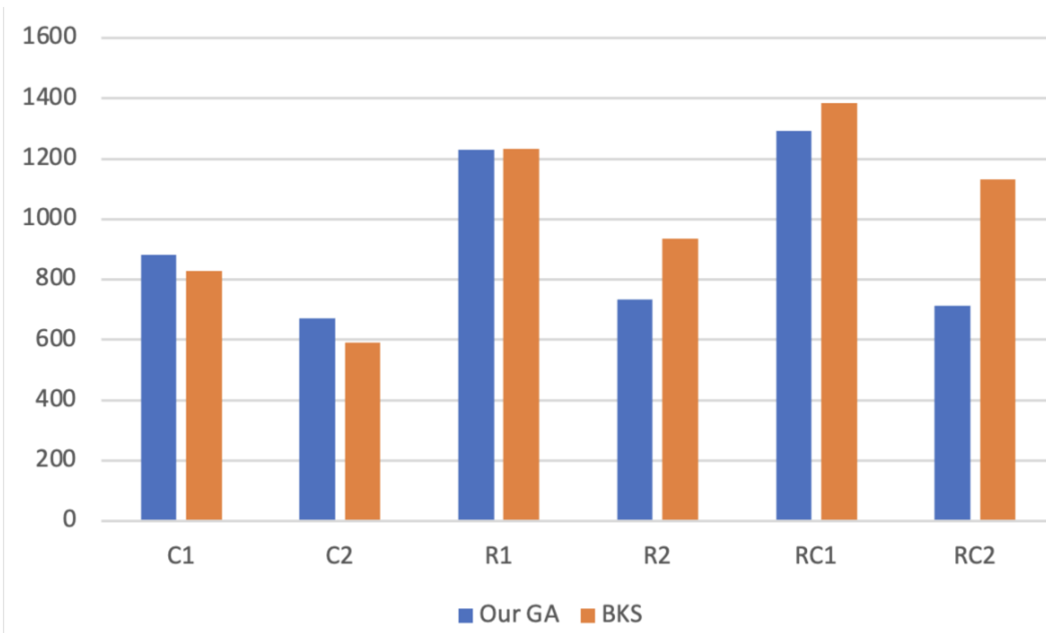

Figure 4.5 Overall Performance Comparison on Average of Instances

4.2 Performance Comparison of Seven Different Mutations

Since the proposed GA used seven different mutations depending on the random probability, this following section analyses the performance of each mutation operator on the C sets, R sets and RC sets. Both of Figure 4.6 and Figure 4.7 describes the separate performance of the mutations in heatmap and line chart respectively. M1 to M7 represents the mutation 1 to mutation 7.

There is no doubt that all wide time window instance groups such as C2, R2 and RC2 have lower travel cost with any mutation compared to narrow time window instances C1, R1 and RC1. That is because unlike narrow time window instances, wide time window instances have less chance to violate time constraint.

The two figures show mutation 1 and mutation 6 can achieve the desirable results among seven mutations. From the perspective of time windows, mutation 1 has better performance on 70% of narrow time window instances: R1 and RC1 while mutation 6 has better performance on 70% of wide time window instances: C2 and RC2. Again, from the perspective of customer locations, mutation 1 and mutation 2 has outstanding results on R sets while mutation 6 has more influence on the results of C and RC sets.

Therefore, for the overall conclusion of this section, mutation 1 is good on randomly generated customer locations and narrow time window instances. That shows that Constrained Route Reversal Mutation is a good choice when the random customers have tight time constraints because this kind of mutation saves the route corruption as it only reverses the order of 2 to 3 customers in the chosen sub-route. On the other hand, mutation 6 is good on clustered customer locations and wide time window instances. This shows the situation when the customers are clustered and, they have loose time constraints, swapping two customers that are two offsets away in the same sub-route under the low probability rate can balance between exploitation and exploration as the customers are not too close but also just two offsets away.

