UNIVERSITY OF THE WITWATERSRAND

MASTERS THESIS

Computational Logistics of the Vehicle Routing Problem with Time Windows

Author:
Krupa PRAG
Student number: 782681

Supervisors:
Dr Matthew WOOLWAY
Prof Byron JACOBS

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Computer Science and Applied Mathematics

Declaration of Authorship

I, Krupa PRAG
Student number: 782681, declare that this thesis titled, "Computational Logistics of the Vehicle Routing Problem with Time Windows" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF THE WITWATERSRAND

Abstract

Science
Computer Science and Applied Mathematics

Master of Science

Computational Logistics of the Vehicle Routing Problem with Time Windows

by Krupa PRAG
Student number: 782681

In the field of Operations Research (OR), optimising the logistics of the distribution problem is of considerable interest, given its prevalence and importance in today's interconnected and globalised world. The procedure of servicing a set of customers within their respective specified time windows, whilst minimising both the number of dispatched vehicles and total travelled distance is known as the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW is classified as both a Combinatorial Optimisation Problem (COP) and Multi-objective Optimisation Problem (MOP), as well as, is categorised to be a part of the $\mathcal{NP}-\mathcal{H}$ class. Identifying that a fair comparison of the performance of applied solution techniques cannot directly be made, as the results recorded in the literature are not generated under standardised experimental conditions, a comparative review was conducted. In this research, two metaheuristic solution techniques, Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) algorithm, are comparatively reviewed. After conducting experiments on the same hardware and software, the results record the procured solution's routes, total travelled distance, accumulated wait time and CPU time. The results were studied to draw conclusions and highlight correlations amongst the dataset types, applied solution techniques and solution evaluation metrics. Comparing the obtained results, the GA was found to be more robust and produce better quality solutions in comparison to the PSO algorithm. In contrast to the solutions obtained using the PSO algorithm, the GA procured solutions which bear resemblance to each other, irrespective of the employed solution evaluation metric, and the solutions were found to converge after a fixed number of iterations. As the problem size scales, inherently the difficulty of the problem increases and intrinsically the produced solutions and CPU time are found to be less consistent in comparison to that of smaller sized datasets. In terms of the CPU time taken in procuring a solution, the PSO algorithm is found to outperform the GA. The stochastic nature of the PSO algorithm is reflected in the inconsistent solutions it produces. The employed solution evaluation metric schemes are both formulated as weighted sum averages which prioritise the number of dispatched vehicles. Due to the similarity of the solution evaluation metrics, their influence on the applied solution techniques were found to be indifferent.

Acknowledgements

I would firstly like to express my profound gratitude to the Almighty, Bhagawan Swaminarayan, and my Gurus, Pramukh Swami Maharaj and Mahant Swami Maharaj, for blessing me with the opportunity, strength and courage to pursue my academic endeavours.

I express my sincere appreciation to my supervisors, Dr M. Woolway and Prof B. Jacobs. Without your insight, time and continuous encouragement, this work would not have been possible.

I am truly grateful to my parents and my sister for constantly providing me with unfailing support and immeasurable love. Thank you!

I am humbled and thankful to have been awarded funding from Bankseta, the Postgraduate Merit Award from the University of the Witwatersrand and the Centre of Excellence - Mathematical and Statistical Sciences.

Finally, I would like to thank the University of the Witwatersrand for all the provided resources.

Contents

List of Figures

List of Tables

Nomenclature

Graphs

G	Graph with vertices from **C** and edges from **A**

Sets

A	Arc set of graph G
C	Vertex set of Graph G
V	Vehicle set
Φ	Empty set
T	List of chromosomes selected through the tournament selection scheme
X	Position set
U	Velocity set

Solomon Benchmarking Data Sets Variables

γ	Percentage of customers to receive time windows (Solomon benchmarking data set)

VRPTW Variables

a	Arc between two vertices
c	Vertex in **C**; either the depot or a customer
i	Indexes vertices c in **C**
j	Indexes vertices c in **C**
nb	Arbitrary vertex reference
v	Vehicle in **V**
r	Indexes vehicle v in **V**
n	Number of customers
d	Distance between two vertices in **C**
D	Total accumulated travelled distance by all dispatched vehicles
e	Earliest time to start servicing a customer
l	Latest time to start servicing a customer
q	Load demand
s	Service time
t	Time a vehicle arrives at a location
w	Waiting time to service a customer
R	Route
S	Solution search space

Ω	All VRPTW constraints

Metric Variables

Λ	Metric A Objective function
Θ	Metric B Objective function
υ	Normalised distance Metric B

GA Variables

Γ	Population
κ	Individual/chromosome in population Γ
b	Indexes chromosomes κ
Z	Number of chromosomes in population Γ
ξ	Dimension of chromosome κ
η	Indexes dimension of chromosome κ, $\{\eta \in [1, \xi] \mid \eta \in \mathbb{N}\}$
τ	Generation or time-step
rad	Empirical radius
ϑ	Indexes routes
Ψ	Data type of elements that make-up chromosome κ
g	General fitness function
ψ	Arbitrary counter

PSO Variables

P	Population
N	Number of particles in population P
m	Dimension of particle
δ	Indexes dimension of particle h, $\{\delta \in [1, m] \mid \delta \in \mathbb{N}\}$
$\mathbf{x}$	Position of particle
$\mathbf{u}$	Velocity of particle
k	Iteration or time-step
φ	Arbitrary constant coefficient
ϖ	Remaining time to serve a customer
f	General fitness function
H	Tournament selection function
M	Difference in position subtraction of arc sets
$\mathbf{pb}$	Personal best position
$\mathbf{gb}$	Global best position

Miscellaneous Variables

ζ	Integer which approximates $\gamma \times n$
$rand$	Random value between 0 and 1 from a uniform distribution

VRPTW Parameters

Q	Vehicle load capacity constraint
TT	Total travel time constraint

GA Parameters

α	Weight coefficient for the number of vehicles in the fitness function Λ
β	Weight coefficient for the total accumulated travel distance in the fitness function Λ
μ	Percentage of initial population encoded using Random Permutation Encoding
ρ	Tournament selection benchmark probability of selecting the *fittest* chromosome
K	Number of chromosomes κ selected for the tournament selection set T
ϱ	Probability of mutation
Υ	Predefined terminating number of generations
θ	Crossover rate

PSO Parameters

σ_1	Random weight value for cognitive term
σ_2	Random weight value for social term
ς_1	Weight for cognitive term, between
ς_2	Weight for social term, between
ϕ	Probability of greedy initialisation
ω	Inertia weight coefficient for CLPSO velocity update. Linearly decreases from
P_c	Current learning probability
sg	Global terminating criterion (25 customers; 50 and 100 customers)
rg	PSO criterion
$\mathbf{S}_U$	Crisp set of from the arcs from the velocity set
$\mathbf{S}_X$	Crisp set of from the arcs from the position set
$\mathbf{S}_A$	Crisp set of from the arcs from the arc set

Result Variables

NV	Number of dispatched vehicles
DIST	Total distance travelled
FIT	Fitness value
NVD	Deviation percentage to the benchmark best value of the number of dispatched vehicles
DD	Deviation percentage to the benchmark best value of the total distance travelled
FitD	Deviation percentage to the benchmark best value of the fitness value

Chapter 1

Introduction

In the twenty-first century, acknowledging the reality of an exponentially more globalised and interconnected world also infers acknowledging the pertinent need to investigate Operational Research (OR) problems. In the field of OR, distribution management is identified as one of the most comprehensive strategic decision issues that need to optimised, for long-term efficient operation of the supply chain networks. For this reason, it is proposed to further investigate an array of mathematical optimisation techniques and engineering strategies which are used to optimise the distribution procedure.

1.1 Problem Overview

A distribution problem, the Truck Dispatching Problem, was first formally presented by Dantzig and Ramser (1959). The Truck Dispatching problem models a fleet of homogeneous gasoline delivery trucks from a bulk terminal to a number of service stations. The problem seeks to find a way to assign stations to trucks in such a manner that the station demands are satisfied whilst minimising both the number of vehicles dispatched and total distance travelled by these vehicles. The context is that of delivering goods from a central depot to their respective customers. Finding an optimal configuration of vehicles and respectively a set of service routes, for each of the vehicles to traverse, is known as the Vehicle Routing Problem (VRP). The VRP is a generalisation and an extension of the well-known Travelling Salesman Problem (TSP). A representation of the VRP is given in Figure 1.1. Variants of the VRP allows for the problem to made more realistic, but at the same time more complex to solve. One particular variant of the VRP is the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW is the VRP with an additional constraint of having to service customers within a specified time interval. Seminal work developing applications of optimisation methods to the VRPTW was conducted by Solomon (1987).

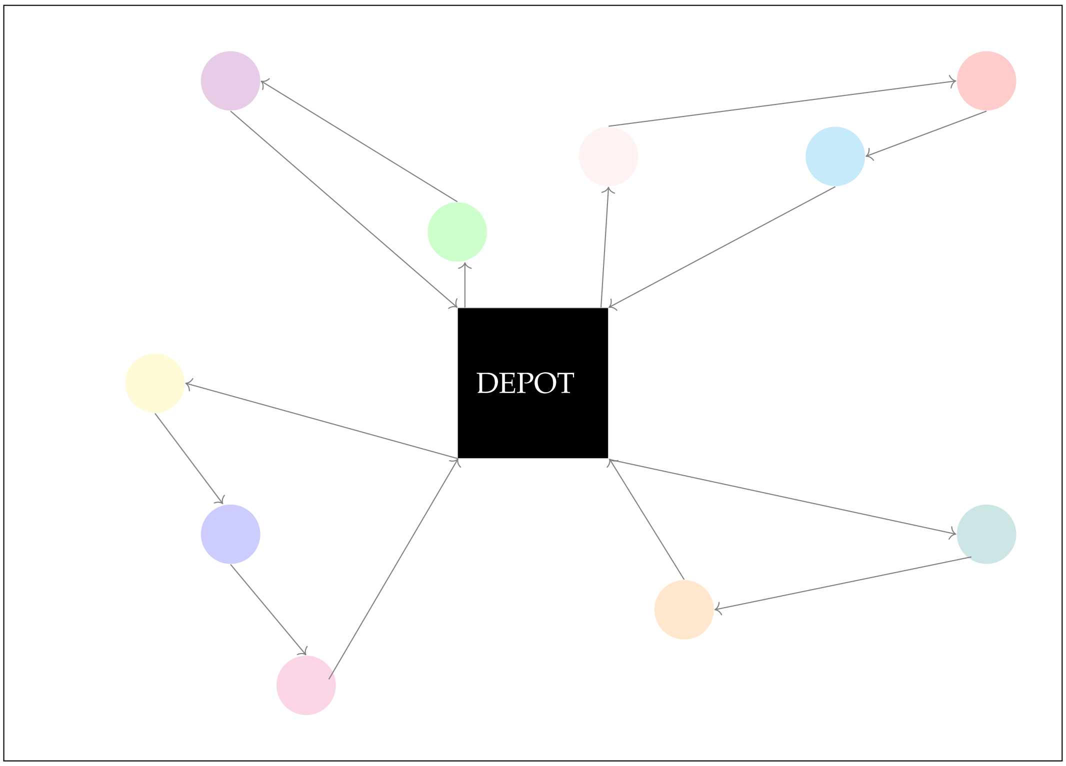


FIGURE 1.1: Representation of Vehicle Routing Problem

1.1.1 Problem Classification

The objective of the VRPTW is to service each customer within the customer set according to their specified time window using a minimum number of delivery vehicles, whilst also minimising the travel distance of each of these dispatched vehicles. The optimality of a solution to the VRPTW depends on:

- The number of dispatched delivery vehicles.
- The assignment of customers to delivery vehicles.
- The sequence in which customers are visited is dependent on the particular delivery vehicles to which they were assigned.

Considering that all of the aforementioned criteria are interdependent and the problem is modelled using integer based variables, then the VRPTW is classified as a Combinatorial Optimisation Problem (COP), an Integer Programming Problem (IP) and a Multi-objective Optimisation Problem (MOP).

Combinatorial Optimisation Problem

COP is a classification for multiple logistical problems found in various industries. Specifically, COP is defined by Rutenbar (1989) as the study of the best selection and configuration of a collection of objects adhering to some objective function.

Integer Programming Problem

An IP problem is defined by Nemhauser and Wolsey (1988) as a mathematical optimisation or feasibility program in which all of the variables are restricted to be integers.

Multi-objective Optimisation Problem

An MOP is a mathematically modelled optimisation problem with multiple objective functions which have to be simultaneously optimised.

1.1.2 Computational Complexity

To classify the complexity of the VRPTW, it is crucial to understand principal concepts of computational complexity theory.

Computational Complexity Theory

Church and Turing (1937) and Fortnow and Homer (2014) state that Computational Complexity Theory is the study of using standard measures to quantify both, the difficulty of tasks and the efficiency of schema applied to solve these tasks. Seminal work on this subject has been done by Gödel, Church, Turing, Kleene and Post, originally undertaken during the 1930s in attempt to answer Hilbert's Entscheidungsproblem, stated by Church and Turing (1937). The Entscheidungsproblem is a well-known decision problem of mathematics. A decision problem poses a question to which the answer is either 'true' or 'false' for an infinite set of inputs. The decision of the output value is governed by logical statements which verify if each step in an algorithm can be carried out by a finitary mathematical agent.

In 'computational complexity theory', the term 'computational' refers to using a computer to model and solve a task. The term 'complexity' refers to the study of a problem's (or a class of problems') computational resource requirements to complete a task. Hogan (2011) states that two common computational resources include time and storage space. If a significant amount of resources are required to complete a task then the task is regarded inherently difficult.

Turing machines, first described by Turing (1937), are simple abstract computational devices intended to help investigate the extent and limitations of what can be computed. Turing questioned what does it mean for a task to be computable. A task is deemed tractable if it can be processed by a Turing machine. For elaboration on how a task is deemed tractable, the reader is referred to Vitányi (2012).

There are a variety of Turing machines. Two particular Turing machines are Deterministic Turing machines and Non-deterministic Turing machines.

- Deterministic Turing machines
  A deterministic Turing machine is an elementary Turing machine, which uses a fixed set of logical statements to determine it's future transitions, as described by De Mol (2018).

- Non-deterministic Turing machines
De Mol (2018) states that a non-deterministic Turing machine is a deterministic Turing machine with an added characteristic of non-determinism. This type of Turing machine allows for numerous possible future transitions from a given state. Hence, the formulation of the Turing machine is repeated using all the possible options until a successful outcome is obtained.

Turing machine models are not intended as practical technology to solve tasks, but rather as theoretical devices which systematically analyse if tasks can be algorithmically solved. Turing machines are also used to classify computational problems into complexity classes. Complexity classes are distinguished by the abstract measurement of the rate of growth in the required resources as the task's input set increases. The relationship among the various complexity classes are shown in Figure 1.2. The classes are defined by Hogan (2011) as follows:

- $\mathcal{P}$: Polynomial time solvable decision problems.
There exists a polynomial time algorithm that solves every problem in $\mathcal{P}$.

- $\mathcal{NP}$: Non-deterministic polynomial time decision problems.
The $\mathcal{NP}$ class contains decision problems for which the problem can be solved by a non-deterministic Turing machine in polynomial time. That is the validity of any proposed solution is generated in a non-deterministic manner, while a deterministic algorithm verifies the proposed solution to the problem.

- $\mathcal{NP} - \mathcal{H}$: Non-deterministic polynomial time hard decision problems.
$\mathcal{NP}$ hardness is a defining property of the $\mathcal{NP}$ class of problems. Informally, if the problem is at least as *hard* as any $\mathcal{NP}$ problem, although does not necessarily have to be in $\mathcal{NP}$, claims Hogan (2011). Formally, Knuth (1974) states that a problem M is $\mathcal{NP} - \mathcal{H}$ when every problem L in $\mathcal{NP}$ can be reduced in polynomial time to M. Assuming that to obtain a solution for M it takes 1 unit time, then M's solution can be used to solve L in polynomial time. As a consequence, finding a polynomial algorithm to solve any $\mathcal{NP} - \mathcal{H}$ problem would give polynomial algorithms for all the problems in $\mathcal{NP}$, which is unlikely as many of them are considered difficult.

- $\mathcal{NP} - \mathcal{C}$: Non-deterministic polynomial time complete decision problems.
$\mathcal{NP} - \mathcal{C}$ falls in the greater classification of $\mathcal{NP}$ problems. A problem is considered to be $\mathcal{NP} - \mathcal{C}$ if it is in $\mathcal{NP}$ and every problem in $\mathcal{NP}$ can be transformed to $\mathcal{NP} - \mathcal{C}$ in polynomial times. Thus, $\mathcal{NP} - \mathcal{C}$ consist of all the problems that belong to both $\mathcal{NP}$ and $\mathcal{NP} - \mathcal{H}$.

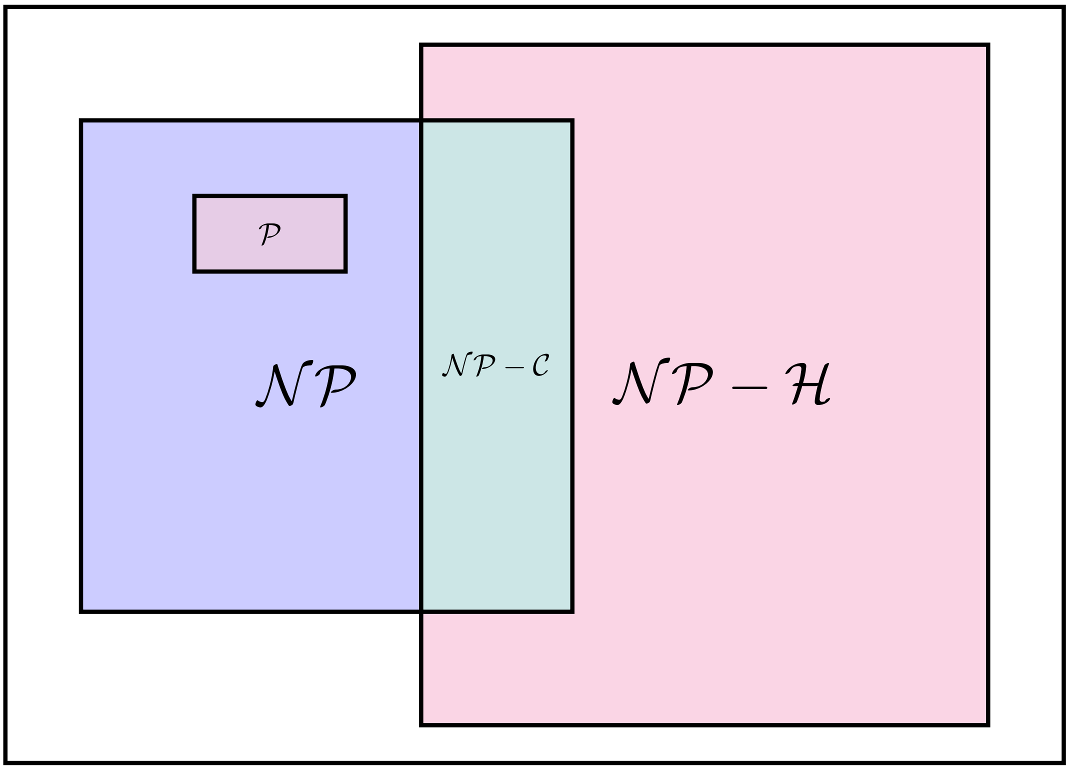


FIGURE 1.2: Diagram of Class

In Complexity Theory, one of the most prominent unresolved questions is whether there exists polynomial time algorithms for solving $\mathcal{NP}-\mathcal{C}$ problems. By corollary, if there does exist polynomial time algorithms for solving $\mathcal{NP}-\mathcal{C}$ problems, then all $\mathcal{NP}$ problems can be solved. This problem is commonly known as $\mathcal{P}$ versus $\mathcal{NP}$. Simply, $\mathcal{NP}-\mathcal{C}$ means that all known exact solution techniques that can be applied to these problems require an exponentially increasing number of enumerations to be computed as the problem grows. It is not feasible to implement exact or deterministic techniques to $\mathcal{NP}-\mathcal{C}$ problems. Thus, the problems are approached using approximate algorithms or non-deterministic techniques. In comparison to exact algorithms, approximate algorithms are less computationally demanding in obtaining a solution (not necessarily the optimal solution). The reason is that approximate algorithms avoid enumeration of considering all possible solutions in a search space. These algorithms usually entail relaxing some of the problem's constraints in order to find a solution. The solutions obtained by approximate algorithms are proven to be within a provable distance of the optimal solution.

Problem's Complexity

In conjunction to computational complexity theory, the VRPTW is a difficult problem for which to find solutions. The VRP is classified by Tripathi and Bhawna (2006) to be a $\mathcal{NP}-\mathcal{H}$ problem. Since the VRP is $\mathcal{NP}-\mathcal{H}$, by restriction, the VRPTW is also classified to be $\mathcal{NP}-\mathcal{H}$. Furthermore, Emde Boas and Savelsbergh (1984) have proved that even finding a feasible solution to the VRPTW when the number of vehicles is fixed is itself a $\mathcal{NP}-\mathcal{C}$ problem.

1.2 Solution Techniques

Considering the multi-objective VRPTW and its complexity, it would not be feasible to apply deterministic methods to find solutions to the problem as the size of the customer set scales, because the computational time required would exponentially scale. If computation time is prioritised over solution quality then non-deterministic techniques can be employed to provide approximate solutions to the problem.

The two main classifications which algorithms are classified into are:

- Deterministic.
- Non-deterministic.

Methods under both these classification come with their own advantages and disadvantages.

1.2.1 Deterministic Techniques

Deterministic algorithms always return the same results and can be solved in a finite number of steps without any random variation. Deterministic techniques are also commonly known as exact methods.

1.2.2 Non-deterministic Techniques

Non-deterministic techniques compute solutions which are approximate. At each iteration of the search process, the solution point is evaluated using either a concurrent algorithm or a probabilistic algorithm. The nature of these two classes of algorithms is the result of the non-uniform behaviour which non-deterministic algorithms exhibit. These two sub-algorithms determine the manner in which solutions are obtained in non-deterministic algorithms.

Sub-classifications of non-deterministic techniques are:

- Heuristic Techniques.
- Metaheuristic Techniques.

Heuristic Techniques

Heuristic techniques are employed to *quickly* obtain an approximate results. They prioritise computational time yet jeopardise the solution's optimality, completeness, and efficiency. Pearl (1984) states that heuristic techniques solutions are taken to be the lower bound solution, which can usually be used to navigate subsequent decisions on how to approach the problem. The Greedy Approach, which is a heuristic technique, makes locally optimal choices with the hope of finding the global optimal solution.

Metaheuristic Techniques

Metaheuristic techniques are high-level approaches that guide various other heuristic techniques in searching for solutions within a broad set of problem domains. Some examples of techniques that will be further investigated are: the Genetic Algorithm (GA) by Yesodha and Amudha (2012) and Junkermeier (2015), Simulated Annealing (SA) Algorithm by Rutenbar (1989), Tabu Seach (TS) by Glover (1986) and Glover and Laguna (1997), Particle Swarm Optimisation (PSO) by Eberhart and Kennedy (1995) and Hannan, Akhtar, Begum, Basri, Hussain, and Scavino (2018) and Ant Colony Optimisation (ACO) by Dorigo, Caro, and Gambardella (1999) and Arias-Rojas, Jiménez, and Montoya-Torres (2012) . With the intention of improving the performance of these metaheuristic techniques across a wide range of similar problems, hybridisations and modified variants of these well-known metaheuristic techniques are introduced by Yesodha and Amudha (2012).

1.3 Problem Statement and Motivation

The distribution procedure is a typical OR scenario occurring in various commodity industries. Applications of the VRP given a particular time window constraint is prevalent in the following examples given by Cordeau J-F. and Semet (2002): bank related deliveries, courier services, industrial refuse collection, fast-food deliveries, school bus routing, security patrol services, cleaning service agents and Just In Time manufacturing. There is extensive literature on the VRPTW, considering the prevalent application of the problem. A host of both deterministic and non-deterministic technique applications to the VRPTW are comprehensively developed in the literature. Acknowledging the VRPTW's complexity as the problem size scales, non-deterministic techniques will only be studied in this thesis. When considering these various metaheuristic optimisation techniques in conjunction to each other, it is not easy to directly compare the obtained respective results. Specifically, to compare the computational resource of time as the results obtained are not generated under standardised experimental conditions. Thus, for all experiments the same machine, programming language and programming style would be required to be used.

To conduct a comparative review of heuristic technique applications to the VRPTW, that means evaluating and comparing the particular applied algorithm's performances. A number of aspects of each of the algorithms form a part of the evaluation criteria. Examples of such criteria given by Barr, Golden, Kelly, Resende, and Stewart (1995) and Cordeau, Gendreau, Laporte, Potvin, and Semet (2002) include: running time, quality of solution, ease of implementation, robustness, and flexibility.

A description by Bräysy and Gendreau (2005a) of each of these criteria aspects are given:

- Flexibility
  An algorithm should be able to accommodate for changes in the model, the constraints and the objective function.
- Robustness
  An algorithm should be able to be applied to variant models where the characteristics of the problem are varied.

- Time
  How long it takes the heuristic to produce good quality solutions.
- Quality
  Solution obtain to the objective function within some set time, measured relative to the standard optimal solution.

1.3.1 Aims

The aims of the research is to present a comparative review of the application of various metaheuristic techniques and metrics to the VRPTW. The research proposes to collate previously applied heuristic optimisation techniques and quantitative measures which were used in finding solutions to the VRPTW under standardised experimental condition. Thus, allowing for direct comparison and evaluation.

1.3.2 Objectives

The objectives of this thesis are:

- To compare the solution quality of the applied solution techniques and metrics by considering the solutions obtained with respect to the time taken to find the solution.
- To compare the computational efforts required by the applied solution techniques, that is the complexity of the solution schema and the CPU time.
- For each of the data sets with varying characteristics, identify correlations to the applied solution technique and metric.
- Investigate the effects of the varying applied metrics to solutions obtained.

1.3.3 Problem formulation

The goal of the VRPTW entails designing a set of minimum cost routes, originating and terminating at a central depot, for a fleet of vehicles which service a set of customers with known demands. The customers must be assigned exactly once to vehicles such that the vehicles capacities are not exceeded. The service at the customer must begin within the defined time window which is the earliest and latest time when the customer permits the start of being serviced.

The VRPTW is a multi-objective optimisation problem as it's goal is to minimise not only the number of vehicles dispatched, but also the waiting time to service a customer, total travel time, and accumulated total travelled distance incurred by the fleet of vehicles.

Problem Set-up

The VRPTW is represented by a fleet of homogeneous vehicles denoted by $\mathbf{V}$, and directed graph $G = (\mathbf{C}, \mathbf{A})$. In graph G, the vertex set is represented by $\mathbf{C}$ and the arc set $\mathbf{A}$. In the vertex set $c_i \in \mathbf{C}$, $\{i \in [0, n] \mid i \in \mathbb{N}_0\}$, where c_0 denotes the depot and $c_i \; \forall \; i \in [1, n]$ denote the n customers which are to be served.

Each customer $c_i \in \mathbf{C}$, where $i \in [1, n]$, has delivery details associated to them. These details are tabulated in Table 1.1. Note, all these variables are assumed to be non-negative.

Customer Detail Variables	
Variable	Description
q_i	Demand.
s_i	Service time.
e_i	Earliest time to start start servicing customer c_i.
l_i	Latest time to start start servicing customer c_i.

TABLE 1.1: Customer Detail Variables

For the depot c_0, the details associated differ to that of the customers. These details are tabulated in Table 1.2.

Depot Detail Variables	
Variable	Description
q_0	$q_0 = 0$.
s_0	$s_0 = 0$.
e_0	Earliest time that any vehicle is allowed to be dispatched from the depot c_0.
l_0	Latest time that any vehicle is allowed to return the depot c_0.

TABLE 1.2: Depot Detail Variables

In graph G, the arc set is given as $\mathbf{A} = \{\langle c_i, c_j\rangle | c_i, c_j \in \mathbf{C}, i \neq j\}$. Each arc $\langle c_i, c_j\rangle$ is associated with a Euclidean distance d_{ij} between the two vertices c_i and c_j, where $d_{ij} = d_{ji}$. Due to the condition that service routes begin and terminates at the depot, the arc set starts and ends at c_0 for each service route. The design of routes to service all the customers requires a binary decision variable a^r_{ij}.

$$a^r_{ij} = \begin{cases} 1, & \text{if vehicle } r \text{ travels directly from } c_i \text{ to } c_j, \\ 0, & \text{otherwise.} \end{cases}$$

A vehicle v in $\mathbf{V}$ is indexed by r, where $\{r \in [1, |\mathbf{V}|] \mid r \in \mathbb{N}\}$. A binary decision variable is used to indicate by which vehicle r a customer c_i is serviced.

$$b^r_i = \begin{cases} 1, & \text{if customer } i \text{ is serviced by vehicle } r, \\ 0, & \text{otherwise.} \end{cases}$$

For any vehicle $v_r \in \mathbf{V}$ there are two constraints:

- The maximum time for which any vehicle is allowed to travel is denoted by TT, where $TT > 0$.
- The maximum capacity that any vehicle may be loaded with is denoted by Q, where $Q > 0$.

Mathematical Model

The goal of the VRPTW is to to design a set of minimal cost routes such that each customer is visited exactly once and the time windows and capacity constraints observed. The VRPTW's primary objective is to reduce the number of dispatched vehicles. The secondary objective of the VRPTW is to reduced the total travelled distance by the dispatched vehicles. These two objectives are respectively expressed by Equations (1.1) and (1.2) .

$$\min\ Z_1 = |\mathbf{V}|, \tag{1.1}$$

and

$$\min\ Z_2 = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{r=1}^{|\mathbf{V}|} a_{ij}^r d_{ij}. \tag{1.2}$$

subject to,

$$\sum_{i=0}^{n} a_{ij}^r = b_j^r \quad \forall\, r = 1, \ldots, |\mathbf{V}|, \quad \forall\, j = 1, \ldots, n \tag{1.3}$$

$$\sum_{j=0}^{n} a_{ij}^r = b_i^r \quad \forall\, r = 1, \ldots, |\mathbf{V}|, \quad \forall\, i = 1, \ldots, n \tag{1.4}$$

$$\sum_{i=0}^{n} b_i^r \times q_i \leq Q, \quad \forall\, r \in 1, \ldots, |\mathbf{V}| \tag{1.5}$$

$$\sum_{r=1}^{|\mathbf{V}|} b_i^r = 1, \quad \forall\, i = 1, \ldots, n \tag{1.6}$$

$$\sum_{r=1}^{|\mathbf{V}|} b_0^r = |\mathbf{V}| \tag{1.7}$$

$$t_i + w_i + s_i + t_{ij} = t_j, \quad \forall\, i, j = 0, \ldots, n, \quad i \neq j \tag{1.8}$$

$$e_j \leq t_j + w_j \leq l_j, \quad \forall\, j = 0, \ldots, n \tag{1.9}$$

$$w_i = \max\{e_i - t_i, 0\} \quad \forall\, i = 0, \ldots, n. \tag{1.10}$$

The constraints given in Equation (1.3) and Equation (1.4) denote that exactly one arc enters and leaves each vertex associated with a customer. The constraint given by Equation (1.5) states that each vehicle must not be loaded with more than its carrying capacity Q. The constraint given by Equation (1.6) stands for that each customer can only be served by one vehicle. The constraint given by Equation (1.7) represents that all routes start from the depot. Formula (1.8) defines t_j, the time at which the customer c_j starts to be serviced; t_j is the sum of: the time of arrival at customer c_i, denoted by t_i, the waiting time or idle time before starting to service customer c_i within it's specified time window, the time to service customer c_i, denoted by s_i, the travel time between customer c_i and c_j, denoted by t_{ij}. The time window within which customer c_j starts to be serviced is denoted by Formula (1.9), where e_j is the earliest time at which customer c_j can start to be serviced and l_j is the latest time at which the customer can start to be serviced, thus the time at which customer c_j starts to be serviced is the sum of the time of arrival at customer c_j and the waiting time to service customer c_j . The waiting time is calculated using Formula (1.10).

The layout of this thesis is as follows. Chapter 1 is an introduction to the problem investigated. Chapter 2 reviews the comprehensive literature on the VRPTW problem. Chapter 3 is the methodology which gives details of the dataset used, and descriptions of the metaheuristic algorithms applied to the VRPTW. Results are presented in Chapter 4. Finally, conclusions and recommendations are given Chapter 5.

Chapter 2

Literature Review

The Vehicle Routing Problem with Time Windows (VRPTW) has emerged as a subject for progress in accommodating realistic generalisations and constraints of the basic vehicle routing model. Solomon (1984) identified that time windows naturally arise in problems faced by businesses that work on fixed time schedules. Specific examples include: bank deliveries, postal deliveries, industrial refuse collection, dial-a-rider service and school bus routing and scheduling.

For many years the spatial problem of routing vehicles has been intensively studied in the literature of Operational Research (OR), as claimed by Raff (1983). Over time there has been development on the VRPTW, which encompasses both spatial and temporal aspects of routing vehicles. Initial work dealing with the VRP with the time window constraint appeared in the form of case studies formulated by Cook and Russell (1978), Knight and Hofer (1968), and Pullen and Webb (1967).

A particular case study presented by Pullen (1967), developed systems for duty scheduling of van drivers with tight time constraints. Tight time constraints are narrow time windows which are to be strictly adhered to. However, the tight time constraint results in occurrences of vehicles having idle time and empty-running time. Empty-running time is when vehicles do not have a load on their return journeys. The heuristic solution presented focuses on the allocation of jobs to vehicles to reduce both idle and empty-running time between customers. The procedure is evaluated by simulation. Another case study presented by Knight and Hofer (1968) involves a contract transport company. The problem is dominated by time windows ranging from 15 minutes to an entire day, and averaging between 1 to 2 hours. A heuristic manual system was developed to increase the utilisation of vehicles as measured by the average number of customers a vehicle has to service per hour and the total number of vehicles used.

Early surveys of solution techniques focusing on exact techniques to solve the VRPTW are presented by Golden and Assad (1986), Desrochers and Soumis (1988),Desrosiers, Dumas, Solomon, and Soumis (1995), and Cordeau, Laporte, and Mercier (2001a). Further details on these exact methods are discussed by Cook and Russell (1978), Larsen (1999), and Cordeau, Laporte, and Mercier (2001a).

Considering the limitations of the overall performance of deterministic algorithms by Solomon and Desrosiers (1988), and the intrinsic difficulty in solving the $\mathcal{NP}-\mathcal{H}$ VRPTW, Solomon studied heuristic techniques to improve the quality of the constructed routes and because of the promise they hold in solving realistic size problems.

To the author's knowledge, a comprehensive comparative survey reviewing the applications of various solution techniques to the VRPTW under standardised experimental conditions has not been conducted. Identifying this, the research is initiated by forming a review of the growing body of literature on the VRPTW. Considering the broad application of the VRPTW for time-sensitive and large-scaled cases in conjunction to the problem's $\mathcal{NP} - \mathcal{H}$ complexity classification, Tripathi and Bhawna (2006) found that the focus of applied solution schemes turned from deterministic techniques to non-deterministic techniques. This is a result of prioritising obtaining solutions in practical time. However, this means accepting approximate solutions rather than exact solutions.

Heuristic techniques are particularly of interest because they can provide good solutions with a low computational effort, but also because they are a important component of all metaheuristics for the VRPTW. In order to improve the approximate solutions obtained by heuristic techniques, metaheuristic techniques are used as a preventative measure of being trapped at local optima solutions, as recorded by Bräysy and Gendreau (2005b), and Braekers, Ramaekers, and Van Nieuwenhuyse (2016).

This chapter considers the timeline for both deterministic and non-deterministic techniques applied to the VRPTW. Heuristic and metaheuristic techniques are both elaborated on as part of the non-deterministic techniques.

2.1 Exact Techniques

As a simplification, Solomon (1984) considers the VRPTW with a single vehicle. The VRPTW then reduces to the the Travelling Salesman Problem with Time Windows (TSPTW). The TSPTW is classified to be in the $\mathcal{NP} - \mathcal{C}$ complexity class, as proved by Savelsbergh (1985). In search of solutions to the TSPTW, Dynamic Programming technique and Branch and Bound Algorithm were both respectively applied by Christofides, Mingozzi, and Toth (1981b) and Baker (1983). To maximise the lower bounds within the Dynamic Programming application, state-space relaxations and the Branch and Bound method were recursively applied. Hence, exploiting the structure through allowing relaxations to the proposed model. An observation made by Baker (1983) was the tightness of time windows impacted the number of vertices being examined in the Branch and Bound tree. Customer sets with 50 customers were solvable by the methods provided by Christofides, Mingozzi, and Toth (1981b) and Baker (1983). A Branch-and-Price algorithm described by Solomon and Desrosiers (1988), which was improved by Desrochers, Desrosiers, and Solomon (1992a) and Irnich and Villeneuve (2006) to improve lower bounds on k-cycle eliminations, to solve for optimality more than 15 unsolved Solomon problem instances with 25, 50 and 100 customers. However, this exact method fails to solve several instances with 25 customers, as claimed by Baldacci, Mingozzi, and Roberti (2012).

2.1.1 Exact Techniques for the Vehicle Routing Problem

Three broad categories which classify exact algorithms for the VRP are presented by Laporte (1992) are:

- Direct Tree Search Method.
- Dynamic Programming.
- Integer Linear Programming.

Tree Search Method

The assignment lower bound and a related Branch and Bound algorithm are considered by Laporte, Mercure, and Nobert (1986). This method exploits the relationship between the VRP and one of its relaxations, the m-TSP. This method is based on the formulation given by Lenstra and Kan (1975). The k-degree center tree and a related algorithm was developed by Christofides, Mingozzi, and Toth (1981a) for symmetrical VRPs. The scheme is based on the k-degree center tree relaxation of the m-TSP. The lower bounds imposed have successfully solved the VRPs ranging in size from 10 to 25 customers.

Dynamic Programming

Dynamic Programming was first proposed for the VRP by Eilon and Watson-Gandy (1971). The VRP is considered for a fix number of m vehicles. The minimum cost of the routing procedure is determined by using recursive methods. An alternative to reducing the routing cost is by using state-space relaxation, which is a method introduced by Christofides, Mingozzi, and Toth (1981a). Optimal solutions are thus obtained by considering the general Dynamic Programming along with other relaxations. Customer sets containing 10 to 25 customers were solved in this manner.

Integer Linear Programming

Set partitioning was first introduced to find solutions to the VRP by Balinski and Quandt (1964). The suggested method entails using a column generation algorithm. Since the solution to the VRP must be a set of integers, the proposed method must be used in conjunction with Branch and Bound algorithm. A variation to this method has been successfully applied by Desrosiers, Soumis, and Desrochers (1984) and was extended to the VRPTW by Desrochers, Desrosiers, and Solomon (1992b). It has been observed that the method performs better when the problem has tighter constraints as the number of feasible columns are reduced.

2.1.2 Exact Techniques for the Vehicle Routing Problem with Time Windows

The exact methods for the VRPTW can be classified into three categories, as per the review paper done by El-sherbeny (2010):

- Lagrange Relaxation-based Methods.
- Column Generation.
- Dynamic Programming.

Lagrange Relaxation-based Methods

The Lagrange Relaxation Method approximates difficult constrained optimisation problems by relaxing some of the problem's constraints. The relaxations are compensated with Lagrange multipliers, which penalise violations of the inequality constraints. These additional penalty terms enforce the relaxed constraints in the optimisation problem. Variations of this method being applied to the VRPTW are done by Fisher, Jörnsten, and Madsen (1997), Kohl and Madsen (1997), and

El-sherbeny (2010). The problem is formulated using a k-tree approach. The problem Lagrangian relaxes all constraints except the constraints ensuring that at most one arc is joining customers c_i and c_j. The problem is then solved by El-sherbeny (2010) with a minimum degree-constrained k-tree problem as a sub-problem and the Lagrange multipliers are set using a sub-gradient approach.

Column Generation

Column Generation is an efficient algorithm for solving linear programming problems which contain too many variables to be explicitly solved. As most of the variables will be non-basic and assume a value of zero in the optimal solution, only a subset of variables need to be considered in theory when solving the problem. Column Generation leverages this idea to generate only the variables which have the potential to improve the objective function. One or more of the variables currently not in the linear program may be added to the problem's formulation to improve the linear program solution. That being to find variables with negative reduced cost (in the instance of a minimisation problem). Column Generation used to solve the VRP was first done by Agarwal, Mathur, and Salkin (1989). It was first used to solve the m-TSP with time windows by Desrosiers, Soumis, and Desrochers (1984). However, a more effective model was applied by to the VRPTW which solved more instances to optimality by Desrochers, Desrosiers, and Solomon (1992b).

Dynamic Programming

The Dynamic Programming method is used to simplify complicated problems by reducing it to simpler sub-problems in a recursive manner. Not all decision problems can be taken apart in this manner, however, decisions which span apart often can be recursively split apart. If the problem can be optimally solved by breaking it into sub-problems and then recursively finding the optimal solution to the sub-problems, then it is said to be in a sub-structure format. Thus, if sub-problems can be nested recursively inside larger problems, so that Dynamic Programming methods are applicable, then there is a relation between the value of the larger problem and the values of the sub-problems. Specific to the VRPTW, branching decisions are taken on vehicle-customer allocations. The Dynamic Programming approach for VRPTW is presented for the first time by Kolen, Rinnooy Kan, and Trienekens (1987). The method calculates lower bounds using Dynamic Programming and state-space relaxation based on the k-tree relaxation given in Fisher (1994). A customer set containing up to customers are optimally solved using this method presented by Kohl, Desrosiers, Madsen, Solomon, Kohl, Desrosiers, Madsen, and Solomon (1999) and Kolen, Rinnooy Kan, and Trienekens (1987). Dynamic Programming method for the VRPTW was also presented by Christofides and Beasley (1984). Branch and Bound is applied together with the Dynamic Programming algorithm to obtain optimal solutions. Each node in the Branch and Bound tree corresponds to three sets:

- A set of fixed feasible routes starting and ending at the depot.
- A set of partially built routes starting at the depot.
- A set of customers forbidden to be next on the set of partially built routes starting at the depot.

2.2 Heuristic Techniques

Heuristic techniques are designed to speed up the process of finding satisfactory solutions to problems. Exact solutions are traded for approximate solutions as a results of the lower computational efforts required by heuristic methods. Heuristic techniques provide sufficient results with immediate goals. Two traditional heuristic approaches which have been applied to the VRPTW are:

- The Route Construction Method
  Generates a set of routes from the start.

- The Route Improvement Method
  Produces an improvement on solutions already produced.

2.2.1 Route Construction Method

Route construction heuristics serially select nodes (or arcs) until a feasible solution has been created. Nodes are chosen based on some cost minimisation criterion, often subject to the restriction that the selection does not create a violation of vehicle capacity or time window constraints. Sequential methods construct one route at a time, while parallel methods simultaneously build several routes. Some route construction heuristics considered are:

- Route-First Cluster-Second Scheme.

- Time Orientated Nearest-Neighbour Scheme.

- I1 Scheme.

Route-First Cluster-Second Scheme

A route-first cluster-second scheme using a giant-tour heuristic was proposed by Solomon (1986). First, the customers are scheduled into one giant tour, which is then divided into a number of smaller routes. The initial giant tour could be generated as a travelling salesman tour without considering the capacity and time window constraints. No computational results are given in the paper for the heuristic. Several heuristics for the VRPTW are described by Solomon (1984). One of the methods is an extension to the savings heuristic of Clarke and Wright (1964). The savings method, originally developed for the classical VRP, is a well-known route construction heuristic. It begins with a solution in which every customer is supplied individually by a separate route. Combining the two routes which respectively serve customers c_i and c_j results in a cost savings in terms of distance. Let r_1 be the route in which c_i is being served, and r_2 the route in which c_j is being served. The distance of r_1 and r_2 are respectively given by D_1 and D_2. Where $D_1 = d_{0i} + d_{0i}$ and $D_2 = d_{0j} + d_{0j}$. Combining r_1 and r_2, the updated route's distance is given as $d_{i0} + d_{ij} + d_{0j}$. Where d_{i0} is the distance between the c_i and the depot, d_{0j} is the distance between the the depot and the customer c_j, and d_{ij} is the distance between c_i and c_j. This scheme select the arc $\langle c_i, c_j \rangle$ linking customers c_i and c_j which has a maximum distance saving, subject to the requirement that the combined route is feasible as done in Clarke and Wright (1964). With this convention, the route combination operation is iteratively applied. In combining routes one can simultaneously form partial routes for all vehicles or sequentially add customers to a given route until the vehicle is fully loaded. To account for both the spatial and temporal proximity of customers, a limit to the waiting time of the route is set.

Time Orientated Nearest-Neighbour Scheme

A time-orientated nearest-neighbour iteratively selects the *closest* neighbour. The metric used to measure the closeness of any pair of customers attempts to account for both geographical and temporal closeness of customers. This method starts every route by finding an unallocated customer closet to the depot. At every subsequent iteration, the heuristic searches for the customers closest to the last customer added to the route and adds it at the end of the route. A new route is started each time the search fails to find a feasible insertion place, unless there are no remaining unallocated customers.

I1 Scheme

The I1 heuristic scheme is claimed to be the most successful of the sequential insertion heuristics proposed by Bräysy and Gendreau (2005b). A route is initialised with a 'seed' customer. Iteratively the remaining customers are inserted into this route until it is full with respect to the scheduling horizon and/or capacity constraint. If unallocated customers remain, the initialisations and insertion procedure are then repeated until all customers are serviced. The seed customers are selected by finding either the geographically furthest unallocated customer in relation to the depot or the unallocated customer with the lowest allowed starting time for service.

2.2.2 Route Improvement Methods

Classical local search methods form a general class of approximate heuristics based on the concept of iteratively improving the solution to a problem by exploring neighbouring solutions. Local search algorithm designs usually have the following specified:

- How the initial feasible solution is generated.
- What move-generation mechanism is used.
- Acceptance criterion.
- Stopping condition.

The move generation mechanism creates the neighbouring solutions by changing one attribute or a combination of attributes of a given solution. An example of these attributes could be arcs connecting a pair of customers, as referenced by Bräysy and Gendreau (2005b). Once a neighbouring solution is identified, it is compared against the current solution. If the neighbouring solution is better, it replaces the current solution, and the search continues. Two common acceptance strategies used in the context of the VRPTW are first-accept and best-accept. The first-accept strategy selects the first neighbour that satisfies the predefined acceptance criterion. The best-accept strategy examines all neighbours satisfying the criterion and selects the best among them.

The local optimum produced by any local search procedure can be very far from the global optimal solution. Local search methods perform myopic searches because they sequentially accept solutions that improves the objective function value. Thus, the outcome is dependent on generated initial solution and the neighbourhood generation mechanism. Most iterative improvement methods given by Bräysy and Gendreau (2005b) have been applied to vehicle routing and scheduling problems are edge-exchange algorithms.

Edge-Exchange Neighbourhoods

The edge-exchange neighbourhoods for a single solution of the orientation of a sequence of two or more route are the set of tours that can be obtained from an initial tour by replacing a set of k of its edges by another set of k edges. Such a replacement is called k-exchange and a tour that cannot be improved by k-exchange is said to be k-optimal. Verifying k-optimality requires $\mathcal{O}(n^k)$ time. It tries to improve the tour by replacing two of its edges by two other edges and iterates until no further improvement is possible.

Neighbourhood-operators

Neighbourhood-operators reviewed by El-sherbeny (2010):

- Relocate operator
  Move one customer from one route to another.
- Exchange operator
  Interchange two customers within a route.
- 2-Opt operator
  Changing one segment of a route with another segment from another route.
- Or-Opt operator
  A continuous segment of customers is moved from one position on a route to another.
- K-node interchange operator
  Sequentially each customer c_i is considered. Customer c_i and its successor c_j and the two customers closest to c_i and c_j but not on the same route are removed. The neighbourhood is defined by trying to insert these four vertices in any other possible way.
- λ-interchange operator
  A subset of customers of size $\leq \lambda$ is exchanged with a subset of customers of size $\leq \lambda$ from another route.
- Shift-sequence operator
  A customer is moved from one route to another after checking all possible positions of insertion. If an insertion is feasible by removing another customer c_j, it is removed and inserted in another route. This procedure is repeated until feasibility is restored.

Variations, modifications and extensions to the edge-exchange neighbourhood local search can be found in Antes and Derigs (1995), Baker and Schaffer (1986), Bräysy and Gendreau (2005b), Rutenbar (1989), Ichoua, Gendreau, and Potvin (2003), and Van Landeghem (1988).

2.3 Metaheuristic Techniques

Metaheuristics are general solution procedures which explore a solution space to identify good solutions. Metaheuristic techniques are often embedded with some of the standard route construction methods and improvement heuristic techniques. In a major departure from classical approaches, metaheuristics allow deteriorating and even infeasible intermediate solutions in the course of the search process. A metaheuristic is a higher-level procedure or heuristic designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimisation problem, especially with limited computational run time. Compared to other optimisation algorithms and iterative methods, metaheuristics do not guarantee that a globally optimal solution. Many metaheuristics implement some form of stochastic optimisation. Hence, the solutions found are dependent on the set of random variables generated and force exploration of the search space.

Metaheuristics which have been applied to the VRPTW include:

- Tabu Search.
- Genetic Algorithm.
- Particle Swarm Optimisation.
- Hybrid Metaheuristic Techniques.

2.3.1 Tabu Search

TS explores the solution space by moving at each iteration from a solution B to the best solution in a subset of its neighbourhood $N(B)$. Contrary to classical descent methods, the current solution may deteriorate from one iteration to the next. Poorer solutions may be accepted only to avoid paths already investigated. This ensures new regions of a problem's solution space will be explored with the goal of avoiding local minima and ultimately finding the global optimal solution. To avoid cycling, solutions possessing some attributes of recently explored solutions are temporarily declared *tabu* or forbidden. The duration that an attribute remains tabu is called its tabu tenure, and it can vary over different intervals of time. The tabu status can be overridden if certain conditions are met; this is called the aspiration criterion. A possible aspiration criterion could be when a tabu solution is better than any previously seen solution, this tabu solution is accepted. Finally, various techniques are often employed to diversify or intensify the search process. TS is a local search metaheuristic introduced by Glover (1986), Glover (1989), Glover (1990), and Ichoua, Gendreau, and Potvin (2003).

The first application of TS to the VRPTW was done by Garcia, Potvin, and Rousseau (1994). The TS they developed uses Solomon's I1 insertion heuristic to create an initial solution and 2-opt and Or-opt exchanges for improvement.

The initial solution is typically created with a cheap insertion heuristic. The most commonly used insertion heuristic is done by Solomon (1987). An exception has been made by Chiang and Russell (1996), who uses a parallel version of the insertion heuristic done by Eberhart and Kennedy (1995). The savings heuristic defined in Clarke and Wright (1964) are used by De Backer, Furnon, Prosser, Kilby, and Shaw (1997) and Schulze and Fahle (1999). A modified version of Solomon's insertion heuristic, initially proposed by Thangiah, Osman, and Sun (1994) and Cordeau, Laporte, and Mercier (2001a) use a modified version of the sweep heuristic developed by Gillett and Miller (1974), is presented in Tan, Lee, Zhu, and Ou (2001a). The concept of a holding list, a data structure containing the unserviced customers was introduced by Lau, Sim, and Teo (2003). In the beginning all customers are in the holding list, and simple relocate and exchange operators are used to transfer customers back and forth from the holding list.

To reduce the complexity of the search, some authors propose special strategies for limiting the neighbourhood. Various instances of these strategies can be found done by Garcia, Potvin, and Rousseau (1994) and Taillard, Badeau, Gendreau, Guertin, Potvin, Taillard, and Guertin (1997b). Another frequently used strategy to speed up the search is to implement the proposed algorithm in parallel on several processors. Various instances of this solution approach were applied by Taillard, Badeau, Gendreau, Guertin, and Potvin (1997a), Taillard, Badeau, Gendreau, Guertin, Potvin, Taillard, and Guertin (1997b), Garcia, Potvin, and Rousseau (1994), and Schulze and Fahle (1999). To cross the barriers of the search space created by time window constraints, some authors allow infeasibilities during the search. For instance, violations allowed for each constraint type (load, duration, and time windows constraints) are done by Cordeau, Laporte, and Mercier (2001a) and Lau, Sim, and Teo (2003). The violations of constraints are penalised in the cost function, and the parameter values regarding each type of violation are dynamically adjusted.

Most of the proposed TS use specialized diversification and intensification strategies to guide the search. For example, 'adaptive memory' is proposed by Rochat and Taillard (1995). The adaptive memory is a pool of routes taken from the best solutions visited during the search. Its purpose is to provide new starting solutions for the TS through selection and combination of routes extracted from the memory. The selection of routes from the memory is probabilistically done and the probability of selecting a particular route depends on the value of the solution to which the route belongs. The selected tours are improved using TS and are subsequently inserted back into adaptive memory. Subsequently, the same strategy to tackle the VRP with soft time windows was used by Taillard, Badeau, Gendreau, Guertin, Potvin, Taillard, and Guertin (1997b). In this problem, lateness at customer locations is allowed, although a penalty is incurred and added to the objective value. To diversify the search by penalising frequently performed exchanges and intensifying the search by reordering the customers within the best routes, were done using Solomon's I1 insertion heuristic recorded by Taillard, Badeau, Gendreau, Guertin, Potvin, Taillard, and Guertin (1997b). A similar strategy for diversification is used by Schulze and Fahle (1999) and Cordeau, Laporte, and Mercier (2001a), however, the intensification is used to reduce waiting time by forbidding certain customers from moving into another route. A similar strategy to adaptive memory is proposed by Schulze and Fahle (1999), wherein all routes generated by the TS heuristic are collected in a pool. At the termination of the local optimisation steps, the worst solution is replaced by a new one created by solving the set-covering problem on the routes in the pool using the Lagrangian relaxation-based heuristic done by Beasley (1990).

A table comparing various implementations of the TS till 2005 can be found in the survey paper by Braysy and Gendreau (2005).

2.3.2 Genetic Algorithm

The GA evolves a population of candidate solutions encoded as chromosomes by creating offspring or progeny through a given number of generations or until some convergence criteria are met. Such criteria may refer to a maximum number of generations, or convergence to a homogeneous population composed of similar progeny. The fittest chromosome generated through this process is then decoded, yielding the optimal or close to optimal solution.

The creation of a new generation of progeny involves four major steps or phases: initialisation, elitism, crossover and mutation. The method emphasises genetic quality while maintaining genetic diversity. Fitness refers to a measure of the value of the objective function which is to be minimised or maximised while exploring the solution space. The recombination or reproduction process makes use of chromosomes of selected parents to produce offspring that will form the next generation. Mutation consists of randomly modifying some chromosome(s) of a single progeny or chromosome at a time to further diversify the solution space and ensure, or preserve genetic diversity. The occurrence of mutation is generally associated with a low probability. A new generation is created by repeating the selection, reproduction, and mutation processes until some stopping condition is met. The set of chromosomes to be created and replaced depends on the elitism strategy applied. In some cases, all chromosomes in the old population are replaced by new ones, and in some cases a set of old chromosomes are preserved. A proper balance between genetic quality and diversity is therefore required within the population to support efficient search.

The GA is an adaptive heuristic search method based on population genetics. The basic concepts were developed by Booker, Goldberg, and Holland (1989), while the practicality of using the GA to solve complex problems was demonstrated by De Jong (1975) and Booker, Goldberg, and Holland (1989).

The GA was first applied to the VRPTW by Thangiah (1995). This GA approach seeks good clusters of customers, within a 'cluster-first, route-second' problem-solving strategy. The routes within each cluster are then constructed with cheapest insertion heuristics, and also λ-exchanges are applied to improve solution quality.

Various hybridised GA method have been produced in attempt to improve both the computational run time and solutions obtained through the GA. These include using GA with various improvement selection heuristics, as well as, other metaheuristics such as TS, Ant Colony Optimisation (ACO) and Hill Climbing (HC), are done by Homberger and Gehring (1999), Homberger and Gehring (2005), Tan, Lee, Ou, and Lee (2001b), Mester (2002), and Le Bouthillier and Crainic (2005).

Important decisions for the GA are: how the fitness value is calculated, the selection scheme for selecting a pair of individuals (parents) for recombination and how recombination is performed. Recombination is a stage in the GA which allows for characteristics of parent chromosomes to be mixed to form offspring. A host of these schemes have been applied by Tan, Lee, Ou, and Lee (2001b), Gehring and Homberger (2001), Homberger and Gehring (2005), and Thangiah, Osman, and Sun (1994).

A comparative table can be found in a survey paper by Braysy and Gendreau (2005), which compares various GA implementations applied to the VRPTW.

2.3.3 Particle Swarm Optimisation

The PSO algorithm is a member of the wide category of Swarm Intelligence (SI) methods for solving global optimisation problems, is a simulation of a simplified social system of birds flocking or fish schooling. This optimisation technique was created by Kennedy (2011). PSO provides a population-based stochastic search procedure in which individuals called particles change their position (state) with time. A population (swarm) of candidate solutions (particles) are updated by *moving* these particles around the search space in accordance to some mathematical formulae relative to the particle's position and velocity. In order to get the swarm to move toward the best solution, each particle's movement to its updated position is influenced by the local and global best position in the search space. The PSO metaheuritic is capable of searching large spaces by each of the particles in the swarm and does not require the use of the problem's gradient, hence the problem does not need to be differentiable like many other classic optimisation techniques such as Global Descent and Quasi-Newton method. The PSO algorithm does not guarantee global optimal solutions to be found because its an stochastic algorithm. For the PSO to be efficiently explore and exploit search spaces, and prevent getting trapped at local optimum positions in a region, different topological schemes may be used to control the flow of information among particles. To prevent premature convergence to a local optimum, yet still have a good rate to reaching an optimum solution, the choice of the method's parameters is crucial. Some important parameters to consider for the PSO include: the number of particles in a swarm, dimension of particles, learning factors, stopping condition and inertia weight. Details on parameter selection are given Shi and Eberhart (1998) and Eberhart and Shi (2000). PSO is attested to be a member of the Swarm Intelligence methods as the swarm adheres to its basic principles which are:

- Proximity
  Performs simple space and time computations.
- Quality
  Responds to quality factors in the environment.
- Diversity
  Does not commit to its activities along excessively narrow channels.
- Stability
  Does not necessarily change every time the environment changes.
- Adaptability
  Changes behaviour when computation cost is not prohibitive.

Similarly to the GA, PSO is an evolutionary algorithm, however, it does not have any evolution operator such as crossover or mutation. The particles of the swarm are updated by moving the particles.

An application of the PSO algorithm to the VRPTW has been done by Zhu, Qian, Li, and Zhu (2006). However, the method was tested on a very small problem of only 8 customers to service. This is far from a realistic sized problem. Thus, the VRPTW has been re-approached by Ai and Kachitvichyanukul (2009). This work makes use of the finding of the application of the PSO to the Capacitated VRP conducted by Kachitvichyanukul (2009). From the experiment results captured by Ai and Kachitvichyanukul (2009), it is found that the PSO time performance is relatively short and linearly proportional to the problem size and it is a useful method to be implemented by practical practitioners. It has also been suggested to further investigate improving the method such that is suitable for a spectrum of problem cases which include data sets that have data that is very sparsely or densely populated, as well as, find means to reduce the empirical computational time such that it is efficient at calculating solutions for large problem cases.

The PSO method has been applied in hybrid form to many related industry problems. Two such instances are, the flow shop scheduling problem by Tasgetiren, Liang, Sevkli, and Gencyilmaz (2007) and the job scheduling problems by Sha and Hsu (2006). Hybrid cases of the PSO technique consider PSO in conjunction to methods such as: Variable Neighbour Search (performs a local search) and Smallest Position Value (a heuristic which transforms position values of particles to job sequences). There are many other extensions of the application of the PSO to the VRPTW. Pongchairerks and Kachitvichyanukul (2009) present a two-level PSO executes which solves MOP in two parts.

2.3.4 Hybrid Metaheuristic Techniques

Simulated Annealing

The Simulated Annealing (SA) technique was first introduced by Kirkpatrick and Gelatt (1983). This technique arises from a physics problem based on a physical annealing process. The SA algorithm can be reshaped to be applied to a spectrum of optimisation problems by tailoring the mathematical constructions of the standard structure. The SA algorithm is a stochastic relaxation technique. This iterative improvement strategy attempts to perturb some current sub-optimal solution towards a more feasible direction. SA for the VRPTW was developed by Chiang and Russell (1996). Various hybridisations of the SA algorithm have been applied to the VRPTW, some of these include:

- A fast SA method based on two-interchanges with best-accept strategy and a monotonously decreasing cooling scheme was developed by Tan and Khoshnevis (2000). Once a set temperature is reached, special temperature resets based on the initial temperature and the current temperature that produced the current best solution. The initial solution is created using a modification of the push-forward insertion heuristic proposed by Thangiah, Osman, and Sun (1994).

- A tabu-embedded SA restart metaheuristic was proposed by Chou, Han, Li, and Lee (2003). Initial solutions are created by the insertion and extended sweep heuristics in Solomon (1987). Three neighbourhood operators based on shifting and exchanging customer segments between and within routes are integrated with a SA procedure that is coerced to restart from the current best solution multiple times. Solomon's insertion procedure is used to reduce the number of routes and to narrow the search by reordering routes and trying to insert customers into other routes. Finally, the search is diversified by performing some random shifts and exchanges of customer segments.

- A basic SA is hybridised with the TS of developed by Tan and Khoshnevis (2000). The initial solution is created with Solomon's I1 insertion heuristic, and the neighbourhood is searched with λ-exchanges using the first-accept strategy. A linear cooling schedule is used, and the search is diversified by randomly shifting and interchanging customers between randomly selected routes.
- A two-stage hybrid metaheuristic is presented by Bent and Van Hentenryck (2004). The first stage is a basic SA used to minimise the number of routes, and the second stage focuses on distance minimisation using the large neighbourhood search. The SA randomly uses the traditional move operators: 2-opt, Or-opt, relocation, exchange and a special evaluation criteria for minimising the number of routes. In addition to route size and minimal delay introduced by Homberger and Gehring (1999), the sum of squares of route sizes is used to favour inserting customers from short to larger routes.

Local Search

The Guided Local Search (GLS) for VRPTW was introduced by Kilby, Prosser, and Shaw (1999). GLS is a memory-based technique developed by Voudouris and Tsang (1998). It operates by augmenting the cost function with a penalty term based on how close the search moves to previously visited local minima, thus encouraging diversification. GLS moves out of local minima by penalising particular solution features it considers should not occur in a near-optimal solution. The penalty awarded is the number of times the feature has already been penalised. The more often a feature appears and is penalised, the less likely it is to be penalised further. The authors choose arcs as the feature to penalise. In the initial solution, no visits are allocated to any vehicle. A penalty is associated with not performing a visit, and so the search process constructs a solution in the process of minimising cost using four different local searches. The local search operators used are 2-opt, relocate and exchange with best-accept strategy.

Ant Colony Optimisation

ACO is also a method classified under the SI family. It is based on the behaviour of ants seeking a path between its colony and source of food. This method was proposed by Dorigo, Caro, and Gambardella (1999). In order to find food, many species randomly explore regions seeking food. Once food is found, ants return to their colony laying a pheromone on the taken path. Pheromone trails can be seen as a indirect low-level communication, as other ants who also are seeking food may use the trail and reinforce the trail if they eventually find food. The strength of the pheromone on the trail is proportional to how frequently a trail is used. That means if a trail has not been used for a long time the pheromones will start evaporating,which will generally be the case for lengthier trails to food sources. Pheromone fading is advantageous to prevent the convergence to a locally optimal solution. ACO is a probabilistic technique, hence the randomness induced allows for diversification i.e. allowing the exploration of routes that have not yet been explored. The algorithm also includes an intensification term which refers to the intensity of the pheromone levels on routes. As ACO is an iterative method, once the stopping condition is met or the last iteration has been completed, the shortest route is selected among the set of routes which is then the final proposed solution.

An ACO approach with a hierarchy of two cooperative artificial ant colonies is used by Gambardella, Taillard, and Agazzi (1999). The first colony is used to minimise the number of vehicles, while the second colony minimises the total travelled distance. The two colonies cooperate through updating the best solution found, and in case the new best solution contains fewer vehicles, both colonies are reinitialised with the reduced number of vehicles.

A comparative table can be found in the survey paper done by Bräysy and Gendreau (2005b), which compares various miscellaneous hybrid metaheuristic implementations applied to the VRPTW.

Chapter 3

Methodology

This chapter discusses the research design for a comparative study of metaheuristic solution techniques used to find solutions to the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW problem instances used are from Solomon's benchmarking data sets. In order to evaluate the computational capabilities of various optimisation solution techniques applied to the VRPTW, a set of test problems were designed and presented by Solomon (1984). This was necessary, given that no benchmark problem sets were available in the prior VRPTW literature. Details of the development of Solomon's benchmarking data sets are given in Section 3.1. The metaheuristic techniques considered to find solutions to the VRPTW are:

- Genetic Algorithm (GA).
- Particle Swarm Optimisation (PSO).

Numerous optimisation metaheuristic technique applications to the VRPTW have been reviewed in prior literature by Braysy and Gendreau (2005). However, these applications are mainly variants or hybridisations of heuristic techniques and not the standard algorithms. Tabu Search (TS) has also been applied to the VRPTW by Taillard, Badeau, Gendreau, Guertin, and Potvin (1997a); Cordeau, Laporte, and Mercier (2001b). Due to TS's poor performance reported in the literature, in terms of solution quality when compared to the best recorded results to date, it is not included or replicated in this thesis. The general method of the metaheuristic techniques considered and the method specific to the VRPTW are outlined in Section 3.2.

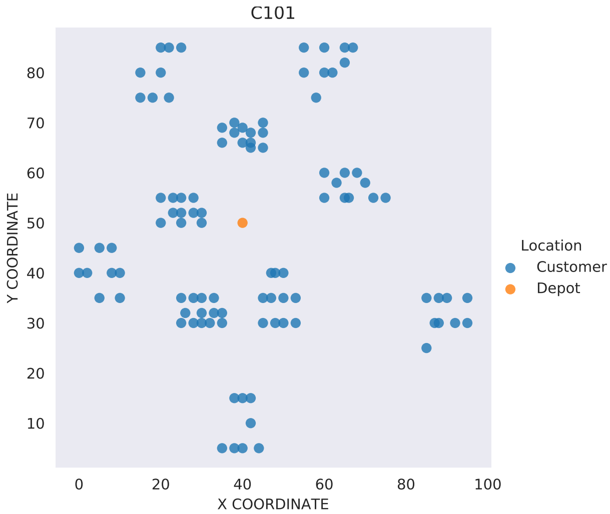
3.1 Solomon Benchmarking Data Sets

Solomon introduced the benchmarking data for the VRPTW in Solomon (1984). The generation of the Solomon Benchmarking instances are based on the standard set of routing test problems data given by Christofides, Mingozzi, and Toth (1979). To highlight factors which influence the performance of the solution techniques applied to the problem, each of the problem sets' characteristics are varied. These factors include:

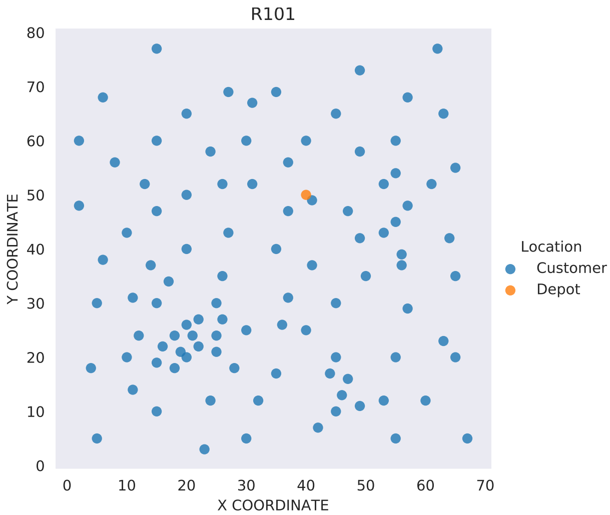
- Geographical arrangement of customers.
- Time window tightness and positioning.
- Number of customers serviced by the same vehicle.

Solomon's benchmarking data sets are divided into classes based on the problem set's spatial and temporal configurations, as stated by Bräysy and Gendreau (2005b). In terms of spatial classes, C1 and C2, have customers located in clusters, and in the R1 and R2 classes, the customers are randomly positioned. The RC1 and RC2 classes contain a mix of both random and clustered geographically located customers. Given the spatial aspects of the data set, the temporal classes of the data set are now addressed. The problems R1, C1 and RC1 have short scheduling horizons, while the problems R2, C2 and RC2 are representative of 'long-haul' deliveries which have longer scheduling horizons. The spatial and temporal selection configuration are interdependent in the data generation. The methods used to obtain the random and clustered classes are respectively described in Section 3.1.1 and Section 3.1.2

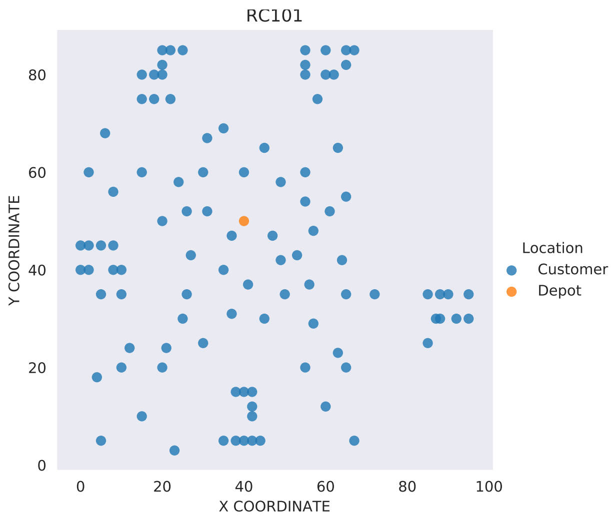
Examples of Solomon's data set classes (clustered, random and random-clustered) are illustrated in Figure 3.1.



(A) Clustered Data Set Class: C101



(B) Random Data Set Class: R101



(C) Random Clustered Data Set Class: RC101

FIGURE 3.1: Topology Examples of Data Set Classes

The six classes of the Solomon benchmarking problems are: R1, R2, C1, C2, RC1 and RC2. Each of these classes contain between 8 and 12 problem instances. Each problem has either 25, 50 or 100 customers, a central depot, a vehicle capacity constraint and a total route time constraint. All problems in any one class have the same customer locations and vehicle capacity; where only the time windows in which to service each of the customers differ.

The details specified for each customer in Solomon's data set is given as follows:

Solomon Data Set: Customer Details							
Customer id	x coordinate	y coordinate	Demand	Earliest Start Time	Latest Start Time	Service Time	Time Window

Each data set's information pertaining to the fleet and customer information can be found at http://web.cba.neu.edu/~msolomon/problems.htm, as well, as at https://github.com/KrupaPrag/VRPTW.

For each customer in the data set the following are stipulated: customer's location, load demand, service time, and service time window (earliest and latest time to start servicing a customer).

3.1.1 Random Class

A random uniform distribution is used in generating the data for the random classes of the Solomon benchmarking instances. The method used for the random generation of the time window constraints for the data sets R1, R2, RC1 and RC2 consisting of n customers is now presented.

1. Select a percentage γ of customers to receive time windows, where $0 \leq \gamma \leq 1$.
2. Given that n represents the number of customers in a customer set, generate n random numbers from the random uniform distribution over the interval $(0, 1)$. These random numbers are then paired with each of the n customers. Based on the random value associated with each of the customers, order the customer list in ascending order. This approach creates a random permutation of customers.
3. Let ζ be the integer that most closely approximates $\gamma \times n$. Then the first ζ customers from the randomly permuted list of customers are assigned time windows. The time windows have a randomly generated center and width. The center of the time window for customer c_i, $\{i \in [1, n] \, | i \in \mathbb{N}\}$, is a uniformly distributed, randomly generated number in the interval:

$$(e_0 + t_{0,c_i},\ l_0 - t_{c_i,0} - s_{c_i}).$$

   The width of the time window for customer c_i is half the width as a normally distributed random number.

3.1.2 Clustered Class

The problem sets C1, C2, RC1 and RC2 are composed of structured problems as the customers appear in clusters and the time windows are positioned around the arrival times at customers. This approach permits the identification of cluster-by-cluster solution which provides an additional means of evaluating the performance of applied heuristics.

The 3-opt routine described by Lin (1965) is used on each cluster to create routes and then produce schedules by selecting an orientation for each cluster. The time window constraints are generated by choosing the center as the arrival time at each customer; the width and density are derived in Section 3.1.1.

In terms of time window density (the percentage of customers with time windows), the problems have 25%, 50%, 75%, and 100% time windows. The length of the route-time constraint acts like a capacity constraint which, together with the vehicle capacity constraint, allows only a few customers to be serviced by a particular vehicle. Short horison problems have vehicles that have small capacities and short route times, and cannot service many customers at one time. In contrast, the classes R2, C2 and RC2 which have long scheduling horisons; this characteristic permits many customers to be serviced by the same vehicle. Hence requiring fewer vehicles to service the customers in group 1's data sets.

All the test problems are 100-customer euclidean problems. Travel times between customers are taken equal to the corresponding distances. The fleet is assumed to be homogeneous.

3.2 Metaheuristic Techniques

The metaheuristics applied to the VRPTW in this thesis are the GA and the PSO algorithm. These techniques are respectively described in Section 3.2.1 and Section 3.2.4.

3.2.1 Genetic Algorithm

The GA is a stochastic method used to find solutions to various optimisation problems. The GA belongs to the larger class of Evolutionary Algorithms (EA). EAs are generic population-based metaheuristic methods that facilitate optimisation processes by making use of mechanisms that mimic the natural evolutionary process, as described by Engelbrecht (2006a). Evolution through natural selection describes a process whereby progressive change occurs over generations in the inherited genetic material and characteristics of a species population. These changes have positive implications on the population, which is the goal of these methods.

The foundation of the GA is based on Darwin's theory of evolution by natural selection. In nature, evolution is the process whereby species that have characteristics best adapted for their environment are inclined to survive and reproduce. Over successive generations, these favourable characteristics become dominant in the population as they are passed down to their progeny from generation to generation. Through computational models, EAs extend the idea of organisms improving their adaptation capabilities over successive generations to survive dynamically changing and reforming environments. Engelbrecht (2006a) states that adaptation occurs through reproduction, mutation, competition and symbiosis.

The process of natural selection is driven by selective pressure. Selective pressures are environmental factors which may restrict reproduction in a species population, and leads to evolutionary changes in a population. A famous example of the effects of selective pressure on a species in an ecosystem is the evolution of long neck and long leg giraffes. As a results of food sources being subjected to large competition in the giraffe habitat, selection pressure favoured specimens in the population with both longer necks and legs. As these taller individuals could reach higher vegetation, they were able to possibly survive any struggle for existence. This means that the surviving giraffe, the healthiest and taller giraffe, is most likely to pass these traits to the offspring they reproduce. Hence, the tall phenotype becomes dominant over generations. This example is described in further detail by Darwin (2004) and Simmons and Scheepers (1996).

In attempt to tackle real-world problems using EAs, the problem would have to be constructed to represent a population-based search space. Thus, to find a solution to the optimisation problem, it is required to project the problem onto the components of the GA in order to apply the GA's operators. In the GA, a population is representative of a search space S. A population Γ consists of Z individual organisms. Each individual is referred to as a chromosome κ, which is representative of a candidate solution to the optimisation problem. The problem's variables, for which an optimal configuration is sought, are referred to as genes. Genes are unit information which make up and characterise a chromosome. Each of the problem's ξ variables are represented as a dimension in the chromosome. A representation of a population Γ of Z individual ξ-dimensional chromosomes are given in Figure 3.2. Each chromosome κ is indexed by b, $\{b \in [1, Z] \mid b \in \mathbb{N}\}$. The dimension's of each chromosome κ is indexed by η, $\{\eta \in [1, \xi] \mid \eta \in \mathbb{N}\}$.

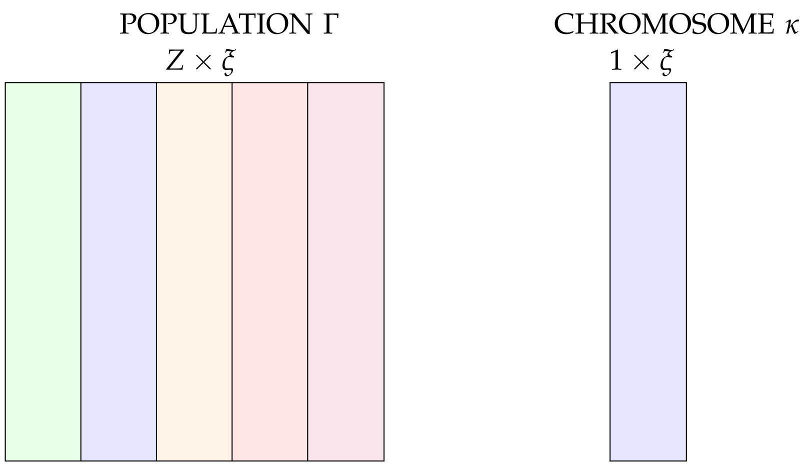


FIGURE 3.2: Population Based Structure

Traditionally, a standard representation of each candidate solution is an array of bits. That is a string of 0's and 1's. An example of a binary representation of candidate solutions can be given as:

Binary Chromosome Representation

0 1 0 0 1 0 1 1 1 1

In the GA, the process of searching a search space is represented by a population of chromosomes undergoing an evolutionary process. This process starts from an initial population of randomly generated individuals. The progressive change of the population occurs by iteratively applying an update to the population at each unit time-step. Each iteration or unit time-step τ is referred to as a generation. In each generation, the fitness of every individual in the population is evaluated. The fitness function g, is usually the value of the objective function of the optimization problem being solved, which is used to evaluate and quantify the design of the chromosome. The *fitter* individuals are stochastically selected from the current generation's population, undergo modification and form part of the next generation's population. The modification entails applying the following operators to the population: selection, crossover and mutation. The population is repeatedly updated by these operators until some terminating criterion is met. The general structure of the GA is given by Algorithm 1.

Algorithm 1 Generic Evolutionary Algorithm

procedure GENERIC EVOLUTIONARY ALGORITHM(Z, ξ, g) ▷ Number of individuals, dimension of each individual, fitness function.
    $\tau \leftarrow 0$ ▷ Initialise generation counter.
    Create and initialise a ξ-dimensional population Γ consisting of Z individuals
    Evaluate the fitness g at τ of each individual κ_b, where $b \in [1, Z]$
    while stopping condition(s) not true **do**
        Perform reproduction to create offspring
        Evaluate the fitness g at τ of each individual κ_b, where $b \in [1, Z]$
        $\tau \leftarrow \tau + 1$
        Select new population Γ at τ
    end while
    return *Fittest* individual in Γ
end procedure

The evolutionary search process is influenced by the following main components of the EA, as listed by Engelbrecht (2006b) are:

- Encoding of the initial population.
- Fitness function.
- Selection.
- Reproduction.
- Stopping criterion.

Encoding of the Initial Population

EAs progressively change a population of candidate solutions as it explores a search space. The first step in applying an EA to solve optimisation problems is to encode an initial population of feasible candidate solutions based on the EA's structure. Diaz-Gomez and Hougen (2007) claims that the probability of finding *good* solutions to the problem is dependent on the initial population. It must be ensured that the initial population represents a wide region of the search space to avoid neglecting uncovered regions during the search process, states Engelbrecht (2006b) and Diaz-Gomez and Hougen (2007). The performance of the algorithm is influenced by the measure of the population's diversity, as it guides the algorithm to avoid premature convergence, claim Burke, Gustafson, and Kendall (2004), Zitzler, Deb, and Thiele (2000), and Lobo and Lima (2005). The computational complexity and exploration abilities are impacted by the set-up of the initial population. The diversity of a population is increased by having a larger population as it broadens the exploration space. However, the larger the population, the higher the computational complexity per generation. In contrast, smaller populations limit the search space whilst being computationally less demanding per generation, it may take many more generations to converge than it would for a larger population.

The diversity in characteristics of the chromosomes in the population are dependent on the encoding schemes used to generate these initial feasible solutions. A standard means of encoding an initial population is to assign random values from the allowed domain to each of the genes of the chromosomes. An initial population may also be *seeded* with feasible solutions based on the information given about the problem and it's solution generation, as described by Diaz-Gomez and Hougen (2007). These *seeded* solutions are solutions in the region where optimal solutions are likely to be found.

Fitness Function

As a result of chromosome representation not necessarily representing the objective function, a fitness function g is defined to provide a means to measure the characteristic design of the chromosome. The fitness function defined by Equation (3.1) maps a chromosomes to a scalar value, whereby Ψ represents the data type of the elements of the ξ_η-dimensional chromosome.

$$g : \Psi^{\xi_\eta} \rightarrow \mathbb{R}\,, \xi_\eta \in \mathbb{R}. \tag{3.1}$$

It is important to emphasises the role of the fitness function in EAs. The evolutionary operators usually make use of the fitness function to evaluate a chromosome and make a decision in selecting a chromosome.

Selection

Selection plays an important role in improving a population from generation to generation. Selection schemes mimic selection pressures found in natural habitats. Selection ensures that favourable characteristics or genetic material are encompassed in generating an updated population, to accentuate solutions. EAs apply selection schemes to ensure that the individuals with favourable characteristics influence the updated population by either directly or indirectly passing their characteristic traits to the next generations' population. In the case of direct influence, a selected number of individuals are directly passed from the current generations' population to the next without any modification. Indirect influence of these favourable characteristic traits would be to stochastically select the top ranking fittest individuals to form part of the pool of the breeding individuals. These individuals are referred to as parental chromosomes. Through reproduction, offspring are produced which encompass modified characteristic traits of their parents.

There are a number of selection schemes that have been developed to be used in the selection stage of the GA. Engelbrecht (2006a) states the most commonly used schemes are: random selection, tournament selection, proportional selection and elitism.

- **Random selection**
  Each individual has the same probability to be selected from a given population and proceed to the next generation. This scheme does not consult the fitness value of the chromosome, hence has a low selective pressure in comparison to other selection schemes.

- **Tournament selection**
  In the tournament selection scheme, a number of individuals are randomly selected from a population. These selected individuals are compared against each other with respect to their fitness. The individual with the best fitness value is selected and returned by the operator. The incorporation of the random selection scheme in this method reduces the selection pressure and prevents the best individuals from being dominant in the EA.

- **Proportional selection**
  The proportional selection scheme biases selection towards the most fit individuals. A probability distribution proportional to the fitness is created and the individuals are selected by sampling the distribution. A popular sampling method used in proportional selection is Roulette wheel sampling. In Roulette Wheel sampling, the fitness values of each individual in a population is normalised (i.e. divide each individual's fitness value by the maximum fitness value in the population). Since the *size* of each individual is proportional to the normalised selection probability, the probability distribution can be compared to a Roulette wheel. Similar to spinning a Roulette wheel and recording which slice ends up at the top, corresponding to the GA, the recorded slice is the individual selected. The Roulette wheel selection scheme results in high genetic variance in the updated population, however, does not necessarily incorporate the fittest individual in its selection.

- **Elitism**
  The process of elitism ensures that the fittest individual is copied to the next generations' population without undergoing any mutation. The diversity of a population reduces when a large number of individuals directly survive to the next generation without any modification.

Reproduction

In the GA, once the initial population has been encoded to represent the optimisation problem, the next stage is to iteratively update the population through an evolutionary process. The next generations' population of the solutions are generated using the crossover and mutation operators. For each new offspring generated, a pair of potential chromosomes are selected from the pool of breeding chromosomes previously selected using a selection scheme. An offspring is produced by applying the crossover and mutation operator to the pair of parent chromosomes. In general, the application of the genetic operators improve the average fitness of the population. The reason for this is that the pool of breeding individuals have the most favourable characteristics of the population obtained through the selection scheme. Crossover and mutation are seen as the main genetic operators in the reproduction phase, however, some other operators used include: regrouping, colonisation-extinction or migration, as described by Engelbrecht (2006a)

- **Crossover**
  In the GA's reproduction stage, the crossover operator combines genetic material of two parental individuals to generate a new offspring as a means to stochastically generate solutions for the next generation's population. This operator represents sexual reproduction in biology. The crossover operation results in two offspring, each carrying some genetic information from both parents. There are a number of crossover methods: single point crossover, two-point and *k*-point crossover, uniform crossover, and crossover for ordered lists.

  - **Single point crossover**
    A random point is picked in both parent chromosomes. This is called the crossover point or chiasma. Bits to one side of this point are swapped between the parent chromosomes. This results in two offspring, each carrying some genetic information from both parents. This is illustrated in Figure 3.3.

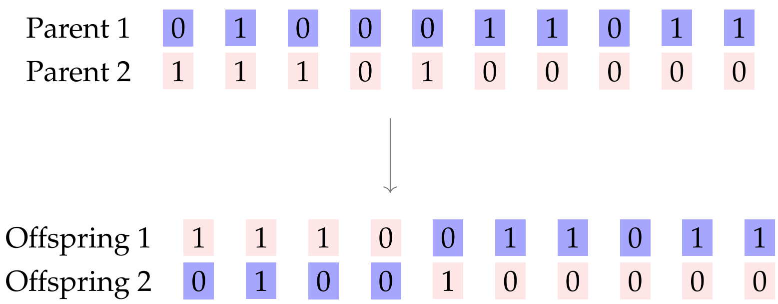


FIGURE 3.3: Single Point Crossover

  - **Two-point crossover**
    In two-point crossover, two crossover points are picked randomly from the parent chromosomes. The bits in between the two points are swapped between the parent organisms. This is illustrated in Figure 3.4.

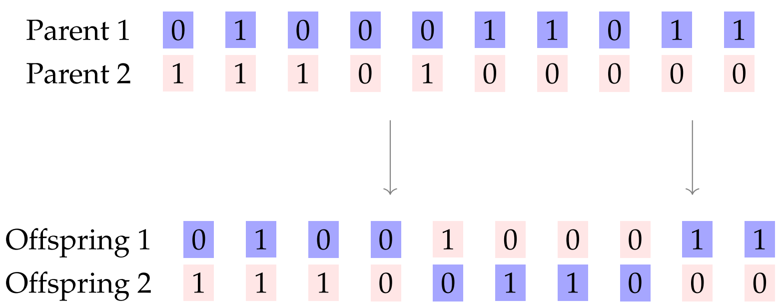


FIGURE 3.4: Two-point Crossover

  - **Uniform crossover**
    In uniform crossover, the exchange of genetic material occurs per bit and not in segments like point crossover. Hence, each bit forming part of the offspring is independently chosen from two parents with respect to a given distribution. This also means that the proximity in the string of bits does not necessarily influence if traits are inherited in conjunction to each other.

- **Mutation**
  Genetic diversity in subsequent generations' populations is brought about by the mutation operation. The mutation operator alters one or more of the individual's characteritics from it's initial state. Mutation occurs in the reproduction stage of the GA according to a defined mutation probability ϱ. The probability of mutation occurring is important as, if this probability is too high, the search may evolve to be a primitive random search; whilst, if too low, it could lead to not bringing about adequate genetic diversity.

  An example of a bit wise mutation occurs, given that the mutation probability is satisfied. An illustration of the mutation operator is shown in Figure 3.5.

Initial State of offspring 1:	1	1	1	0	0	1	1	0	1	1
Mutated State of offspring 1:	1	0	1	0	0	1	1	0	1	1

FIGURE 3.5: Mutation

Stopping Criterion

In EAs, the evolutionary operators, selection, crossover and mutation, are iteratively applied until a condition is satisfied. Common terminating criterion include:

- Updating the population for a set number of generations.
- Terminating the updating procedure when the highest ranking solution's fitness is reaching or has reached a plateau.
- A solution is found that satisfies minimum criteria.
- Allocated computation time is reached.

3.2.2 Genetic Algorithm for the Vehicle Routing Problem with Time Windows

The VRPTW's objective is to design routes for delivery vehicles to service customers at their specific geographic locations, with various demands, within predefined time windows without violating the total capacity and total time constraint for each vehicle. The goal is to execute the delivery procedure while minimising both the number of dispatched vehicles and the total travelled distance by the vehicles. Considering the nature of the problem, the VRPTW is classified to be both a Combinatorial Optimisation Problem (COP) and a Multi-objective Optimisation Problem (MOP). EAs are widely applied to problems of these classification, state Ombuki, Ross, and Hanshar (2006a). The GA, a particular EA is an adaptive metaheuristic that stimulates the optimisation process with mechanisms based on natural evolution. The success of any EA is determined by whether or not the problem can be projected on the structure of EA's solution space, and if the general application of the evolutionary process could progressively navigate the search space to find *good* solutions to the problem. In order to find solutions to the VRPTW using the GA, it is pertinent to first transform the problem to be adapted to the GA's solution space structure so that it's operators can be applied.

In the GA, the population Γ is represents a search space. The population comprises of candidate solutions κ_b, $\{b \in [1, Z], b \in \mathbb{N}\}$. These candidate solutions or chromosomes are feasible solutions to the optimisation problem. Specific to the VRPTW, each chromosome is comprised of multiple routes designed for vehicles to service all the customers, without violating any of the problem's constraints. Each of these routes are a configuration of a set of customers that the particular vehicle sequentially services. The population undergoes an evolutionary process by applying the standard GA's operators: selection, crossover and mutation. These operators are tailored specific to the VRPTW such that when applied they produce feasible solutions. Thus, avoiding the need to check for constraint violations and further needing to apply repair mechanisms. Thus, increasing the efficiency of the algorithm.

An example candidate solution κ comprises of multiple feasible service routes R_ϑ. The index ϑ, is a number between 1 and the total number of routes found in κ. An example representation of a candidate solution comprising of three service routes which services ten customers is shown in Figure 3.6.

Route:		R_1			R_2				R_3				R_4	
Route List:	5	8	6	|	1	|	9	3	7	4	|	10		2

FIGURE 3.6: Example of a Candidate Solution

The application of the GA to find solutions to the VRPTW, discussed in this section, is based on the implementation method presented by Ombuki, Ross, and Hanshar (2006a).

Encoding of the Initial Population

The initial population structure of the GA for the VRPTW, is encoded in two phases. The first phase generates a population of individual feasible solutions. The second phase applies a routing scheme to each individual in the population.