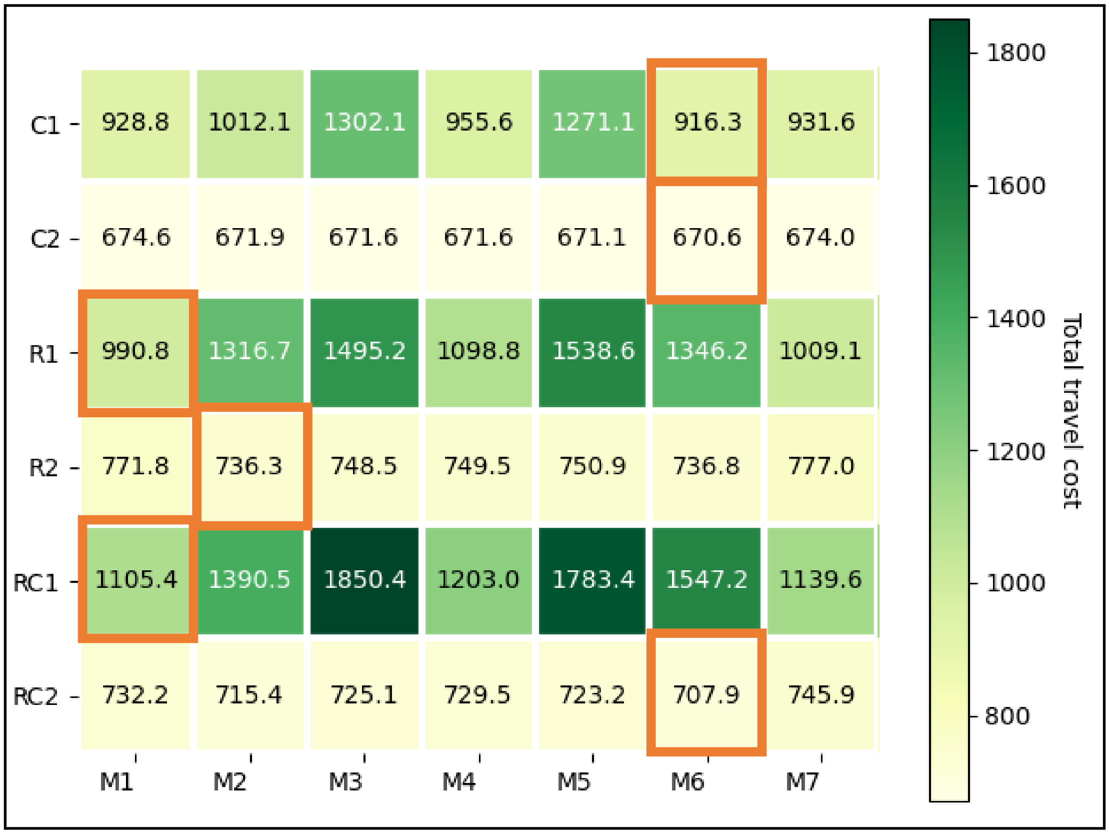

Figure 4.6 Performance of Seven Different Mutations in Proposed GA (Heatmap)

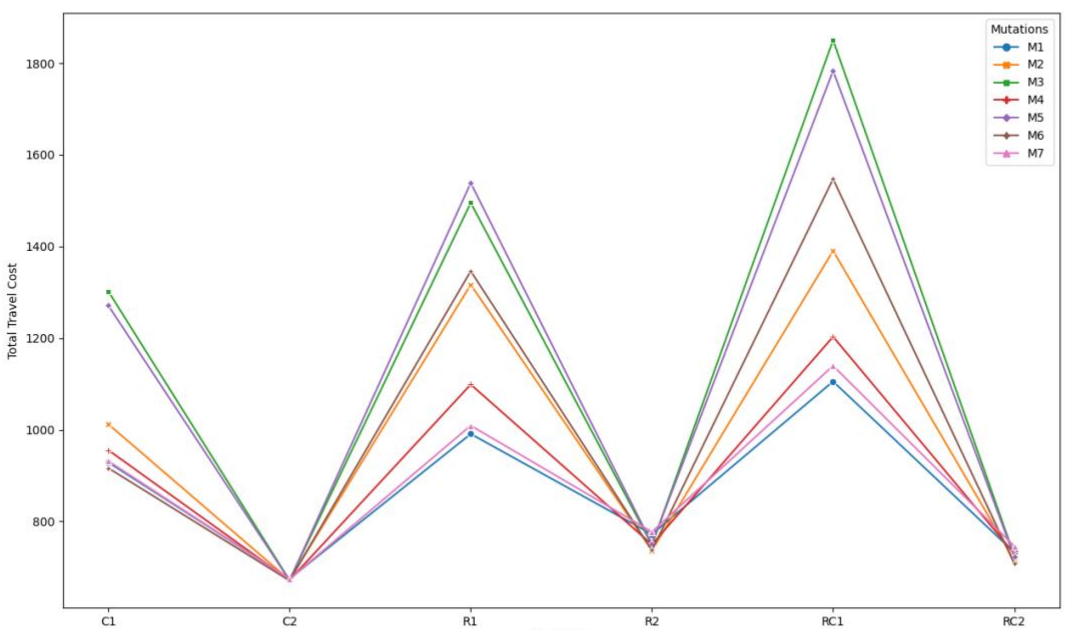

Figure 4.7 Performance of Seven Different Mutations in Proposed GA (Line Chart)

4.3 Route Structure Visualization

Figure 4.8 and Figure 4.9 display the example vehicle topology of C sets: C103 and C205 respectively. As they have clustered customer locations, the route structures show the specific pattern of groups. C1 sets have narrow time windows so they use more vehicles than C2 sets. Contrast, C2 sets use only few vehicles compared to C1 as they have wider time frame.