- **Phase I**
  In the first phase of initialising a population, candidate solutions to the VRPTW are generated. Each candidate solution is defined by multiple routes scheduled to service all the customers. Each route is representative of a vehicle dispatched to service a set of customers. The population's chromosomes (i.e. feasible route schedules) are encoded using two methods:

  - Random Permutation Encoding.
  - Greedy Encoding.

  A percentage μ of the population's individuals are encoded using Random Permutation Encoding, and $1 - \mu$ percent of the population's individuals are encoded using the Greedy Encoding method.

  - **Random Permutation Encoding**
    For the random permutation encoding, generate a list of randomly ordered customers (RCL) from the list of $|\mathbf{C}|$ serviceable customers. First, a route list is initialised. Through an iterative process, the route list with

multiple service routes is populated. The encoding begins by initialising a new route. The *RCL* is sequentially iterated. Given that the encountered customer in *RCL* does not violate any of the problem's constraints, it is appended to the current route. However, if the encountered customer cannot be appended to the current route, a new route is initialised with this customer. This process is repeated until all the customer's in *RCL* are used in populating the service routes in route list.

Consider the following example. Let a given example customer set consist of 5 customers. The details associated with each of the customers are given in Table 3.1.

Example Customer Data Set				
Customer	Demand	Service Time	Earliest Start Time	Latest Start Time
c_1	5	10	5	25
c_2	8	10	3	15
c_3	3	10	21	40
c_4	10	10	30	55
c_5	12	10	22	38

TABLE 3.1: Example Customer Data Set

The associated distance matrix stipulating the distance between any two locations (customers and/ or the depot c_0) are given in Table 3.2.

Vertex	$\mathbf{c_0}$	$\mathbf{c_1}$	$\mathbf{c_2}$	$\mathbf{c_3}$	$\mathbf{c_4}$	$\mathbf{c_5}$
$\mathbf{c_0}$	∞	3	4	2	3	1
$\mathbf{c_1}$	3	∞	2	3	5	3
$\mathbf{c_2}$	4	2	∞	3	1	4
$\mathbf{c_3}$	2	3	3	∞	3	2
$\mathbf{c_4}$	3	5	1	3	∞	8
$\mathbf{c_5}$	1	3	4	2	8	∞

TABLE 3.2: Example Distance Matrix

Let the load capacity constraint for this example be 15 units, and the total travel time constraint be 50 units.

Given the customer list:

$$c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5.$$

The list is randomly ordered to return an *RCL*:

$$c_4 \quad c_2 \quad c_5 \quad c_1 \quad c_3.$$

Sequentially condsidering the customers in *RCL* and the Random Permutation Encoding scheme, the solution generated is given in Table 3.3.

Example Solution by Random Permutation Encoding				
Route ID	Route	Load	Total Time	Reason for New Route Initialisation
R_1	0-c_4-0	10	16	First customer.
R_2	0-c_2-0	8	18	Time constraint violation.
R_3	0-c_5-0	12	12	Load constraint violation.
R_4	0-c_1-c_3-0	8	33	Time constraint violation.

TABLE 3.3: Example Solution by Random Permutation Encoding

The pseudocode of the Random Permutation Encoding of an chromosome in a population is given in Algorithm 2.

Algorithm 2 Random Permutation Encoding

procedure RANDPERMUTATIONENCODING($\mathbf{C}$, $distMat$,Q) ▷ Input: customer data set, distance matrix, vehicle capacity.
  $numCust \leftarrow |\mathbf{C}|$
  $\kappa \leftarrow$ Initialise route list
  $\vartheta \leftarrow 1$
  $R_\vartheta \leftarrow$ Initialise route
  $RCL \leftarrow$ Randomly ordered $\mathbf{C}$ ▷ Customers randomly arranged.
  for $\psi \leftarrow 0$ to $numCust$ **do**
    $c_\psi \leftarrow RCL\,[\delta]$
    if Appending c_δ to R_ϑ does not violate the problem constraints **then**
      c_ψ is appended to R_ϑ
    else
      $\vartheta \leftarrow \vartheta + 1$
      c_ψ is appended to R_ϑ
    end if
  end for
  return κ ▷ A list of ordered customers specific to the route they belong, to form a chromosome.
end procedure

- **Greedy Encoding:**
  Once an initial route list has been initialised, the list of customers to be serviced are iteratively considered to populate the route list. The nearest feasible neighbour within some empirical raduis *rad* to the current position is selected to be appended to the current route and is removed from the list of customers to be serviced. This process is repeated until no such neighbours exist, however, a new route is initialised with this customer if the customer list is not empty.

  The Greedy Encoding method is given in Algorithm 3:

Algorithm 3 Greedy Encoding

procedure GREEDYENCODING($\mathbf{C}, distMat, Q, rad$) ▷ Input: customer data, distance matrix, vehicle capacity, empirical radius.
    $numCust \leftarrow |\mathbf{C}|$
    $\kappa \leftarrow$ Initialise route list
    $\vartheta \leftarrow 1$
    $R_\vartheta \leftarrow$ Initialise route
    while $\mathbf{C} \neq \Phi$ **do** ▷ While customer set not empty.
        if A feasible customer c_ψ within rad exists **then**
            Append c_ψ tot R_ϑ
            Remove c_ψ from $\mathbf{C}$
        else
            $\vartheta \leftarrow \vartheta + 1$
            $R_\vartheta \leftarrow$ Initialise route
        end if
    end while
    return κ ▷ A list of ordered customers specific to the route they belong, to form a chromosome.
end procedure

- **Phase II:**
  Once the initial population is encoded using the Random Permutation Encoding method or the Greedy Encoding method, a routing scheme is applied to each individual in the population. Phase II of the initial encoding attempts to reduce the cost to service the customers; that is to reduce the number of vehicles dispatched and their respective travel distances. Given a chromosome made of b routes, then $\{\vartheta' \in [1, b-1] \mid b \in \mathbf{N}\}$. The last customer of each route $R_{\vartheta'}$ is relocated to be the first customer of the next route $R_{\vartheta'+1}$. Updated $R_{\vartheta'}$ and $R_{\vartheta'+1}$ are accepted if the summed updated distance is less than the original routes summed distance, or if the two routes have been reduced to a single route, else the original routing topology is maintained.

Selection

At every iteration of the generating stage, individuals are selected through a selection scheme from the current generations' population which will directly or indirectly influence the genetic composition of the next generations' population. The selection schemes select individuals with respect to their evaluated fitness value. Two particular selection schemes used in generating the next generations' population are:

- Elitism.
- Tournament selection.

The influence and layout of these two schemes are now discussed.

- **Elitism:**
  The elite model carries the fittest individual, without alteration, from the current generations' population to the next generations' population. This ensures that the best solution to the problem does not deteriorate in fitness over successive generations. However, the best chromosome is faced to compete with the new fittest individual produced through reproduction.

- **Tournament selection:**
  The breeding population is selected through tournament selection. This scheme randomly selects a set of K individuals from the current generation's

population. These individuals make up the tournament set. A random number *rand* between 0 and 1 is generated. If *rand* is less then some threshold probability ρ, the fittest individual from the tournament set is selected, else any individual from the tournament set is selected. Two chromsomes are selected through this process to be parental chromsomes to be used in reproduction.

The pseudocode of the tournament selection scheme for the parent pair selection is given in Algorithm 4:

Algorithm 4 Parent Pair Selection

```
procedure ParentPair(K, ρ, Z, Γ)   ▷ Input: number of chromosomes selected for the tournament set,
benchmark probability for selection,number of chromosomes in the population, population
    Parents ← initialise a list to store the parent pair
    for ψ ← 0 to 2 do
        r ← a random number between 0 and 1
        T ← a list of K randomly selected chromosomes from the Z chromosomes in the population Γ
        if rand < ρ then
            Parents [ψ] ← fittest chromosome in T
        else
            Parents [ψ] ← randomly select chromosome from T
        end if
    end for
    return Parents                                   ▷ Parent chromosomes.
end procedure
```

Reproduction

The reproduction operators, crossover and mutation, are tailored specific for the VRPTW. This stage of the evolutionary process aims to minimise the total cost of servicing customers by minimising both the number of vehicles to be dispatched and their accumulated travelled distance. This procedure is carried out simultaneous to checking that the feasibility constraints are not being violated. The tailored crossover and mutation operators are now discussed.

- **Crossover:**
  The crossover operation requires two parent chromosomes (Parent 1 (P_1) and Parent 2 (P_2)). These parent chromosomes are selected from the current generation's population using the tournament selection scheme given in Algorithm 4. In conjunction to the example below, the steps of the crossover operator is explained as follows:

  1. Randomly select a route from each of the parent chromosomes, P_1 and P_2. Let Route A (R_A) and Route B (R_B) be respectively selected from P_1 and P_2. In the example, R_3 and R_2 are respectively selected from P_1 and P_2.
  2. The customers found in the selected routes, R_A and R_B, must be removed from the opposing parent chromosomes. That is the customers in R_A are removed from P_2 and the customers in R_B are removed from P_1.
  3. The customers removed from the particular parent chromosome must then be reinserted into the same parent chromosome by seeking all possible positions in the chromosome for which the problem's constraints are not violated. The customers are then inserted into the chromosome at a position which possibly improves the chromosome's fitness value (i.e. minimising the route cost). It must be noted that the if

the customer considered for insertion does not find a feasible position in any of the current routes, a new route is initialised with this customer.

An example application of the crossover operator is illustrated in Figure 3.7.

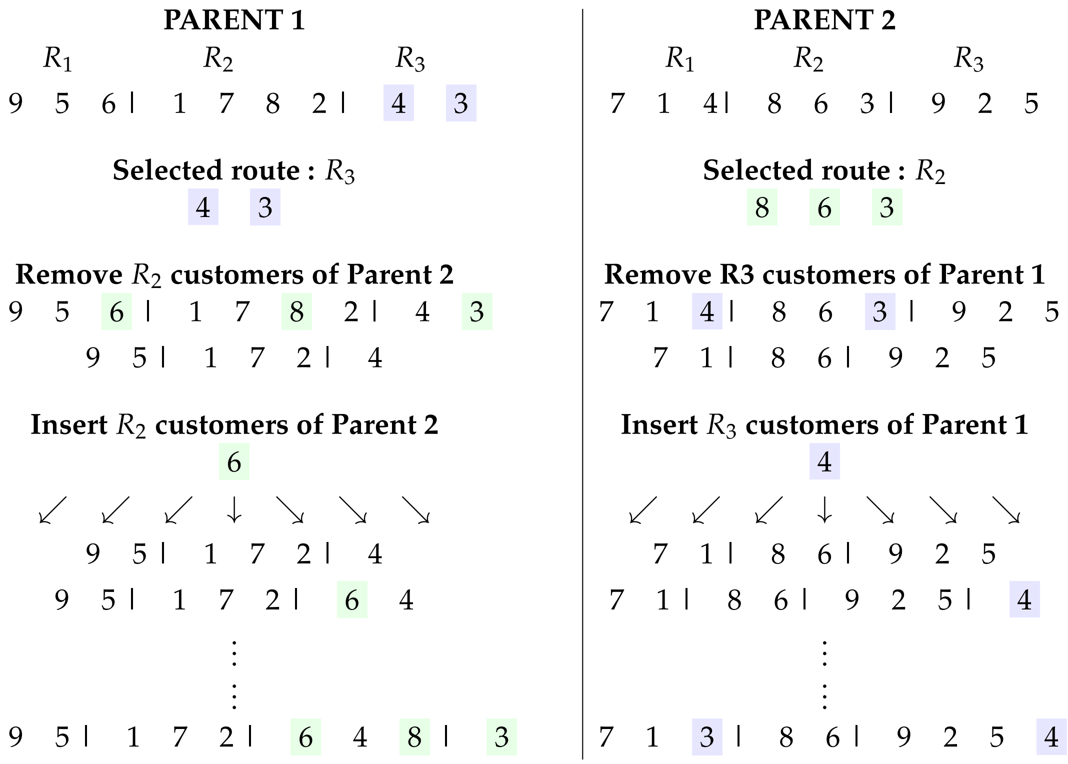


FIGURE 3.7: Example of the Crossover Operation

- **Mutation:**
  In order to break the GA free from fixation at any given point in the search space, mutation is implemented. A constrained route reversal mutation is applied, which is an adaptation of the simple more commonly used mutation called inversion. In this particular GA implementation, mutation is carried out by randomly selecting a route in the newly generated child chromosome which is of length two or more. A subset of the route is selected which is either of length two or three. The subset route is then reversed to create a mutated route. This mutated route is evaluated. If no constraints are violated and the route fitness is better than the current route's fitness then the mutation is accepted and the route is updated. This operation aids the search from converging at a local optima.

  For example, consider a particular chromosome comprising of multiple routes as shown in Figure 3.8.

R_1 R_2 R_3

9 5 | 1 7 8 2 | 4 3

FIGURE 3.8: Example of a Chromosome

Randomly select a route that is of length two or more. Let R_2 be chosen for this example. If the route's length is greater than length two, select a subset of the route which is of length two or three. The selection from R_2 is shown in Figure 3.9.

9 5 | 1 7 8 2 | 4 3

FIGURE 3.9: Selection of a subset of a route for mutation

Reverse the selected sub-route of R_2 and verify if any violation constraints are incurred. If none, compare the altered route's cost to the original. If the altered route's cost is lower, update the route to the altered route. The selected sub-route of R_2 route reversal is illustrated in Figure 3.10.

1 2 8 7

FIGURE 3.10: Sub-route reversal

The updated candidate solution after mutation is illustrated in Figure 3.11.

9 5 | 1 2 8 7 | 4 3

FIGURE 3.11: Candidate solution after mutation

Stopping Criterion

The GA applied to the VRPTW terminates the evolutionary process after a predetermined number of generations Y.

3.2.3 Genetic Algorithm Implementation Overview

The main features of the GA are tied together in the GA implementation which seeks solutions to the VRPTW. An overview of the GA is given in Algorithm 5.

Algorithm 5 Genetic Algorithm (GA)

procedure GENETICALG(Z, Y, g) ▷ population size, stopping condition, fitness function.
  $\Gamma \leftarrow$ initialise population of size Z using initial encoding schemes
  Calculate the fitness g for each chromosome κ and rank them
  for $\psi \leftarrow 0$ to Y **do**
    Keep the highest ranked chromosomes from Γ for Γ'
    Apply crossover method to generate the remaining chromosomes of the new population Γ'
    Apply mutation operation to each of the chromosomes in the new population Γ'
    Calculate the fitness g of each chromosome κ in Γ' and rank them
    $\Gamma \leftarrow \Gamma'$
  end for
  return Highest ranked chromosome in final generation's population Γ
end procedure

3.2.4 Particle Swarm Optimisation

The PSO algorithm is a global search method used in finding solutions to global optimisation problems. The concept of the PSO algorithm originates from the simplified social system of a flock of birds searching a finite area for food, without being coordinated by a control system. This means none of the birds have any knowledge of the locality of the food, but each know how far they are from the food at any given unit time-step. The problem solving behaviour which emerges from the interaction and cooperation of each individual is referred to as Swarm Intelligence (SI). SI is a property of a system whereby the collective behaviour of unsophisticated individuals locally relate with their environment and neighbouring individuals causing coherent functional global patterns to emerge in the complex swarm system, as described by Engelbrecht (2006b).

PSO is a population-based stochastic search procedure which is based on the behaviour simulation of a flock of birds where social sharing of information takes place. Formally, a swarm is commonly referred to as a population. Each individual in the population is called a particle. A particle is representative of a candidate solution. As a swarm would fly towards more promising regions of a landscape; similarly, particles in a population explore a search space. The movement of the particles in a search space is governed by a few simple formulae which use the particle's personal memory and the swarm's global memory. Personal memory refers to the best position held by each particle in the search space, while global memory refers to the best position discovered over the iterations up until the current iteration by the entire swarm. At each time-step the population generation is updated, that is if improved positions are discovered then personal and global memory are respectively updated.

The question arises, how is it determined whether or not the updated position, for a particular particle, is *better* that it's personal best known position? In order to quantify the performance of the population's particles moving in a search space, the PSO algorithm utilises a reward scheme. Each particle in the population is evaluated based on it's current position $\mathbf{x}$, which is used to obtain it's fitness value calculated using the problem's objective function f. For global optimisation problems the search for a solution in a search space S, is identified by the discovery of the global optimiser of the real-valued m-dimensional objective function $f : \mathbb{R}^m \to \mathbb{R}$. Given a real-valued objective function $f(\mathbf{x})$, which is defined on $\mathbf{x} \in S$. Without loss of generality, it is assumed that the m-dimensional objective function f is being minimised.

Particle Swarm Optimisation Algorithm's Components

To generalise the PSO algorithm, it's components are now formalised. Given a population P consisting of N particles, where each particle is uniquely indexed by h, $\{h \in [1, N] \mid h \in \mathbb{N}\}$. At any given discrete time-step or iteration k, where $k \in \mathbb{N}_0$, each particle h has the following information stored in it's personal memory:

- $\mathbf{x}_k^h$: current position.
- $f(\mathbf{x}_k^h)$: fitness value corresponding to $\mathbf{x}_k^h$.
- $\mathbf{pb}_k^h$: personal best position.
  A position visited by particle h over the previous k iterations with the minimum corresponding fitness value in the set $\{\mathbf{x}_0^h, \mathbf{x}_1^h, \ldots, \mathbf{x}_{k-1}^h, \mathbf{x}_k^h\}$.

- $f(\mathbf{pb}_k^h)$: fitness value corresponding to $\mathbf{pb}_k^h$.
  Where $\mathbf{pb}_k^h$ is temporarily the optimal position of particle h discovered over k iterations.

The algorithm creates a global memory for the entire population of particles by recording the population's best with the following:

- $\mathbf{gb}_k$: global best position.
  The best position discovered over the k iterations for the entire population. That is the best position from all the N particle's personal best positions $\{\mathbf{pb}_k^1, \mathbf{pb}_k^2, \ldots, \mathbf{pb}_k^N\}$.
- $f(\mathbf{gb}_k)$: fitness value corresponding to $\mathbf{gb}_k$.

Initialisation

At the start of the algorithm, when $k = 0$, the particles are randomly distributed over the search space S. Considering an m-dimensional search space, m random values within the search space parameters are generated defining the coordinates of each particle h's position. The particle's personal initial best position is set to its current positions (randomly generated positions), i.e. $\mathbf{pb}_0^h = \mathbf{x}_0^h$, $\forall\, h \in [1, N]$. The initial global best position is defined as the best location held by one of the particles at the start of the algorithm, i.e. $\mathbf{gb}_0$ is set to the position with the minimum corresponding fitness from the set $\{\mathbf{pb}_0^1, \mathbf{pb}_0^2, \ldots, \mathbf{pb}_0^N\}$. The initial velocity $\mathbf{u}_0^h$ for each particle $h \in [1, N]$ is defined as an m-dimensional vector. These values are sampled from a uniform distribution within bounds of the search space. The initial random velocities associated with each particle are used to set the swarm into motion.

Position Update

Once the population is initialised, the search for the optima begins by iteratively updating the population generation. In order to update the population, each particle's position for the next time step $k + 1$ is calculated. The updated position of particle h, $\mathbf{x}_{k+1}^h$, is it's current position $\mathbf{x}_k^h$ with an added velocity $\mathbf{u}_k^h$ calculated at the next time-step $k + 1$. Mathematically, this is represented by Equation (3.2).

$$\mathbf{x}_{k+1}^h = \mathbf{x}_k^h + \mathbf{u}_{k+1}^h. \tag{3.2}$$

Velocity Update

The optimisation process is driven by the velocity component. It refers to both:

- The experiential knowledge of the particle.
- The socially exchanged information from the particle's neighbours' experiences in the population.

The velocity update is defined by the Equation (3.3).

$$\mathbf{u}_{k+1}^h = \mathbf{u}_k^h + \varsigma_1 \sigma_1 (\mathbf{pb}_k^h - \mathbf{x}_k^h) + \varsigma_2 \sigma_2 (\mathbf{gb}_k - \mathbf{x}_k^h). \tag{3.3}$$

The updated velocity is calculated to be proportional to the sum of three distinct swarm components:

- **Inertia component:**
  Particle h's velocity at time step k, $\mathbf{u}_k^h$, serves as a memory of the previous flight direction, that being the motion at time-step k. This memory term can be seen as a momentum, which prevents the particle from any extreme change in direction and biases it toward the current direction of motion.

- **Cognitive component:**
  The cognitive component, $\varsigma_1\sigma_1(\mathbf{pb}_k^h - \mathbf{x}_k^h)$, is the experiential knowledge of a particle. It is proportional to the distance of the particle from its personal best position discovered over k unit time steps. The coefficient, $\varsigma_1\sigma_1$, is a product of a set weight ς_1 of the cognitive component and σ_1, a random number from a uniform distribution between 0 and 1. The cognitive component quantifies the performance of particle h relative to it's past performances. An individual's memory of the position that was best suited for the particle is resembled by the cognitive component. The effect of this term is that particles are drawn back to their own best positions, resembling the tendency of individuals to return to situations or places that satisfied them most in the past.

- **Social component:**
  The social component, $\varsigma_2\sigma_2(\mathbf{gb}_k - \mathbf{x}_k^h)$, quantifies the performance of particle h relative to a group of particles, or neighbours. The coefficient, $\varsigma_2\sigma_2$, is a product of a set weight ς_2 for the social component and a random number σ_2 between 0 and 1. Conceptually, the social component resembles a group norm or standard that individuals seek to attain. The effect of the social component is that each particle is also drawn towards the best position found by the particle's neighbourhood.

The contribution of the cognitive and social components are respectively weighed by a stochastic amount, $\varsigma_1\sigma_1$ or $\varsigma_2\sigma_2$. The effects of these weights are discussed in detail by Engelbrecht (2006b).

Illustration of Particle Position Update

The effect of the velocity update applied to the particle's current position can easily be illustrated in a two-dimensional vector space. Consider a single particle in a two-dimensional search space. An example of the particle's movement is illustrated in Figure 3.12. The Figure shows how the three components (inertia component, cognitive component and social component) contribute to calculate an updated velocity, $\mathbf{u}_{k+1}^h$, which moves the particle to its updated position, $\mathbf{x}_{k+1}^h$.

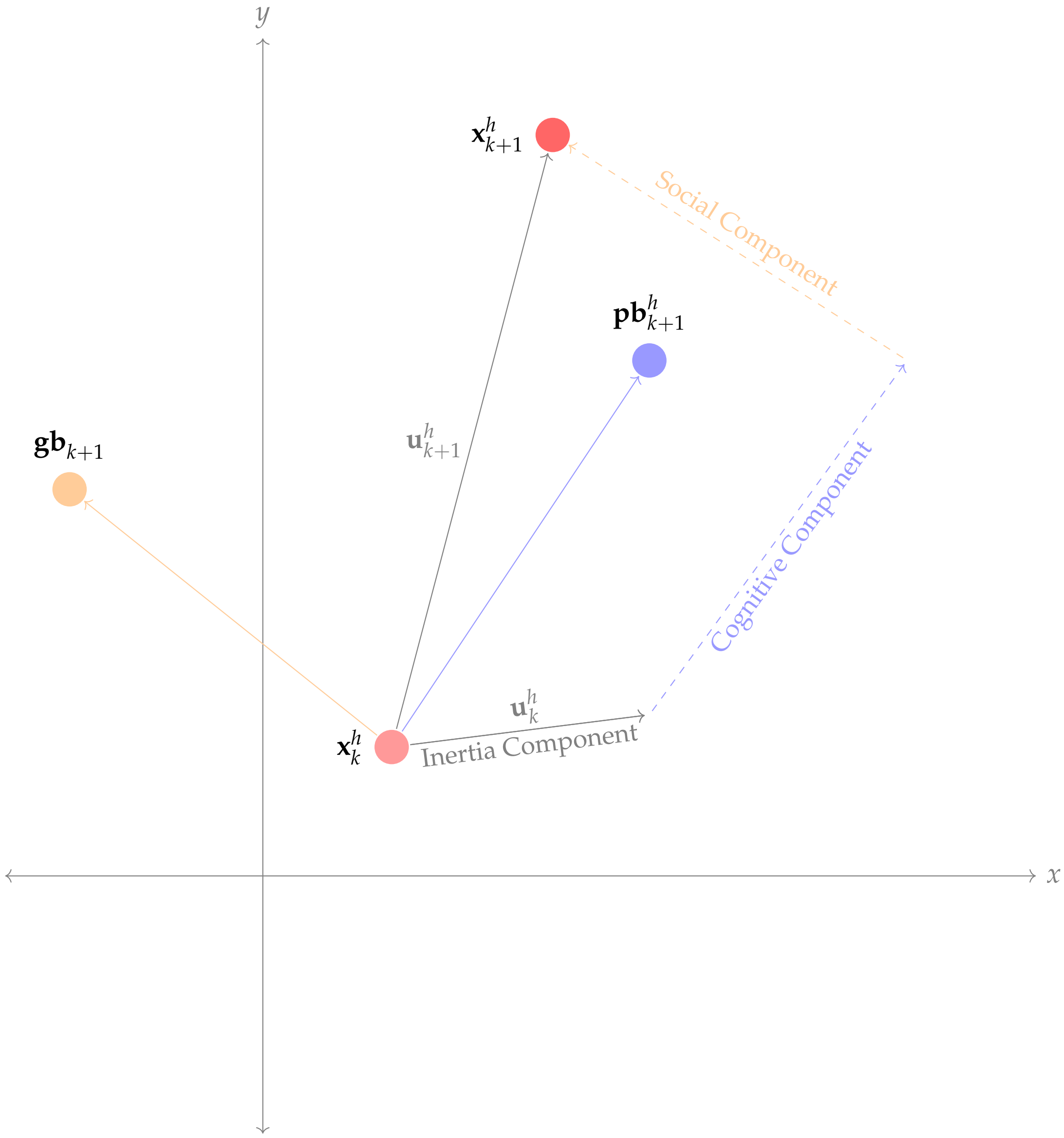


FIGURE 3.12: Movement of a Particle in a Swarm

It must be noted that it is of course possible for a particle to overshoot the global best position, mainly due to the momentum term. This results in two scenarios:

- As a result of overshooting the current global best, the new position may be a better position than the current global best. In this case the new particle position will become the new global best position and all particles will be drawn towards it.
- The new position is still worse than the current global best particle. In subsequent time-steps the cognitive and social components will cause the particle to change direction back towards the global best.

The cumulative effect of all the position updates of a particle is that each particle converges to a point on the line that connects the global best position and the personal best position of the particle. An example of the update of the swarm's particles from iteration $k = 0$ to the time-step where the algorithm terminates is shown in Figure 3.13.

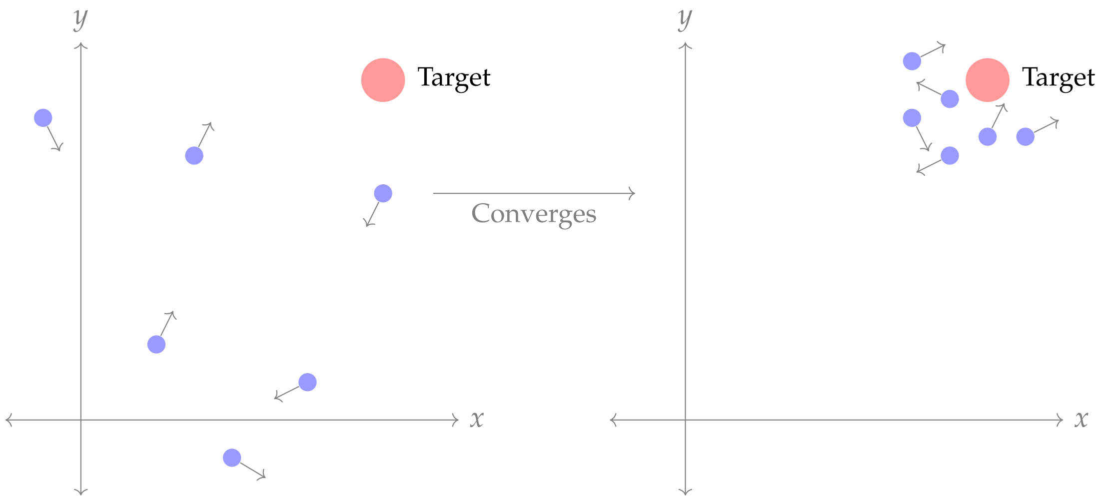


FIGURE 3.13: Convergence of the PSO Algorithm

The pseudocode of the symbiotic cooperative algorithm, PSO algorithm, is given in Algorithm 6.

Algorithm 6 Particle Swarm Optimisation Algorithm (PSO)

procedure PSO(N, m, f) ▷ Input: number of particles, dimension of the search space and objective function.
  $k \leftarrow 0$
  for each particle h in range N **do**
    Initialise the particle's position $\mathbf{x}_0^h$ and velocity $\mathbf{u}_0^h$ with a uniformly distributed random vector of dimension m
    $\mathbf{pb}_0^h \leftarrow \mathbf{x}_0^h$ ▷ Initial personal best position of the particle is the current position.
  end for
  $\mathbf{gb}_0 \leftarrow \text{argmin}(\{f(\mathbf{pb}_0^1), \ldots, f(\mathbf{pb}_0^N)\})$ ▷ Initial global best position of the population is the best of all initial positions.
  while Stopping criterion not met **do**
    for each particle h in range N **do**
      Calculate updated velocity, $\mathbf{u}_{k+1}^h$, according to Equation (3.2)
      Update particle position , $\mathbf{x}_{k+1}^h$, according to Equation (3.3)
      $\mathbf{u}_k^h \leftarrow \mathbf{u}_{k+1}^h$
      $\mathbf{x}_k^h \leftarrow \mathbf{x}_{k+1}^h$
      Calculate fitness value $f(\mathbf{x}_k^h)$
      if $f(\mathbf{x}_k^h) < f(\mathbf{pb}_k^h)$ **then**
        $\mathbf{pb}_k^h \leftarrow \mathbf{x}_k^h$ ▷ Update particle's personal best position.
        if $f(\mathbf{pb}_k^h) < f(\mathbf{gb}_k)$ **then**
          $\mathbf{gb}_k \leftarrow \mathbf{pb}_k^h$ ▷ Update population's global best position.
        end if
      end if
    end for
    $k \leftarrow k + 1$
  end while
  return $\mathbf{gb}_k$ ▷ Returns the global best position after k iterations.
end procedure

3.2.5 Particle Swarm Optimisation for the Vehicle Routing Problem with Time Windows

The PSO algorithm is more commonly used to find solutions to continuous optimisation problems rather than discrete ones. As the operators of the initial PSO algorithm were designed for a m-dimensional continuous space. However, modifying the PSO algorithm allows for it to be extended to discrete optimisation problems such as the VRPTW. A novel set-based PSO (S-PSO) method applicable to combinatorial optimisation problems (COPs) in discrete spaces, is presented by Liang, Qin, Suganthan, and Baskar (2006b); Chen, Zhang, Chung, Zhong, Wu, and Shi (2010); Gong, Zhang, Liu, Huang, Chung, and Shi (2012a). Two principle characteristics of the S-PSO method are:

- Characterising the discrete search space of COPs by a set-based scheme.
- Respectively representing candidate solutions and velocity, as a crisp set and sets with probabilities.

To tailor the arithmetic PSO update operators for S-PSO, the original operations are defined to be applied on crisp sets and sets with probabilities. Thus, the S-PSO method can follow the general PSO algorithm structure in a discrete search space. Crisp sets are sets where an element is either included or excluded from a set.

There are a number of variants based on the proposed S-PSO method. One particular discrete variations of the S-PSO algorithm is the the Comprehensive Learning PSO (CLPSO). To prevent premature convergence to a solution to the VRPTW, the CLPSO method is employed. The application of the CLPSO in the S-PSO to the VRPTW is based on the work done by Liang, Qin, Suganthan, and Baskar (2006b); Gong, Zhang, Liu, Huang, Chung, and Shi (2012a). Details of the S-PSO method are now discussed.

The solution space S for the VRPTW is an undirected complete graph $G = (\mathbf{C}, \mathbf{A})$. The graph G is an undirected graph with pairs of distinct vertices from $\mathbf{C}$ which are connected by unique edges from set $\mathbf{A}$. Specific to the problem, the vertices represent each customer in the customer set and the depot; an edge represents the path between any two adjacent vertices.

The generalisation of the particle's position $\mathbf{x}$ and velocity $\mathbf{u}$, specific to the VRPTW, are now discussed. Given that a population P has N particles, each particle in the population is indexed by h, where $h \in [1, N]$. The time-step or iteration considered is denoted by k, where $k \in \mathbb{N}_0$.

Position Representation

Each particle's position in the S-PSO for the VRPTW is representative of a candidate solution. Each particle's position is represented by a crisp set of arcs. These arcs are a subset of $\mathbf{A}$. The position of the particle is represented by $\mathbf{x}_k^h$. Each of the m-dimension positional components in the search space are denoted by $x_k^{h_\delta}$, $\delta \in [1, m]$. Hence, the position of a particle can be expended as:

$$\mathbf{x}_k^h = \left[x_k^{h_1}, \ldots, x_k^{h_m}\right].$$

Each dimensional arc set $x_k^{h_\delta}$ comprises of two adjacent arcs adjacent to node δ. The arc set is represented by Equation 3.4.

$$x_k^{h_\delta} = [\langle nb_1, \delta \rangle, \langle \delta, nb_2 \rangle]. \tag{3.4}$$

In Equation (3.4) δ is the index of the current dimension; nb_1 is the previous node of δ, and nb_2 is the subsequent node to node δ. The condition associated to Equation (3.4) is that $nb_1, nb_2 \in \{1, 2, \ldots, \delta - 1, \delta + 1, \ldots, m\}$ where $nb_1 \neq nb_2$.

The position of each particle represents a directed Hamilton circle of the complete graph G. The collection of Hamilton circles are representative of routes to be followed by each of the vehicles to be dispatched. By the definition of the particle's position and the constraint-based decoder given in Figure 3.14, each particle in the population represents a feasible solution of the S-PSO for VRPTW

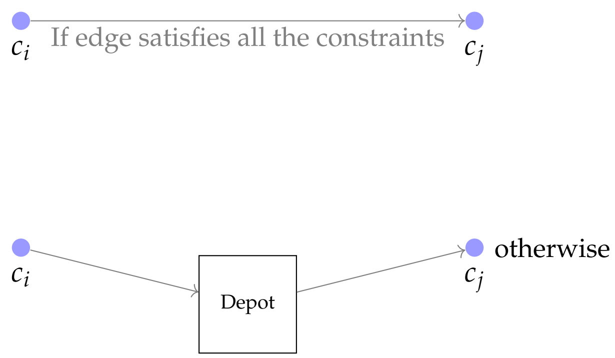


FIGURE 3.14: S-PSO for VRPTW Decoder

For each arc in the Hamilton circle the depot is the starting point of the outer arc. If the arc satisfies all the VRPTW constraints, the arc is preserved; otherwise, the depot is inserted between the two nodes, and the arc is replaced by two new arcs. When the vehicle along a particular Hamiltonian circle is unable to serve the next customer because of the capacity or the time-window constraints, a new vehicle route is initialised. An example of this decoding method is illustrated in Figure 3.15, which uses the decoding stipulated in Figure 3.14.

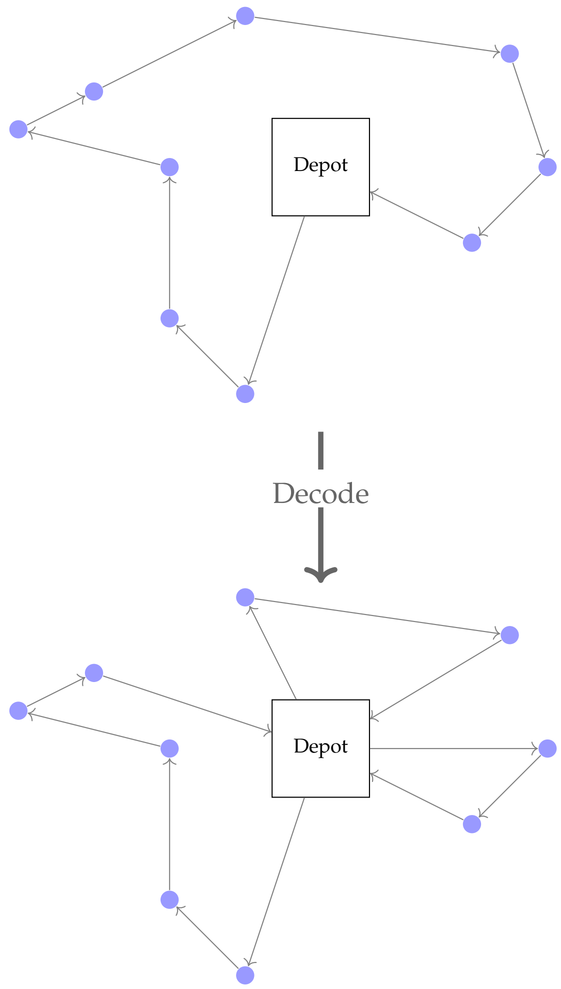


FIGURE 3.15: S-PSO for VRPTW Positioning Decoded

Velocity Representation

The velocity is represented by a set of arc probabilities. Each particle's m-dimensional velocity component is represented by $u_k^{h_\delta}, \forall\, \delta \in [1, m]$. The velocity of a particle h is defined over the m-dimensional space as:

$$\mathbf{u}_k^h = \left[u_k^{h_1}, \ldots, u_k^{h_m}\right].$$

Formally, the probability set of the arcs in dimension δ is represented by Equation 3.5.

$$u_k^{h_\delta} = \{\langle c_i, c_j \rangle / p(c_i, c_j) | \langle c_i, c_j \rangle \in A^\delta\}. \tag{3.5}$$

The arcs in the set A^δ are all the arcs adjacent to node δ in the complete graph G. The probability of arc $\langle c_i, c_j \rangle$ is denoted by $p(\langle c_i, c_j \rangle)$, where $p(\langle c_i, c_j \rangle) \in [0, 1]$. This probability is considered when arc $\langle c_i, c_j \rangle$ is selected in updating the particle's position. The arc $\langle c_i, c_j \rangle$ is omitted from $u_k^{h_\delta}$ if $p(\langle c_i, c_j \rangle) = 0$.

S-PSO for VRPTW

An overview of the application of the S-PSO algorithm to the VRPTW is given in Algorithm 7.

Algorithm 7 S-PSO for VRPTW (Overview)

procedure S-PSO(N, m, f, rg, sg) ▷ Input: number of particles, dimension of the search space, objective function, stopping conditions for pbest and gbest
  $k \leftarrow 0$
  $gbestCounter \leftarrow 0$
  $pbestCounter[1:N] \leftarrow 0$
  for each particle h in range N **do**
    Initialisation using Algorithm 8 ▷ Initialise particle h's position and velocity.
  end for
  while $gbestCounter < sg$ **do** ▷ Stopping criterion.
    for each particle h in range N **do**
      if $pbestCounter[h] > rg$ **then**
        Calculate updated velocity, $\mathbf{u}^h_{k+1}$, using PSO
        Update particle position , $\mathbf{x}^h_{k+1}$
        $\mathbf{x}^h_k \leftarrow \mathbf{x}^h_{k+1}$
        $\mathbf{u}^h_k \leftarrow \mathbf{u}^h_{k+1}$
        if $f(\mathbf{x}^h_k) < f(\mathbf{pb}^h_k)$ **then**
          $\mathbf{pb}^h_k \leftarrow \mathbf{x}^h_k$ ▷ Update particle's personal best position.
          $pbestCounter[h] \leftarrow 0$
          if $f(\mathbf{pb}^h_k) < f(\mathbf{gb}_k)$ **then**
            $\mathbf{gb}_k \leftarrow \mathbf{pb}^h_k$ ▷ Update population's global best position.
            $gbestCounter \leftarrow 0$
          else
            $gbestCounter \leftarrow gbestCounter + 1$
          end if
        else
          $pbestCounter[h] \leftarrow pbestCounter[h] + 1$
        end if
      end if
      Calculate updated velocity, $\mathbf{u}^h_{k+1}$, using CLPSO
      Update particle position , $\mathbf{x}^h_{k+1}$
      $\mathbf{x}^h_k \leftarrow \mathbf{x}^h_{k+1}$
      $\mathbf{u}^h_k \leftarrow \mathbf{u}^h_{k+1}$
      if $f(\mathbf{x}^h_k) < f(\mathbf{pb}^h_k)$ **then**
        $\mathbf{pb}^h_k \leftarrow \mathbf{x}^h_k$ ▷ Update particle's personal best position.
        $pbestCounter[h] \leftarrow 0$
        if $f(\mathbf{pb}^h_k) < f(\mathbf{gb}_k)$ **then**
          $\mathbf{gb}_k \leftarrow \mathbf{pb}^h_k$ ▷ Update population's global best position.
          $gbestCounter \leftarrow 0$
        else
          $gbestCounter \leftarrow gbestCounter + 1$
        end if
      else
        $pbestCounter[h] \leftarrow pbestCounter[h] + 1$
      end if
    end for
    $k \leftarrow k + 1$
  end while
  return $\mathbf{gb}_k$ ▷ Returns the global best position after k iterations.
end procedure

Initialisation

The process of initialising a population of N particles entails initialising each particle's position and velocity over the m-dimensional search space at time-step $k = 0$. The initialisation phase makes use of two initialisation heuristics:

- Nearest Neighbour Heuristic (NNH).
- Random initialisation.

The application of S-PSO to the VRPTW defines the initialisation as given in Algorithm 8.

Algorithm 8 Initialisation

```
1:  procedure PSO INITIALISATION(N, m, C, φ)     ▷ Input: number of particles, dimension of the search space,
    customer set and probability of greedy initialisation.
2:      for each particle h in range [1, N] do
3:          if rand < φ then                 ▷ rand is a random number from a uniform distribution between [0,1].
4:              x_k^h ← initialised by NNH(C, m)          ▷ Greedy Method: Nearest Neighbour Heuristic.
5:          else
6:              x_k^h ← initialised by Random(C, m)       ▷ Random Method.
7:          end if
8:      end for
9:      return X = {x_k^1, ..., x_k^N}        ▷ Returns the position of each particle in the population.
10: end procedure
```

- **Nearest Neighbour Heuristic**

  The NNH is a greedy method for initialising the candidate solutions of a population. The algorithm is initialised with the customer set **C**, which contains the customer's respective details: location, demand and service times. The method initialises each route starting at the depot. The NNH recursively appends the *nearest* feasible neighbour to the route, until no such neighbour exists. If the customer set **C** is not empty, then a new route is initialised, else the algorithm is terminated. The *nearest* neighbour is determined using Equation (3.6). This equation is calculated from the current customer i to each unattended customer, and the *nearest* neighbour j is then selected to be the customer which has the minimum τ_{ij} value.

  $$nn_{ij} = \tau_{ij} + \varpi_{ij} = (d_{ij} + w_j) + (l_j - t_j). \tag{3.6}$$

  The terms of Equation (3.6) are defined:

  - τ_{ij} represents the time span between the time when the vehicle completes the service for customer i and the time when the vehicle begins servicing customer j . The components of τ_{ij} are given as follows:
    - d_{ij} is the travel time or distance between customer i and j.
    - w_j is the waiting time of a vehicle at customer j until e_j.
  - ϖ_{ij} defines the the remaining time for which customer j has to be served. The components of ϖ_{ij} are given as follows:
    - l_j is the latest time customer j can start to be served, and
    - t_j is the time when the vehicle arrives at customer j.

- **Random Initialisation**

  The random initialisation method initialises candidate solutions of a population in a random manner. This method randomly selects a customer from the customer set **C**, if feasible to append to the current route it is appended, else a new route is initialised. This is repeated until the customer set **C** is empty.

Position Update

The position update follows the general PSO position update given by Equation (3.2). However, the operator '+' is redefined for the S-PSO algorithm being applied to the VRPTW. In updating the position, velocity **u** is converted into a crisp set by Equation (3.7).

$$Cut(\mathbf{u}_k^h) = \Big\{ \langle c_i, c_j \rangle | \langle c_i, c_j \rangle / p(c_i, c_j) \in \mathbf{u}_k^h \text{ and } p(c_i, c_j) \geq rand \Big\} \tag{3.7}$$

The random number, *rand*, is from a uniform distribution over the interval $[0,1]$. The arc $\langle c_i, c_j\rangle$ is kept in the velocity set if and only if its corresponding probability $p(c_i, c_j)$ is not less than the random value, *rand*. Hence, the arc with a larger $p(c_i, c_j)$ is more likely to be preserved in a crisp set after the conversion.

The updated position $\mathbf{x}^h_{k+1}$, is built in a constructive way, during which the capacity and time window constraints are taken into account. The constraints are denoted as Ω. Initially, $\mathbf{x}^h_{k+1}$ is set to be an empty set, in accordance with each vehicle starting at depot 0. To construct the vehicle routes, the method selects an arc adjacent to node 0, and then repeatedly looks for adjacent arcs to the current node. Suppose the current customer that the vehicle is servicing is c_i, then the next customer or node c_j that the vehicle will next visit, comes from one of three crisp sets:

- $\mathbf{S}_U = \{c_j | \langle c_i, c_j\rangle \in \mathbf{u}^h_k, \langle c_i, c_j\rangle \text{ satisfies } \Omega\}$.
- $\mathbf{S}_X = \{c_j | \langle c_i, c_j\rangle \in \mathbf{x}^h_k, \langle c_i, c_j\rangle \text{ satisfies } \Omega\}$.
- $\mathbf{S}_A = \{c_j | \langle c_i, c_j\rangle \in \mathbf{A}, \langle c_i, c_j\rangle \text{ satisfies } \Omega\}$.

If there are nodes available in $\mathbf{S}_U$, c_j is selected from $\mathbf{S}_U$. Otherwise, if there are nodes available in $\mathbf{S}_X$, c_j is selected from $\mathbf{S}_X$. Otherwise, c_j is selected from $\mathbf{S}_A$. After selecting c_j, arc $\langle c_i, c_j\rangle$ is added to $\mathbf{x}^h_{k+1}$. When there are no available nodes in all $\mathbf{S}_U$, $\mathbf{S}_X$ and $\mathbf{S}_A$, or the constraints Ω of the VRPTW cannot be satisfied, a new route is created. At this point, the depot node is inserted after c_i and c_j is re-selected, thus, ensuring the feasibility of $\mathbf{x}^h_{k+1}$. The pseudocode of the position update is given in Algorithm 9

Algorithm 9 Position Update

procedure POSITION UPDATE(N, m, Ω) ▷ Input: Number of particles, search space dimensions and constraints.
  $c_i \leftarrow 0$
  $k \leftarrow 0$
  $curr \leftarrow 0$
  $\mathbf{S}_U = \{c_i | \langle c_i, c_j\rangle \in \mathbf{U}^h, \langle c_i, c_j\rangle \text{ satisfies } \Omega\}$
  $\mathbf{S}_X = \{c_i | \langle c_i, c_j\rangle \in \mathbf{X}^h, \langle c_i, c_j\rangle \text{ satisfies } \Omega\}$
  $\mathbf{S}_A = \{c_i | \langle c_i, c_j\rangle \in \mathbf{A}, \langle c_i, c_j\rangle \text{ satisfies } \Omega\}$
  while the construction of $\mathbf{x}^h_{k+1}$ is not completed **do**
    if $\mathbf{S}_U \neq \Phi$ **then**
      Select c_j in $\mathbf{S}_U$, and add $\langle curr, c_j\rangle$ to $\mathbf{x}^h_k$
      $c_i \leftarrow c_j$
      $curr \leftarrow c_j$
    else if $\mathbf{S}_X \neq \Phi$ **then**
      Select c_j in $\mathbf{S}_X$, and add $\langle curr, c_j\rangle$ to $\mathbf{x}^h_k$
      $c_i \leftarrow c_j$
      $curr \leftarrow c_j$
    else if $\mathbf{S}_A \neq \Phi$ **then**
      Select c_j in $\mathbf{S}_A$, and add $\langle curr, c_j\rangle$ to $\mathbf{x}^h_k$
      $c_i \leftarrow c_j$
      $curr \leftarrow c_j$
    else
      $c_j \leftarrow 0$
      $c_i \leftarrow 0$ ▷ Initialise new route.
    end if
    Update $\mathbf{S}_U$, $\mathbf{S}_X$ and $\mathbf{S}_A$
  end while
  return $\mathbf{x}^h_{k+1}$ ▷ Returns service route for particle h.
end procedure

Velocity Update

The CLPSO velocity update is used at every iteration to update a particle's position. However, if a particle's personal best value has not altered for a set number of iterations, then the particle has an additional velocity update. This particle then has the PSO velocity update applied to it. The CLPSO and PSO methods are now discussed.

- **CLPSO Position Update**

  The CLPSO is a variant of the PSO algorithm, which is used to update each particle's position at every iteration k in the S-PSO for the VRPTW. The CLPSO uses a novel velocity update rule to prevent premature convergence to a solution. The CLPSO velocity update rule is stated in Equation (3.8).

$$u_{k+1}^{h_\delta} = \omega \times u_k^{h_\delta} + \varsigma \times \sigma(pb_k^{h_\delta} - x_k^{h_\delta}). \tag{3.8}$$

  In Equation 3.8, ω is the inertia weight coefficient, ς is the acceleration coefficient that determines the relative weight of the cognitive component and σ is a random number generated over the uniform distribution over the range $[0, 1]$. The particle is indexed by h and the dimension considered is denoted by δ. The personal best position used is not necessarily the considered personal best position for the particular dimension at time-step k. This condition allows for the particle to learn from another particle's personal best position. The personal best position selected for the velocity update is determined using the learning probability P_c and a tournament selection function H. A random number $rand$ from a uniform distribution over the range $[0, 1]$ is generated. If $P_c < rand$, then the personal best position used corresponds to the considered particle i.e., pb_k^h. Otherwise, the tournament selection function H is used to determine which particle's personal best position from which the particle learns. The tournament selection procedure followed by function H is given by Liang, Qin, Suganthan, and Baskar (2006b). The tournament selection employed has the following outline:

  1. Randomly choose two particles out of the population which excludes the particle whose velocity is being updated.
  2. Compare the fitness values of these two particles' personal best position and select the fitter of the two. Since this is a minimisation problem, the particle with a smaller fitness value is selected.
  3. Use the selected particle's personal best position as the exemplar from which to learn, $\forall\ \delta \in [1, m]$.

  The tournament selection procedure allows for particle's to learn from fitter exemplars and avoid accumulating search time in directions which are unfavourable.

  The operators employed by the S-PSO for the VRPTW are novel and are defined on sets and probabilities, instead of arithmetic operations as in the original PSO algorithm. The definition of the operators being applied are now given.

- **Coefficient $\times$ velocity:**
  This operator is defined by Equation (3.9) which determines the probability of an arc existing between two vertices, c_i and c_j using Equation (3.10).

$$\omega \times u_k^{h_\delta} = \{\langle c_i, c_j\rangle / p'(c_i, c_j) | \langle c_i, c_j\rangle \in A^\delta\}, \tag{3.9}$$

$$p'(c_i, c_j) = \begin{cases} 1, & \text{if } \omega \times p(\langle c_i, c_j\rangle) > 1, \\ \omega \times p(\langle c_i, c_j\rangle), & \text{otherwise.} \end{cases} \tag{3.10}$$

- **Velocity + velocity:**
  The probability of an arc between vertices for two particles h and h' for dimension δ are manipulated by this operator, as shown in Equation (3.11).

$$u_k^{h_\delta} + u_k^{h'_\delta} = \{\langle c_i, c_j\rangle / \max(p_k^{h_\delta}(c_i, c_j), p_k^{h'_\delta}(c_i, c_j)) | \langle c_i, c_j\rangle \in A^\delta\}. \tag{3.11}$$

- **Position - position:**
  The subtraction of two arc sets are defined by this operator, as shown in Equation (3.12).

$$x_k^{h_\delta} - x_k^{h'_\delta} = M^\delta = \{\langle c_i, c_j\rangle \mid \langle c_i, c_j\rangle \in x_k^{h_\delta} \text{ and } \notin x_k^{h'_\delta}\}. \tag{3.12}$$

- **Coefficient $\times$ (position - position):**
  This operator converts a crisp set given by M^δ into a set with probabilities, as shown in Equation (3.13). The conditional probability condition is given by Equation (3.14).

$$\varphi \times M^\delta = \{\langle c_i, c_j\rangle / p'(c_i, c_j) \mid \langle c_i, c_j\rangle \in A^\delta\}, \tag{3.13}$$

$$p'(c_i, c_j) = \begin{cases} 1, & \text{if } \langle c_i, c_j\rangle \in M^\delta \text{ and } \varphi > 1, \\ \varphi, & \text{if } \langle c_i, c_j\rangle \in M^\delta \text{ and } 0 \leq \varphi \leq 1, \\ 0, & \text{if } \langle c_i, c_j\rangle \notin M^\delta. \end{cases} \tag{3.14}$$

An example of the application of the defined operators is now given. For example, suppose that:

$$u_k^{h_1} = \{\langle 1,3\rangle/0.5,\ \langle 1,4\rangle/0.2,\ \langle 4,1\rangle/0.3\},$$

$$x_k^{h_1} = \{\langle 5,1\rangle,\ \langle 1,3\rangle\},$$

$$pb_k^{f_h(1)} = \{\langle 4,1\rangle,\ \langle 1,3\rangle\},$$

$$\omega = 0.7,\ \varsigma = 2.0, \text{ and } \sigma = 0.3.$$

Then,

$$\omega \times u_k^{h_1} = \{\langle 1,3\rangle/0.35,\ \langle 1,4\rangle/0.14,\ \langle 4,1\rangle/0.21\},$$

$$pb_k^{f_h(1)} - x_k^{i_1} = \{\langle 4,1\rangle\}, \text{ and}$$

$$\varsigma \times \sigma \times (pb_k^{f_h(1)} - x_k^{h_1}) = \{\langle 4,1\rangle/0.6\}.$$

The new velocity component of $u_{k+1}^{h_1}$ is deduced to be :

$$u_{k+1}^{h_1} = \omega \times u_k^{h_1} + \varsigma \times \sigma \times (pb_k^{f_h(1)} - x_k^{h_1}) \tag{3.15}$$
$$= \{\langle 1,3\rangle/0.35,\ \langle 1,4\rangle/0.14,\ \langle 4,1\rangle/0.6\}. \tag{3.16}$$

By learning from the $pb_k^{f_h(1)}$, the corresponding probability of the arc $\langle 4,1\rangle$ in the velocity set is improved. This example highlights the significance of updating the arcs with probabilities from the velocity set of the S-PSO and serves the purpose proposed by the original PSO velocity update, that is adjusting the particle's position by learning from the historical best information.

- **PSO Velocity Update**

  The '+' operator defined for the set-based method is utilised when applying the general PSO velocity update which is given by Equation (3.3). The difference between the CLPSO update and this velocity update is that now the global best component is considered in updating the particle's velocity.

Local Search

A simple local search scheme is applied once each of the particle's positions have been updated. This scheme is applied with the objective of reducing the number of vehicles dispatched to service customers. The local search is considered to accelerate the convergence of the swarm. The local search follows the following procedure for each particle h in the population:

1. Choose the route R that has the fewest customers.
2. For each customer in R, attempt to insert the customer from the other vehicle routes on the premise such that the insertion doesn't change the start and end time of the service for all the customers in which the route is being inserted and satisfies the constraints of the VRPTW.
3. If all the customers in R can be inserted into other routes, route R is deleted, and the position of the particle h is updated. Otherwise, the position of the particle h remains the same.

Figure 3.16 is an example representation of a candidate solution to the VRPTW. Following the local search scheme, R is set to be Route 3 as it has the fewest customers in the route. The scheme then tries to insert the customers in Route 3 into the Routes 1 and 2.

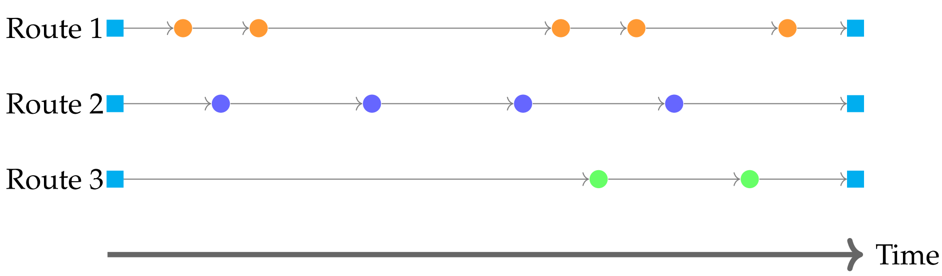


FIGURE 3.16: Routes Before Local Search Scheme Application

If the insertion of all the customers in Route 3 can successfully be inserted into Route 1 and 2 as shown by Figure 3.17, then Route 3 is deleted. The position of the particle considered is updated.

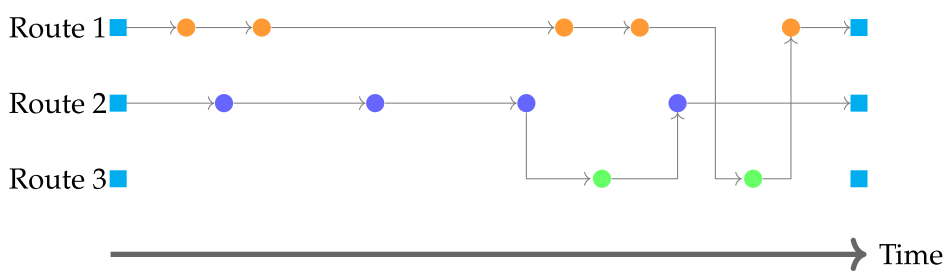


FIGURE 3.17: Additional Local Search

Conditional Statements

Conditional satements or expressions state that if a set of rules are met then perform some predefined tasks. The S-PSO for the VRPTW makes use of two particular conditional statement in the algorithm. The stopping gap and the refreshing gap condition.

- **Stopping gap condition:**
  The stopping gap is a flexible terminal condition which terminates the S-PSO algorithm if the global best fitness value ceases to improve for a certain number of iterations sg.

- **Refreshing gap condition:**
  The refreshing gap condition states that an additional PSO velocity update is to be applied to the current personal best position if the personal best position fitness has not improved for a certain number of iterations rg. This method's formulation is given by Liang, Qin, Member, Suganthan, Member, and Baskar (2006a).

3.3 Metrics

As the VRPTW is a MOP it aims to minimise routing cost which entails minimising both the number of vehicles dispatches to service customers and their accumulated travelled distance. Two particular metrics are considered in evaluating the solution designs. These two metrics are discussed respectively in Section 3.3.1 and Section 3.3.2.

3.3.1 Metric A

Metric A is constructed using a weighted sum method presented by Ombuki, Ross, and Hanshar (2006a). This method evaluates a solution to the VRPTW by adding the problems objective function with weighted coefficients, hence transforming the objective function of the VRPTW into a single optimisation problem. This metric evaluates a solution as Λ which is defined by Equation (3.17). The value of Λ is proportional to the sum of the number of vehicles $|\mathbf{V}|$ dispatched to service all the customers $\mathbf{C}$ and the summed distances D of the accumulated distance travelled by the dispatched vehicles. The parameters α and β weight these two objectives; number of dispatched vehicles and total travelled distance.

$$\Lambda = \alpha \cdot |\mathbf{V}| + \beta \cdot D. \tag{3.17}$$

3.3.2 Metric B

Metric B evaluation method of a solution to the VRPTW is presented by Gong, Zhang, Liu, Huang, Chung, and Shi (2012a). The solution design is evaluated and scored based on both the number of vehicles $|\mathbf{V}|$ used to service all the customers and the accumulated distance D travelled by these vehicles. This metric sums the number of vehicles used and the normalised value of the accumulated distance D travelled as Θ, defined in Equation (3.18). The definition of the normalisation function is given in Equation (3.19).

$$\Theta = |\mathbf{V}| + \text{normalise}(D). \tag{3.18}$$

$$\text{normalise}(D) = \arctan(D)/(\pi/2). \tag{3.19}$$

Both Metric A and Metric B's primary objective is the number of vehicles dispatched and the secondary objective is the total travelled distance. These metrics award a greater penalty to the number of dispatched vehicles in comparison to the total travelled distance. Hence, this thesis will analyse the effects of the structuring of the metrics.

Chapter 4

Results

This chapter records the results of the computational experiments conducted to investigate optimisation solution techniques, solution evaluation metrics and their computational performance in obtaining solutions to the Vehicle Routing Problem with Time Windows (VRPTW). Problem instances containing 25, 50 and 100 customers are considered. Results are recorded for each of Solomon's benchmarking problems for the three spatial classes (Clustered (C), Random (R) and Random Clustered (RC)), under the two temporal categories (tight time windows (1) and *long-haul* or wide time windows (2)). A detailed description of the Solomon benchmarking problem instances are described in Section 3.1 of Chapter 3. It is pertinent to reiterate that the customers in each problem instance within a spatial class, have the same spatial arrangement but have varying time window durations in which they are to be serviced. The time window duration is the length of the time frame which a customer can be served, from which the standard service time is subtracted, such that the remaining time indicates the duration of time flexibility in servicing a customer. The results record components of the procured solutions to the VRPTW by applying the two metaheuristic optimisation solution techniques, the Genetic Algorithm (GA) and the Particle Swarm Optimisation (PSO) algorithm for two solution evaluation metrics, Metric A and Metric B. The applied solution techniques are described in Section 3.2 of Chapter 3, followed by the formulation of the solution evaluation metrics in Section 3.3 of Chapter 3.

For each of the spatial categories of the Solomon Benchmark problem instances, the following components of the produced solutions are recorded for each of the solution evaluation metrics employed by the optimistion solution techniques:

- NV: Number of dispatched vehicles.
- DIST: Total travelled distance by the dispatched vehicles.
- FIT: Solution evaluation value awarded to a produced solution by the applied a metric scheme.
- Wait Time: Total time that dispatched vehicles stand idle when on service routes in order to meet customer's specified service time windows. The wait time is given in unit time.
- CPU Time: Computational Processing Unit Time taken to procure a solution to a problem instance. The CPU time is given in seconds.

The best recorded results in the literature to Solomon's benchmarking VRPTW for datasets containing 25, 50 and 100 customers can be found at `http://web.cba.neu.edu/~msolomon/problems.htm` and the updated results are recorded by Gong, Zhang, Liu, Huang, Chung, and Shi (2012a) and Ombuki, Ross, and Hanshar (2006b). These values record the best recorded values in the literature on the VRPTW and the best recorded results obtained by applying the PSO algorithm given by Gong, Zhang, Liu, Huang, Chung, and Shi (2012a) and the GA given by Ombuki, Ross, and Hanshar (2006b). It must be noted that the results obtained by applying the PSO algorithm by Gong, Zhang, Liu, Huang, Chung, and Shi (2012b) record results for datasets containing 25, 50 and 100 customers, as well as the time taken to produce these solutions, whilst the GA applied by Ombuki, Ross, and Hanshar (2006b) only records the results for datasets containing 100 customers. Hence, the obtained solutions are compared to these respective benchmark values. To evaluate the obtained solution relative to the recorded best value, a deviation percentage value is calculated. The deviation is calculated as shown in Equation 4.1:

$$\left(\frac{\text{Obtained} - \text{Benchmark Best}}{\text{Benchmark Best}}\right) \times 100. \tag{4.1}$$

The fitness evaluation benchmark value is calculated as per the defined metric formulations using the best recorded number of dispatched vehicles and their respective summed total travel distance. The following deviation values are recorded with their specific highlighting.

- DD: Deviation of the total distance by dispatched vehicles relative to the benchmark best values.
    - **Bold**: Deviation is 0, as solution value is equal to the recorded benchmark value.
    - Green: Deviation is $< 0\%$. Where the total travelled distance by the dispatched vehicles is less than the recorded benchmark best.
    - Red: Deviation is $> 10\%$. Where the total travelled distance by the dispatched vehicles is more than the recorded benchmark best, with a percentage deviation $> 10\%$.
- FitD: Deviation of the fitness value awarded to the produced solution relative to the benchmark best values.
    - Green: Deviation is $< 0\%$. The awarded fitness value to the obtained solution is equal or an improvement to the benchmark solution, as the awarded fitness value is equal or lower than the computed benchmark best value.
    - Red: Deviation is $> 0\%$. The awarded fitness value to the obtained solution is worse than the benchmark solution, as the awarded fitness value is greater than the computed benchmark best value.

4.1 Experimental Settings

The parameter values employed by the applied solution techniques, the GA and PSO algorithm, which are respectively described in Section 3.2.1 and Section 3.2.4 of Chapter 3 are now given. The parameter values are respectively given in Table 4.1 and Table 4.2.

The parameter values stipulated in Table 4.1 are obtained from the application of the GA to the VRPTW conducted by Ombuki, Ross, and Hanshar (2006b)

Variable	Variable Description	Value
Z	Number of chromosomes in population Γ	300
α	Weight coefficient for the number of vehicles in the fitness function Λ	100
β	Weight coefficient for the total accumulated travel distance in the fitness function Λ	0.001
μ	Percentage of initial population encoded using Random Permutation Encoding	90%
ρ	Tournament selection benchmark probability of selecting the *fittest* chromosome	0.8
K	Number of chromosomes κ selected for the tournament selection set T	4
ϱ	Probability of mutation	0.1
Υ	Predefined terminating number of generations	350
θ	Crossover rate	0.8
rad	Empirical radius used in selecting the nearest neighbours in the initialisation of candidate solutions	$\frac{(max - min)}{2}$

TABLE 4.1: Parameter Values of the Applied GA to the VRPTW

The parameter values stipulated in Table 4.2 are obtained from the application of the GA to the VRPTW conducted by Gong, Zhang, Liu, Huang, Chung, and Shi (2012b).

Variable	Variable Description	Value
N	Number of particles in population P	20
σ_1	Random weight value associated to the cognitive term	2
σ_2	Random weight value associated to the social term	2
ς_1	Weight for cognitive term, between	(0,1)
ς_2	Weight for social term, between	(0,1)
ϕ	Probability of greedy initialisation	0.3
ω	Inertia weight coefficient for CLPSO velocity update. Linearly decreases from	0.9 to 0.4
P_c	Current learning probability	2
sg	Global terminating criterion (25 customers; 50 and 100 customers)	1000;10000
rg	PSO criterion	7

TABLE 4.2: Parameter Values of the Applied PSO algorithm to the VRPTW

The applied solution technique algorithms are coded in Python 3® and are run on an AMD Ryzen™ Threadripper 2950x 16-core processor machine. Each experiment was independently run 30 times. The best result obtained from the 30 experiments are recorded, with the respective time taken to obtain the solution as the CPU time.

The implementation of the algorithms have used parallel computing methods from Python's Multiprocessing package.

The Python scripts and the Solomon data sets can be found at
`https://github.com/KrupaPrag/VRPTW.git`.

4.2 25 Customers

This section contains and discusses the results obtained to the VRPTW for Solomon's data set containing 25 customers.

	Benchmark Result		Benchmark PSO Result			GA										PSO									
						Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
C101	**3**	**191.30**	3	191.81	7.80	**3**	191.81	0.27	440.76	214.57	**3**	191.81	0.27	440.76	224.06	**3**	191.81	0.27	440.76	34.83	**3**	191.81	0.27	440.76	33.16
C102	**3**	**190.30**	3	190.74	7.80	**3**	190.74	0.23	595.32	201.01	**3**	190.74	0.23	595.32	257.80	**3**	213.03	11.94	686.04	37.32	**3**	210.96	10.85	674.45	38.87
C103	**3**	**190.30**	3	190.74	8.40	**3**	190.74	0.23	595.32	198.85	**3**	190.74	0.23	595.32	254.11	**3**	203.08	6.72	941.00	40.90	**3**	200.77	5.50	849.13	41.30
C104	**3**	**186.90**	3	187.45	8.40	**3**	190.74	2.05	594.69	195.99	**3**	190.74	2.05	594.69	249.96	**3**	202.24	8.21	1032.65	52.34	**3**	200.58	7.32	1029.94	72.02
C105	**3**	**191.30**	3	191.81	7.80	**3**	191.81	0.27	417.76	215.91	**3**	191.81	0.27	417.76	271.85	**3**	191.81	0.27	461.69	33.57	**3**	191.81	0.27	461.69	37.60
C106	**3**	**191.30**	3	191.81	7.80	**3**	191.81	0.27	447.76	282.62	**3**	191.81	0.27	447.76	276.41	**3**	191.81	0.27	447.76	32.84	**3**	191.81	0.27	505.69	32.13
C107	**3**	**191.30**	3	191.81	7.20	**3**	191.81	0.27	373.76	274.15	**3**	191.81	0.27	373.76	246.58	**3**	191.81	0.27	373.76	35.01	**3**	191.81	0.27	373.76	34.44
C108	**3**	**191.30**	3	191.81	8.40	**3**	191.81	0.27	372.76	256.61	**3**	191.81	0.27	372.76	250.12	**3**	193.17	0.98	413.01	47.12	**3**	193.17	0.98	413.01	43.37
C109	**3**	**191.30**	3	191.81	7.80	**3**	191.81	0.27	283.76	252.54	**3**	191.81	0.27	283.76	248.01	**3**	204.82	7.07	378.00	43.40	**3**	204.82	7.07	378.00	47.68
C201	**2**	**214.70**	2	215.54	7.20	**2**	215.54	0.39	1996.34	324.23	**2**	215.54	0.39	1996.34	342.49	**2**	215.54	0.39	1996.34	31.63	**2**	215.54	0.39	1996.34	31.30
C202	**2**	**217.70**	1	223.32	7.20	1	223.31	2.58	639.70	394.61	1	223.31	2.58	639.70	400.55	1	223.31	2.58	639.70	30.69	1	230.67	5.96	546.49	32.74
C203	**2**	**214.70**	1	223.31	11.40	1	246.57	14.84	588.50	221.77	1	234.81	9.37	731.10	229.48	1	247.90	15.46	821.74	41.54	1	240.19	11.87	637.14	42.53
C204	**2**	**214.70**	1	221.28	12.60	1	213.93	-0.36	765.90	270.19	1	213.93	-0.36	765.90	272.24	1	239.20	11.41	499.46	45.31	1	248.02	15.52	586.30	47.66
C205	**2**	**213.10**	1	297.45	6.60	1	298.84	40.24	456.42	263.70	1	298.30	39.98	551.23	264.83	1	297.45	39.58	457.81	28.66	1	297.45	39.58	457.81	29.82
C206	**2**	**214.70**	1	285.39	6.60	1	286.79	33.57	452.90	241.65	1	286.79	33.57	452.90	243.49	1	290.47	35.29	454.29	32.28	1	285.39	32.93	454.29	32.36
C207	**2**	**214.70**	1	274.78	6.60	1	274.78	27.98	712.43	247.03	1	274.78	27.98	712.43	241.92	1	274.78	27.98	712.43	32.66	1	277.15	29.09	382.39	30.00
C208	**2**	**214.50**	1	229.84	6.60	1	229.84	7.15	841.21	234.12	1	229.84	7.15	841.21	249.93	1	230.01	7.23	365.26	31.81	1	230.01	7.23	365.26	31.39

TABLE 4.3: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Clustered Problem Instances containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Fitness		PSO Benchmark Fitness		GA				PSO			
					Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
C101	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.19	0.00	4.00	0.00
C102	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.21	0.01	4.00	0.00
C103	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.20	0.00	4.00	0.00
C104	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.20	0.01	4.00	0.00
C105	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.19	0.00	4.00	0.00
C106	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.19	0.00	4.00	0.00
C107	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.19	0.00	4.00	0.00
C108	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.19	0.00	4.00	0.00
C109	**300.19**	**4.00**	300.19	4.00	300.19	0.00	4.00	0.00	300.20	0.00	4.00	0.00
C201	**200.21**	**3.00**	200.22	3.00	200.22	0.00	3.00	0.00	200.22	0.00	3.00	0.00
C202	**200.22**	**3.00**	100.22	2.00	100.22	-49.94	2.00	-0.50	100.22	-49.94	2.00	-0.50
C203	**200.21**	**3.00**	100.22	2.00	100.25	-49.93	2.00	-0.50	100.25	-49.93	2.00	-0.50
C204	**200.21**	**3.00**	100.22	2.00	100.21	-49.95	2.00	-0.50	100.24	-49.93	2.00	-0.50
C205	**200.21**	**3.00**	100.30	2.00	100.30	-49.90	2.00	-0.50	100.30	-49.90	2.00	-0.50
C206	**200.21**	**3.00**	100.29	2.00	100.29	-49.91	2.00	-0.50	100.29	-49.91	2.00	-0.50
C207	**200.21**	**3.00**	100.27	2.00	100.27	-49.92	2.00	-0.50	100.27	-49.92	2.00	-0.50
C208	**200.21**	**3.00**	100.23	2.00	100.23	-49.94	2.00	-0.50	100.23	-49.94	2.00	-0.50

TABLE 4.4: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Clustered Problem Instances containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Result		Benchmark PSO Result			GA										PSO									
						Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
R101	**8**	**617.10**	8	618.33	9.00	8	618.33	0.20	500.55	378.91	8	618.33	0.20	500.55	299.58	8	619.17	0.34	514.86	42.21	8	619.17	0.34	528.51	39.63
R102	**7**	**547.10**	7	548.11	14.40	7	548.11	0.18	458.26	338.58	7	548.11	0.18	458.26	260.19	7	567.30	3.69	503.87	50.70	7	567.30	3.69	503.87	39.32
R103	**5**	**454.60**	4	473.39	8.40	4	473.39	4.13	105.86	266.32	4	473.39	4.13	105.86	210.80	4	528.22	16.20	118.22	51.42	4	518.43	14.04	84.86	61.77
R104	**4**	**416.90**	4	418.30	10.80	4	417.96	0.25	104.55	250.70	4	417.96	0.25	104.55	203.80	4	449.30	7.77	161.99	48.73	4	447.00	7.22	141.41	59.58
R105	**6**	**530.50**	5	556.72	8.40	5	556.72	4.94	148.42	250.19	5	556.72	4.94	148.42	251.50	5	556.72	4.94	148.42	46.23	5	556.72	4.94	148.42	39.42
R106	**3**	**465.40**	5	466.48	10.20	5	466.48	0.23	240.34	227.35	5	466.48	0.23	240.34	224.92	5	501.70	7.80	256.63	60.65	5	494.81	6.32	254.98	50.63
R107	**4**	**424.30**	4	425.27	12.00	4	425.27	0.23	123.67	210.95	4	425.27	0.23	123.67	210.04	4	463.87	9.33	111.27	47.27	4	460.16	8.45	132.95	50.50
R108	**4**	**397.30**	4	405.39	12.60	4	398.29	0.25	90.61	203.72	4	398.29	0.25	90.61	203.68	4	434.83	9.45	130.18	57.70	4	431.72	8.66	131.20	52.02
R109	**5**	**441.30**	4	460.52	9.00	4	460.52	4.36	52.76	224.96	4	460.52	4.36	52.76	223.34	4	460.52	4.36	52.76	37.88	4	474.04	7.42	51.45	42.82
R110	**4**	**444.10**	4	445.80	9.00	4	445.80	0.38	99.03	214.94	4	448.29	0.94	99.03	211.94	4	449.82	1.29	120.18	56.34	4	449.82	1.29	101.81	39.91
R111	**5**	**428.80**	4	429.70	9.60	4	429.70	0.21	98.14	212.88	4	429.70	0.21	98.14	213.36	4	450.81	5.13	119.10	48.67	4	453.79	5.83	110.86	59.00
R112	**4**	**393.00**	4	394.10	10.20	4	394.10	0.28	82.98	205.80	4	394.10	0.28	82.98	205.00	4	413.02	5.09	108.83	60.75	4	410.96	4.57	108.83	57.61
R201	**4**	**463.30**	2	523.66	9.00	2	523.66	13.03	922.72	214.14	2	523.66	13.03	922.72	219.42	2	543.83	17.38	902.55	39.75	2	542.60	17.12	879.08	35.49
R202	**4**	**410.50**	2	455.53	12.60	2	457.82	11.53	963.85	210.17	2	457.74	11.51	973.93	218.19	2	511.88	24.70	1059.81	43.55	2	502.12	22.32	1006.47	38.68
R203	**3**	**391.40**	2	408.89	13.80	2	406.24	3.79	1063.52	209.81	2	408.69	4.42	1029.61	221.45	2	477.19	21.92	936.70	40.72	2	444.60	13.59	1104.34	42.11
R204	**2**	**355.00**	1	389.91	12.60	1	397.67	12.02	244.33	223.77	1	390.10	9.89	241.90	238.13	1	445.12	25.39	217.20	51.75	1	441.41	24.34	220.91	77.02
R205	**3**	**393.00**	1	501.83	7.20	1	502.07	27.75	130.86	210.44	1	503.66	28.16	98.53	214.04	1	525.49	33.71	85.52	37.91	1	525.53	33.72	107.40	37.50
R206	**3**	**374.40**	1	413.21	9.60	1	421.36	12.54	210.63	227.42	1	413.18	10.36	208.60	231.41	1	465.09	24.22	223.15	48.37	1	466.27	24.54	262.18	43.11
R207	**3**	**361.60**	1	402.28	11.40	1	416.57	15.20	182.52	228.16	1	418.51	15.74	171.01	230.46	1	450.40	24.56	172.14	49.58	1	448.81	24.12	251.91	61.62
R208	**1**	**328.20**	1	329.33	8.40	1	338.09	3.01	271.85	235.59	1	331.80	1.10	242.11	221.43	1	425.36	29.61	213.56	46.93	1	398.80	21.51	251.86	50.12
R209	**2**	**370.70**	1	438.24	9.00	1	427.67	15.37	155.00	212.34	1	418.25	12.83	155.00	214.85	1	484.17	30.61	117.56	35.97	1	480.51	29.62	137.32	44.52
R210	**3**	**404.60**	1	513.98	12.00	1	514.16	27.08	126.84	219.90	1	513.31	26.87	127.69	217.91	2	487.04	20.38	933.29	38.36	2	492.56	21.74	901.13	48.09
R211	**2**	**350.90**	1	361.69	8.40	1	363.93	3.71	160.45	234.81	1	361.69	3.07	147.09	231.61	1	446.43	27.22	53.67	39.69	1	451.64	28.71	96.95	43.78

TABLE 4.5: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Random Problem Instances containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Fitness		PSO Benchmark Fitness		GA				PSO			
					Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
R101	**800.62**	**9.00**	800.62	9.00	800.62	0.00	9.00	0.00	800.62	0.00	9.00	0.00
R102	**700.55**	**8.00**	700.55	8.00	700.55	0.00	8.00	0.00	700.57	0.00	8.00	0.00
R103	**500.45**	**6.00**	400.47	5.00	400.47	-19.98	5.00	-0.20	400.53	-19.97	5.00	-0.20
R104	**400.42**	**5.00**	400.42	5.00	400.42	0.00	5.00	0.00	400.45	0.01	5.00	0.00
R105	**600.53**	**7.00**	500.56	6.00	500.56	-16.65	6.00	-0.17	500.56	-16.65	6.00	-0.17
R106	**300.47**	**4.00**	500.47	6.00	500.47	66.56	6.00	0.67	500.50	66.58	6.00	0.67
R107	**400.42**	**5.00**	400.43	5.00	400.43	0.00	5.00	0.00	400.46	0.01	5.00	0.00
R108	**400.40**	**5.00**	400.41	5.00	400.40	0.00	5.00	0.00	400.43	0.01	5.00	0.00
R109	**500.44**	**6.00**	400.46	5.00	400.46	-19.98	5.00	-0.20	400.46	-19.98	5.00	-0.20
R110	**400.44**	**5.00**	400.45	5.00	400.45	0.00	5.00	0.00	400.45	0.00	5.00	0.00
R111	**500.43**	**6.00**	400.43	5.00	400.43	-19.98	5.00	-0.20	400.45	-19.98	5.00	-0.20
R112	**400.39**	**5.00**	400.39	5.00	400.39	0.00	5.00	0.00	400.41	0.00	5.00	0.00
R201	**400.46**	**5.00**	200.52	3.00	200.52	-49.93	3.00	-0.50	200.54	-49.92	3.00	-0.50
R202	**400.41**	**5.00**	200.46	3.00	200.46	-49.94	3.00	-0.50	200.51	-49.92	3.00	-0.50
R203	**300.39**	**4.00**	200.41	3.00	200.41	-33.28	3.00	-0.33	200.48	-33.26	3.00	-0.33
R204	**200.35**	**3.00**	100.39	2.00	100.40	-49.89	2.00	-0.50	100.45	-49.87	2.00	-0.50
R205	**300.39**	**4.00**	100.50	2.00	100.50	-66.54	2.00	-0.67	100.53	-66.54	2.00	-0.67
R206	**300.37**	**4.00**	100.41	2.00	100.42	-66.57	2.00	-0.67	100.47	-66.55	2.00	-0.67
R207	**300.36**	**4.00**	100.40	2.00	100.42	-66.57	2.00	-0.67	100.45	-66.56	2.00	-0.67
R208	**100.33**	**2.00**	100.33	2.00	100.34	0.01	2.00	0.00	100.43	0.10	2.00	0.00
R209	**200.37**	**3.00**	100.44	2.00	100.43	-49.88	2.00	-0.50	100.48	-49.85	2.00	-0.50
R210	**300.40**	**4.00**	100.51	2.00	100.51	-66.54	2.00	-0.67	200.49	-33.26	3.00	-0.33
R211	**200.35**	**3.00**	100.36	2.00	100.36	-49.91	2.00	-0.50	100.45	-49.86	2.00	-0.50

TABLE 4.6: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Random Problem Instances containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Result		Benchmark PSO Result			GA										PSO									
						Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
RC101	**4**	**461.10**	4	462.16	7.80	4	462.16	0.23	107.89	243.02	4	462.16	0.23	107.89	242.65	4	462.16	0.23	92.66	39.80	4	462.16	0.23	107.89	39.23
RC102	**3**	**351.80**	3	352.74	9.00	3	352.74	0.27	56.88	220.34	3	352.74	0.27	56.88	218.85	3	352.98	0.34	69.16	41.89	3	352.98	0.34	69.16	40.35
RC103	**3**	**332.80**	3	333.92	9.00	3	333.92	0.34	96.17	204.83	3	333.92	0.34	96.17	205.44	3	338.25	1.64	94.76	44.39	3	339.19	1.92	93.81	40.43
RC104	**3**	**306.60**	3	307.14	9.00	3	307.14	0.18	136.95	194.82	3	307.14	0.18	136.95	197.36	3	319.99	4.37	109.81	40.78	3	322.17	5.08	100.98	43.39
RC105	**4**	**411.30**	4	412.38	7.80	4	413.53	0.54	189.57	228.94	4	413.53	0.54	189.57	229.96	4	418.70	1.80	162.24	42.26	4	417.60	1.53	163.34	40.70
RC106	**3**	**345.50**	3	346.51	8.40	3	347.31	0.52	20.75	209.12	3	347.31	0.52	20.75	210.12	3	356.28	3.12	27.23	39.04	3	354.87	2.71	27.84	36.84
RC107	**3**	**298.30**	3	298.95	8.40	3	298.95	0.22	36.53	201.98	3	298.95	0.22	36.53	201.24	3	309.63	3.80	55.76	41.37	3	316.34	6.05	44.81	42.35
RC108	**3**	**294.50**	3	294.99	8.40	3	295.74	0.42	73.70	201.72	3	296.19	0.57	42.64	213.68	3	305.09	3.60	36.21	52.55	3	307.47	4.41	52.35	48.59
RC201	**3**	**360.20**	2	432.30	8.40	2	432.30	20.02	966.62	236.32	2	432.30	20.02	966.62	234.78	2	460.58	27.87	928.13	37.10	2	458.20	27.21	930.51	36.64
RC202	**3**	**338.00**	2	376.12	10.20	1	548.42	62.25	74.69	229.23	1	548.42	62.25	74.69	228.02	2	399.56	18.21	1079.70	52.68	2	397.11	17.49	1081.26	35.91
RC203	**3**	**326.90**	1	432.55	12.00	1	432.82	32.40	173.30	237.08	1	435.87	33.34	263.64	237.85	2	378.58	15.81	1041.44	39.68	2	377.41	15.45	1023.77	36.12
RC204	**3**	**299.70**	1	327.33	8.40	1	333.90	11.41	265.60	234.14	1	329.45	9.93	301.04	238.44	1	361.78	20.71	330.33	54.99	1	349.51	16.62	332.90	44.80
RC205	**3**	**338.00**	2	386.15	8.40	2	386.15	14.25	970.40	220.80	2	386.15	14.25	970.40	215.43	2	429.33	27.02	927.22	39.41	2	422.35	24.96	934.20	44.38
RC206	**3**	**324.00**	1	482.02	8.40	1	478.72	47.75	124.67	224.85	1	482.02	48.77	121.05	213.69	1	492.02	51.86	111.05	59.12	1	501.10	54.66	123.81	45.22
RC207	**3**	**298.30**	1	478.97	9.60	1	478.97	60.57	90.37	226.84	1	478.97	60.57	90.37	211.16	1	510.43	71.11	75.81	44.87	1	521.13	74.70	82.96	45.79
RC208	**2**	**269.10**	1	309.85	9.60	1	314.54	16.89	183.39	228.30	1	311.54	15.77	196.22	208.72	1	363.79	35.19	134.89	44.24	1	344.08	27.86	251.98	47.32

TABLE 4.7: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Random Clustered Problem Instances containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Fitness		PSO Benchmark Fitness		GA				PSO			
					Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
RC101	**400.46**	**5.00**	400.46	5.00	400.46	0.00	5.00	0.00	400.46	0.00	5.00	0.00
RC102	**300.35**	**4.00**	300.35	4.00	300.35	0.00	4.00	0.00	300.35	0.00	4.00	0.00
RC103	**300.33**	**4.00**	300.33	4.00	300.33	0.00	4.00	0.00	300.34	0.00	4.00	0.00
RC104	**300.31**	**4.00**	300.31	4.00	300.31	0.00	4.00	0.00	300.32	0.00	4.00	0.00
RC105	**400.41**	**5.00**	400.41	5.00	400.41	0.00	5.00	0.00	400.42	0.00	5.00	0.00
RC106	**300.35**	**4.00**	300.35	4.00	300.35	0.00	4.00	0.00	300.36	0.00	4.00	0.00
RC107	**300.30**	**4.00**	300.30	4.00	300.30	0.00	4.00	0.00	300.31	0.00	4.00	0.00
RC108	**300.29**	**4.00**	300.29	4.00	300.30	0.00	4.00	0.00	300.31	0.00	4.00	0.00
RC201	**300.36**	**4.00**	200.43	3.00	200.43	-33.27	3.00	-0.33	200.46	-33.26	3.00	-0.33
RC202	**300.34**	**4.00**	200.38	3.00	100.55	-66.52	2.00	-0.67	200.40	-33.28	3.00	-0.33
RC203	**300.33**	**4.00**	100.43	2.00	100.43	-66.56	2.00	-0.67	200.38	-33.28	3.00	-0.33
RC204	**300.30**	**4.00**	100.33	2.00	100.33	-66.59	2.00	-0.67	100.36	-66.58	2.00	-0.67
RC205	**300.34**	**4.00**	200.39	3.00	200.39	-33.28	3.00	-0.33	200.43	-33.27	3.00	-0.33
RC206	**300.32**	**4.00**	100.48	2.00	100.48	-66.54	2.00	-0.67	100.49	-66.54	2.00	-0.67
RC207	**300.30**	**4.00**	100.48	2.00	100.48	-66.54	2.00	-0.67	100.51	-66.53	2.00	-0.67
RC208	**200.27**	**3.00**	100.31	2.00	100.31	-49.91	2.00	-0.50	100.36	-49.89	2.00	-0.50

TABLE 4.8: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Random Clustered Problem Instances containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	GA								PSO							
	Metric A				Metric B				Metric A				Metric B			
Dataset	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10
C1	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	88.89	11.11	0.00	0.00	88.89	11.11
C2	0.00	12.50	37.50	50.00	0.00	12.50	50.00	37.50	0.00	0.00	37.50	62.50	0.00	0.00	37.50	62.50
R1	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	91.67	8.33	0.00	0.00	91.67	8.33
R2	0.00	0.00	27.27	72.73	0.00	0.00	36.36	63.64	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00
RC1	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00
RC2	0.00	0.00	0.00	100.00	0.00	0.00	12.50	87.50	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00

TABLE 4.9: The percentage (%) of problem instances per class containing 25 Customers classified under the stipulated total travelled distance deviation ranges, for the Applied Solution Techniques and Employed Evaluation Metrics

	GA			PSO		
Dataset	Metric A = Metric B	Metric A < Metric B	Metric A > Metric B	Metric A = Metric B	Metric A < Metric B	Metric A > Metric B
C1	100.00	0.00	0.00	66.67	0.00	33.33
C2	75.00	0.00	25.00	37.50	37.50	25.00
R1	91.67	8.33	0.00	33.33	16.67	50.00
R2	9.09	27.27	63.64	0.00	36.36	63.64
RC1	87.50	12.50	0.00	25.00	50.00	25.00
RC2	50.00	25.00	25.00	0.00	25.00	75.00

TABLE 4.10: The percentage (%) of problem instances containing 25 Customers with the total travelled distance per class classified under the stipulated metric relations, for the Applied Solution Techniques and Employed Evaluation Metrics.

	GA						PSO					
	Metric A			Metric B			Metric A			Metric B		
Dataset	Equal	< 0	> 0	Equal	< 0	> 0	Equal	< 0	> 0	Equal	< 0	> 0
C1	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
C2	0.00	87.50	12.50	0.00	87.50	12.50	0.00	87.50	12.50	0.00	87.50	12.50
R1	0.00	33.33	66.67	0.00	33.33	66.67	0.00	33.33	66.67	0.00	33.33	66.67
R2	0.00	90.91	9.09	0.00	90.91	9.09	0.00	90.91	9.09	0.00	90.91	9.09
RC1	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
RC2	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00

TABLE 4.11: Summary of Fitness Deviation Classifications of Fitness relative to benchmark fitness.