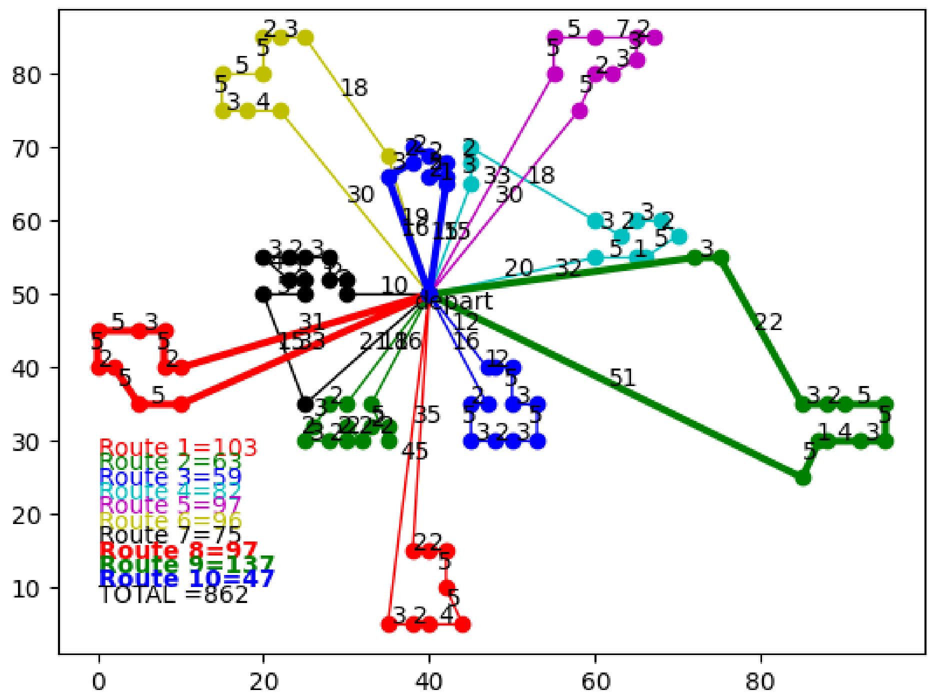

Figure 4.8 Vehicle Topology of instance C103.

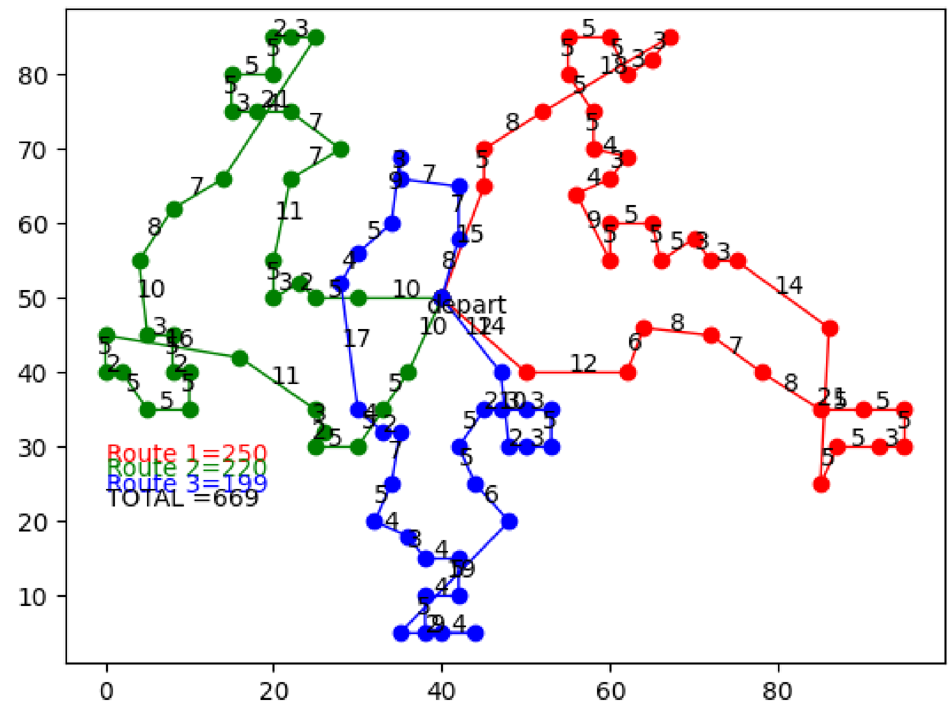

Figure 4.9 Vehicle Topology of instance C205.

Figure 4.10 and Figure 4.11 display the example vehicle topology of R sets: R103 and R203 respectively. As they have randomly generated customer locations, the route structures do not have the specific pattern of groups. R1 sets have narrow time windows so they use more vehicles than R2 sets. Meanwhile, R2 sets use fewer vehicles than R1 as they have wide time windows.

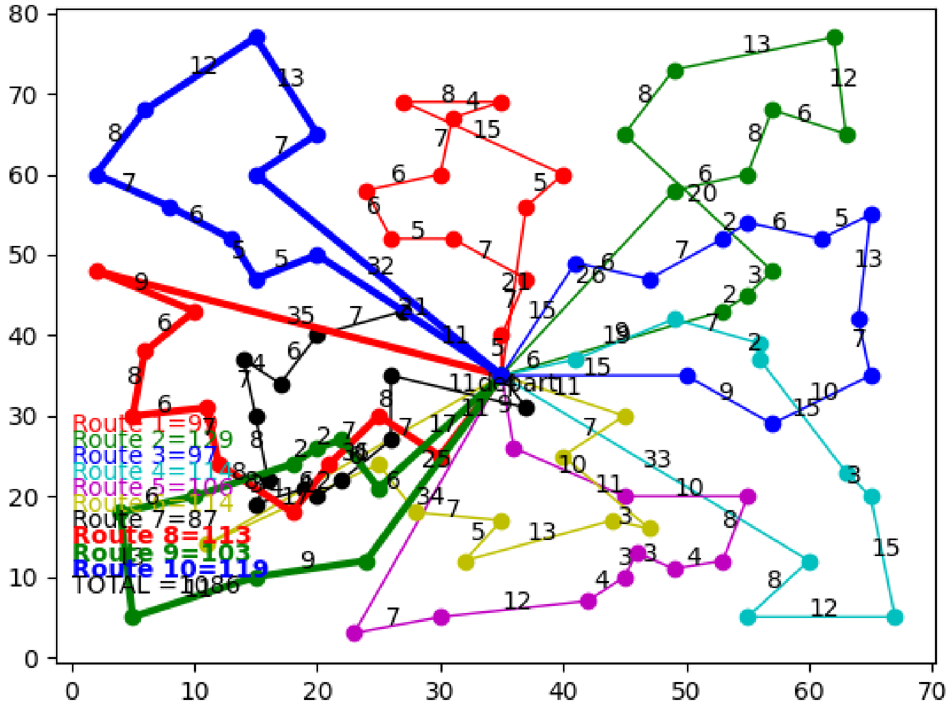

Figure 4.10 Vehicle Topology of instance R103.