FIGURE 4.1: Wait Time (Unit Time) for Problem instances per class containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics

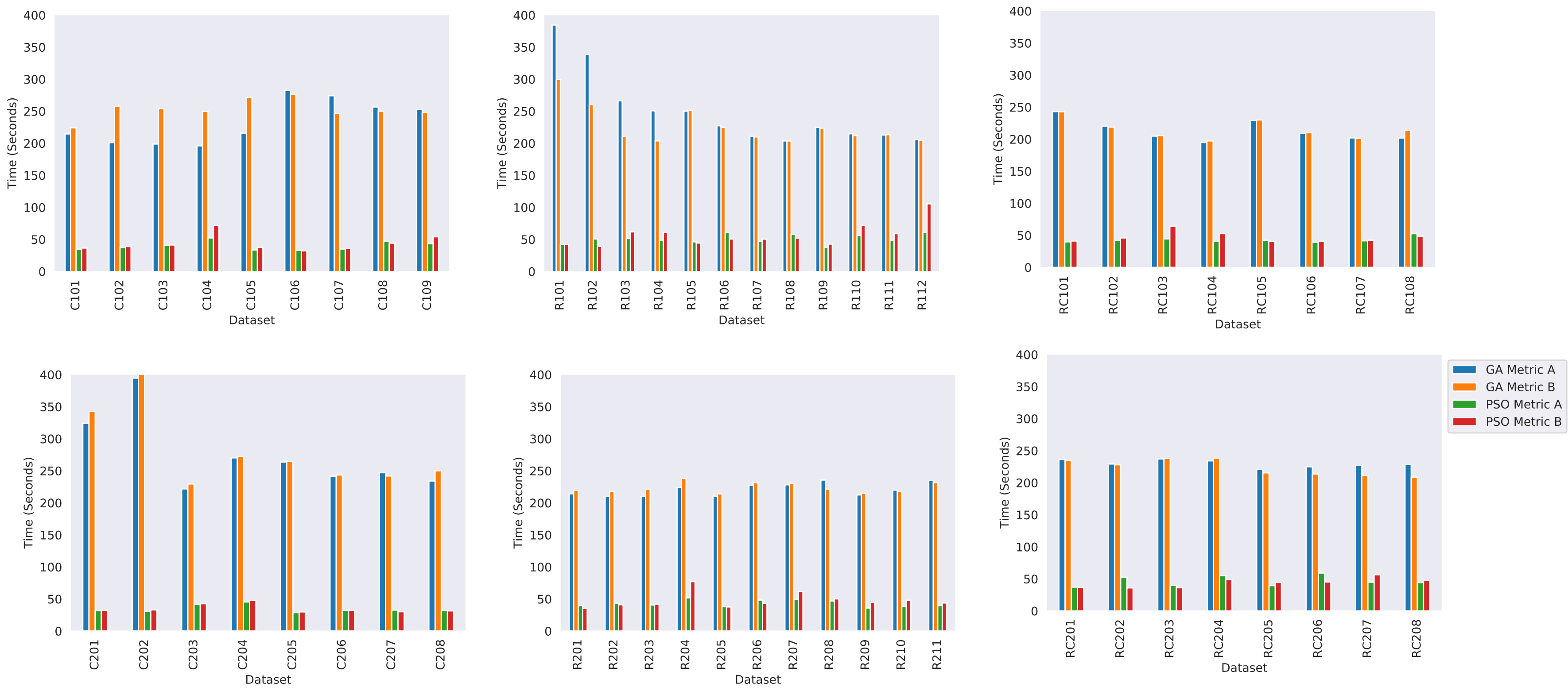


FIGURE 4.2: CPU Time (Seconds) for Problem instances per class containing 25 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics

4.2.1 Clustered

From the recorded results in Table 4.3 and Table 4.4 to the problem instances of spatially clustered customers spatially clustered, the following are noted for the problem instances in temporal class 1 and 2.

Temporal Class 1

- From the results obtained to the solutions to the problem instances in C1, the following are noted:
    - The number of dispatched vehicles in the obtained solutions correlate to both the recorded best and PSO benchmark values.
    - The corresponding total travelled distance percentage deviation to the benchmark value is less than 1% when the GA is applied and less than 15% when the PSO algorithm is applied. The PSO benchmark distance is is greater than the best benchmark value, hence the obtained solutions' total travelled distance deviation is less than that compared to deviation from the the benchmark total travelled distance when compared to the PSO total travelled distance. The solutions obtained through the application of the GA better resemble the PSO benchmark results than the results of the solutions obtained by the application of the PSO algorithm.
    - Comparing the solutions produced by the applied solution techniques, the distance deviation percentage is greater when the PSO algorithm is applied than when the GA is applied.
    - In Table 4.9 it is recorded that for 100% of the problem instances obtain a distance deviation percentage between $0\% - 10\%$ when the GA is applied. When the PSO algorithm is employed with Metric A, the solutions have a distance deviation percentage between $0\% - 10\%$ for 88.89% of it's problem instances and 11.11% of the problem instances has a distance deviation greater than 10%; when the PSO algorithm is employed using Metric B the distance deviation is between $0\% - 10\%$ for 100% of the problem instances.
- It is significant to note that the solutions produced by the GA are indifferent to the employed metric scheme, hence allowing for the GA to be claimed to be more robust. In contrast, the PSO algorithm produces varied results under the respective employed metric schemes. Let C102 be considered as an example. The resultant topologies of the produced solutions are given in Figure 4.7.

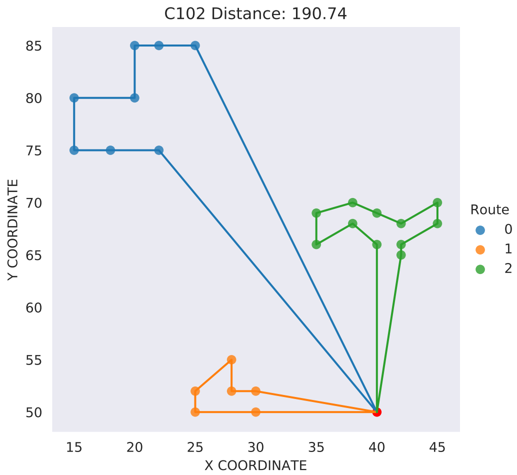


FIGURE 4.3: Topology of a Solution to C102 Obtained Using GA with Metric A

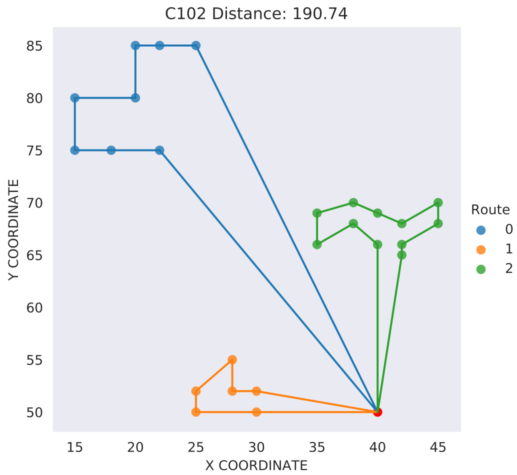


FIGURE 4.4: Topology of a Solution to C102 Obtained Using GA with Metric B

FIGURE 4.5: Topology of a Solution to C102 Obtained Using PSO with Metric A

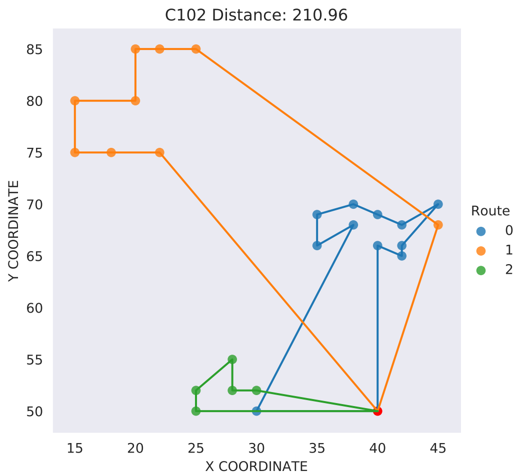


FIGURE 4.6: Topology of a Solution to C102 Obtained Using PSO with Metric B

FIGURE 4.7: Topology Plots of C102

The GA produces the same solution irrespective of the employed metric scheme. Hence, the resultant service route designs in Figure 4.3 and Figure 4.4 are identical. However, the resultant routes service different customers for either of the employed solution evaluation metrics applied with the PSO algorithm, as shown in Figure 4.5 and Figure 4.6. Other problem instances with solutions of this nature include, C103 and C104.

- It is worthy to note that the solutions obtained for C101 are indifferent under the respective solution techniques and evaluation metrics. This is not true for C102. To understand the reason for the variability in the obtained solution, the time window duration of the two problem instances are considered.

FIGURE 4.8: Time Window Duration of Customers in C101 and C102

The time window durations of customers in C101 and C102 are shown in Figure 4.8. It is noted that C102 has a few customers that have wider time window durations in comparison to C101. Wider time window durations mean there is flexibility of when a customer can be serviced, which may be the result of the variability in the obtained solutions by the two applied solution techniques and employed evaluation schemes.

Temporal Class 2

- From the results obtained to the solutions to the problem instances in C2, the following are noted:
  - The obtained solutions dispatch an equal or fewer service vehicles in comparison to the benchmark value. The resultant number of dispatched vehicles are equal to the PSO benchmark vehicle value.
  - The total travelled distance by the dispatched vehicles are greater than the benchmark value, as a consequence of dispatching fewer vehicles. However, the problem instance C204, is an exception. In comparison to the PSO benchmark distance value, the total travelled distance has a lower deviation to the PSO benchmark total travelled distance value, as a result of the resultant distance values corresponding to the PSO benchmark vehicle value.
- The problem instance C204 is particularly of interest as its solution dispatches fewer vehicles and has a total travelled distance which is less than the benchmark recorded best values when the GA is applied. Through the application of the solution techniques and employed evaluation metric schemes, the produced solutions route topologies are illustrated in Figure 4.39
.

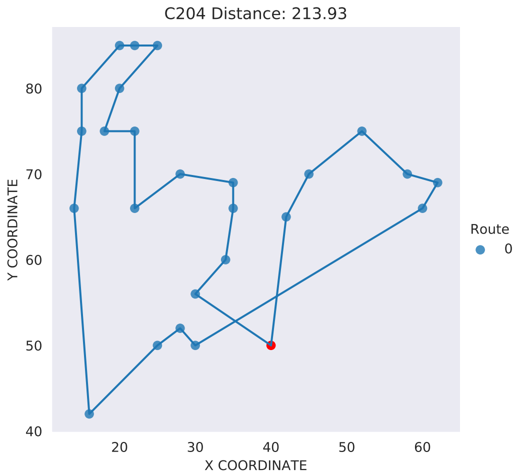


FIGURE 4.9: Topology of a Solution to C204 Obtained Using GA with Metric A

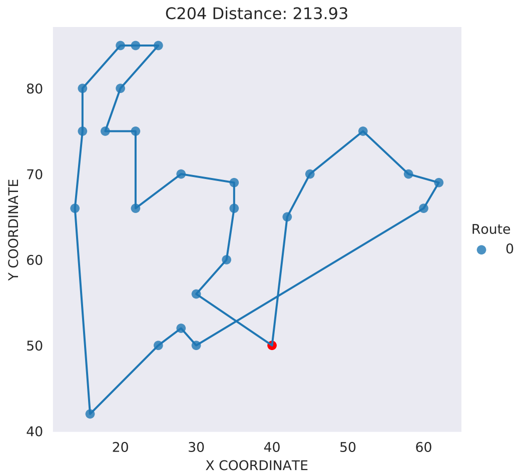


FIGURE 4.10: Topology of a Solution to C204 Obtained Using GA with Metric B

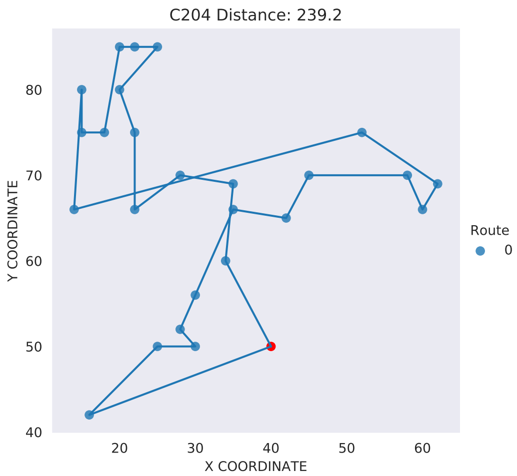


FIGURE 4.11: Topology of a Solution to C204 Obtained Using PSO with Metric A

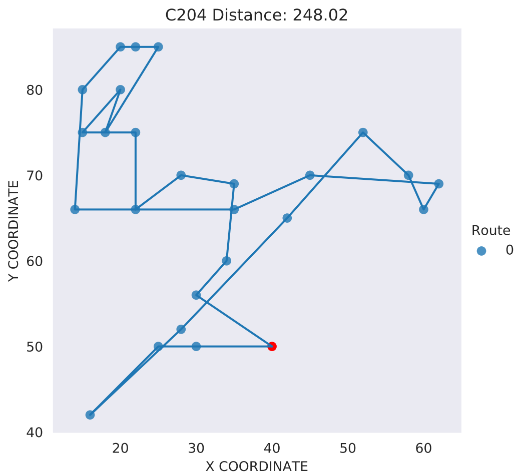


FIGURE 4.12: Topology of a Solution to C204 Obtained Using PSO with Metric B

FIGURE 4.13: Topology Plots of C204

The time window duration for customers in C204 is given in Figure 4.40.

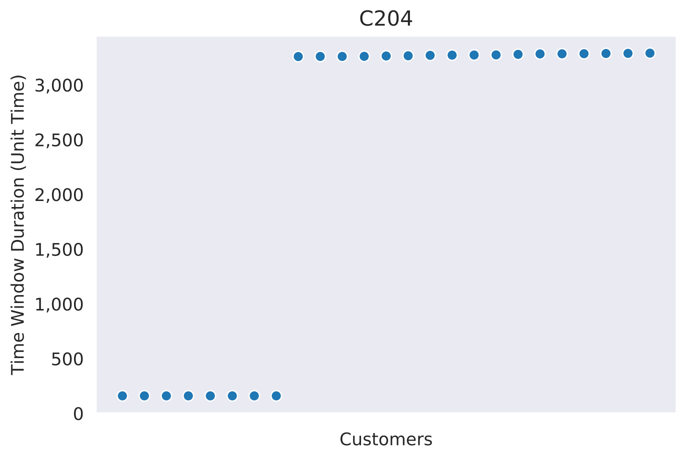


FIGURE 4.14: Time Window Duration of Customers in C204

From the description of Solomon's Benchmarking problem instances, C2 is classified under the *long-haul* temporal class. The wide time window durations of the majority of the customers in C204 may be a possible factor which allows for the variability in the obtained solutions It is further highlighted that the total wait time incurred when applying the GA is greater than that incurred when the PSO algorithm is applied. The wait time for the solutions obtained using the GA for both Metric A are Metric B is 765.9 time units, and using the PSO algorithm with Metric A and Metric B are respectively 499.46 and 586.3 time units. Thus, the less total travelled distance for the solutions obtained using the GA is compensated by a greater wait time incurred.

4.2.2 Random

The randomly spatially organised customers results are recorded in Table 4.5 and Table 4.6.

Temporal Class 1

- From the results obtained to the problem instances in R1, the following are noted:
    - The total travelled distance for all the R1 problem instances obtained using the GA are between $0\% - 10\%$, for 100% of the problem instances and 91.67% of the problem instances obtain a solution using the PSO algorithm have a distance deviation percentage between $0\% - 10\%$. The total distance travelled values have a closer resemblance to the PSO benchmark total travelled distance values as a result of the solutions having an equal number of dispatched vehicles to the PSO benchmark value. Furthermore, the solutions obtained using the GA have a total travelled distance which is less than that of the solutions obtained using the PSO algorithm, hence have a lower deviation to the benchmark distance values.
    - The solutions produced dispatch an equal number or fewer vehicles than the benchmark value, with the exception of R106. It is significant to note

that the PSO benchmark value of the number of dispatched vehicles is greater than the best recorded benchmark value.

- The problem instance R106 is of particular interest as it is the only problem instance with a solution which dispatches more vehicles than the benchmark value in the R1 class. However, the resultant number of dispatched vehicles and corresponding distance are equal to the PSO benchmark value for this problem instance. The topology of the produced solutions are shown in Figure 4.19.

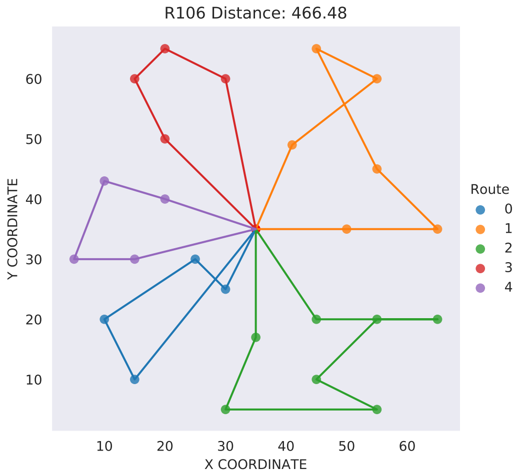


FIGURE 4.15: Topology of a Solution to R106 Obtained Using GA with Metric A

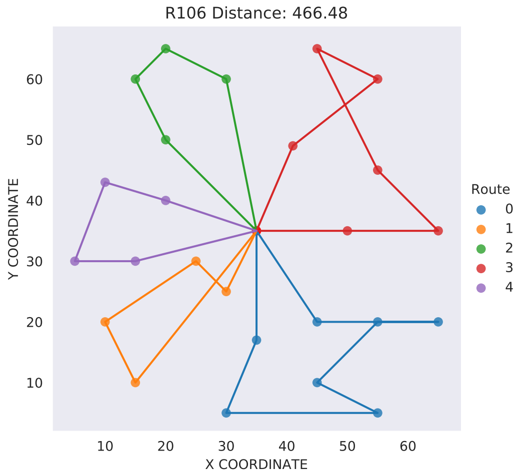


FIGURE 4.16: Topology of a Solution to R106 Obtained Using GA with Metric B

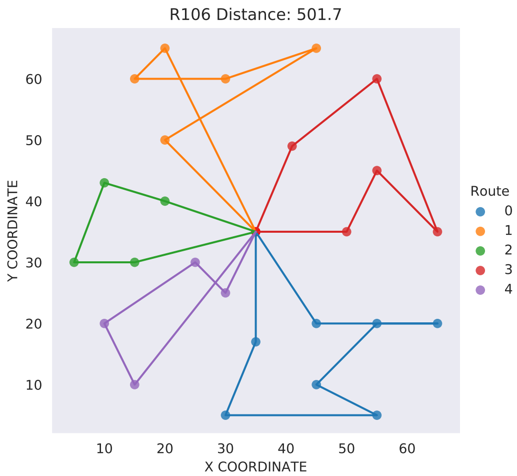


FIGURE 4.17: Topology of a Solution to R106 Obtained Using PSO with Metric A

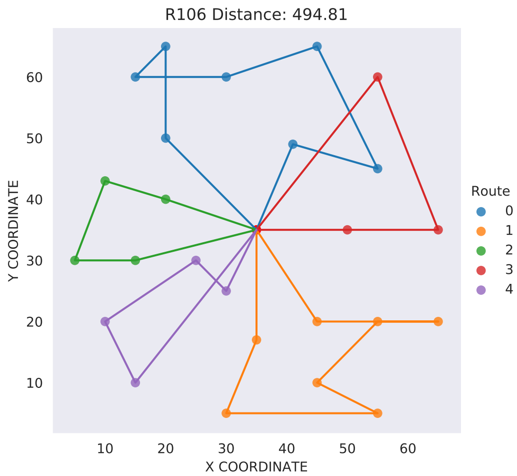


FIGURE 4.18: Topology of a Solution to R106 Obtained Using PSO with Metric B

FIGURE 4.19: Topology Plots of R106

The solutions produces the same route design under either solution evaluation metric scheme when the GA is applied, as seen in Figure 4.15 and Figure 4.16. However, the solution obtained by applying the PSO algorithm under either of the metric schemes are different. Route 1 and 3 in Figure 4.17 visit the same customers but in a different routing structure to Route 0 and 3 in Figure 4.18. The time window duration of the customers in R106 are shown in Figure 4.20.

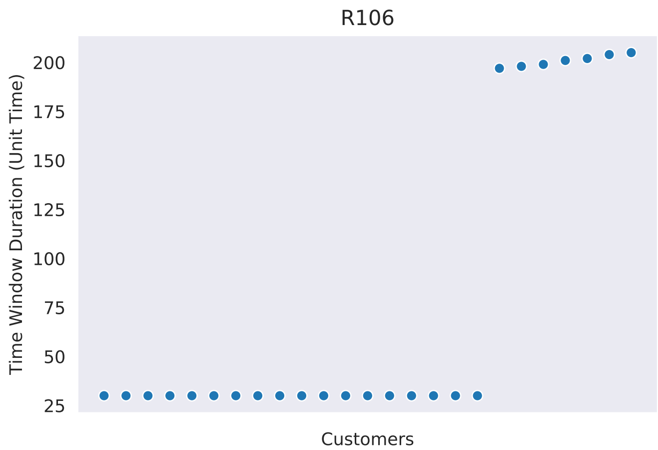


FIGURE 4.20: Time Window Duration of Customers in R106

As shown in Figure 4.20, R106 has customers which have very tight time windows. However, the few with wide time windows allow for variability in the route construction. The total wait time incurred using the GA is less than that incurred by applying the PSO algorithm. Furthermore, a lesser total distance is travelled by the dispatched vehicles in the solutions produced using the GA.

Temporal Class 2

- From the results obtained to the solutions to the problem instance in R2, the following are noted:
  - The total number of dispatched vehicles of the obtained solutions are less than or equal to the benchmark value, and equal to the PSO benchmark value.
  - Dispatching fewer vehicles than the benchmark value is compensated by the greater total distance having to be travelled by the dispatched vehicles. The GA has 27.27% of its problem instances which have a distance deviation between $0\% - 10\%$ and 72.73% of the problem instances with a distance deviation $> 10\%$. A 100% of the problem instances have a distance deviation $> 10\%$ when applying the PSO algorithm. As a result of the number of dispatched vehicles being equal to the PSO benchmark value, the total travelled distance bears closer resemblance to the PSO distance benchmark value. The total travelled distance obtained to the solutions using the GA is less than that obtained using the PSO algorithm. As a result, the distance deviation of the solutions obtained using the GA are less than that obtained for the solutions obtained using the PSO algorithm.
- The problem instance R210 is of interest as the solutions produced using the two solution techniques dispatch a distinct number of vehicles. The solutions

obtained by the application of the GA dispatched two less vehicles than the benchmark value. However, by applying the PSO algorithm, the solution dispatches one less vehicle in comparison to the benchmark value. The topologies of the obtained solutions are given in Figure 4.25.

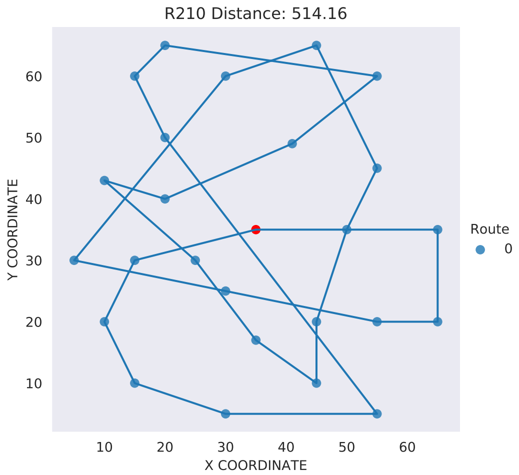


FIGURE 4.21: Topology of a Solution to R210 Obtained Using GA with Metric A

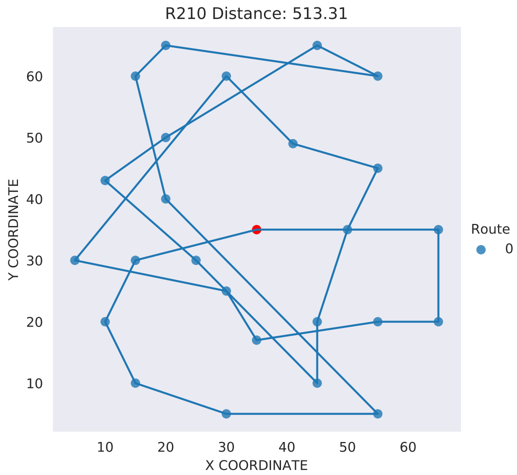


FIGURE 4.22: Topology of a Solution to R210 Obtained Using GA with Metric B

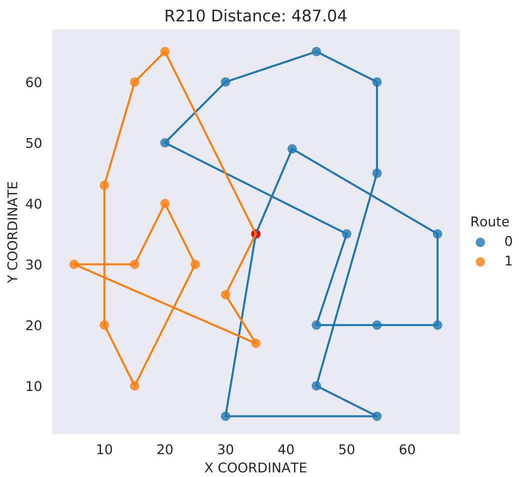


FIGURE 4.23: Topology of a Solution to R210 Obtained Using PSO with Metric A

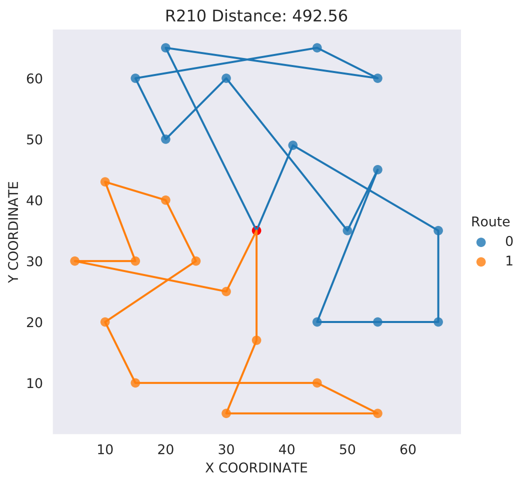


FIGURE 4.24: Topology of a Solution to R210 Obtained Using PSO with Metric B

FIGURE 4.25: Topology Plots of R210

The time window duration of the customers in R210 are given in Figure 4.26.

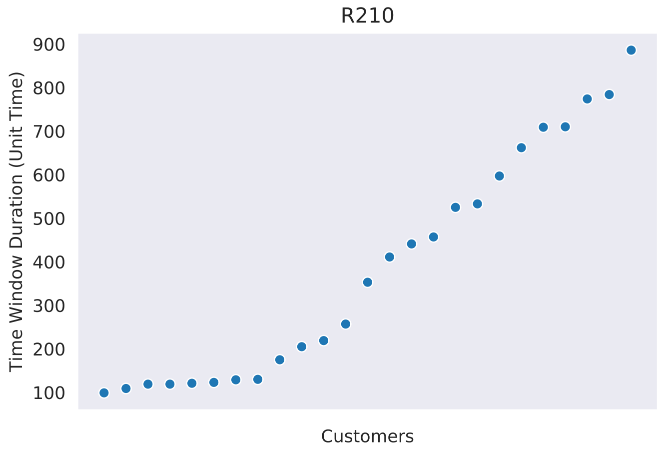


FIGURE 4.26: Time Window Duration of Customers in R210

From Figure 4.25 it is noted that although more vehicles are dispatched by the PSO algorithm, the total travelled distance is less than that obtained by the GA. Furthermore, the GA incurs a less total wait time than the PSO algorithm. Considering the time window durations of each of the customers in R210, shown in Figure 4.26, it is noted that the customers have varying time window durations between 100 and 900 unit time, which allows for flexibility in the placement of customers into service routes.

4.2.3 Random Clustered

The randomly clustered spatially organised customers results are recorded in Table 4.7 and Table 4.8.

Temporal Class 1

- From the results obtained to the problem instances in RC1, the following are noted:
  - The number of dispatched vehicles in the recorded solutions are consistent to both the recorded benchmark and PSO benchmark value, for all the problem instances in RC1.
  - The total travelled distance percentage deviation to the benchmark distance value is between $0\% - 10\%$ for the obtained solutions for both the applied solution techniques with either employed solution evaluation metric. The PSO benchmark distance value is greater than the best known benchmark value, hence the obtained resultant total travelled distance deviation to the PSO benchmak distance value is lower than that compared to the benchmark value. Furthermore, it is worth noting that in general the resultant PSO algorithm's total travelled distance is greater than that of the solution's produced using the GA.

Temporal Class 2

- From the results obtained to the solutions to the problem instance in RC2, the following are noted:

- The number of dispatched vehicles of the obtained solutions are less than the benchmark value for all the problem instances in RC2, but are equal to the PSO benchmark value.
- Dispatching fewer vehicles than the benchmark value is compensated by the greater resultant total travelled distance than the benchmark distance value. For the GA employing Metric A and the PSO algorithm with either metric, 100% of the problem instances have a distance deviation percentage $> 10\%$. The solutions obtained using the GA with Metric B, 12.5% of the problem instances have a distance deviation percentage between $0\% - 10\%$ and 87.5% of the problem instances have a distance deviation percentage $> 10\%$. Since the PSO benchmark vehicle value and the obtained solution values are equal, the total travelled distance of the obtained solutions bear resemblance to the PSO benchmark distance value.

- The problem instance RC207 is highlighted as although dispatches two less vehicles than the benchmark recorded value, the total distance percentage deviation is the highest in RC2 irrespective of the applied solution techniques. The solution route topologies of RC207 are shown Figure 4.62.

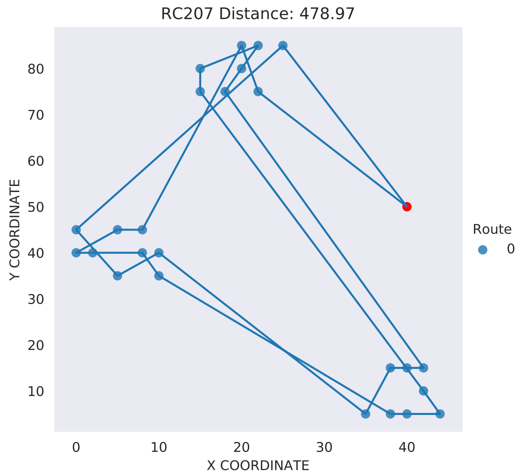


FIGURE 4.27: Topology of a Solution to RC207 Obtained Using GA with Metric A

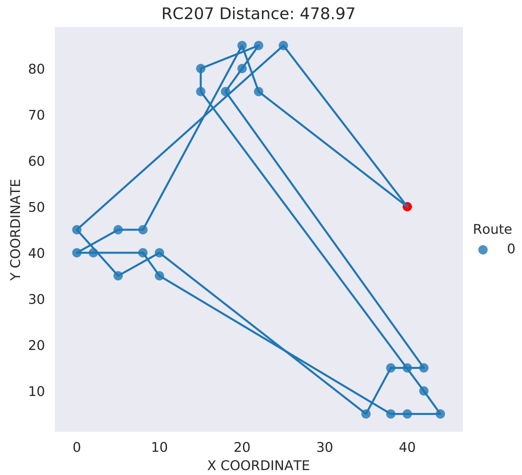


FIGURE 4.28: Topology of a Solution to RC207 Obtained Using GA with Metric B

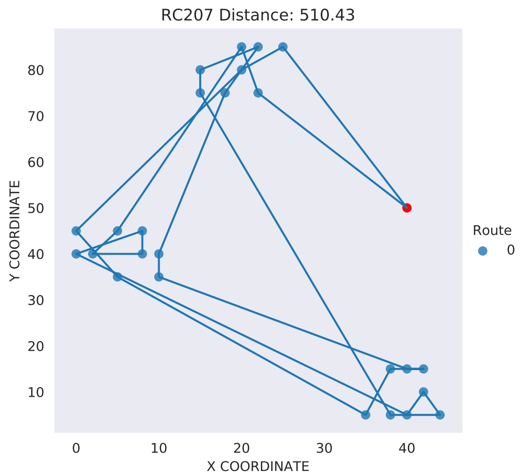


FIGURE 4.29: Topology of a Solution to RC207 Obtained Using PSO with Metric A

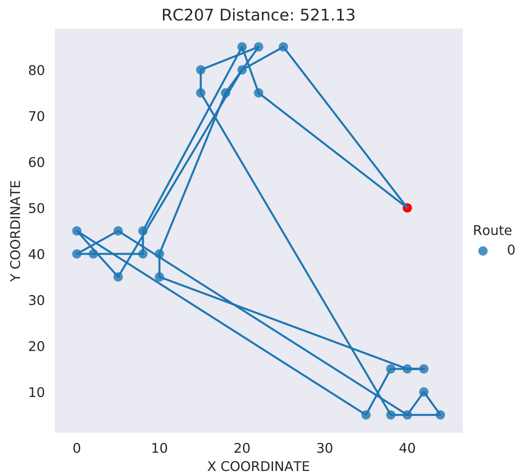


FIGURE 4.30: Topology of a Solution to RC207 Obtained Using PSO with Metric B

FIGURE 4.31: Topology Plots of RC207

The time window duration of the customers in RC207 are given in Figure 4.63.

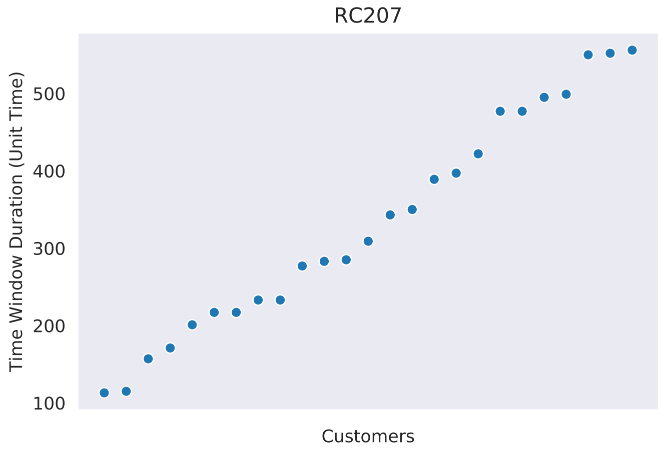


FIGURE 4.32: Time Window Duration of Customers in RC207

From Figure 4.62 it is reiterated that the solutions produced using the GA are indifferent to the employed metric scheme. However, the solutions produced using the PSO algorithm are different with the different employed metric schemes. The PSO algorithm incurs less wait time in comparison to the solutions obtained using the GA. Considering the time window durations of each of the customers in RC207, shown in Figure 4.63, it is noted that the customers have varying time window durations between 100 and 600 time units, which allows for flexibility in the placement of customers into service routes, the variability in the obtained solutions and decrease in the number of dispatched vehicles compensated by incurring a greater total travelled distance.

4.3 50 Customers

This section contains and discusses the results obtained to the VRPTW for Solomon's data set containing 50 customers.

	Benchmark Result		Benchmark PSO Result			GA										PSO									
						Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
C101	**5**	**362.4**	5	363.25	117.6	**5**	416.29	14.87	118.72	325.5	**5**	416.29	14.87	118.72	314.67	**5**	421.86	16.41	151.18	213.8	**5**	421.86	16.41	151.18	458.62
C102	**5**	**361.4**	5	361.17	124.8	**5**	388.29	7.44	256.79	326.26	**5**	388.29	7.44	256.79	375.05	**5**	418.93	15.92	558.94	480.96	**5**	418.67	15.85	716.77	477.21
C103	**5**	**361.4**	5	361.17	144.6	**5**	385.75	6.74	326.97	309.54	**5**	387.01	7.09	356.74	405.25	**5**	421.02	16.5	877.27	578.15	**5**	421.02	16.5	877.27	597.58
C104	**5**	**358**	5	358.88	185.4	**5**	391.13	9.25	227.37	311.09	**5**	389.45	8.78	226.93	392.69	**5**	457.79	27.87	774.78	877.75	**5**	420.7	17.51	1062.11	865.22
C105	**5**	**362.4**	5	363.25	121.8	**5**	396.68	9.46	119.18	326.38	**5**	396.68	9.46	119.18	387.74	**5**	363.25	0.23	119.18	480.67	**5**	363.25	0.23	119.18	488.32
C106	**5**	**362.4**	5	363.25	121.2	**5**	407.59	12.47	0	323.75	**5**	407.59	12.47	0	392.25	**5**	363.25	0.23	67.18	316.96	**5**	363.25	0.23	67.18	516.43
C107	**5**	**362.4**	5	363.25	129	**5**	398.01	9.83	94.18	316.41	**5**	398.01	9.83	94.18	425.6	**5**	363.25	0.23	94.18	334.51	**5**	363.25	0.23	94.18	444.48
C108	**5**	**362.4**	5	363.25	129	**5**	399.03	10.11	16.51	300.3	**5**	399.16	10.14	160.38	406.2	**5**	380.1	4.88	15.75	515.89	**5**	375.51	3.62	17.18	529.79
C109	**5**	**362.4**	5	363.25	126	**5**	389.64	7.52	1.87	302.65	**5**	393.76	8.65	21.54	397.44	**5**	378.74	4.51	0	627.09	**5**	376.66	3.94	0	580.35
C201	**3**	**360.2**	2	444.96	115.8	2	444.96	23.53	1177.17	1278.55	2	444.96	23.53	1177.17	757.91	2	444.96	23.53	1177.17	241.82	2	444.96	23.53	1177.17	440.49
C202	**3**	**360.2**	2	403.81	140.4	2	403.81	12.11	1218.33	1539.45	2	403.81	12.11	1218.33	694.42	2	528.92	46.84	1265.3	377.36	2	528.92	46.84	1265.3	377.36
C203	**3**	**359.8**	2	402.52	149.4	2	410.63	14.13	1212.82	2061.54	2	416.17	15.67	1121.7	633.13	2	568.04	57.88	1587.78	595.26	2	522.76	45.29	1607.01	598.78
C204	**2**	**350.1**	2	356.77	255.6	**2**	357.83	2.21	1095.48	705.92	**2**	361.52	3.26	765.9	784.63	**2**	545.81	55.9	1531.81	834.38	**2**	528.34	50.91	1620.12	1537.13
C205	**3**	**359.8**	2	429.12	118.8	2	445.71	23.88	1189.43	1575.63	2	444.08	23.42	1189.43	592.68	2	444.96	23.67	1097.17	394.69	2	441.53	22.72	1257.3	466.33
C206	**3**	**359.8**	2	412.5	122.4	2	409.61	13.84	1332.3	1579.17	2	409.61	13.84	1332.3	558.79	2	448.52	24.66	1067.17	495.34	2	443.21	23.18	1245.52	616.72
C207	**3**	**359.6**	2	426.13	131.4	2	398.33	10.77	1224.06	569.93	2	398.54	10.83	1224.06	566.84	2	477.35	32.75	1250.88	565.28	2	474.85	32.05	1221.97	652.56
C208	**2**	**350.5**	2	352.29	125.4	**2**	352.12	0.46	1277.13	1736	**2**	352.12	0.46	1277.13	577.56	**2**	368.72	5.2	1029.46	388.11	**2**	368.72	5.2	1029.46	368.49

TABLE 4.12: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Clustered Problem Instances containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark		PSO Benchmark		GA				PSO			
	Fitness		Fitness		Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
C101	**500.36**	**6.00**	500.36	6.00	500.42	0.01	6.00	0.00	500.42	0.01	6.00	0.00
C102	**500.36**	**6.00**	500.36	6.00	500.39	0.01	6.00	0.00	500.42	0.01	6.00	0.00
C103	**500.36**	**6.00**	500.36	6.00	500.39	0.00	6.00	0.00	500.42	0.01	6.00	0.00
C104	**500.36**	**6.00**	500.36	6.00	500.39	0.01	6.00	0.00	500.46	0.02	6.00	0.00
C105	**500.36**	**6.00**	500.36	6.00	500.40	0.01	6.00	0.00	500.36	0.00	6.00	0.00
C106	**500.36**	**6.00**	500.36	6.00	500.41	0.01	6.00	0.00	500.36	0.00	6.00	0.00
C107	**500.36**	**6.00**	500.36	6.00	500.40	0.01	6.00	0.00	500.36	0.00	6.00	0.00
C108	**500.36**	**6.00**	500.36	6.00	500.40	0.01	6.00	0.00	500.38	0.00	6.00	0.00
C109	**500.36**	**6.00**	500.36	6.00	500.39	0.01	6.00	0.00	500.38	0.00	6.00	0.00
C201	**300.36**	**4.00**	200.44	3.00	200.44	-33.27	3.00	-0.33	200.44	-33.27	3.00	-0.33
C202	**300.36**	**4.00**	200.40	3.00	200.40	-33.28	3.00	-0.33	200.53	-33.24	3.00	-0.33
C203	**300.36**	**4.00**	200.40	3.00	200.41	-33.28	3.00	-0.33	200.57	-33.22	3.00	-0.33
C204	**200.35**	**3.00**	200.36	3.00	200.36	0.00	3.00	0.00	200.55	0.10	3.00	0.00
C205	**300.36**	**4.00**	200.43	3.00	200.45	-33.26	3.00	-0.33	200.44	-33.27	3.00	-0.33
C206	**300.36**	**4.00**	200.41	3.00	200.41	-33.28	3.00	-0.33	200.45	-33.26	3.00	-0.33
C207	**300.36**	**4.00**	200.43	3.00	200.40	-33.28	3.00	-0.33	200.48	-33.25	3.00	-0.33
C208	**200.35**	**3.00**	200.35	3.00	200.35	0.00	3.00	0.00	200.37	0.01	3.00	0.00

TABLE 4.13: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Clustered Problem Instances containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark		Benchmark			GA										PSO									
	Result		PSO Result			Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
R101	**12**	**1044**	11	1100.72	141.6	**12**	1050.38	0.61	619.93	505.47	**12**	1054.75	1.03	610.41	604.67	**12**	1069.05	2.4	595.74	500.06	**12**	1069.05	2.4	595.74	500.06
R102	**11**	**909**	10	923.71	157.8	10	934.68	2.82	398.06	435.24	10	930.98	2.42	393.83	389.01	10	996.02	9.57	470.14	551.45	10	992.05	9.14	453.42	553.72
R103	**9**	**772.9**	8	790.17	244.2	8	786.23	1.72	283.03	364.46	8	786.23	1.72	283.03	420.71	8	841.09	8.82	388.05	545.69	8	857.91	11	399.09	532.04
R104	**6**	**625.4**	6	631.58	268.8	**6**	638.57	2.11	56.37	307.94	**6**	631.39	0.96	100.52	379.89	**6**	715.36	14.38	139.76	632.38	**6**	715.36	14.38	139.76	956.96
R105	**9**	**899.9**	8	983.49	159	**9**	917.6	1.97	241.24	533.07	**9**	914.31	1.6	228.92	308.02	**9**	956.64	6.31	215.63	632.41	**9**	952.47	5.84	201.8	561.93
R106	**5**	**793**	7	865.93	174	8	800.63	0.96	169.91	372.16	8	812.15	2.42	158.84	483.16	7	891.44	12.41	95.55	587.55	7	880.56	11.04	110.58	526.54
R107	**7**	**711.1**	6	737.1	202.8	6	750.78	5.58	48.16	319.56	6	745.36	4.82	37.4	374.3	**7**	795.21	11.83	253.22	623.61	**7**	787.55	10.75	277.78	666.76
R108	**6**	**617.7**	6	624.29	268.8	**6**	623.95	1.01	44.59	314.33	**6**	622.52	0.78	66.04	402.12	**6**	715.31	15.8	97.21	758.68	**6**	724.23	17.25	107.31	751.66
R109	**8**	**786.8**	7	806.97	218.4	**8**	818.77	4.06	171.28	468.92	7	848.74	7.87	67.29	485.6	7	846.59	7.6	70.9	552.78	7	881.36	12.02	70.34	647.96
R110	**7**	**697**	7	720.4	195.6	**7**	723.4	3.79	178.35	345.88	**7**	731.24	4.91	165.85	440.72	**7**	800.37	14.83	234.76	810.5	**7**	793.85	13.9	167.05	634.36
R111	**7**	**707.2**	6	756.35	247.2	**7**	722.42	2.15	113.17	330.02	**7**	715.42	1.16	116.62	428.31	**7**	793.12	12.15	171.34	794.73	**7**	810.46	14.6	224.47	588.3
R112	**6**	**630.2**	6	638.49	244.2	**6**	641.85	1.85	87.09	419.54	**6**	639.07	1.41	91.09	422.57	**6**	713.07	13.15	75.89	839.57	**6**	688.91	9.32	69.43	721.76
R201	**6**	**791.9**	2	953.29	190.2	2	953.29	20.38	243.65	1209.4	2	953.29	20.38	243.65	484.69	3	995.77	25.74	853.73	506.29	3	997.97	26.02	933.42	501.9
R202	**5**	**698.5**	2	815.23	250.2	2	827.31	18.44	357.8	1746.02	2	830.23	18.86	373.88	569.33	2	1004.29	43.78	373.61	402.58	2	1004.29	43.78	373.61	412.63
R203	**5**	**605.3**	2	668.36	289.2	2	685.21	13.2	520.44	2045.32	2	685.93	13.32	500.53	611.49	2	897.09	48.21	462.09	443.69	2	875.08	44.57	559.34	412.27
R204	**2**	**506.4**	2	518.57	319.8	**2**	516.21	1.94	594.64	2619.4	**2**	519.18	2.52	578.46	780.83	**2**	679.48	34.18	473	529.25	**2**	656.52	29.65	535.97	422.42
R205	**4**	**690.1**	2	756.38	289.8	2	750.89	8.81	400	1623.83	2	770.79	11.69	425.9	535.6	2	950.34	37.71	222.59	521.48	2	950.34	37.71	222.59	420.58
R206	**4**	**632.4**	2	661.55	256.2	2	675.32	6.79	464.34	1914.36	2	682.23	7.88	405.48	567.23	2	838.73	32.63	411.86	513.82	2	838.73	32.63	411.86	385.12
R207			2	593.95	349.2	2	616.04		420.5	2071.73	2	616.94		413.37	625.57	2	806.03		468.95	641.5	2	806.03		468.95	454.26
R208			2	508.41	309.6	2	505.62		527.29	2901.3	2	512.3		447.94	848.62	2	651.08		701.06	663.8	2	651.08		701.06	445.93
R209	**4**	**600.6**	2	658.28	315	2	680.18	13.25	364.91	906.45	2	679.69	13.17	332.39	544.45	2	854.83	42.33	320.01	552.27	2	854.83	42.33	320.01	424.74
R210	**4**	**645.6**	2	670.99	244.2	2	678.99	5.17	437.05	64.21	2	679.49	5.25	440.04	623.29	2	910.45	41.02	430.03	547.3	2	879.49	36.23	481.52	437.68
R211	**3**	**535.5**	2	562.74	295.2	2	561.06	4.77	381.63	2320.81	2	579.24	8.17	393.48	684.01	2	784.74	46.54	290.12	809.91	2	784.74	46.54	290.12	759.36

TABLE 4.14: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Random Problem Instances containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark		PSO Benchmark		GA				PSO			
	Fitness		Fitness		Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
R101	**1201.04**	**13**	1101.1	12	1201.05	0	13	0	1201.07	0	13	0
R102	**1100.91**	**12**	1000.92	11	1000.93	-9.08	11	-0.09	1001	-9.08	11	-0.09
R103	**900.77**	**10**	800.79	9	800.79	-11.1	9	-0.11	800.84	-11.09	9	-0.11
R104	**600.63**	**7**	600.63	7	600.64	0	7	0	600.72	0.01	7	0
R105	**900.9**	**10**	800.98	9	900.92	0	10	0	900.96	0.01	10	0
R106	**500.79**	**6**	700.87	8	800.8	59.91	9	0.6	700.89	39.96	8	0.4
R107	**700.71**	**8**	600.74	7	600.75	-14.27	7	-0.14	700.8	0.01	8	0
R108	**600.62**	**7**	600.62	7	600.62	0	7	0	600.72	0.02	7	0
R109	**800.79**	**9**	700.81	8	800.82	0	8	-0.12	700.85	-12.48	8	-0.12
R110	**700.7**	**8**	700.72	8	700.72	0	8	0	700.8	0.01	8	0
R111	**700.71**	**8**	600.76	7	700.72	0	8	0	700.79	0.01	8	0
R112	**600.63**	**7**	600.64	7	600.64	0	7	0	600.71	0.01	7	0
R201	**600.79**	**7**	200.95	3	200.95	-66.55	3	-0.67	301	-49.9	4	-0.5
R202	**500.7**	**6**	200.82	3	200.83	-59.89	3	-0.6	201	-59.86	3	-0.6
R203	**500.61**	**6**	200.67	3	200.69	-59.91	3	-0.6	200.9	-59.87	3	-0.6
R204	**200.51**	**3**	200.52	3	200.52	0	3	0	200.68	0.09	3	0
R205	**400.69**	**5**	200.76	3	200.75	-49.9	3	-0.5	200.95	-49.85	3	-0.5
R206	**400.63**	**5**	200.66	3	200.68	-49.91	3	-0.5	200.84	-49.87	3	-0.5
R207			200.59	3	200.62		3		200.81		3	
R208			200.51	3	200.51		3		200.65		3	
R209	**400.6**	**5**	200.66	3	200.68	-49.91	3	-0.5	200.85	-49.86	3	-0.5
R210	**400.65**	**5**	200.67	3	200.68	-49.91	3	-0.5	200.91	-49.85	3	-0.5
R211	**300.54**	**4**	200.56	3	200.56	-33.27	3	-0.33	200.78	-33.19	3	-0.33

TABLE 4.15: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Random Problem Instances containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Result		Benchmark PSO Result			GA										PSO									
						Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
RC101	**8**	**944**	8	945.58	231.6	**8**	950.65	0.7	161.91	411.92	**8**	950.65	0.7	161.91	468.12	**8**	977.05	3.5	154.7	618.53	**8**	977.05	3.5	154.7	643.01
RC102	**7**	**822.5**	7	823.97	207	**7**	876.36	6.55	98.72	430.04	**7**	901.7	9.63	73.56	473.87	**7**	836.89	1.75	146.02	869.48	**7**	836.89	1.75	146.02	752.32
RC103	**6**	**710.9**	6	712.91	211.2	7	796.65	12.06	137.52	431.46	7	800.64	12.62	141.08	439.25	**6**	731.17	2.85	124.21	1080.16	**6**	731.17	2.85	124.21	1080.16
RC104	**5**	**545.8**	5	546.51	206.4	6	652.58	19.56	130.23	272.2	6	647.03	18.55	147.03	408.97	**5**	581.03	6.45	88.42	880.44	**5**	581.03	6.45	88.42	958.48
RC105	**8**	**855.3**	8	856.97	143.4	**8**	859.55	0.5	268.92	243.03	**8**	863.72	0.98	255.66	500.65	**8**	906.52	5.99	204.78	1158.1	**8**	906.52	5.99	204.78	1192.82
RC106	**6**	**723.2**	6	724.65	136.8	7	840.19	16.18	97.32	235.59	7	839.31	16.05	87.66	465.55	**6**	751.51	3.91	19.54	812.51	**6**	751.51	3.91	19.54	990.99
RC107	**6**	**642.7**	6	645.7	139.8	7	753.54	17.25	137.39	426.35	**6**	750.81	16.82	68.59	433.79	**6**	713.16	10.96	92.59	1636.44	**6**	713.16	10.96	92.59	1494.39
RC108	**6**	**598.1**	6	599.17	141.6	**6**	710.03	18.71	38.26	548.95	**6**	690.95	15.52	38.26	433.2	**6**	661.88	10.66	76.13	1137.22	**6**	652.29	9.06	118.15	824.2
RC201	**5**	**684.8**	3	838.76	159	3	848.4	23.89	1005.29	875.68	3	848.79	23.95	1006.62	453.01	3	912.3	33.22	989.29	432.22	3	912.3	33.22	989.29	487.58
RC202	**5**	**613.6**	2	867.26	250.2	2	867.83	41.43	304.83	1403.87	2	870.74	41.91	301.92	557.74	3	919.57	49.86	1145.84	465.43	3	810.86	32.15	1239.8	454.33
RC203	**4**	**555.3**	2	674.44	231	2	685.24	23.4	467.75	1819.32	2	677.36	21.98	475.63	648.11	2	815.07	46.78	440.45	385.47	2	815.07	46.78	440.45	347.63
RC204	**3**	**444.2**	2	479.22	192	2	485.71	9.35	675.61	2334.05	2	482.09	8.53	652.55	704.57	2	591.48	33.16	573.84	655.61	2	582.81	31.21	580.12	485.38
RC205	**5**	**630.2**	3	765.02	164.4	3	768.35	21.92	1189.93	1057.4	3	769.01	22.03	1097.83	431.21	3	936.63	48.62	986.29	402.65	3	936.63	48.62	986.29	501.3
RC206	**5**	**610**	2	755.13	168.6	2	761.7	24.87	326.92	1352.77	2	759.09	24.44	329.53	477.73	2	917.14	50.35	221.44	423.85	2	917.14	50.35	221.44	404
RC207	**4**	**558.6**	2	655.81	300.6	2	680.82	21.88	316.19	1749.55	2	681.09	21.93	441.6	511.91	2	867.17	55.24	296.91	422.33	2	855.12	53.08	383.52	482.74
RC208	**99**	**99**	2	498.79	287.4		524.91		389.68	627.6		523.87		381.63	454.26		708.84		352.42	873.87		686.86		368.56	603.09

TABLE 4.16: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Random Clustered Problem Instances containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Fitness		PSO Benchmark Fitness		GA				PSO			
					Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
RC101	**800.94**	**9.00**	800.95	9.00	800.95	0.00	9.00	0.00	800.98	0.00	9.00	0.00
RC102	**700.82**	**8.00**	700.82	8.00	700.88	0.01	8.00	0.00	700.84	0.00	8.00	0.00
RC103	**600.71**	**7.00**	600.71	7.00	700.80	16.66	8.00	0.17	600.73	0.00	7.00	0.00
RC104	**500.55**	**6.00**	500.55	6.00	600.65	20.00	7.00	0.20	500.58	0.01	6.00	0.00
RC105	**800.86**	**9.00**	800.86	9.00	800.86	0.00	9.00	0.00	800.91	0.01	9.00	0.00
RC106	**600.72**	**7.00**	600.72	7.00	700.84	16.67	8.00	0.17	600.75	0.00	7.00	0.00
RC107	**600.64**	**7.00**	600.65	7.00	700.75	16.67	7.00	0.00	600.71	0.01	7.00	0.00
RC108	**600.60**	**7.00**	600.60	7.00	600.71	0.02	7.00	0.00	600.66	0.01	7.00	0.00
RC201	**500.68**	**6.00**	300.84	4.00	300.85	-39.91	4.00	-0.40	300.91	-39.90	4.00	-0.40
RC202	**500.61**	**6.00**	200.87	3.00	200.87	-59.88	3.00	-0.60	300.92	-39.89	4.00	-0.40
RC203	**400.56**	**5.00**	200.67	3.00	200.69	-49.90	3.00	-0.50	200.82	-49.87	3.00	-0.50
RC204	**300.44**	**4.00**	200.48	3.00	200.49	-33.27	3.00	-0.33	200.59	-33.24	3.00	-0.33
RC205	**500.63**	**6.00**	300.77	4.00	300.77	-39.92	4.00	-0.40	300.94	-39.89	4.00	-0.40
RC206	**500.61**	**6.00**	200.76	3.00	200.76	-59.90	3.00	-0.60	200.92	-59.87	3.00	-0.60
RC207	**400.56**	**5.00**	200.66	3.00	200.68	-49.90	3.00	-0.50	200.87	-49.85	3.00	-0.50
RC208			200.50	3.00	200.52		3.00		200.71		3.00	

TABLE 4.17: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Random Clustered Problem Instances containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	GA								PSO							
	Metric A				Metric B				Metric A				Metric B			
Dataset	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10
C1	0.00	0.00	66.67	33.33	0.00	0.00	66.67	33.33	0.00	0.00	55.56	44.44	0.00	0.00	55.56	44.44
C2	0.00	0.00	25.00	75.00	0.00	0.00	25.00	75.00	0.00	0.00	12.50	87.50	0.00	0.00	12.50	87.50
R1	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	41.67	58.33	0.00	0.00	33.33	66.67
R2	0.00	0.00	45.45	36.36	0.00	0.00	36.36	45.45	0.00	0.00	0.00	81.82	0.00	0.00	0.00	81.82
RC1	0.00	0.00	37.50	62.50	0.00	0.00	37.50	62.50	0.00	0.00	75.00	25.00	0.00	0.00	87.50	12.50
RC2	0.00	0.00	12.50	75.00	0.00	0.00	12.50	75.00	0.00	0.00	0.00	87.50	0.00	0.00	0.00	87.50

TABLE 4.18: The percentage (%) of problem instances per class containing 50 Customers classified under the stipulated total travelled distance deviation ranges, for the Applied Solution Techniques and Employed Evaluation Metrics

	GA			PSO		
Dataset	Metric A = Metric B	Metric A < Metric B	Metric A > Metric B	Metric A = Metric B	Metric A < Metric B	Metric A > Metric B
C1	44.44	33.33	22.22	55.56	0.00	44.44
C2	50.00	37.50	12.50	37.50	0.00	62.50
R1	0.00	41.67	58.33	16.67	33.33	50.00
R2	9.09	81.82	9.09	63.64	9.09	27.27
RC1	12.50	37.50	50.00	87.50	0.00	12.50
RC2	0.00	50.00	50.00	50.00	0.00	50.00

TABLE 4.19: The percentage (%) of problem instances containing 50 Customers with the total travelled distance per class classified under the stipulated metric relations, for the Applied Solution Techniques and Employed Evaluation Metrics.

	GA						PSO					
	Metric A			Metric B			Metric A			Metric B		
Dataset	Equal	< 0	> 0	Equal	< 0	> 0	Equal	< 0	> 0	Equal	< 0	> 0
C1	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
C2	0.00	75.00	25.00	0.00	75.00	25.00	0.00	75.00	25.00	0.00	75.00	25.00
R1	0.00	25.00	75.00	0.00	33.33	66.67	0.00	25.00	75.00	0.00	25.00	75.00
R2	0.00	72.73	9.09	0.00	72.73	9.09	0.00	72.73	9.09	0.00	72.73	9.09
RC1	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
RC2	0.00	87.50	0.00	0.00	87.50	0.00	0.00	87.50	0.00	0.00	87.50	0.00

TABLE 4.20: Summary of Fitness Deviation Classifications of Fitness relative to benchmark fitness.

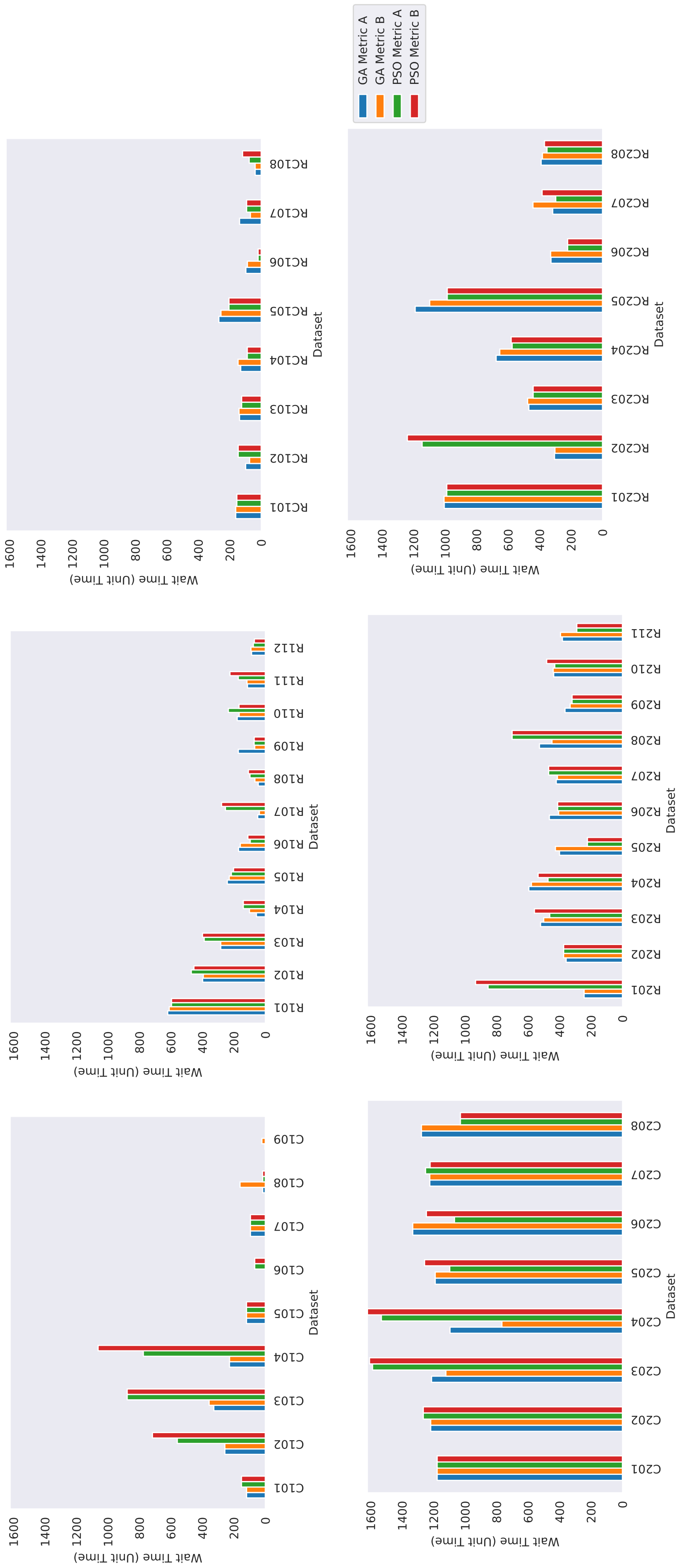


FIGURE 4.33: Wait Time (Unit Time) for Problem instances per class containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics

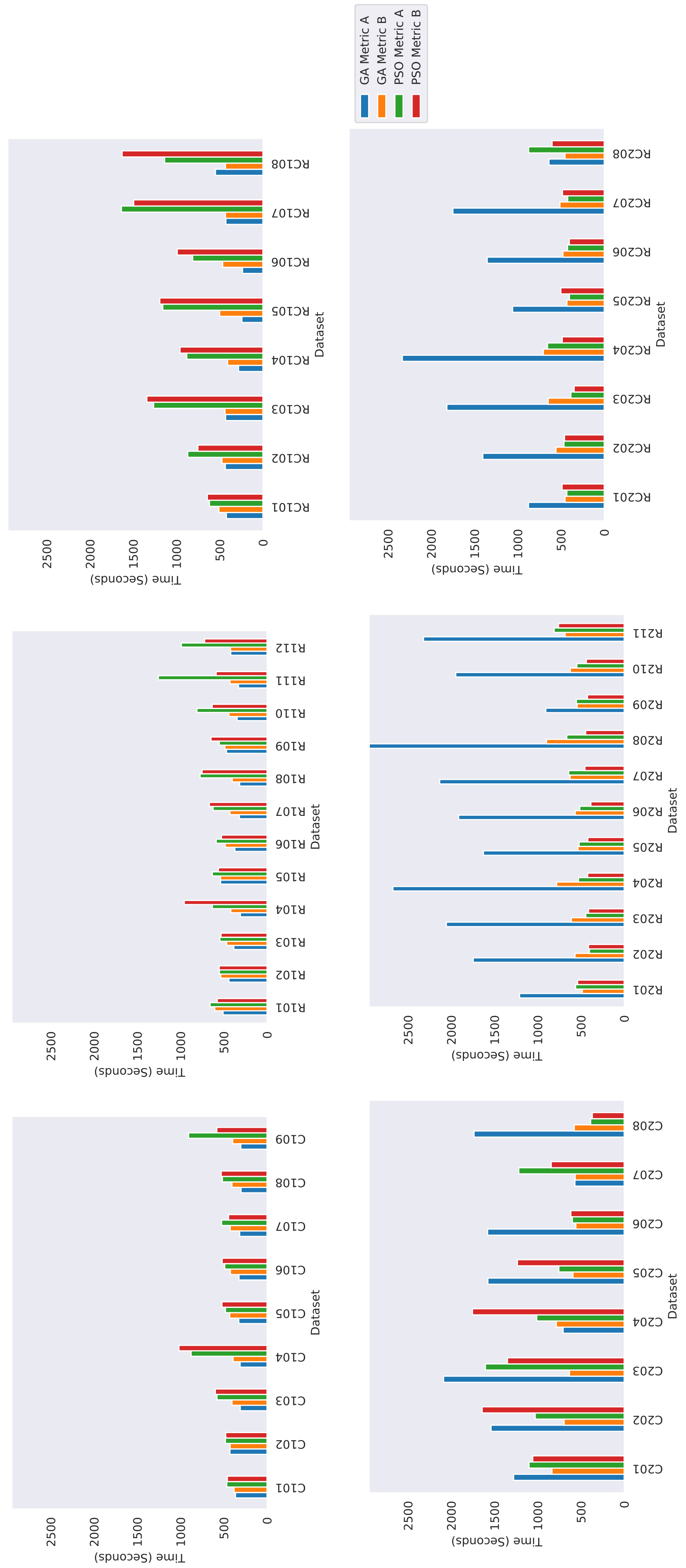


FIGURE 4.34: CPU Time (Seconds) for Problem instances per class containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics

4.3.1 Clustered

From the recorded results in Table 4.12 and Table 4.13 to the problem instances with customers spatially clustered, the following are noted for the problem instanced in temporal class 1 and 2.

Temporal Class 1

- From the results obtained to the solutions to the problem instances in C1, the following are noted:
  - The number of dispatched vehicles in the obtained solutions correlate to both the recorded best and PSO benchmark values.
  - The corresponding total travelled distance percentage deviation to the benchmark value is greater than 0% for either applied solution technique.
  - In Table 4.18 it is recorded that 66.67% of the problem instances obtain a distance deviation percentage between $0\% - 10\%$, and 33.33% of the problem instances have a percentage distance deviation greater than 10% when the GA is applied. When the PSO algorithm is applied, a distance deviation percentage between $0\% - 10\%$ is obtained for 55.56% of it's problem instances and for 44.44% of the problem instances it has a distance deviation of greater than 10%.
- It is significant to note that the solutions produced by the GA are not indifferent to the employed metric scheme as highlighted for the C1 class containing 25 customers. To investigate the reason for the variability which now occurs in the produced solution in the same spatial and temporal class, the problem instance C104 is considered. The resultant topologies of the produced solutions are given in Figure 4.39.

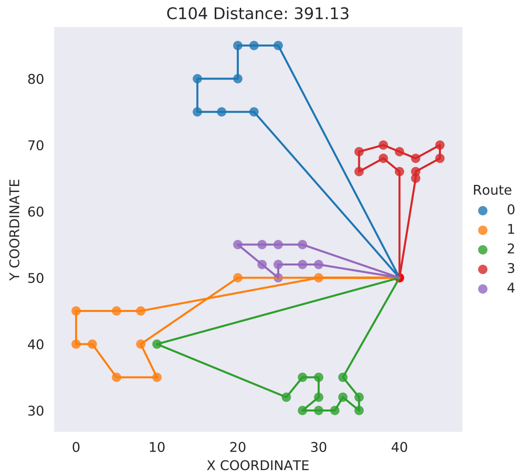


FIGURE 4.35: Topology of a Solution to C104 Obtained Using GA with Metric A

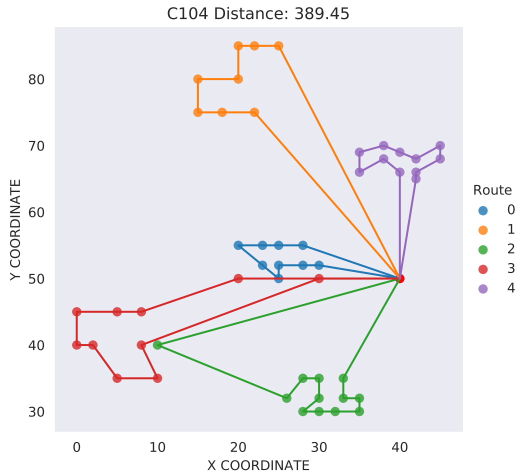


FIGURE 4.36: Topology of a Solution to C104 Obtained Using GA with Metric B

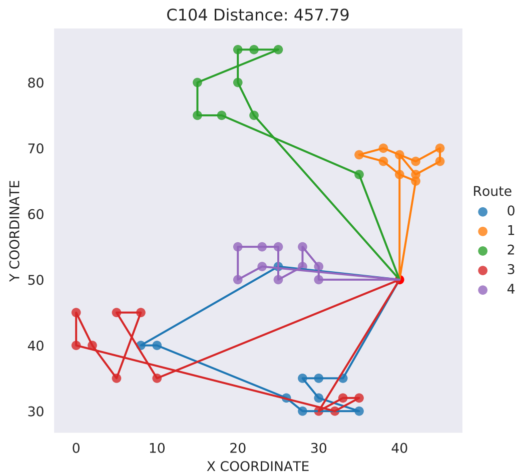


FIGURE 4.37: Topology of a Solution to C104 Obtained Using PSO with Metric A

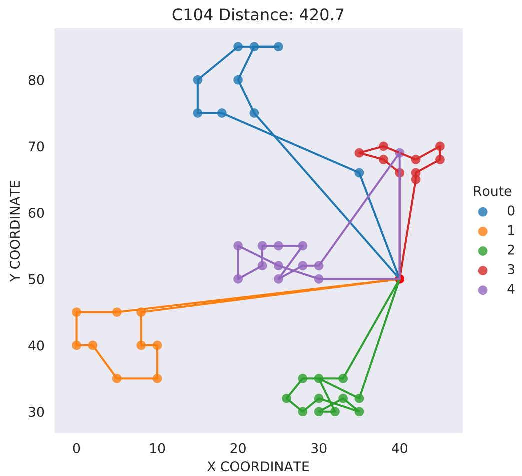


FIGURE 4.38: Topology of a Solution to C104 Obtained Using PSO with Metric B

FIGURE 4.39: Topology Plots of C104

The resultant constructed service routes shown in Figure 4.39 are distinct for each of the applied solution techniques and metric schemes. The solution obtained using the PSO algorithm with Metric B, shown in Figure 4.38 has routes which are most clustered in comparison to the other topological solutions shown in Figure 4.39. However, the total distance obtained using the GA with Metric B is the least in comparison to the rest of the obtained solutions for this problem instance. Furthermore, the total incurred wait time for the solutions obtained through the application of the GA is less than that when using the PSO algorithm. Comparing the routes obtained when applying the GA, in Figure 4.35 and Figure 4.36, each of the routes service the same cluster of customers. However, there is a difference in the resultant total travelled distance. This is a result of the different ordering of when customers are visited in each of the corresponding constructed routes. The time window duration for each of the customers in C104 problem instance are shown in Figure 4.40.

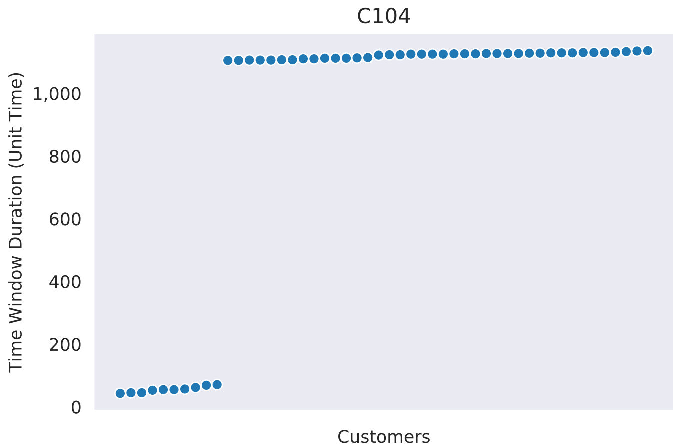


FIGURE 4.40: Time Window Duration of Customers in C104

The time window durations of customers in C104 are shown in Figure 4.40. It is noted that C104 has a few customers with tight time window durations which are less than 200 time units, whilst th majority of customers have a wide time window duration between 1000 and 1200 time units. The wider time windows and larger dataset consequentially allows flexibility of when to service the customers in the dataset. It must be highlighted that although the wide time windows would allow for variability in the route designs, in contrast to solutions produced under the PSO algorithm, the solutions produced under the GA bear resemblance to each other. Thus, the stochastic nature of the PSO algorithm is highlighted. Factors contributing to the variability in the PSO produced solutions are wide time window durations and the probabilistic nature of the PSO algorithm.

- The solutions produced to problem instance C108 is of particular interest. The solution obtained by applying the GA with Metric A has a total travelled distance which is 0.13 distance units less than that obtained by applying the GA with Metric B. Furthermore, the total wait time incurred for the solution obtained by the GA with Metric A is 10.2% of that of the solution obtained by applying the GA with Metric B. Comparing the solutions obtained by applying the PSO algorithm, both the total travelled distance and wait time are less than the corresponding results of the GA solutions. The topologies of the designed routes obtained by applying the GA and PSO algorithm with the two metric formulations are shown in Figure 4.45.

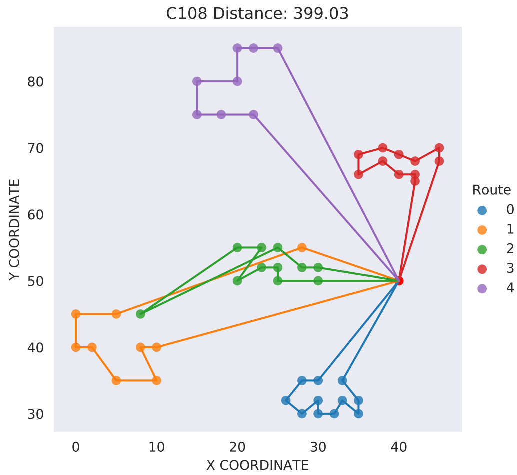


FIGURE 4.41: Topology of a Solution to C108 Obtained Using GA with Metric A

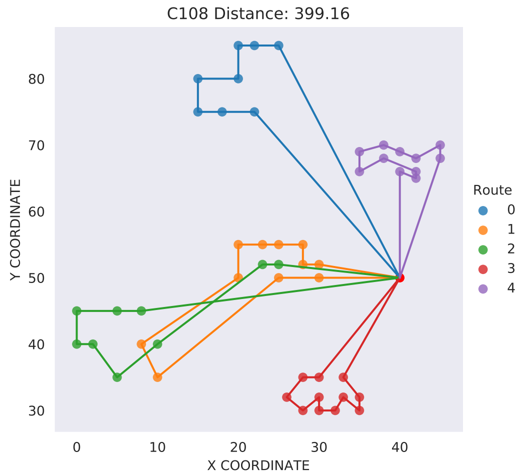


FIGURE 4.42: Topology of a Solution to C108 Obtained Using GA with Metric B

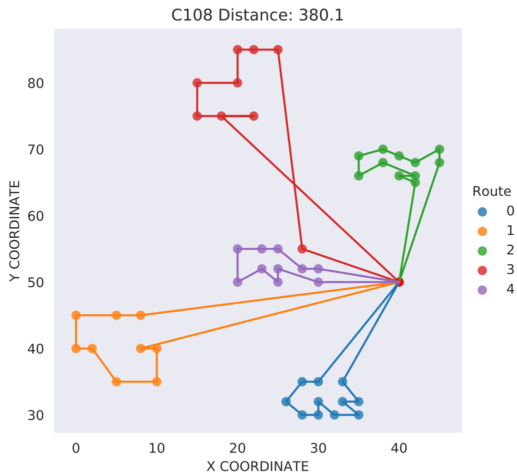


FIGURE 4.43: Topology of a Solution to C108 Obtained Using PSO with Metric A

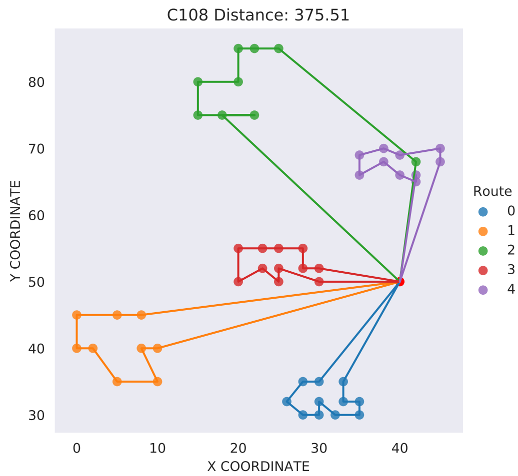


FIGURE 4.44: Topology of a Solution to C108 Obtained Using PSO with Metric B

FIGURE 4.45: Topology Plots of C108

From the results recorded in Table 4.12, it is significant to highlight that the wait time for the solution produced using the GA with Metric A is 16.51 time units and when Metric B is employed the wait time is 160.38 time units. This pronounced wait time difference is attributed to the difference in the design of Routes 1 and 2 shown in Figure 4.41 and Figure 4.42. The solutions obtained by applying the PSO algorithm are found to better cluster the customers, hence reducing the total travelled distance.

Temporal Class 2

- From the results obtained to the solutions to the problem instances in C2, the following are noted:

- ◦ The obtained solutions dispatch an equal or fewer number of service vehicles in comparison to the benchmark value. The resultant number of vehicles to procured by the solution techniques are equal to the PSO benchmark value.
- ◦ The total travelled distance by the dispatched vehicles are greater than that recorded as the benchmark value, as a consequence of dispatching fewer vehicles. For the GA, 75% of the problem instances and 87.5% of the problem instances with solutions obtained by applying the PSO algorithm have a distance deviation which is greater than 10%. The total distance travelled result of the solutions obtained by applying the GA, bears resemblance to the PSO benchmark distance values. The distance result of the solutions obtained by applying the PSO algorithm are greater in comparison to the GA distance result, hence have a greater distance deviation to the benchmark distance values.
- ◦ The total incurred wait time is greater than that for the clustered temporal class one as a consequence of dispatching fewer service vehicles.

- The solutions produced to problem instance C203 is of particular interest as the PSO solutions incur a greater total travelled distance and total wait time than the solutions produced using the GA algorithm. The topologies of the designed routes obtained by applying the GA and PSO algorithm with the two metric formulations are shown in Figure 4.50.

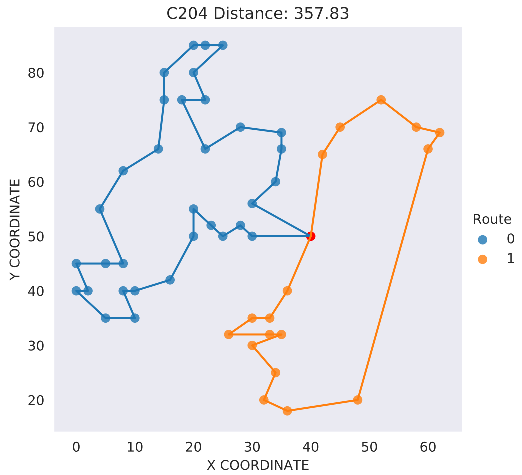


FIGURE 4.46: Topology of a Solution to C204 Obtained Using GA with Metric A

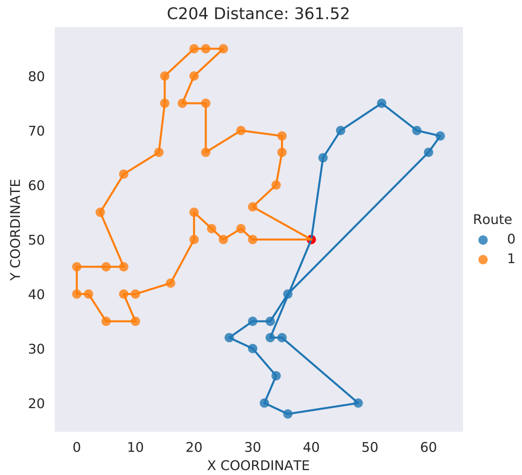


FIGURE 4.47: Topology of a Solution to C204 Obtained Using GA with Metric B

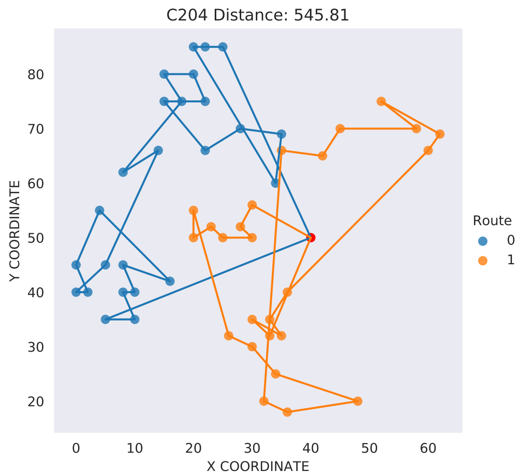


FIGURE 4.48: Topology of a Solution to C204 Obtained Using PSO with Metric A

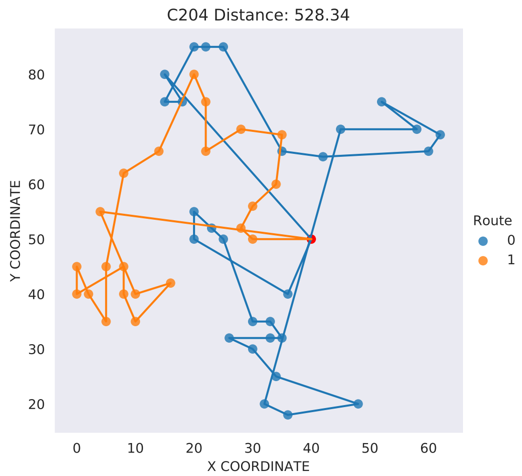


FIGURE 4.49: Topology of a Solution to C204 Obtained Using PSO with Metric B

FIGURE 4.50: Topology Plots of C204

From the topological solutions shown in Figure 4.50, it is found that the produced route solutions by applying the GA are visibly clustered in contrast to the routes constructed by applying the PSO algorithm. Furthermore, the GA solutions, Figure 4.46 and Figure 4.47, route differences are distinctly visible in contrast to the PSO solutions produced using the two diffrent solution evaluation metrics. Considering the nature of the applied algorithms it is noted that the GA performs crossover by finding the best position to place a customer in routes in a bruteforce manner, whilst the PSO's stochastic nature of probabilistically updating a route is accentuated in the respective more organised and random route structures. The time window durations of customers in C204 are shown in Figure 4.51.

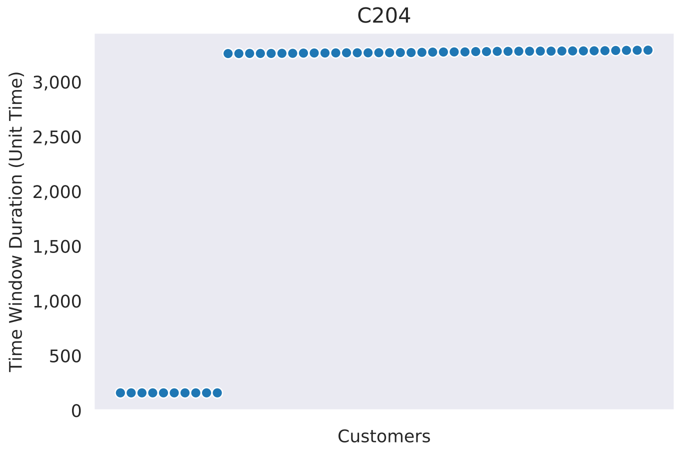


FIGURE 4.51: Time Window Duration of Customers in C204

From the time window durations of customers in C204 are shown in Figure 4.51, it is reiterated that the wide time windows are a contributing factor in the variability of the obtained solutions.

4.3.2 Random

The randomly spatially organised customers results are recorded in Table 4.14 and Table 4.15.

Temporal Class 1

- From the results obtained to the problem instances in R1, the following are noted:
  - The total travelled distance for all the R1 problem instances have a distance deviation between 0% – 10%, for 100% of the problem instances when the GA is employed. However, when the PSO is applied, results with distance deviation greater than 10% are also obtained.
  - The solutions produced dispatch an equal number or fewer vehicles than the benchmark value, and an equal number to the PSO benchmark value. However, this holds with the exception of problem instance R106.
- The problem instance R106 is of particular interest as its the only problem instance with a solution which dispatches more vehicles than both the best and PSO benchmark values in the R1 class. The service routes constructed are are shown in Figure 4.56.

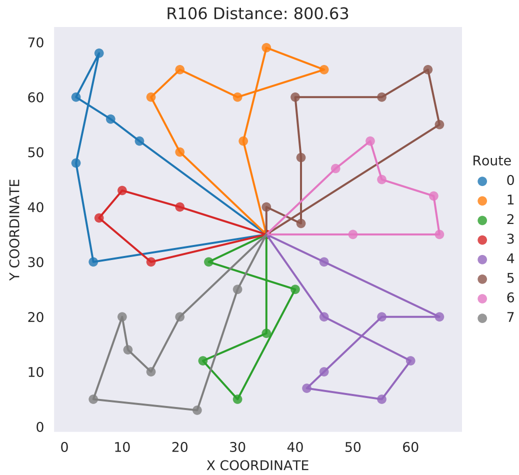


FIGURE 4.52: Topology of a Solution to R106 Obtained Using GA with Metric A

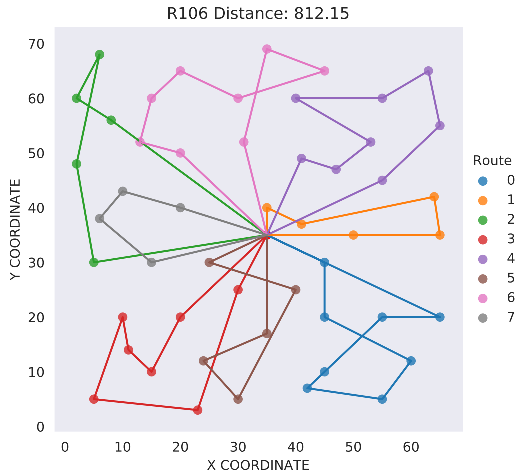


FIGURE 4.53: Topology of a Solution to R106 Obtained Using GA with Metric B

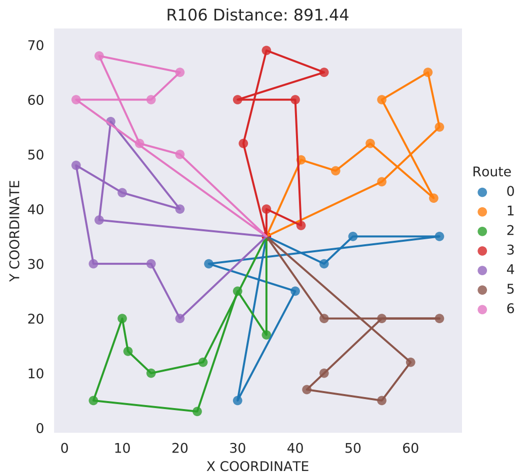


FIGURE 4.54: Topology of a Solution to R106 Obtained Using PSO with Metric A

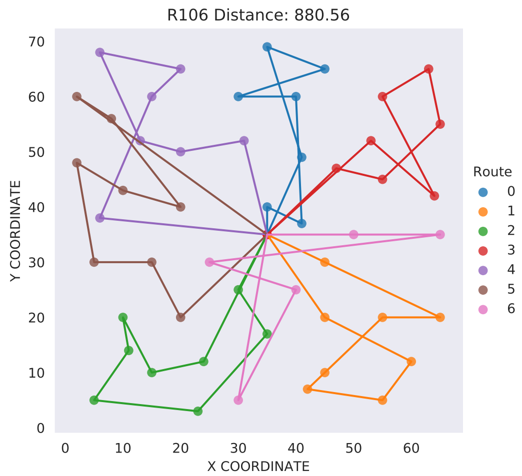


FIGURE 4.55: Topology of a Solution to R106 Obtained Using PSO with Metric B

FIGURE 4.56: Topology Plots of R106

The solutions produced using the GA with Metric A and Metric B, are respectively shown in Figure 4.52 and Figure 4.53. The only difference in the two solutions are corresponding Routes 5 and 6 in Figure 4.52 to Routes 1 and 4 in Figure 4.53. These routes service the same customers but customer in the corresponding routes differ. However, the PSO produces solution which have multiple routes which service different sets of customers with each dispatched vehicle as can be compared in Figure 4.54 and Figure 4.55. In contrast to the solutions produced by the PSO algorithm, the GA route solutions bear resemblance to each other. The time window duration of the customers in R106 are given in Figure 4.57.

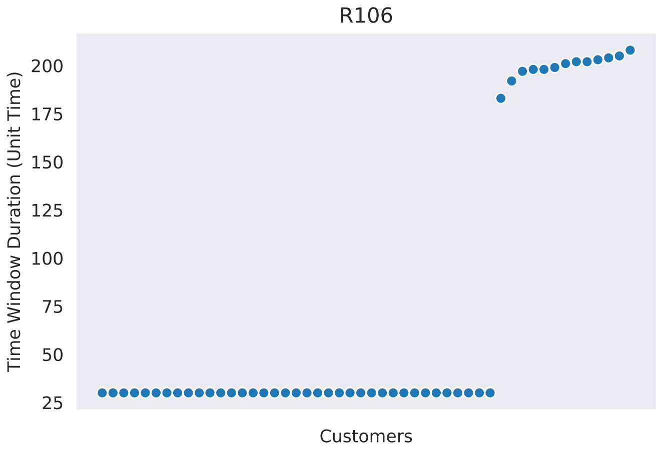


FIGURE 4.57: Time Window Duration of Customers in R106

As shown in Figure 4.57, R106 has customers which have very tight time windows. However, the few with wide time windows allow for variability in the route construction. It is significant to note that since the majority of the customers in this problem instance have tight time windows, the variability in the solutions procured by the application of the PSO algorithm highlights the influence of the stochastic nature of the PSO algorithm in obtaining solutions. The total wait time incurred using the GA is less than that incurred by applying the PSO algorithm. Furthermore, a lesser total distance is travelled by the dispatched vehicles in the solutions produced using the GA.

Temporal Class 2

- From the results obtained to the solutions to the problem instance in R2, the following are noted:
  - The total number of dispatched vehicles of the obtained solutions are less than the benchmark value. The solutions obtained using the GA dispatch an equal number of vehicles to the PSO benchmark value.
  - Dispatching fewer vehicles is compensated by the greater total distance having to be travelled by the dispatched vehicles. Although the best recorded does not record the best benchmark value for R207 and R208, hence the percentage of problem instances classified under the stipulated distance deviation classifications exclude these problem instances. As a result of the number of dispatched vehicles being equal to the number of dispatched vehicles recorded as the PSO benchmark, the total travelled distance bears greater resemblance to the PSO benchmark distance value. The total travelled distance of the solutions obtained by applying the PSO algorithm is greater than that of the GA solutions, hence these PSO solutions have a greater distance deviation than that calculated for the GA solutions, comparative to the PSO benchmark distance.