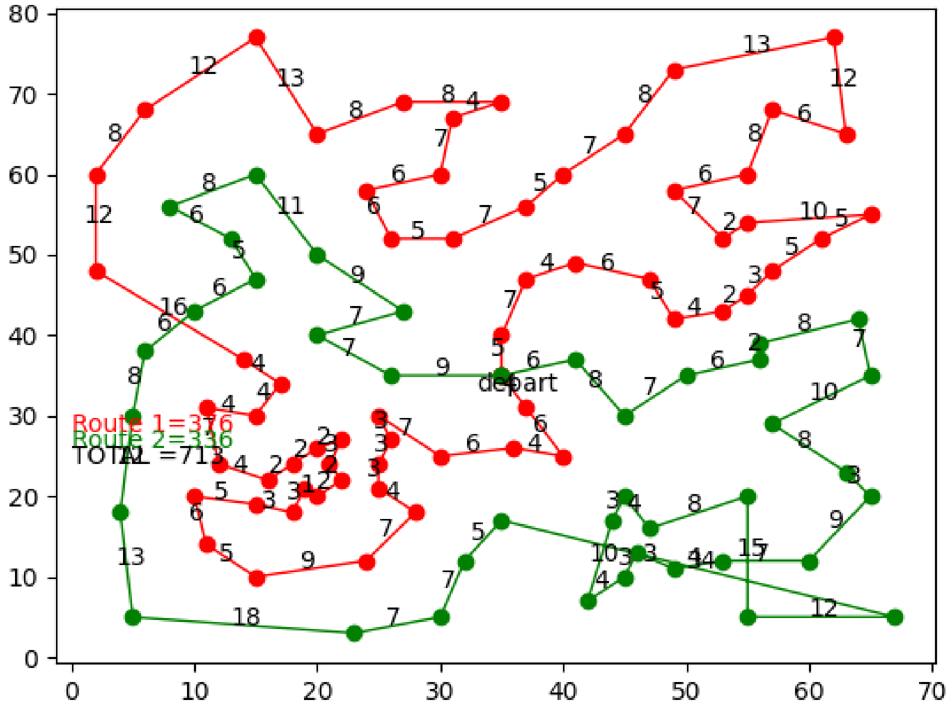

Figure 4.11 Vehicle Topology of instance R203.

Figure 4.12 and Figure 4.13 display the example vehicle topology of RC sets: RC106 and RC207 respectively. Since RC sets are the mixture of R and C sets, randomly generated customer locations and clusters, the route structures are not much grouped as in C but not too scattered like in R. R1 sets have tight time constraints, so they must use more vehicles than R2 sets. Meanwhile, R2 sets use fewer vehicles than R1 as they have loose time constraints.

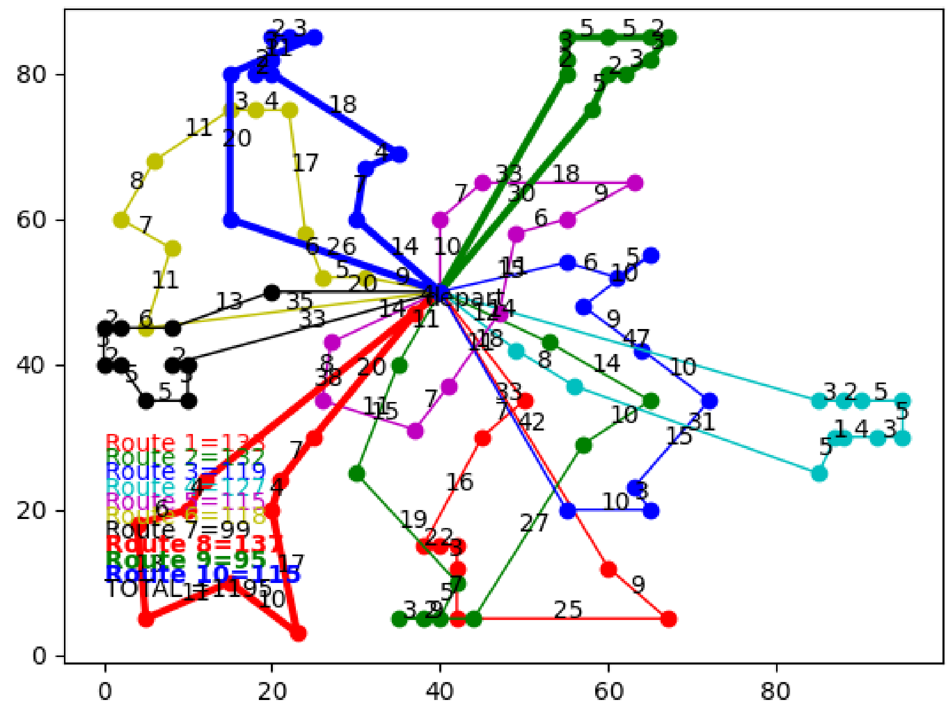

Figure 4.12 Vehicle Topology of instance RC106.