Random Clustered

The randomly clustered spatially organised customers results are recorded in Table 4.16 and Table 4.17.

Temporal Class 1

- From the results obtained to the problem instances in RC1, the following are noted:
  - The number of dispatched vehicles in the recorded solutions are equal to both the benchmark value and the PSO benchmark value for all the problem instances in RC1.
  - The total travelled distance percentage deviation is greater than 0%.

Temporal Class 2

- From the results obtained to the solutions to the problem instance in RC2, the following are noted:
  - The number of dispatched vehicles of the obtained solutions are less than the benchmark value for all the problem instances in RC2, and are equal to the PSO benchmark value.
  - To compensate for the fewer dispatched vehicles, a greater total travelled distance is obtained for the produced solutions. Since the number of dispatched vehicles are equal to the PSO benchmark result, the obtained total travelled distances bear resemblance to the PSO distance benchmark values. The GA solutions' total travelled distance is less than that of the PSO solutions, hence the PSO distance has a greater variation to the PSO benchmark distance value.
- The problem instance RC207 is highlighted as although dispatches two less vehicles than the benchmark recorded value, the total distance percentage deviation is the highest in RC2 irrespective of the applied solution techniques. The solution route topologies of RC207 are shown Figure 4.62.

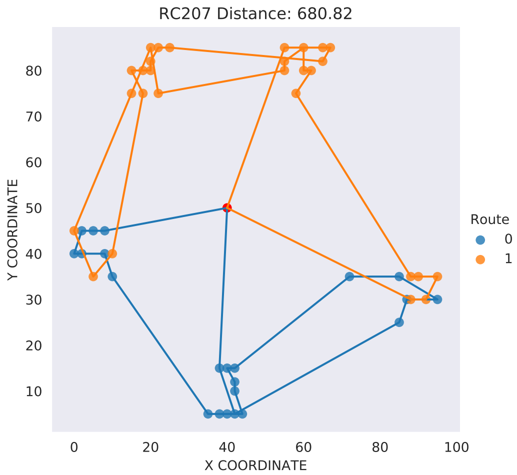


FIGURE 4.58: Topology of a Solution to RC207 Obtained Using GA with Metric A

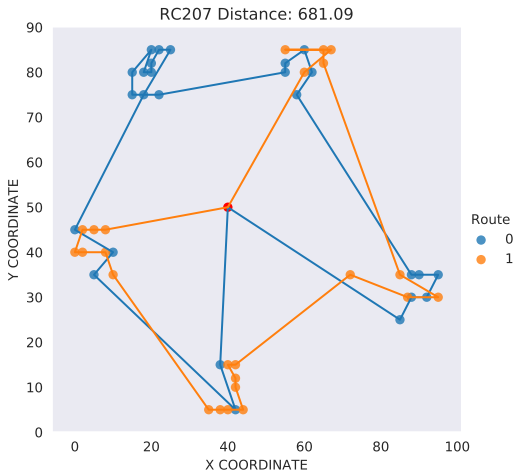


FIGURE 4.59: Topology of a Solution to RC207 Obtained Using GA with Metric B

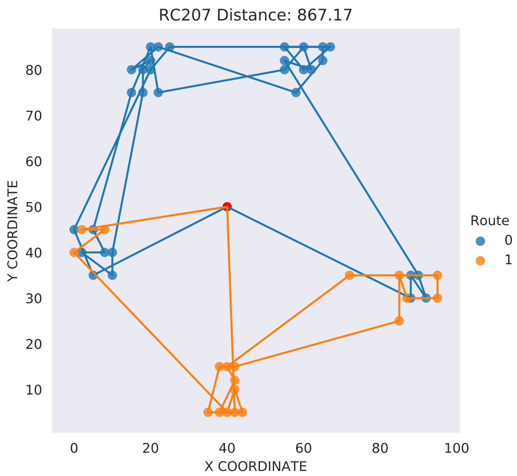


FIGURE 4.60: Topology of a Solution to RC207 Obtained Using PSO with Metric A

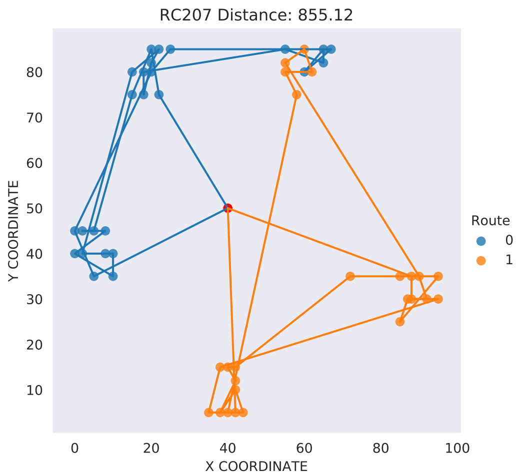


FIGURE 4.61: Topology of a Solution to RC207 Obtained Using PSO with Metric B

FIGURE 4.62: Topology Plots of RC207

The time window duration of the customers in RC207 are given in Figure 4.63.

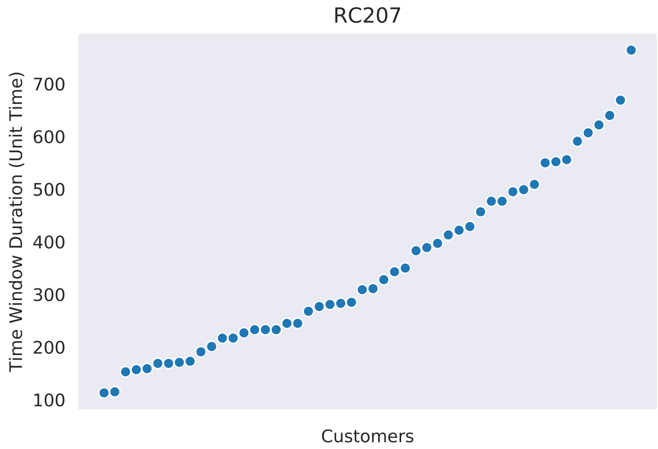


FIGURE 4.63: Time Window Duration of Customers in RC207

From Figure 4.62 the customers seem to be arranged in a semi-clustered manner, however the routes are repeatedly joined between these clusters as a means to avoid dispatching more vehicles. Considering the time window durations of each of the customers in RC207 shown in Figure 4.63 it is noted that the customers have varying time window durations between 100 and 800 time units, which allows for flexibility in the placement of customers into service routes. It is validated that although the customers are spatially arranged at random and

4.4 100 Customers

This section contains and discusses the results obtained to the VRPTW for Solomon's data set containing 100 customers.

	Benchmark Result		Benchmark PSO Result			Benchmark GA Result		GA										PSO									
								Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
C101	**10**	**828.94**	10	828.94	336.00	10	828.94	11	930.37	12.24	1028.23	1024.06	11	927.68	11.91	944.28	1011.28	11	959.00	15.69	742.36	860.30	11	944.02	13.88	842.78	667.76
C102	**10**	**828.94**	10	829.71	581.40	10	828.94	**10**	926.31	11.75	319.51	1004.11	**10**	906.08	9.31	330.76	1019.17	**10**	1210.18	45.99	753.27	1561.15	**10**	1166.09	40.67	683.50	775.53
C103	**10**	**828.06**	10	851.37	1726.80	10	828.06	**10**	912.61	10.21	463.70	959.89	**10**	909.67	9.86	523.87	955.11	**10**	1298.76	56.84	1390.06	2844.53	**10**	1341.38	61.99	1216.88	1466.82
C104	**10**	**824.78**	10	868.52	2404.20	10	824.78	**10**	886.59	7.49	439.73	958.39	**10**	887.24	7.57	194.71	900.22	**10**	1280.25	55.22	1401.80	4181.38	**10**	1248.05	51.32	1382.47	2881.98
C105	**10**	**828.94**	10	828.94	364.80	10	828.94	**10**	919.39	10.91	174.67	951.38	**10**	936.96	13.03	68.01	937.63	**10**	931.86	12.42	265.53	1373.70	**10**	889.45	7.30	95.31	930.51
C106	**10**	**828.94**	10	828.94	379.20	10	828.94	**10**	936.12	12.93	99.91	922.91	**10**	940.85	13.50	99.30	939.83	**10**	1084.50	30.83	93.70	1684.14	**10**	1167.00	40.78	116.10	777.24
C107	**10**	**828.94**	10	828.94	417.00	10	828.94	**10**	920.11	11.00	32.97	941.12	**10**	912.87	10.13	141.05	936.56	**10**	850.31	2.58	0.00	1313.42	**10**	853.80	3.00	80.77	720.17
C108	**10**	**828.94**	10	828.94	759.60	10	828.94	**10**	924.30	11.50	212.37	917.57	**10**	912.06	10.03	143.87	902.92	**10**	1124.76	35.69	193.39	2214.42	**10**	1143.57	37.96	0.00	935.97
C109	**10**	**828.94**	10	828.94	934.80	10	828.94	**10**	910.78	9.87	6.17	911.80	**10**	913.20	10.16	30.24	943.35	**10**	951.27	14.76	102.69	2131.23	**10**	954.95	15.20	285.80	1297.14
C201	**3**	**591.56**	3	591.56	377.40	3	591.56	**3**	591.56	0.00	0.00	2189.51	**3**	591.56	0.00	0.00	2267.63	**3**	591.56	0.00	0.00	1107.02	**3**	591.56	0.00	0.00	522.82
C202	**3**	**591.56**	3	591.56	432.60	3	591.56	**3**	593.18	0.27	9.77	2063.20	**3**	593.18	0.27	9.77	2177.08	4	893.83	51.10	2759.13	1898.03	4	940.65	59.01	3360.93	681.75
C203	**3**	**591.17**	3	591.17	845.40	3	591.17	**3**	613.34	3.75	33.53	2201.33	**3**	616.00	4.20	32.20	2227.07	4	1182.33	100.00	2232.93	2394.11	4	1109.68	87.71	2931.38	898.02
C204	**3**	**590.60**	3	615.43	2285.40	3	596.55	**3**	623.31	5.54	24.42	2451.07	**3**	622.83	5.46	113.57	2315.33	4	1211.45	105.12	2769.55	2914.74	4	1146.37	94.10	3173.03	2226.50
C205	**3**	**588.88**	3	588.88	355.20	3	588.88	**3**	606.32	2.96	0.00	1997.23	**3**	642.44	9.09	12.10	1988.86	**3**	591.56	0.45	0.00	1200.60	**3**	591.56	0.45	0.00	629.81
C206	**3**	**588.49**	3	588.88	385.80	3	588.49	**3**	624.72	6.16	41.98	2026.92	**3**	618.78	5.15	41.98	2017.11	**3**	677.74	15.17	40.70	1510.41	**3**	671.12	14.04	87.65	755.34
C207	**3**	**588.29**	3	591.35	385.80	3	588.29	**3**	615.38	4.60	41.98	2049.99	**3**	634.17	7.80	72.12	2056.18	**3**	680.30	15.64	0.26	1272.01	**3**	702.03	19.33	0.00	713.85
C208	**3**	**588.32**	3	588.49	382.80	3	588.32	**3**	633.73	7.72	45.91	2113.29	**3**	600.46	2.06	50.96	2018.27	**3**	606.28	3.05	0.00	1308.42	**3**	604.11	2.68	0.00	841.92

TABLE 4.21: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Random Clustered Problem Instances containing 50 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark		PSO Benchmark		GA Benchmark		GA				PSO			
	Fitness		Fitness		Fitness		Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
C101	**1000.83**	**11.00**	1000.83	11.00	1000.83	11.00	1100.93	10.00	12.00	0.10	1100.96	10.00	12.00	0.10
C102	**1000.83**	**11.00**	1000.83	11.00	1000.83	11.00	1000.93	0.01	11.00	0.00	1001.21	0.04	11.00	0.00
C103	**1000.83**	**11.00**	1000.85	11.00	1000.83	11.00	1000.91	0.01	11.00	0.00	1001.30	0.05	11.00	0.00
C104	**1000.82**	**11.00**	1000.87	11.00	1000.82	11.00	1000.89	0.01	11.00	0.00	1001.28	0.05	11.00	0.00
C105	**1000.83**	**11.00**	1000.83	11.00	1000.83	11.00	1000.92	0.01	11.00	0.00	1000.93	0.01	11.00	0.00
C106	**1000.83**	**11.00**	1000.83	11.00	1000.83	11.00	1000.94	0.01	11.00	0.00	1001.08	0.03	11.00	0.00
C107	**1000.83**	**11.00**	1000.83	11.00	1000.83	11.00	1000.92	0.01	11.00	0.00	1000.85	0.00	11.00	0.00
C108	**1000.83**	**11.00**	1000.83	11.00	1000.83	11.00	1000.92	0.01	11.00	0.00	1001.12	0.03	11.00	0.00
C109	**1000.83**	**11.00**	1000.83	11.00	1000.83	11.00	1000.91	0.01	11.00	0.00	1000.95	0.01	11.00	0.00
C201	**300.59**	**4.00**	300.59	4.00	300.59	4.00	300.59	-0.00	4.00	-0.00	300.59	-0.00	4.00	-0.00
C202	**300.59**	**4.00**	300.59	4.00	300.59	4.00	300.59	0.00	4.00	0.00	400.89	33.37	5.00	0.33
C203	**300.59**	**4.00**	300.59	4.00	300.59	4.00	300.61	0.01	4.00	0.00	401.18	33.46	5.00	0.33
C204	**300.59**	**4.00**	300.62	4.00	300.60	4.00	300.62	0.01	4.00	0.00	401.21	33.47	5.00	0.33
C205	**300.59**	**4.00**	300.59	4.00	300.59	4.00	300.61	0.01	4.00	0.00	300.59	0.00	4.00	0.00
C206	**300.59**	**4.00**	300.59	4.00	300.59	4.00	300.62	0.01	4.00	0.00	300.68	0.03	4.00	0.00
C207	**300.59**	**4.00**	300.59	4.00	300.59	4.00	300.62	0.01	4.00	0.00	300.68	0.03	4.00	0.00
C208	**300.59**	**4.00**	300.59	4.00	300.59	4.00	300.63	0.02	4.00	0.00	300.61	0.01	4.00	0.00

TABLE 4.22: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Clustered Problem Instances containing 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Result		Benchmark PSO Result			Benchmark GA Result		GA										PSO									
								Metric A					Metric B					Metric A					Metric B				
Dataset	NV	DIST	NV	DIST	CPU Time	NV	DIST	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
R101	**19**	**1645.79**	19	1652.00	439.80	19	1685.27	**19**	1721.76	4.62	856.62	1127.63	**19**	1713.01	4.08	883.98	1140.48	**19**	1739.67	5.70	798.05	1722.98	**19**	1763.33	7.14	722.35	854.20
R102	**17**	**1486.12**	17	1500.81	882.60	18	1523.10	**17**	1539.62	3.60	785.23	1067.25	**17**	1545.89	4.02	676.35	1092.24	**17**	1662.18	11.85	792.67	1590.59	**17**	1694.07	13.99	780.98	846.80
R103	**13**	**1292.68**	14	1242.65	1656.00	13	1348.28	14	1275.26	-1.35	519.60	959.41	14	1275.26	-1.35	621.89	1012.78	14	1448.25	12.03	706.21	2463.74	14	1420.18	9.86	634.20	1377.38
R104	**9**	**1007.24**	10	1042.22	1935.00	10	1010.36	10	1025.26	1.79	196.13	974.49	10	1045.36	3.78	136.25	968.14	11	1248.59	23.96	245.70	3295.61	12	1275.48	26.63	414.75	1506.61
R105	**14**	**1377.11**	14	1385.08	1056.60	15	1427.72	**14**	1476.85	7.24	250.84	1090.04	**14**	1454.23	5.60	270.07	1084.94	15	1555.40	12.95	270.09	2130.83	**14**	1585.00	15.10	202.64	932.69
R106	**12**	**1251.98**	12	1294.87	1366.20	12	1273.62	**12**	1368.12	9.28	135.97	1055.20	**12**	1361.65	8.76	127.69	1068.91	13	1403.48	12.10	382.74	2031.70	13	1403.04	12.07	380.28	825.62
R107	**10**	**1104.66**	11	1123.98	1696.20	11	1100.97	11	1110.47	0.53	253.78	984.54	11	1103.92	-0.07	185.45	976.53	12	1311.46	18.72	373.95	2954.09	12	1286.58	16.47	395.63	1352.73
R108	**9**	**960.88**	10	1011.68	1832.40	10	960.26	10	988.20	2.84	145.44	1000.04	10	986.55	2.67	51.41	972.17	11	1220.80	27.05	242.68	3626.17	11	1249.27	30.01	195.78	2715.05
R109	**11**	**1194.73**	12	1211.63	2509.80	12	1211.81	12	1200.72	0.50	161.10	1032.79	12	1223.03	2.37	156.15	1020.35	12	1432.67	19.92	110.10	2585.31	12	1393.57	16.64	89.59	2607.00
R110	**10**	**1118.59**	11	1190.36	2196.60	11	1146.11	12	1145.28	2.39	213.93	1006.34	12	1145.37	2.39	198.92	1009.97	12	1377.12	23.11	272.02	2893.09	12	1319.71	17.98	254.81	1861.70
R111	**10**	**1096.72**	11	1102.99	1755.60	11	1132.51	11	1115.38	1.70	160.64	993.32	11	1105.44	0.80	224.63	976.43	12	1231.45	12.28	405.92	2445.36	12	1281.15	16.82	351.54	1599.87
R112	**9**	**982.14**	10	1029.12	2464.80	10	985.99	10	1012.72	3.11	108.39	993.17	10	1008.56	2.69	133.36	986.96	11	1228.03	25.04	137.71	4064.15	11	1150.44	17.14	199.98	1506.78
R201	**4**	**1252.37**	4	1274.97	1457.40	4	1276.20	**4**	1352.70	8.01	1022.88	1479.98	**4**	1303.47	4.08	1237.39	1456.58	**4**	1594.37	27.31	797.53	1914.43	**4**	1607.48	28.36	516.62	1036.66
R202	**3**	**1191.70**	3	1247.03	2052.00	4	1087.52	4	1136.62	-4.62	1458.51	1551.34	4	1122.83	-5.78	1455.21	1715.61	4	1465.40	22.97	1423.11	1678.48	4	1472.96	23.60	1359.15	831.72
R203	**3**	**939.54**	3	1052.71	2031.00	3	952.52	**3**	992.23	5.61	690.55	2314.34	**3**	977.22	4.01	719.02	2362.56	4	1371.59	45.99	1349.18	1979.74	**3**	1410.99	50.18	568.40	1869.20
R204	**2**	**825.52**	3	844.16	2337.60	3	766.92	3	795.21	-3.67	829.27	2907.26	3	801.23	-2.94	728.11	2919.96	3	1164.14	41.02	760.05	3216.05	3	1160.52	40.58	733.79	2609.64
R205	**3**	**994.42**	3	1061.46	2125.20	3	1036.08	**3**	1084.08	9.02	508.41	1994.39	**3**	1091.33	9.75	475.70	2046.70	**3**	1559.73	56.85	208.08	2104.31	**3**	1510.17	51.86	163.36	2195.88
R206	**3**	**906.14**	3	1016.35	2702.40	3	921.32	**3**	964.88	6.48	533.86	2372.65	**3**	931.25	2.77	590.14	2404.48	**3**	1401.00	54.61	495.26	2027.31	**3**	1414.11	56.06	496.24	1226.46
R207	**2**	**893.33**	3	946.78	2436.60	3	821.32	3	854.23	-4.38	649.36	2685.04	3	851.71	-4.66	648.78	2674.64	3	1336.47	49.60	582.40	3000.09	3	1322.51	48.04	526.90	2201.57
R208	**2**	**726.75**	2	834.72	2563.20	3	738.41	**2**	929.01	27.83	7.11	3284.41	**2**	830.02	14.21	109.04	3711.91	3	1082.70	48.98	779.44	4502.70	3	1106.69	52.28	685.45	2988.12
R209	**3**	**909.16**	3	1003.19	2502.60	3	928.93	**3**	1034.80	13.82	370.75	2127.51	**3**	966.36	6.29	498.62	2147.71	4	1404.97	54.53	717.12	2294.01	4	1430.55	57.35	811.61	1611.41
R210	**3**	**939.34**	3	1040.54	2619.60	3	983.77	**3**	1009.24	7.44	651.77	2225.72	**3**	1017.51	8.32	628.72	2270.17	4	1494.16	59.07	1135.67	2348.60	**3**	1567.02	66.82	354.53	1858.02
R211	**2**	**892.17**	3	861.32	2724.60	2	786.23	3	845.41	-5.24	491.80	2776.27	3	842.48	-5.57	480.57	2810.14	3	1370.76	53.64	162.61	3687.79	3	1298.57	45.55	242.06	3006.12

TABLE 4.23: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Random Problem Instances containing 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark		PSO Benchmark		GA Benchmark		GA				PSO			
	Fitness		Fitness		Fitness		Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
R101	**1901.65**	**20.00**	1901.65	20.00	1901.69	20.00	1901.72	0.00	20.00	0.00	1901.74	0.00	20.00	0.00
R102	**1701.49**	**18.00**	1701.50	18.00	1801.52	19.00	1701.54	0.00	18.00	0.00	1701.66	0.01	18.00	0.00
R103	**1301.29**	**14.00**	1401.24	15.00	1301.35	14.00	1401.28	7.68	15.00	0.08	1401.45	7.70	15.00	0.08
R104	**901.01**	**10.00**	1001.04	11.00	1001.01	11.00	1001.03	11.10	11.00	0.11	1101.25	22.22	13.00	0.33
R105	**1401.38**	**15.00**	1401.39	15.00	1501.43	16.00	1401.48	0.01	15.00	0.00	1501.56	7.15	15.00	0.00
R106	**1201.25**	**13.00**	1201.29	13.00	1201.27	13.00	1201.37	0.01	13.00	0.00	1301.40	8.34	14.00	0.08
R107	**1001.10**	**11.00**	1101.12	12.00	1101.10	12.00	1101.11	9.99	12.00	0.10	1201.31	20.00	13.00	0.20
R108	**900.96**	**10.00**	1001.01	11.00	1000.96	11.00	1000.99	11.10	11.00	0.11	1101.22	22.23	12.00	0.22
R109	**1101.19**	**12.00**	1201.21	13.00	1201.21	13.00	1201.20	9.08	13.00	0.09	1201.43	9.10	13.00	0.09
R110	**1001.12**	**11.00**	1101.19	12.00	1101.15	12.00	1201.15	19.98	13.00	0.20	1201.38	20.00	13.00	0.20
R111	**1001.10**	**11.00**	1101.10	12.00	1101.13	12.00	1101.12	9.99	12.00	0.10	1201.23	19.99	13.00	0.20
R112	**900.98**	**10.00**	1001.03	11.00	1000.99	11.00	1001.01	11.10	11.00	0.11	1101.23	22.23	12.00	0.22
R201	**401.25**	**5.00**	401.27	5.00	401.28	5.00	401.35	0.03	5.00	0.00	401.59	0.09	5.00	0.00
R202	**301.19**	**4.00**	301.25	4.00	401.09	5.00	401.14	33.18	5.00	0.33	401.47	33.29	5.00	0.33
R203	**300.94**	**4.00**	301.05	4.00	300.95	4.00	300.99	0.02	4.00	0.00	401.37	33.37	4.00	0.00
R204	**200.83**	**3.00**	300.84	4.00	300.77	4.00	300.80	49.78	4.00	0.50	301.16	49.96	4.00	0.50
R205	**300.99**	**4.00**	301.06	4.00	301.04	4.00	301.08	0.03	4.00	0.00	301.56	0.19	4.00	0.00
R206	**300.91**	**4.00**	301.02	4.00	300.92	4.00	300.96	0.02	4.00	0.00	301.40	0.16	4.00	0.00
R207	**200.89**	**3.00**	300.95	4.00	300.82	4.00	300.85	49.76	4.00	0.50	301.34	50.00	4.00	0.50
R208	**200.73**	**3.00**	200.83	3.00	300.74	4.00	200.93	0.10	3.00	0.00	301.08	50.00	4.00	0.50
R209	**300.91**	**4.00**	301.00	4.00	300.93	4.00	301.03	0.04	4.00	0.00	401.40	33.40	5.00	0.33
R210	**300.94**	**4.00**	301.04	4.00	300.98	4.00	301.01	0.02	4.00	0.00	401.49	33.41	4.00	0.00
R211	**200.89**	**3.00**	300.86	4.00	200.79	3.00	300.85	49.75	4.00	0.50	301.37	50.02	4.00	0.50

TABLE 4.24: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Clustered Problem Instances containing 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

Dataset	Benchmark Result		Benchmark PSO Result			Benchmark GA Result		GA										PSO									
								Metric A					Metric B					Metric A					Metric B				
	NV	DIST	NV	DIST	CPU Time	NV	DIST	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time	NV	DIST	DD	Wait Time	CPU Time
RC101	**14**	**1696.94**	15	1641.20	765.60	15	1678.86	15	1681.85	-0.89	366.66	1129.76	15	1700.53	0.21	310.76	1129.92	15	1804.18	6.32	283.44	1438.38	15	1818.85	7.18	247.66	857.95
RC102	**12**	**1554.75**	13	1510.95	1098.60	13	1536.04	13	1592.70	2.44	255.15	1077.53	14	1534.27	-1.32	377.64	1073.68	14	1682.63	8.23	460.48	1436.27	14	1648.07	6.00	574.05	872.72
RC103	**11**	**1261.67**	11	1294.74	1252.20	12	1309.59	12	1339.81	6.19	236.13	1008.33	**11**	1350.60	7.05	162.16	998.19	12	1510.19	19.70	294.02	2025.17	13	1560.59	23.69	448.67	1201.01
RC104	**10**	**1135.48**	10	1190.54	1617.00	10	1154.18	11	1198.75	5.57	268.25	996.47	11	1216.40	7.13	98.82	991.82	12	1474.07	29.82	314.30	1609.09	11	1453.44	28.00	125.51	1505.44
RC105	**13**	**1629.44**	14	1603.71	1080.00	14	1623.33	14	1621.52	-0.49	316.90	1132.33	14	1668.16	2.38	358.22	1118.81	15	1821.71	11.80	421.01	1696.17	15	1788.10	9.74	457.53	863.85
RC106	**11**	**1424.73**	12	1410.93	1424.40	12	1441.46	13	1469.74	3.16	212.22	1035.07	13	1463.63	2.73	253.02	1039.03	13	1564.99	9.84	144.56	2057.40	13	1557.45	9.32	199.06	918.43
RC107	**11**	**1230.48**	11	1249.80	2259.60	11	1271.59	12	1322.38	7.47	215.46	997.17	12	1309.87	6.45	197.78	672.12	12	1543.68	25.45	183.76	2678.89	12	1543.34	25.43	188.14	2611.08
RC108	**10**	**1139.82**	11	1181.87	2167.20	10	1139.82	11	1225.79	7.54	124.29	979.35	11	1210.37	6.19	148.01	967.31	12	1454.86	27.64	278.75	3484.51	12	1453.39	27.51	212.15	3146.66
RC201	**4**	**1406.91**	4	1423.52	1329.00	4	1438.43	**4**	1504.32	6.92	913.55	1408.01	**4**	1494.38	6.22	947.86	1374.61	5	1813.64	28.91	1019.78	2006.50	**4**	1878.78	33.54	474.40	1056.83
RC202	**3**	**1367.09**	4	1193.59	1726.20	4	1181.99	4	1235.41	-9.63	1222.90	1476.63	4	1244.26	-8.98	1228.02	1561.42	4	1680.75	22.94	1118.83	2383.90	4	1687.17	23.41	1059.49	704.39
RC203	**3**	**1049.62**	3	1123.42	2289.00	3	1078.38	**3**	1149.03	9.47	545.92	2031.67	**3**	1136.86	8.31	530.28	2125.50	4	1511.22	43.98	1284.62	3077.03	4	1540.46	46.76	1186.93	1101.37
RC204	**3**	**798.41**	3	894.12	2056.80	3	810.15	**3**	850.89	6.57	623.04	2751.66	**3**	866.73	8.56	606.12	2321.44	**3**	1225.50	53.49	623.11	3668.09	**3**	1229.75	54.02	628.63	1167.61
RC205	**4**	**1297.19**	4	1321.43	2393.40	4	1334.83	**4**	1372.88	5.84	1025.83	1464.33	**4**	1364.86	5.22	1089.20	1436.44	5	1854.08	42.93	1274.51	1965.77	5	1779.58	37.19	1311.27	904.83
RC206	**3**	**1146.32**	3	1307.90	2538.60	3	1203.70	**3**	1351.38	17.89	259.30	1738.58	**3**	1313.65	14.60	248.78	2098.01	4	1747.34	52.43	276.98	2410.09	4	1695.14	47.88	272.32	1011.85
RC207	**3**	**1061.14**	3	1130.37	2748.00	3	1093.25	4	1087.44	2.48	849.10	1873.82	4	1070.78	0.91	864.29	2197.76	4	1661.53	56.58	688.80	3410.39	4	1575.65	48.49	744.07	1578.63
RC208	**3**	**828.14**	3	958.24	3216.00	3	912.76	4	1087.44	31.31	522.67	1873.82	**3**	917.79	10.83	301.46	3229.72	**3**	1594.90	92.59	186.02	4743.15	**3**	1430.86	72.78	255.85	2027.34

TABLE 4.25: The Number of Dispatched vehicles (NV), the Total Travelled Distance of the Dispatched Vehicles (DIST), and it's percentage (%) deviation to the Benchmark Distance (DD), the Total Wait Time (Unit Time) and CPU Time (Seconds) are recorded results for each of the Random Clustered Problem Instances containing 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	Benchmark Fitness		PSO Benchmark Fitness		GA Benchmark Fitness		GA				PSO			
							Metric A		Metric B		Metric A		Metric B	
Dataset	Metric A	Metric B	Metric A	Metric B	Metric A	Metric B	Fit	FitD	Fit	FitD	Fit	FitD	Fit	FitD
RC101	**1401.70**	**15.00**	1501.64	16.00	1501.68	16.00	1501.68	7.13	16.00	0.07	1501.80	7.14	16.00	0.07
RC102	**1201.55**	**13.00**	1301.51	14.00	1301.54	14.00	1301.59	8.33	15.00	0.17	1401.68	16.66	15.00	0.17
RC103	**1101.26**	**12.00**	1101.29	12.00	1201.31	13.00	1201.34	9.09	12.00	0.00	1201.51	9.10	14.00	0.18
RC104	**1001.14**	**11.00**	1001.19	11.00	1001.15	11.00	1101.20	9.99	12.00	0.10	1201.47	20.01	12.00	0.10
RC105	**1301.63**	**14.00**	1401.60	15.00	1401.62	15.00	1401.62	7.68	15.00	0.08	1501.82	15.38	16.00	0.15
RC106	**1101.42**	**12.00**	1201.41	13.00	1201.44	13.00	1301.47	18.16	14.00	0.18	1301.56	18.17	14.00	0.18
RC107	**1101.23**	**12.00**	1101.25	12.00	1101.27	12.00	1201.32	9.09	13.00	0.09	1201.54	9.11	13.00	0.09
RC108	**1001.14**	**11.00**	1101.18	12.00	1001.14	11.00	1101.23	10.00	12.00	0.10	1201.45	20.01	13.00	0.20
RC201	**401.41**	**5.00**	401.42	5.00	401.44	5.00	401.50	0.02	5.00	0.00	501.81	25.01	5.00	0.00
RC202	**301.37**	**4.00**	401.19	5.00	401.18	5.00	401.24	33.14	5.00	0.33	401.68	33.29	5.00	0.33
RC203	**301.05**	**4.00**	301.12	4.00	301.08	4.00	301.15	0.03	4.00	0.00	401.51	33.37	5.00	0.33
RC204	**300.80**	**4.00**	300.89	4.00	300.81	4.00	300.85	0.02	4.00	0.00	301.23	0.14	4.00	0.00
RC205	**401.30**	**5.00**	401.32	5.00	401.33	5.00	401.37	0.02	5.00	0.00	501.85	25.06	6.00	0.25
RC206	**301.15**	**4.00**	301.31	4.00	301.20	4.00	301.35	0.07	4.00	0.00	401.75	33.41	5.00	0.33
RC207	**301.06**	**4.00**	301.13	4.00	301.09	4.00	401.09	33.22	5.00	0.33	401.66	33.42	5.00	0.33
RC208	**300.83**	**4.00**	300.96	4.00	300.91	4.00	401.09	33.33	4.00	0.00	301.59	0.25	4.00	0.00

TABLE 4.26: Fitness Calculated using Benchmark Results and Metric Schemes are compared to the Fitness (Fit) of the obtained results, and its calculated percentage (%) Fitness Deviation (FitD) for each of the Random Clustered Problem Instances containing 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics.

	GA								PSO							
	Metric A				Metric B				Metric A				Metric B			
Dataset	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10	Equal	< 0	0-10	> 10
C1	0.00	0.00	22.22	77.78	0.00	0.00	33.33	66.67	0.00	0.00	11.11	88.89	0.00	0.00	22.22	77.78
C2	0.00	12.50	87.50	0.00	0.00	12.50	87.50	0.00	0.00	12.50	25.00	62.50	0.00	12.50	25.00	62.50
R1	0.00	8.33	91.67	0.00	0.00	16.67	83.33	0.00	0.00	0.00	8.33	91.67	0.00	0.00	16.67	83.33
R2	0.00	36.36	45.45	18.18	0.00	36.36	54.55	9.09	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00
RC1	0.00	25.00	75.00	0.00	0.00	12.50	87.50	0.00	0.00	0.00	37.50	62.50	0.00	0.00	50.00	50.00
RC2	0.00	12.50	62.50	25.00	0.00	12.50	62.50	25.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00

TABLE 4.27: The percentage (%) of problem instances per class containing 100 Customers classified under the stipulated total travelled distance deviation ranges, for the Applied Solution Techniques and Employed Evaluation Metrics

	GA			PSO		
Dataset	Metric A = Metric B	Metric A < Metric B	Metric A > Metric B	Metric A = Metric B	Metric A < Metric B	Metric A > Metric B
C1	0.00	44.44	55.56	0.00	55.56	44.44
C2	25.00	37.50	37.50	25.00	25.00	50.00
R1	0.00	41.67	58.33	0.00	50.00	50.00
R2	0.00	27.27	72.73	0.00	63.64	36.36
RC1	0.00	50.00	50.00	0.00	25.00	75.00
RC2	0.00	25.00	75.00	0.00	50.00	50.00

TABLE 4.28: The percentage (%) of problem instances containing 100 Customers with the total travelled distance per class classified under the stipulated metric relations, for the Applied Solution Techniques and Employed Evaluation Metrics.

	GA						PSO					
	Metric A			Metric B			Metric A			Metric B		
Dataset	Equal	< 0	> 0	Equal	< 0	> 0	Equal	< 0	> 0	Equal	< 0	> 0
C1	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
C2	0.00	12.50	87.50	0.00	12.50	87.50	0.00	12.50	87.50	0.00	12.50	87.50
R1	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
R2	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
RC1	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
RC2	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00

TABLE 4.29: Summary of Fitness Deviation Classifications of Fitness relative to benchmark fitness.

FIGURE 4.64: Wait Time (Unit Time) for Problem instances per class containing 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics

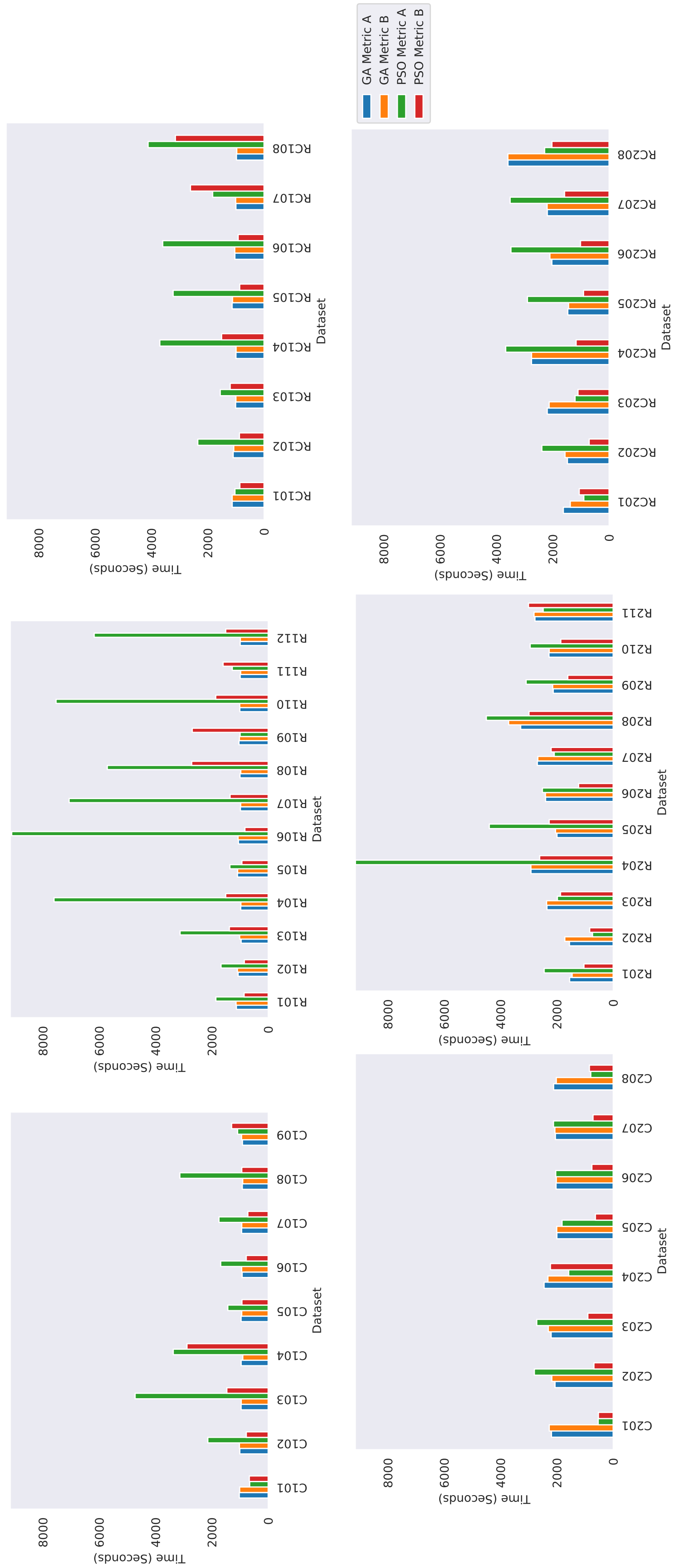


FIGURE 4.65: CPU Time (Seconds) for Problem instances per class containing 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics

4.4.1 Clustered

From the recorded results in Table 4.21 and Table 4.22 to the problem instances with customers spatially clustered, the following are noted for the problem instances in temporal class 1 and 2.

Temporal Class 1

- From the results obtained to the solutions to the problem instances in C1, the following are noted:
  - The number of dispatched vehicles in the obtained solutions correlate to both the recorded benchmark best and PSO benchmark value, except for C101.
  - The corresponding total travelled distance percentage deviation to the benchmark value is greater than 0%, with the majority of the problem instances' solutions with a distance deviation greater than 10% irrespective of the applied solution technique and employed evaluation metric. The total travelled distance for the solutions obtained through the application of the PSO algorithm are greater than that of the GA algorithm.
- The service routes of the produced solutions to the problem instance C103 are illustrated in Figure 4.70.

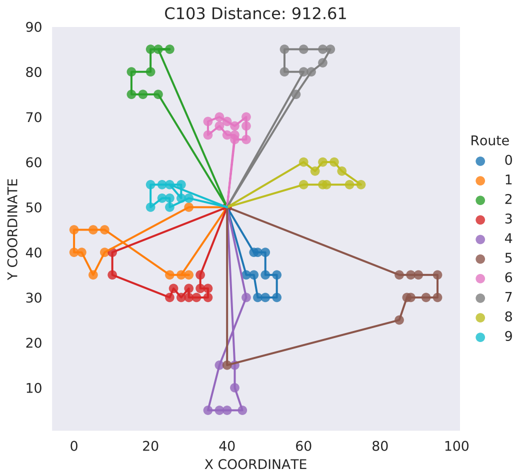


FIGURE 4.66: Topology of a Solution to C103 Obtained Using GA with Metric A

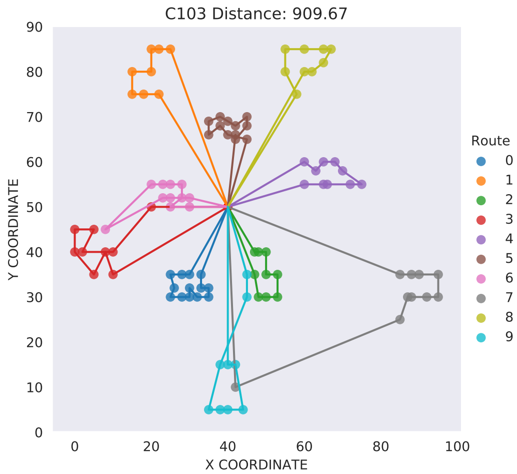


FIGURE 4.67: Topology of a Solution to C103 Obtained Using GA with Metric B

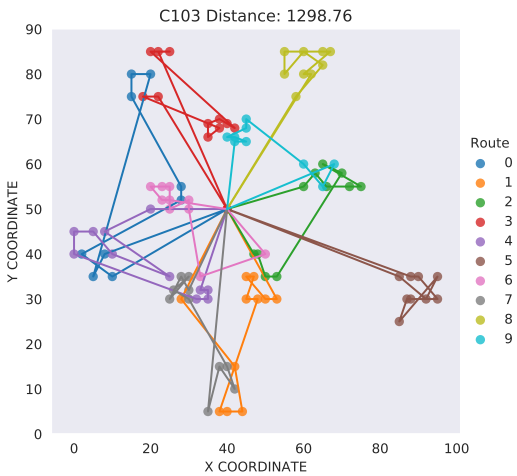


FIGURE 4.68: Topology of a Solution to C103 Obtained Using PSO with Metric A

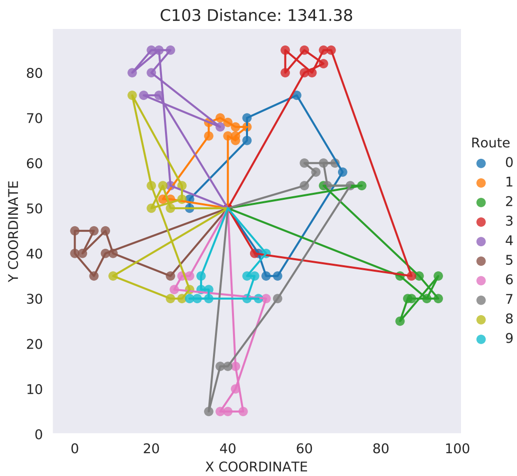


FIGURE 4.69: Topology of a Solution to C103 Obtained Using PSO with Metric B

FIGURE 4.70: Topology Plots of C103

From the resultant routes, Routes 1, 3 and 9 in Figure 4.66 and Routes 0, 3 and 6 in Figure 4.67 service the same customers but with different customer combinations. Hence the difference in the total travelled distance. The resultant routes obtained by applying the PSO algorithm, as shown in Figure 4.68 and Figure 4.69 have multiple differing routes in comparison to the GA solutions. These routes incur a greater total travelled distance as the vehicles service customers in a various spatial cluster arrangements, hence incurring this additional travel distance. Comparing the wait times of these solutions, as recorded in Table 4.21 and shown in Figure 4.64, the wait time is consistent across all applied metrics and solution techniques. Thus, the GA's performance must be highlighted in comparison to the PSO algorithm.

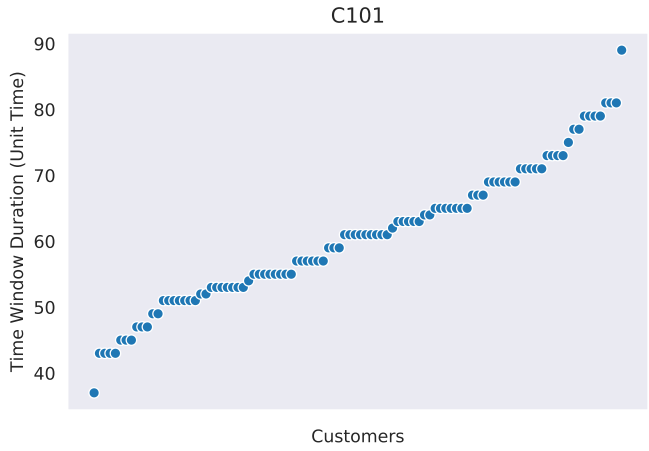


FIGURE 4.71: Time Window Duration of Customers in C101

The time window durations of customers in C103 are shown in Figure 4.71. It is noted that C103 time window durations are short which had indicated less flexibility in the produces solutions for the datasets containing 25 and 50 customers. This is a result of the increase in the size of the dataset, as there is a greater number of customers available to service which induces the flexibility of the placement of customers in the route designs and resulting in the variability in the produced solutions.

Temporal Class 2

- From the results obtained to the solutions to the problem instances in C2, the following are noted:
  - The solutions obtained using the GA dispatch an equal number of service vehicles in comparison to the benchmark values, hence also incurring a totalled travel distance which bears resemblance to the benchmark distance values.
  - The solutions obtained using the PSO algorithm dispatch an equal or greater number of service vehicles in comparison to the benchmark values, and have a resultant totalled travel distance which greater than the benchmark distance values.
- The problem instance C204 is particularly of interest as its solution dispatches an equal number of vehicles to the benchmark value when the GA is applied, but a greater number of vehicles when the PSO algorithm is applied. The solution route topologies obtained using the applied solution techniques and evaluation metrics are shown in Figure 4.76.

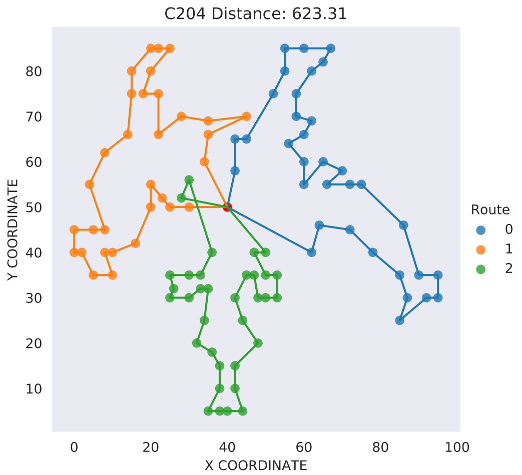


FIGURE 4.72: Topology of a Solution to C204 Obtained Using GA with Metric A

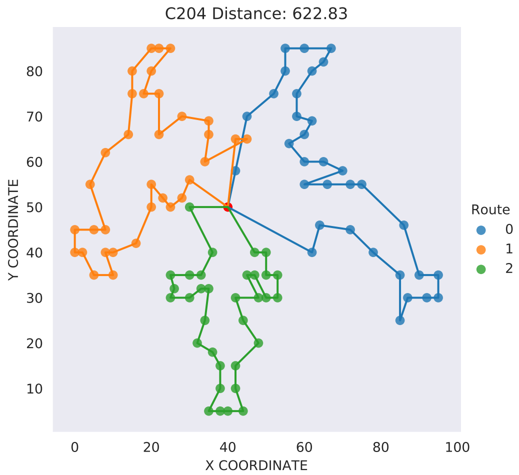


FIGURE 4.73: Topology of a Solution to C204 Obtained Using GA with Metric B

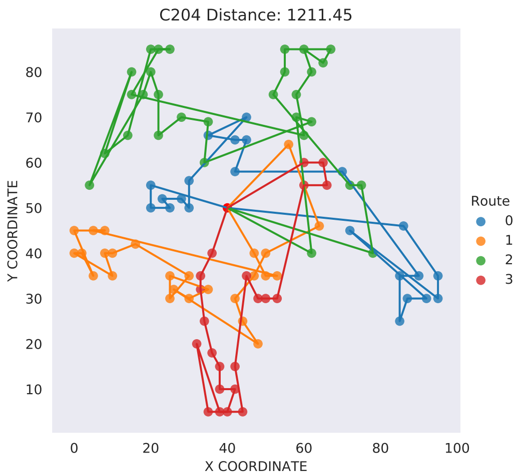


FIGURE 4.74: Topology of a Solution to C204 Obtained Using PSO with Metric A

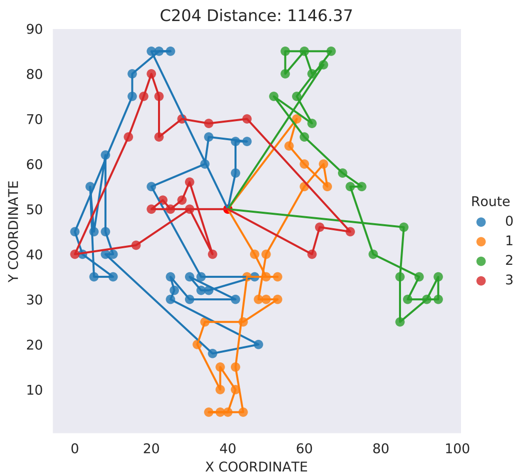


FIGURE 4.75: Topology of a Solution to C204 Obtained Using PSO with Metric B

FIGURE 4.76: Topology Plots of C204

The time window duration for C204 is given in Figure 4.77.

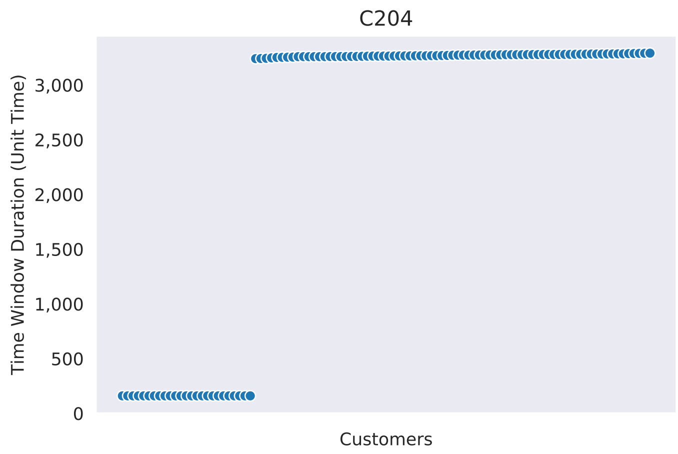


FIGURE 4.77: Time Window Duration of Customers in C204

From the description of Solomon's Benchmarking problem instances, C2 is classified under the *long-haul* temporal class. The variability in the obtained solution to C204 can be attributed to the fact that the majority of the customers in C204 have wide time windows. It is further highlighted that the total wait time incurred when applying the PSO is greater than that incurred when the GA algorithm is applied. The solutions produced using the GA with Metric A and Metric B are respectively shown in Figure 4.72 and Figure 4.73. Comparing these two illustrations, the constructed routes bear resemblance to each other. In contrast to this, the solutions produced using the PSO algorithm with the applied evaluation schemes, shown in Figure 4.74 and Figure 4.75, do not. The reason for the variability allowed in the PSO solution is due to the solution techniques stochastic nature, wide time windows and scaled problem size.

4.4.2 Random

The randomly spatially organised customers results are recorded in Table 4.23 and Table 4.24.

- From the results obtained to the problem instances in R1, the following are noted:
    - The total travelled distance for all the R1 problem instances obtained using the GA are have a lower total travelled distance percentage deviation than that of the solutions obtained using the PSO algorithm.
    - The number of dispatched vehicles are greater than or equal to the best benchmark value, and all the solutions obtained using the GA are equal to the PSO benchmark value. As a result, the corresponding resultant distances bear a closer resemblance to the PSO benchmark distance values.
- The problem instance R103 is of particular interest as its the only problem instance with a solution which dispatches more vehicles than the benchmark value in the R1 class but an equal number to the PSO benchmark value. The resultant total travelled distance obtained by the solution is less than then benchmark value when the GA is applied, however, this is negligible as a

additional vehicle is dispatched. The topology of the produced solutions are shown in Figure 4.82.

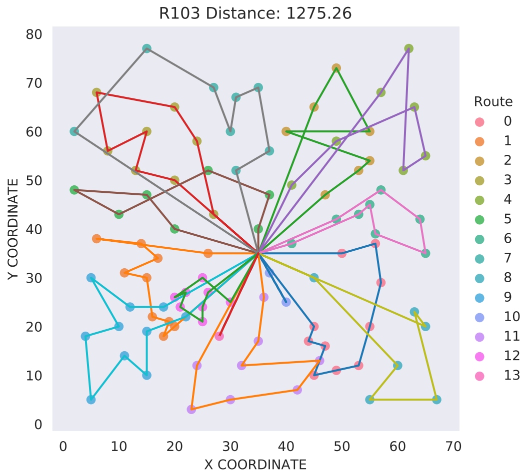


FIGURE 4.78: Topology of a Solution to R103 Obtained Using GA with Metric A

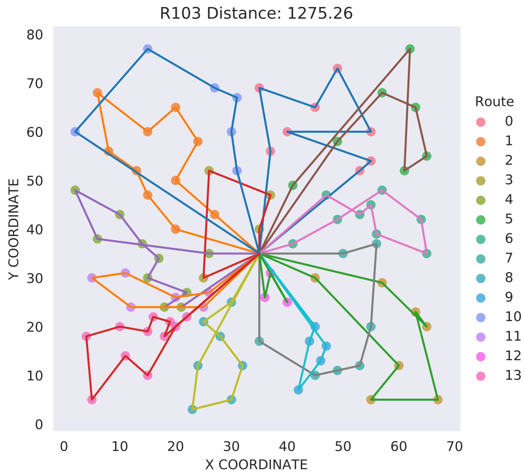


FIGURE 4.79: Topology of a Solution to R103 Obtained Using GA with Metric B

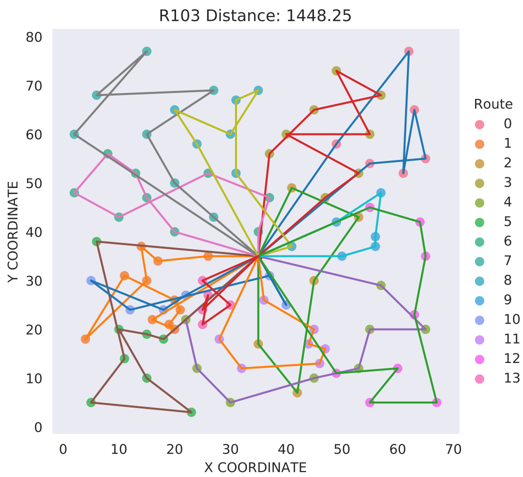


FIGURE 4.80: Topology of a Solution to R103 Obtained Using PSO with Metric A

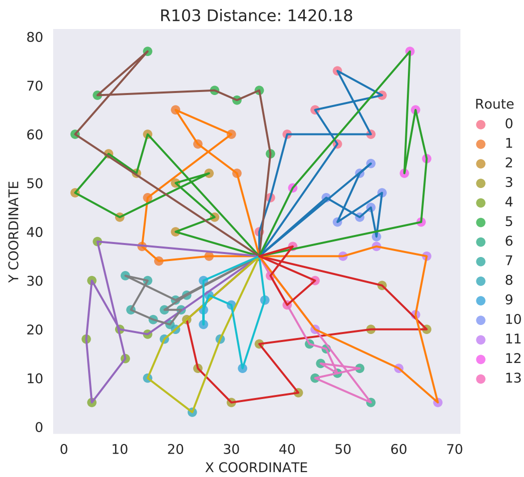


FIGURE 4.81: Topology of a Solution to R103 Obtained Using PSO with Metric B

FIGURE 4.82: Topology Plots of R103

The time window duration of the customers in R103 are given in Figure 4.83.

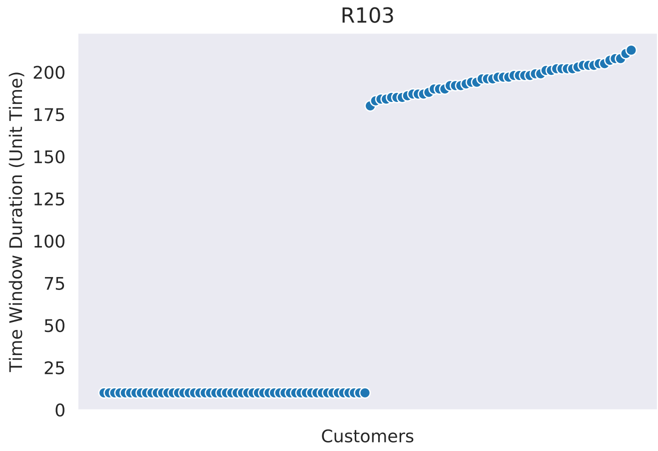


FIGURE 4.83: Time Window Duration of Customers in R103

As shown in Figure 4.83, R103 has half the customers in the dataset with extremely tight time window durations which are less than 25 time units and the half the customers with time windows with durations ranging between 175 and 225 time units. The tight time windows have been seen to govern the consistency of the results, however, the random spatial arrangement and increased dataset size influence the variability in the produced solutions. Although the route constructions differ for the GA solutions shown in Figure 4.78 and Figure 4.79, the total travelled distance is the same, reiterating the robustness and consistency in the GA's performance.

Temporal Class 2

- From the results obtained to the solutions to the problem instance in R2, the following are noted:
  - The total number of dispatched vehicles of the obtained solutions are greater than or equal to the benchmark value, equal to the GA benchmark value and are equal to the PSO benchmark for all instances except for R208 and R2011, when the GA is applied. However, when the PSO algorithm is applied a greater or equal number of vehicles are specified to be dispatched than the benchmark values.
  - Since the number of vehicles dispatched correlate to the benchmark values when the GA is applied, the corresponding total travelled distance too bears resemblance to the recorded benchmark values. Although an equal or greater number of vehicles are dispatched when the PSO algorithm is applied, the corresponding resultant total travelled distance value is greater than the recorded benchmark values.
- The problem instance R203 is of interest as the solutions produced using the two solution techniques dispatch distinct number of vehicles when the PSO algorithm is applied. The solutions obtained by the application of the GA dispatch an equal number of vehicles as the benchmark value, also when the PSO algorithm is applied with Metric B. However, when the PSO algorithm is applied with Metric A one additional vehicle is specified to be dispatched. The topologies of the obtained solutions are given in Figure 4.88.

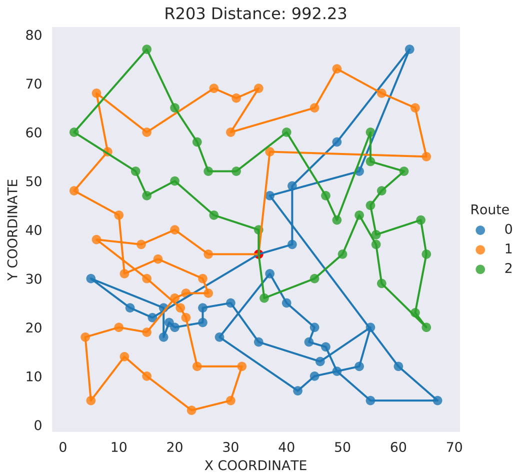


FIGURE 4.84: Topology of a Solution to R203 Obtained Using GA with Metric A

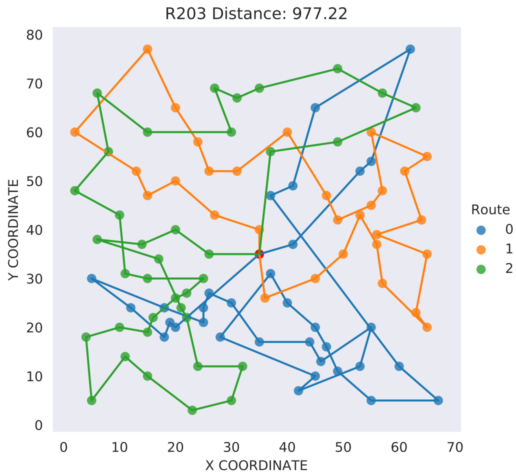


FIGURE 4.85: Topology of a Solution to R203 Obtained Using GA with Metric B

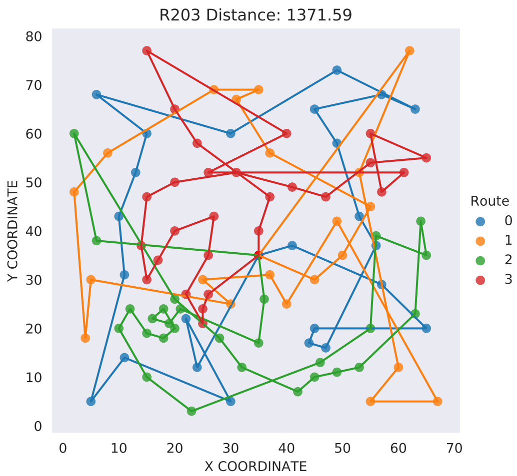


FIGURE 4.86: Topology of a Solution to R203 Obtained Using PSO with Metric A

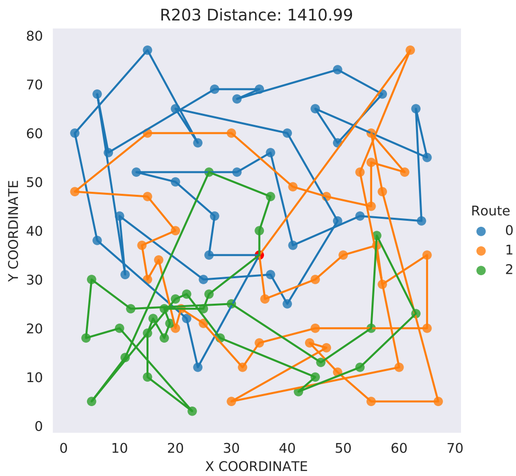


FIGURE 4.87: Topology of a Solution to R203 Obtained Using PSO with Metric B

FIGURE 4.88: Topology Plots of R203

The time window duration of the customers in R203 are given in Figure 4.89.

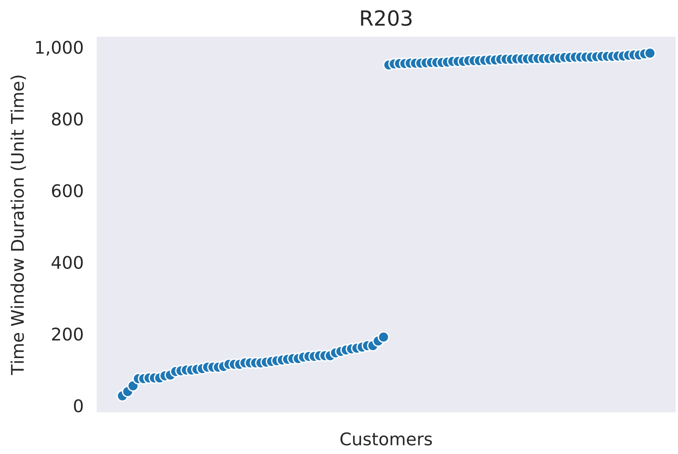


FIGURE 4.89: Time Window Duration of Customers in R203

From Figure 4.88 it is noted that the GA solutions bear resemblance under the two employed metric schemes, in contrast to the PSO solutions. The wide time window durations shown in Figure 4.89, the random spatial arrangement, increased dataset size and the stochastic nature of the PSO algorithm are consequential of the variability in the obtained solutions by the application of the PSO algorithm.

4.4.3 Random Clustered

The randomly clustered spatially organised customers results are recorded in Table 4.25 and Table 4.26.

Temporal Class 1

- From the results obtained to the problem instances in RC1, the following are noted:
    - The number of dispatched vehicles in the recorded solutions are greater than or equal to the benchmark value, and in the instances where the GA or PSO benchmark number of vehicles are greater than the best known benchmark value, then the obtained solutions are equivalent to these values.
    - The total travelled distance percentage deviation is between for the obtained solutions are greater when the PSO algorithm is applied than when the GA algorithm is applied.
- A particular problem instance of interest is RC103. The benchmark number of vehicles is equal to 11 for this problem instance. The obtained solutions resultant number of vehicles is 12 when Metric A is employed with either of the applied solution techniques. The GA benchmark number of vehicles value too is 12. The resultant number of vehicles when the GA is applied with Metric B is equal to the best known and PSO benchmark values. The PSO algorithm employing the solution evaluation metric Metric B specifies to dispatch 13 vehicles, which is greater than any of the benchmark values. The solutions procured by the application of the GA bear resemblance to the GA benchmark distance value. The PSO solution's total travelled distance

percentage deviation is greater than that of the GA's solutions. In Figure 4.94 an illustration of the obtained solutions are given.

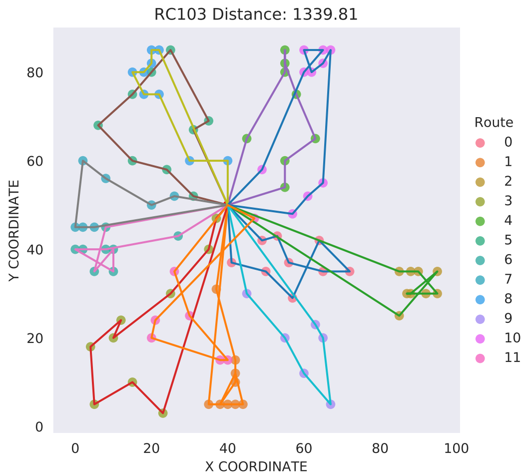


FIGURE 4.90: Topology of a Solution to RC103 Obtained Using GA with Metric A

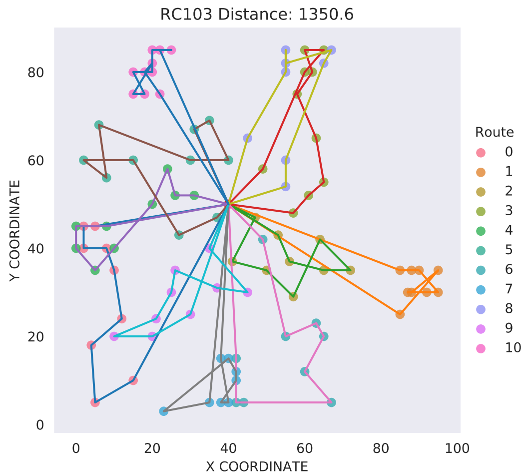


FIGURE 4.91: Topology of a Solution to RC103 Obtained Using GA with Metric B

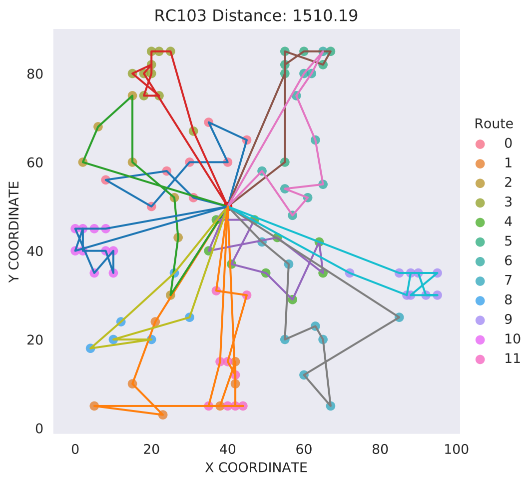


FIGURE 4.92: Topology of a Solution to RC103 Obtained Using PSO with Metric A

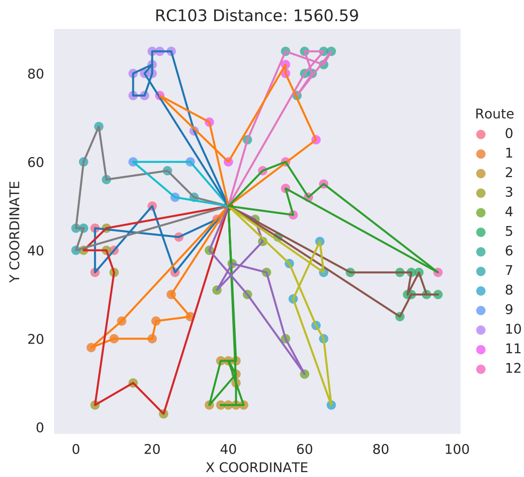


FIGURE 4.93: Topology of a Solution to RC103 Obtained Using PSO with Metric B

FIGURE 4.94: Topology Plots of RC103

The time window duration of the customers in RC103 are given in Figure 4.95.

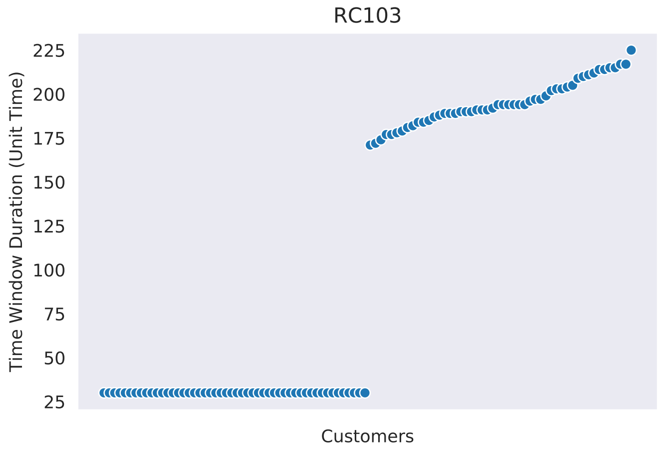


FIGURE 4.95: Time Window Duration of Customers in RC103

Temporal Class 2

- From the results obtained to the solutions to the problem instance in RC2, the following are noted:
  - The number of dispatched vehicles of the obtained solutions are greater than or equal to the benchmark value for all the problem instances in RC2. The resultant number of dispatched vehicles obtained by the application of the GA are equal to the PSO and the PSO and GA benchmark values, except for RC207 and RC208, however, when the PSO algorithm is applied, there are instances where a greater number of dispatched vehicles is obtained.
  - The total travelled distance of the GA solutions bear resemblance to the PSO benchmark distance value. However, the total travelled distance obtained by applying the PSO algorithm is greater than that of the GA, hence has a greater total distance travelled percentage deviation to the benchmark values.

4.5 CPU Time Overview

In Figure 4.96 the CPU times per spatcial and twmporal class are given for 25, 50 and 100 customers. The average CPU times are tabulated in Table 4.30 and visually represented in Figure 4.97 and Figure 4.98.

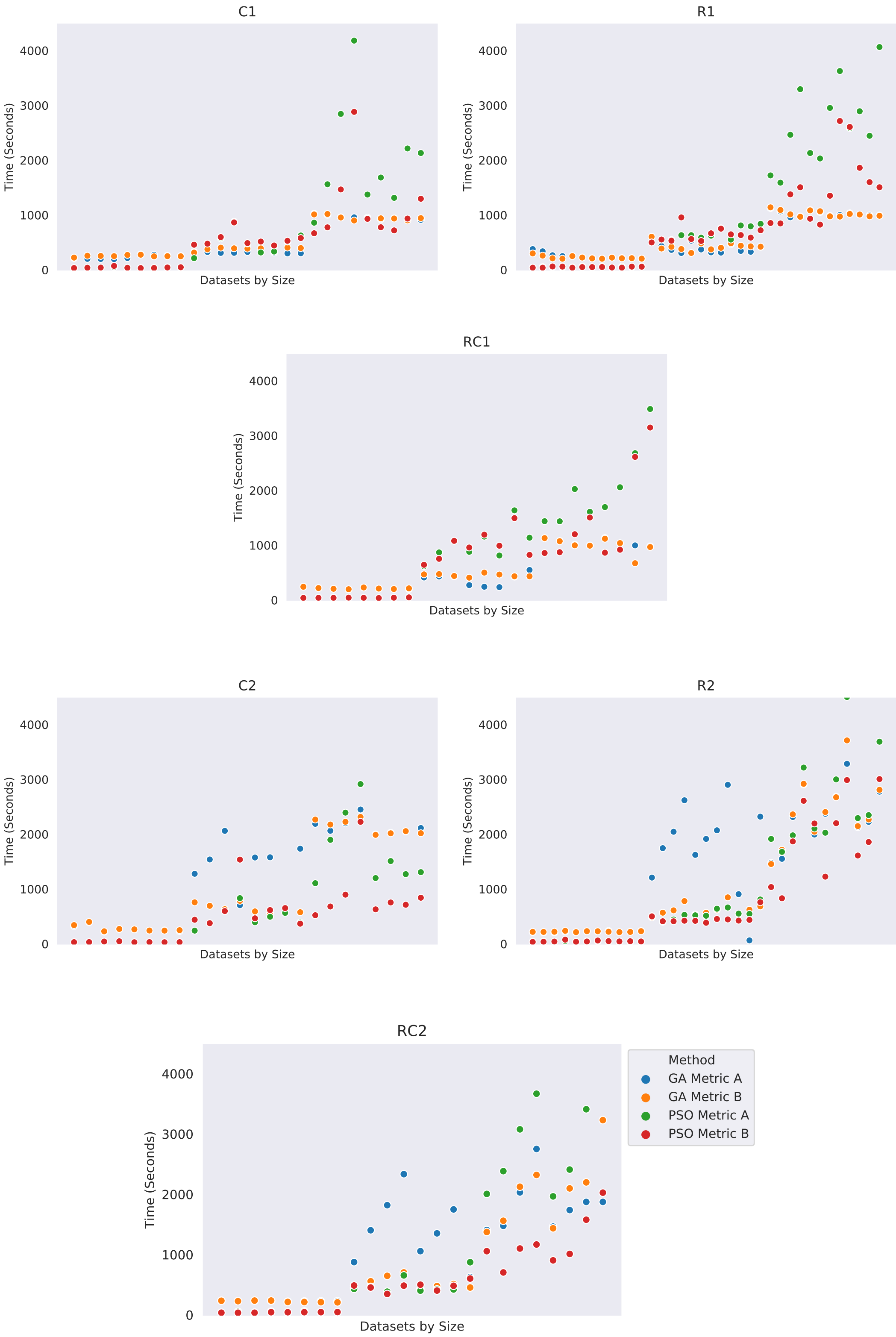


FIGURE 4.96: CPU Time (Seconds) for Problem Instances Per Spatial and Temporal Class Containing 25, 50, 100 Customers, for the Applied Solution Techniques and Employed Evaluation Metrics

	25 Customers				50 Customers				100Customers			
	GA		PSO		GA		PSO		GA		PSO	
Dataset	Metric A	Metric B	Metric A	Metric B	Metric A	Metric B	Metric A	Metric B	Metric A	Metric B	Metric A	Metric B
C1	232.47	253.21	39.7	42.28	315.76	388.54	491.75	550.89	954.58	949.56	2018.25	1161.46
C2	274.66	280.62	34.32	34.72	1380.77	645.75	486.53	632.23	2136.57	2133.44	1700.67	908.75
R1	248.78	226.51	50.71	49.35	393.05	428.26	652.45	636.84	1023.69	1025.82	2650.3	1498.87
R2	220.6	223.54	42.96	47.46	1765.71	625.01	557.44	461.54	2338.08	2410.95	2613.96	1948.62
RC1	213.1	214.91	42.76	41.49	374.94	452.93	1024.11	992.05	1044.5	998.86	2053.24	1497.14
RC2	229.69	223.51	46.51	42.02	1402.53	529.82	507.68	470.76	1827.32	2043.11	2958.12	1194.11

TABLE 4.30: Average CPU Time (Seconds), for the Applied Solution Techniques and Employed Evaluation Metrics

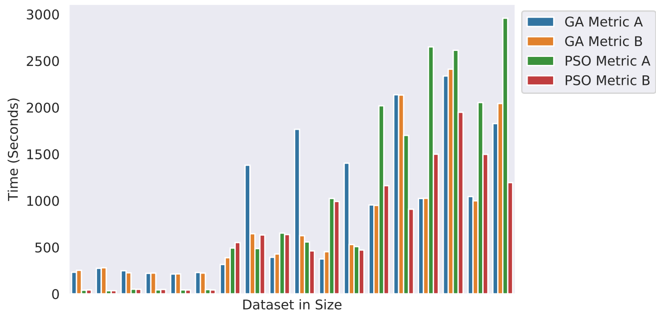


FIGURE 4.97: Average CPU Time (Seconds) for 25, 50 and 100 Customer Datasets for the 6 Dataset Classes, for the Applied Solution Techniques and Solution Evaluation Metrics

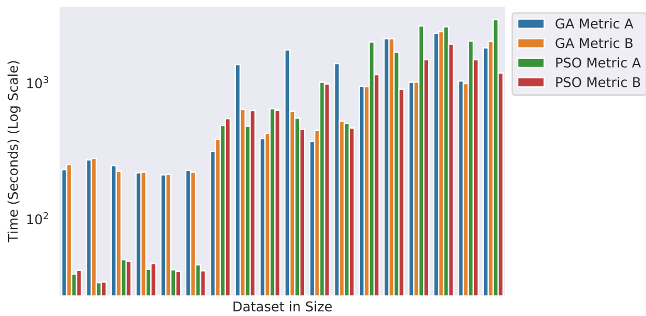


FIGURE 4.98: Average CPU Time (Seconds) on a Logged Scale for 25, 50 and 100 Customer Datasets for the 6 Dataset Classes, for the Applied Solution Techniques and Solution Evaluation Metrics

4.6 Results Overview

Holistically considering the results obtained to Solomon's problem instances of the VRPTW procured by the application of the optimisation solution techniques, the GA and the PSO algorithm, and the employed evaluation metric schemes, Metric A and Metric B, to datasets containing 25, 50 and 100 customers, the following have been observed and summarised from the obtained results.

To analyse the performance of the applied solution techniques, the following were taken into consideration: the impact of the solution evaluation metrics employed with respect to the obtained solutions, the fitness of the obtained solutions and the topological route design of the obtained solutions. The fitness results are respectively summarised and tabulated in Table 4.11, Table 4.20 and Table 4.29, and the metric comparison statistics are given in Table 4.10, Table 4.19 and Table 4.28, for datasets containing 25, 50 and 100 customers. For the results to the datasets containing 25 customers, it is found that the GA produces solutions which are indifferent irrespective to the employed solution evaluation metric. However, as the problem size scales, the results are less consistent but the routes produced by the GA bear resemblance irrespective of the employed metric schemes, unlike the solutions produced using the PSO algorithm. Furthermore, considering the analysis of R103 for a dataset containing 100 customers, it was found that the resultant total distance is the same under the GA despite non-identical route constructions. In general, the results of the obtained solutions through the GA bear closer resemblance to the PSO benchmark values.

One disadvantage of the application of the GA in seeking solutions to the VRPTW is that it has a CPU time which is much greater than that obtained when applying the PSO algorithm. Although the PSO algorithm's performance, in terms of computational time is seen as more favourable, analysing the solutions produced, it is found that in general the obtained solution have a greater resultant total travelled distance, hence a greater distance deviation to the benchmark distance value. The PSO algorithm's solution were also found to be less consistent, as earlier discussed in conjunction to the constructed topological routes.

Questioning which factors contribute to causing variability in the produced solutions, the influence of the customer's time window durations, temporal class, spatial category and the nature of the applied solution techniques were investigated. In general, as the problem size scales, intrinsically the variability in the solutions too increase as a result of the widened search space.

As a naive hypothesis, it was presumed that wider time window durations would allow for flexibility in the constructed service routes, resulting in solutions which display variability. Taking the distance percentage deviation into account, it was found that the solutions were inconsistent between the two applied solution techniques, the GA and the PSO algorithm, rather than in the temporal class with *long-haul* time windows. Closely analysing instances such as R106 (25, 50), C104 (50) and R103 (100) and many other earlier discussed solutions to problem instances belonging to temporal class one, it was found that the solutions produced by the PSO algorithm with either of the employed solution evaluation metrics, in general, did not bear resemblance to each other, irrespective of the time window constraints and spatial arrangement of customers. The design of the applied PSO solution technique substantiates the stochastic nature visible in the produced solutions, as a result of the algorithm making probabilistic selections.

In contrast to the PSO algorithm's design, the solutions produced using the GA are claimed to converge as the solutions produced using either of the employed metric schemes bear resemblance to each other, if not identical. Furthermore, it is important to highlight that the the GA is set to terminate after a fixed number of iterations, unlike the PSO algorithm, which is contradictory to expectations in terms of the convergence of the results to the obtained solutions.

Noting the robustness of the GA, the algorithm is compared on a closer level to the PSO algorithm. In the initial encoding stage of both algorithms, a nearest neighbour method is used as a greedy approach encoding scheme. The GA finds the nearest neighbour in terms of Euclidean distance, whilst the PSO algorithm uses a Nearest Neighbour Heuristic (NNH), which considers both the time window and euclidean distance when selecting the next customer to add to a service routes. The general result of the solutions produced using the GA have a lower total travelled distance than the solutions obtained using the PSO algorithm, could be linked to the initial applied encoding schemes.

Considering the awarded fitness values to the obtained solutions, tabulated in Tables 4.4, 4.6, 4.8, 4.13, 4.15, 4.17, 4.22, 4.24, 4.26, it is found that the metric formulations priorities the number of dispatched vehicles in comparison to the total travelled distance. Although the percentage distance deviation may be greater than 0%, when formulating the fitness value of a solution, the distance deviation is negligible. In instances where the number of dispatched vehicles are fewer than the benchmark value, but have a greater total travelled distance, an overall improved fitness is reflected. It is also significant to highlight that the *arctan* function used in the weighting the total travelled distance in Metric B does not provide to be entirely beneficial as converges to 0.999 as a result of truncation, better solutions may not be selected as an improved distance may be seen as the same after minor transformation.

The wait time is the total accumulated time for which dispatched vehicles stand idle. The wait time was considered in evaluating the quality of a solution. As a result of the NNH, it would be expected that the solutions produced by the PSO algorithm have a reduced wait time. However, this does not consistently occur in the recorded results. As prior discussed for R106 (25). Considering the importance of the wait time in a produced solution, it is important to mention that it would beneficial to factor this feature in employed metric schemes, in a revised metric formulation, to possibly improve the quality of the produced solutions. One significant trend found in terms of the wait time is that the wait time in general is longer for solutions obtained to problem instance in temporal class 2 than temporal class 1. The reason for this is that temporal class 1 has shorter time window durations, thus having to dispatch vehicles such that they consecutively service customers. In contrast, solutions in temporal class 2 incur longer wait times as a result of the wide time window durations which allow for vehicles to travel longer distances to service customers such that fewer vehicles are dispatched and more customers are serviced by each dispatched service vehicle.

Considering the spatial arrangement of the customers in the considered problem instances, as expected, it is found that the clustered problem instances have less variation to the benchmark value and procure solution in less time in comparison to the random and random clustered class. This is visually illustrated in Figures 4.2, 4.34 and 4.65. Furthermore, holistically considering the scale of the CPU time taken in obtaining solution over the three different dataset sizes, it if found that the time scales log normally, as shown in Figure 4.96. A summary of the average times obtained per spatial and temporal class for the applied solution techniques and evaluation metrics are tabulated in Table 4.30. The average CPU times are visually represented in Figure 4.97. From the average CPU times, it can be deduced that the CPU times is consistent irrespective of solution technique and evaluation metric schemes applied for problem instances containing 25 customers. As the problem scales, with the inherent intrinsic difficulty of the problem, the CPU times is more variant. In general, for the 50 customer datasets the CPU time is found to double in relation to the 25 customer datasets, and for the datasets containing 100 customers, the CPU time is approximately 10 times that of the problem instances containing 25 customers. The scaling of the CPU time as the problem size increases is substantiated as in the GA a brute force local search is conducted as shown in Figure 3.7, and the PSO algorithm is not stipulated to terminate after a fixed set of iterations. Comparing the PSO benchmark CPU time to the recoded CPU times corresponding to the procured solutions, the current results do not support the previous times. The PSO benchmark CPU times was recorded for a solution technique coded in C and executed on an Intel Pentium 4 CPU 2.79 GHz with 512 MB RAM. However, the comparison of the CPU time of the GA and PSO can be fairly compared as a result of fair testing conditions.

Critically comparing the implemented solution techniques to the their references, the assumptions made must be highlighted. In replicating the GA for the VRPTW from Ombuki, Ross, and Hanshar (2006b), the paper does not explicitly state the numerical values used in initialising the initial candidate solution. The greedy encoding method, given in Algorithm 3, requires the next customer to be selected to be within some Euclidean empirical in distance. However, the calculation of the range within which the nearest neighbour is to be selected is not specified. Hence, the following calculation has been used:

$$rad = \frac{max - min}{2},$$

where the radius *rad* searched for the nearest neighbour is the difference between the maximum distance between any two vertices in the data set and the minimum distance between any two vertices in the data set, divided by two. This formulation was used as all customers had fallen within this range with respect to each other.

The PSO algorithm for VRPTW applied by Gong, Zhang, Liu, Huang, Chung, and Shi (2012b) does not explicitly state the parameter selection for the implementation, but refers to the parameter values selected by the CLPSO application in Liang et al. (2006b). The selected obtained values are given in Table 4.2.

Chapter 5

Conclusions

Optimising the logistics of the distribution procedure is an important topic of study in the field of Operational Research (OR). A particular, well defined logistics problem is the Vehicle Routing Problem with Time Windows (VRPTW). The VPTW is classified as both a Combinatorial Optimisation Problem (COP) and a Multi-objective Optimisation Problem (MOP). The VRPTW entails having to service a set of customers within their specified time windows, whilst minimising both the number of dispatched service vehicles and total travelled distance of the dispatched vehicles.

In this research, a computational comparative review is conducted of applied optimisation solution techniques and employed solution evaluation formulations to find solutions to the VRPTW. The VRPTW which was initially proposed by Solomon (1984). Surveying the literature on the VRPTW, it was found that both deterministic and non-deterministic optimisation solution techniques have been applied to find solutions to this COP. The performance of the applied solution schemes were directly compared by the obtained results recorded in the literature. Due to the different implementation styles, programming languages and hardware used in conducting the experiments, a fair comparison cannot be claimed. Both deterministic and non-deterministic solution techniques applied to the VRPTW are reviewed in the literature review (Chapter 2). Considering the limitations of the overall performance of deterministic algorithms by Solomon and Desrosiers (1988), and the intrinsic difficulty in solving this $\mathcal{NP}-\mathcal{H}$ problem, Solomon studied heuristic techniques to improve the quality of the constructed routes and because of the potential they hold in solving unstructured realistic size problems.

The two particular metaheuristic optimisation solution techniques studied were the Genetic Algorithm (GA) and the Particle Swarm Optimisation Problem (PSO) which are respectively described by Ombuki, Ross, and Hanshar (2006b) and Gong et al. (2012b). The solution evaluation schemes employed by these solution techniques are given in the respective references and both are weighted average formulations. The problem instances to which the solution techniques and employed evaluation schemes are applied are the Solomon benchmarking problems for the VRPTW.

Summarising the observations of the results recorded in Chapter 4 it is found that as the problem scales, the difficulty in obtaining solutions to the problem instances intrinsically increases. Comparing the performance of the applied solution techniques, it can be claimed that the GA is more robust than the PSO algorithm as the solution procured are consistent irrespective of the employed evaluation metric schemes. The GA is executed for a fixed number of iterations, after which the obtained solutions are found to have converged as they bear resemblance to each other and generally have a total travelled distance within a 5% unit distance range. The PSO algorithm's stochastic nature is reflected in the variability of the solutions obtained by applying the PSO algorithm. Considering the evaluation metric schemes, both metrics are weighted sum formulations which prioritise the number of dispatched vehicles. Since heuristic techniques use fitness schemes in guiding how solutions are iteratively improved, it would be suggested to improve the formulation of the metric schemes to better account for the multiple objectives of the VRPTW.

Considering the Computational Unit (CPU) time taken in procuring solutions, in general, the PSO algorithm procures a solution in a more favourable time than the GA. However, the quicker return of a solution compensates the solution quality. As the problem size scales, the CPU time is found to log normally scale, and found to be less consistent. Comparing the CPU times of fond when using datasets containing 25 customers to those with 50 and 100 customers, it was highlighted that the average time was less consistent as a result of the inherent difficulty of the problem as the problem scales as shown in Figure 4.96 and Figure 4.97. This is attributed to the characteristics of the implemented algorithms; the GA applies a brute force search method and the PSO algorithm's terminating condition.

Comparing the results obtained to the referenced GA and PSO algorithm applied respectively by Ombuki, Ross, and Hanshar (2006b) and Gong, Zhang, Liu, Huang, Chung, and Shi (2012b). It was found that the results obtained bore resemblance to the results obtained by Gong, Zhang, Liu, Huang, Chung, and Shi (2012b) as they too prioritised the objective of the number of vehicles dispatched. As the problem size scaled, the deviation to the benchmark results were found to increase. A comparison of the obtained results to Ombuki, Ross, and Hanshar (2006b) could only be made for the dataset containing 100 customers as results for datasets containing 25 and 50 customers were not recorded.

5.1 Future Work and Recommendations

In attempt to improve the procured solutions, that would inherently mean improving the formulation of the solution evaluation metric scheme as they serve as a guide in updating solutions procured through heuristic techniques. A possible factor to be included in the metric formulation would be the wait time. Wait time is the accumulated time which dispatched vehicles stand idle. Considering that the VRPTW is a MOP, it can be interpreted and represented in multiple different manners based on the application, business needs and prioritisations. Either a bias towards the number of dispatched vehicles or the total travelled distance. Both the employed evaluation schemes priorities reducing the number of dispatched vehicles, as a result of the cost associated with them and the manpower required to drive them. However, more fuel and time may be the compensation as vehicles are travelling longer distances. However, vehicle count can be less important if vehicles to be dispatched were self powered, i.e. if bicycles are being dispatched in support of ecological considerations, the metric scheme would have to account for the cost of manpower, exclude fuel cost, and tailor the costing as per the mode of transport. Thus, it is recommended to design a more versatile evaluation formulation.

Considering that obtained solutions variation to the benchmark values increased as the problem size scales, it would be suggested to increase the number of iterations for which the applied solution technique is applied. Particularly in the application of the GA, as it's performance is validated for smaller sized datasets.

In attempt to gain further insight on the performance of the solution techniques and evaluation metrics, further work would entail applying solution techniques to real sized datasets, use geographical distances, real time traffic updates, and possibly real time time window changes as per the demands and availability of the customers to be serviced.