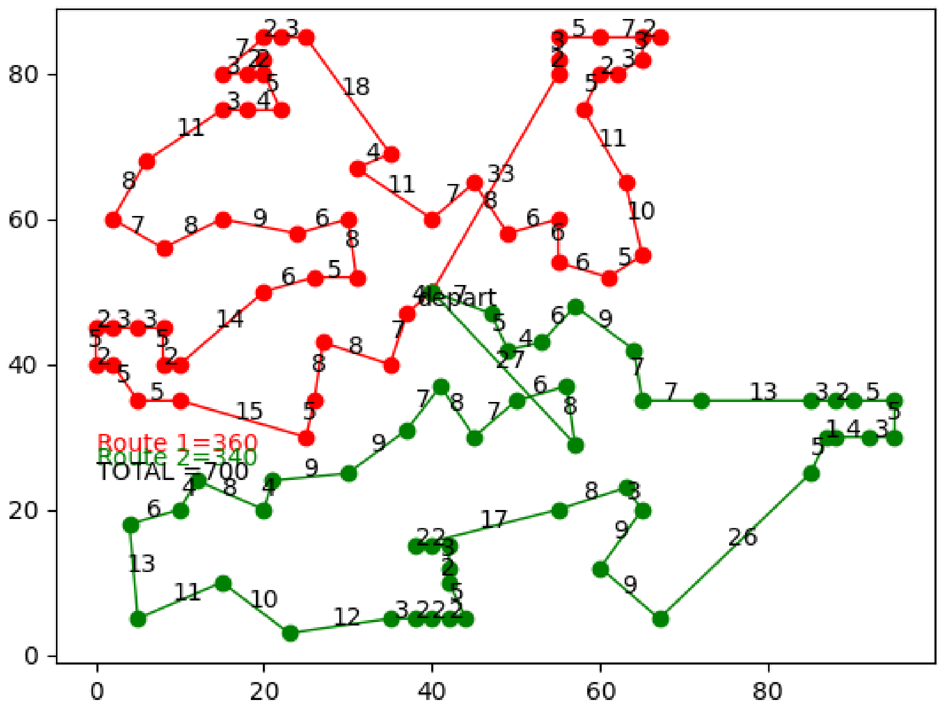

Figure 4.13 Vehicle Topology of instance RC207.

To conclude this section 4, the overall section visualizes the route structure examples taken from each set of C1, C2, R1, R2, RC1 and, RC2. The distance travelled between each customer point in the route is calculated as Euclidean distance defined by Solomon Benchmark. The visualizations show that more vehicles are needed to solve narrow time window problems (C1, R1 and RC1) regardless of the customer locations while the wide time window problems (C2, R2 and RC2) can solve with few vehicles. That is because narrow time window customers have tight time constraints. If the vehicle violates such one constraint at a customer point, the customer cannot take that vehicle and need to allocate to another vehicle which will satisfy the customer time constraint.

CHAPTER 5 CONCLUSION AND DISCUSSION FOR FUTURE WORK

5.1 Conclusion

In this report, state-of-the-art reviews on VRP variants and solution approaches has been discussed in chapter 2. It shows that the time windows variant of VRP are the fundamental variants in the real-world case studies, moreover, in the solution approaches, GA still stands as one of the mostly used solution approaches among the classic and state-of-the-art approaches. This phenomenon describes the fact that the research of VRPTW with GA has not been dead as the new combination of genetic operators can be found as described in chapter 3 of this report.

To conclude this research report, the newly designed crossover which is Remove-Insert Reverse Crossover and the randomly chosen seven different mutations have been discussed for the purpose of implementing new crossover operation and generalization of mutation operations which gives better results on randomly generated customer points, R sets and a mixture of clustered and randomly generated customer points, RC sets. Some experimental results are significantly improved as compared with the best-known previous work. It has shown that adding the provided heuristic information to the mutation phase contributes to the performance of the proposed GA efficiently on the R and RC sets. The proposed GA is appropriate for real-world case studies like delivery systems which have customers from random locations rather than the clustered groups.

5.2 Discussion for future work

As mentioned above, the proposed GA has significant optimal performance on R and RC sets while it can also give competitive results on the C set. This phenomenon occurs as the randomness nature of GA on various thresholds in most of the steps of the proposed operators is used rather than constraining on the specific conditions and also the nature of choosing different mutation operations at a time depending on the random probability. In the future, the performance of the algorithm can be extended by integrating with other heuristics or meta-heuristics. As the trend of hybrid algorithms is also emerging nowadays, future contributions can be considered to modify the proposed GA into hybrid GA for obtaining the optimal solutions in a short time with limited resources.

The well-known VRPTW real-world use cases which can be found from the previous reports consist of cash delivery for the banks [52], waste collection for the environment [53], delivery for the fast-food business [54] and courier service [55]. In the real-world VRP systems, VPRTW use cases are the fundamental and crucial case studies for most of the VRP variants. Therefore, in the future, the proposed algorithm is expected to be applied in one of these real-world scenarios but intended for different variant like Multiple Depots VRPTW problems.

REFERENCES

1. En.wikipedia.org. Vehicle routing problem. [Online], Available at: https://en.wikipedia.org/wiki/Vehicle_routing_problem [2020, October 8].

2. Dantzig, G. B. and Ramser, J. H., 1959, "The Truck Dispatching Problem", **Management Science**, Vol. 6, No. 1, pp. 80–91.

3. Smail, S. N., Ku-Mahamud, K. R. and Abdul-Rahman, S, 2017, “A Review on Delivery Routing Problem and Its Approaches”, **Journal of Theoretical and Applied Information Technology**, Vol. 95, No. 2, pp. 367-380.

4. Kim, G., Ong, Y. S., Heng, C. K., Tan, P. S. and Zhang, N. A, 2015, “City Vehicle Routing Problem (City VRP): A Review”, **IEEE Transactions on Intelligent Transportation Systems**, Vol. 16, No. 4, 1654–1666.

5. Soonpracha, K., Mungwattana, A., Janssens, G. K. and Manisri, T, 2014, “Heterogeneous VRP Review and Conceptual Frameworks”, **Lecture Notes in Engineering and Computer Science**, Vol. 2210.

6. Onut, S., Kamber, M. R., and Altay, G, 2014, “A Heterogeneous Fleet Vehicle Routing Model for Solving the LPG Distribution Problem: A Case Study”, **Journal of Physics: Conference Series**, Vol. 490, No. 1.

7. Panicker, V. V. and Ihsan O, M, 2018, “Solving a Heterogeneous Fleet Vehicle Routing Model- A Practical Approach”, **2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA 2018)**, 6-7 July 2018, Pondicherry, India.

8. Kumar, S. and Panneerselvam, R., 2012, “A Survey on the Vehicle Routing Problem and Its Variants”, **Intelligent Information Management**, Vol. 4, No.3, pp. 66-74.

9. Bello, I., Pham, H., Le, Quoc V., Norouzi, M. and Bengio, S., 2017, “Neural Combinatorial Optimization with Reinforcement Learning”, **5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings**. (Under Review)

10. Nazari, M., Oroojlooy, A., Takáč, M. and Synder, Lawrence V., 2018, “Reinforcement learning for solving the vehicle routing problem”, **Advances in Neural Information Processing Systems**, December 2018, pp. 9839-9849.

11. Awad, H., Elshaer, R., AbdElmo'ez, A. and Nawara, G., 2018, “An Effective Genetic Algorithm for Capacitated Vehicle Routing Problem”, **Proceedings of the International Conference on Industrial Engineering and Operations Management**, 6-8 March 2018, Bandung, Indonesia, pp. 374-384.

12. Azi, N., Gendreau, M. and Potvin, J.-Y, 2010, “An Exact Algorithm for a Vehicle Routing Problem with Time Windows and Multiple Use of Vehicles”, **European Journal of Operational Research**, Vol. 202, No. 3, pp. 756–763.

13. Subramanian, A., Uchoa, E. and Ochi, L. S, 2013, “A Hybrid Algorithm for a Class of Vehicle Routing Problems”, **Computers & Operations Research**, Vol. 40, No. 10, pp. 2519–2531.

14. Liong, C.-Y, Wan, I., Omar and Khairuddin, 2008, “Vehicle Routing Problem: Models and Solutions”, **Journal of Quality Measurement and Analysis**, Vol. 4, pp. 205-218.

15. Taner, F., Galić, A., and Carić, T., 2012, “Solving Practical Vehicle Routing Problem with Time Windows Using Metaheuristic Algorithms”, **PROMET - Traffic&Transportation**, Vol. 24.

16. Ree, S. and Yoon, B. S., 1996, “A Two-Stage Heuristic Approach for the Newspaper Delivery Problem”, **Computers & Industrial Engineering**, Vol. 30, No. 3, pp. 501–509.

17. Repoussis, P., Tarantilis, C., and Ioannou, G., 2008, “An Evolutionary Algorithm for the Open Vehicle Routing Problem with Time Windows”, **Studies in Computational Intelligence**, Vol. 161, pp. 55-75.

18. Yao, B., Hu, P., Zhang, M., and Tian, X., 2014, “Improved Ant Colony Optimization for Seafood Product Delivery Routing Problem”, **PROMET - Traffic&Transportation**, Vol. 26, No. 1, pp. 1-10.

19. Dong, L. W. and Xiang, C. T., 2006, “Ant Colony Optimization for VRP and Mail Delivery Problems”, **4th IEEE International Conference on Industrial Informatics**, 16-18 Aug 2006, Singapore, pp. 1143-1148.

20. Garcia-Najera, A. and Bullinaria, J. A, 2011, “An Improved Multi-Objective Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows”, **Computers & Operations Research**, Vol. 38, No. 1, pp. 287–300.

21. Yan, X., Zhang, C. and Luo, W., 2012, “Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm”, **International Journal of Computer Science**, Vol. 9, No. 6, pp. 264–271.

22. Berger, J., Barkaoui, M. and Bräysy, O. A., 2003, "Route-Directed Hybrid Genetic Approach for the Vehicle Routing Problem with Time Windows", **INFOR**, Vol. 41, No. 2, pp. 179-194.

23. Vrp-rep.org. Home - VRP-REP: the vehicle routing problem repository. [Online], Available at: http://www.vrp-rep.org/ [2020, November 1].

24. Neo.lcc.uma.es. NEO: Networking and Emerging Optimization. [Online], Available at: https://neo.lcc.uma.es/index.html [2020, November 1].

25. Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R. and Parthiban, P., 2009, "Optimization of Multiple Vehicle Routing Problems using Approximation Algorithms", **IJEST**, Vol. 1, No. 3, pp. 129-135.

26. Rabbouch, B., Saâdaoui, F. and Mraihi, R., 2019, "Efficient Implementation of the Genetic Algorithm to Solve Rich Vehicle Routing Problems", **Operational Research**, Vol. 21, pp. 1763–1791.

27. Singh, Varimna, L., Ganapathy, Pundir and Ashok, 2019, "An Improved Genetic Algorithm for Solving Multi Depot Vehicle Routing Problems", **International Journal of Information Systems and Supply Chain Management**, Vol. 12, No. 4, pp. 1-26.

28. Zhang, Y. and Chen, X. D., 2014, "An Optimization Model for the Vehicle Routing Problem in Multi-Product Frozen Food Delivery", **Journal of Applied Research and Technology**, Vol. 12, No. 2, pp. 239–250.

29. Phiboonbanakit, T., Supnithi, T., Horanont, T. and Huynh, V., 2018, "Knowledge-Based Learning for Solving Vehicle Routing Problem", **The 2018 ACM International Joint Conference and 2018 International Symposium**, 8-12 Oct 2018, Singapore, pp. 1103-1111.

30. Wang, X., Xu, C. and Hu, X., 2008, "Genetic Algorithm for Vehicle Routing Problem with Time Windows and A Limited Number of Vehicles", **2008 International Conference on Management Science and Engineering 15th Annual Conference Proceedings**, 10-12 Sept 2008, Long Beach, CA, pp. 128-133.

31. Alipour, M. M., Razavi, S. N., Feizi Derakhshi, M. R. and Balafar, M. A, 2017, "A Hybrid Algorithm Using a Genetic Algorithm and Multi-agent Reinforcement Learning Heuristic to Solve the Traveling Salesman Problem", **Neural Computing and Applications**, pp. 2935–2951.

32. de Lima Junior, F. C., de Melo, J. D. and Neto, A. D. D., 2007, “Using Q-learning Algorithm for Initialization of the GRASP Metaheuristic and Genetic Algorithm”, **2007 International Joint Conference on Neural Networks**, 12-17 Aug 2007, Orlando, FL, pp. 1243-1248.

33. Liu, F. and Zeng, G., 2009, “Study of Genetic Algorithm with Reinforcement Learning to solve the TSP”, **Expert Systems with Applications**, Vol. 36, No. 3, pp. 6995–7001.

34. Niu, Y., He, J., Wang, Z. and Xiao, J., 2014, “A P-Based Hybrid Evolutionary Algorithm for Vehicle Routing Problem with Time Windows”, **Mathematical Problems in Engineering**, Vol. 2014, pp. 1–11.

35. Ghoseiri, K. and Gannadpour, S.F., 2009, “Hybrid Genetic Algorithm for Vehicle Routing and Scheduling Problem”, **2009 Asian Network for Scientific Information**, Vol. 9, No. 3, pp. 79-87.

36. Thangiah, S. R., Osman, I. H., and Sun, T., 1998, “Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows”, **Practical Handbook of Genetic Algorithms: Complex Coding Systems**, Vol. 3, pp. 347-376.

37. Thangiah, S. R., 1995, “Vehicle Routing with Time Windows using Genetic Algorithms”, **Application hanbook of genetic algorithms: new frontiers**, Vol. 2, pp. 253-277.

38. Solomon, M., 1987, “Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints”, **Oper. Res.**, Vol. 35, pp. 254-265.

39. Phannikul, T. and Sindhuchao, S., 2010, “A Customized Tabu Search for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery”, **Thammasat International Journal of Science and Technology**, Vol. 15, No. 2, pp. 70–82.

40. Ji, Y., Yang, H. and Zhou, Y., 2015, “Vehicle Routing Problem with Simultaneous Delivery and Pickup for Cold-chain Logistics”, **2015 International Conference on Modeling, Simulation and Applied Mathematics**, 23-24 August 2015, Phuket, Thailand, pp. 70-74.

41. Li, P., He, J., Zheng, D., Huang, Y. and Fan, C., 2015, “Vehicle Routing Problem with Soft Time Windows Based on Improved Genetic Algorithm for Fruits and Vegetables Distribution”, **Discrete Dynamics in Nature and Society**, Vol.2015, pp. 1–8.

42. Vidal, T., Crainic, T. G., Gendreau, M. and Prins, C., 2014, “Time-Window Relaxations in Vehicle Routing Heuristics”, **Journal of Heuristics**, Vol. 21, No. 3, pp. 329–358.

43. Fu, Z., Eglese, R. and Li, L. Y. O., 2008, “A Unified Tabu Search Algorithm for Vehicle Routing Problems with Soft Time Windows”, **Journal of the Operational Research Society**, Vol. 59, No. 5, pp. 663–673.

44. Holland J.H., 1975, “Adaptation in Natural and Artificial Systems”, **MITPress**, Cambridge, MA.

45. Marinakis, Y., and Marinaki, M, 2010, “A Hybrid Genetic – Particle Swarm Optimization Algorithm for the Vehicle Routing Problem”, **Expert Systems with Applications**, Vol. 37, No. 2, pp. 1446–1455.

46. Liu, X., Jiang, W. and Xie, J., 2009, “Vehicle Routing Problem with Time Windows: A Hybrid Particle Swarm Optimization Approach”, **2009 Fifth International Conference on Natural Computation**, 14-16 Aug 2009, Tianjin, pp. 502-506.

47. Sheng-Hua, X., Ji-Ping, L., Fu-Hao, Z., Liang, W. and Li-Jian, S., 2015, “A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem with Time Windows”, **Sensors**, Vol. 15, pp. 21033-21053.

48. Bhatt, 5 Things You Need to Know about Reinforcement Learning – Kdnuggets, [Online], KDnuggets. Available at: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html [2020 November, 24].

49. Web.cba.neu.ed. Benchmarking Problems, [online], Available at: http://web.cba.neu.edu/~msolomon/problems.htm [2020 December, 8].

50. Ombuki, B., Ross, B.J. and Hanshar, F., 2006, "**Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows**", Vol. 24, No. 1, pp. 17-30.

51. Vladimir, K., Ivan, M., Evgeny, K., Boris, T. and Ilya, R., 2016, "Approaches to Solve the Vehicle Routing Problem in the Valuables Delivery Domain", **Procedia Computer Science**, Vol. 88, pp. 487-492.

52. Boonsam, P., Suthikarnnarunai, N. and Chitphaiboon, W., 2011, “Assignment Problem and Vehicle Routing Problem for an Improvement of Cash Distribution”, **2011 Proceedings of the World Congress on Engineering and Computer Science**, 19-21 Oct 2011, San Francisco, USA, Vol. 2.

53. Byung-In Kim, Seongbae Kim and Surya Sahoo, 2006, “Waste collection vehicle routing problem with time windows”. **Comput. Oper. Res.**, Vol. 33, No. 12, pp. 3624–3642.

54. Russell, R.A, 1995, “Hybrid heuristics for the vehicle routing problem with time windows”, **Transportation Science**, Vol. 29, No. 2, pp.156–166.

55. Henggeler, K., 2015, “Route Optimizer”, **Thesis Book of BSc Geomatic Engineering and Planning**, Swiss Federal Institute of Technology Zurich, Institute of Cartography and Geoinformation, pp. 1 – 46.

56. Elshaer, R. and Awad, H., 2020, “A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants”, **Computers & Industrial Engineering**, Vol. 140.

57. Czuba, P., and Pierzchała, D., 2021, “Machine Learning methods for solving Vehicle Routing Problems”, **Proceedings of the 36th International Business Information Management Association (IBIMA)**, 4-5 Nov 2020, Granada, Spain, pp. 5-7.

58. Tan, S., and Wei-Chang, Y., 2021, "The Vehicle Routing Problem: State-of-the-Art Classification and Review" **Applied Sciences**, Vol. 11, No. 2.

59. De Jong, K.A. and Spears, W.M., 1992, “A formal analysis of the role of multi-point crossover in genetic algorithms”, **Ann Math Artif Intell**, Vol. 5, pp. 1–26.

60. Eiben, A., and Schippers, C., 1998, “On Evolutionary Exploration and Exploitation”, **Fundamenta Informaticae**, Vol. 35, No. 1-4, pp. 35-50.

APPENDIX

Instance	Cost 1	Cost 2	Cost 3	Avg
C101	875	883	888	882
C102	862	888	886	878
C103	883	862	883	876
C104	864	886	868	872
C105	880	875	881	878
C106	867	864	894	875
C107	900	906	898	901
C108	892	888	886	888
C109	877	885	906	889
C201	672	672	672	672
C202	672	672	670	671
C203	672	672	672	672
C204	672	672	672	672
C205	669	669	672	670
C206	672	672	671	671
C207	670	672	672	671
C208	672	672	672	672
R101	1209	1114	1100	1141
R102	1238	1222	1244	1234
R103	1210	1036	1452	1232

R104	1383	1177	1144	1234
R105	1226	1257	1468	1317
R106	1219	1351	1236	1268
R107	1353	1233	1233	1273
R108	1160	1375	1158	1231
R109	1239	1083	1234	1185
R110	1293	1294	1225	1270
R111	1138	1272	1226	1212
R112	1211	1094	1224	1176
R201	730	724	737	730
R202	751	733	746	743
R203	732	722	713	722
R204	733	744	739	738
R205	745	722	728	731
R206	733	724	731	729
R207	737	739	732	736
R208	732	743	723	732
R209	740	731	733	734
R210	730	727	738	731
R211	743	729	721	731
RC101	1258	1431	1387	1358

RC102	1316	1273	1284	1291
RC103	1315	1327	1303	1315
RC104	1199	1262	1265	1242
RC105	1251	1264	1425	1313
RC106	1195	1250	1381	1275
RC107	1201	1218	1331	1250
RC108	1338	1265	1273	1292
RC201	722	701	713	712
RC202	707	716	721	714
RC203	703	706	713	707
RC204	714	736	714	721
RC205	715	707	733	718
RC206	713	716	708	712
RC207	720	710	700	710
RC208	704	708	719	710

CURRICULUM VITAE

NAME	Miss Aye Thant May
DATE OF BIRTH	**10 August 1997**

EDUCATIONAL RECORD

HIGH SCHOOL	High School Graduation Practising High School, Yangon Institute of Education, 2013
BACHELOR'S DEGREE	Bachelor of Computer Science (Software Engineering) University of Information Technology, 2018
MASTER'S DEGREE	Master of Engineering (Computer Engineering) King Mongkut's University of Technology Thonburi, 2021
SCHOLARSHIP	Multi-Intellectuals Scholarship King Mongkut's University of Technology Thonburi, 2019
EMPLOYMENT RECORD	Software Engineer Intern Innovative Global Wave Technology (Yangon), 2018 Machine Learning Engineer Brillar Co. Ltd. (Novotel, Yangon Branch), 2018-2019
PUBLICATION	A. T. May, C. Jariyavajee and J. Polvichai, "An Improved Genetic Algorithm for Vehicle Routing

Problem with Hard Time Windows," **2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)**, 2021, pp. 1-6, doi: 10.1109/ICECET52533.2021.