DOCTOR OF PHILOSOPHY

Agent-based Optimisation Approach for Dynamic Vehicle Routing Problem under Random Breakdowns

Abu Monshar, Anees

Award date:
2023

Awarding institution:
Coventry University

Agent-based Optimisation Approach for Dynamic Vehicle Routing Problem under Random Breakdowns

by

Anees Abu-Monshar

December 2022

Coventry University

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy

Certificate of Ethical Approval

Applicant:	Anees Abu-Monshar
Project Title:	Agent-based Optimisation Approaches for Dynamic Vehicle Routing Problem under Random Breakdowns

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Medium Risk

Date of approval:	29 Apr 2022
Project Reference Number:	P109675

قل رب زدني علما

Acknowledgements

First and foremost, I would like to thank the Almighty who granted me the strength to complete this research journey. Followed by my beloved parents, my mother, Hana Aulama, who never stopped her prayers for me to give my utmost potential, and my father, Mohd Abu-Minshar, who always believed in me and never hesitated to support me financially, believing that science and knowledge are priceless. I'm also grateful to my elder siblings, Hassan, Haneen, Ahmad and Tala, who have always supported me in every step I take, since birth!

Special thanks to Dr. Abdulsattar Al-Alusi, who introduced me to one of his students, my director of studies, Dr. Ammar Al-Bazi, for whom I'm also thankful for his encouragement to start this journey with non-stopping support. Research work passed through three generations. I would also like to thank Dr. Vasile Palade for his help and invaluable input.

I would like to extend my gratitude to our collaborators at Aramex, Raad Kattoura, Rami Haddad and Shaher Abu-Hammour, for their time and effort in providing essential data to further validate this study. This collaboration would not have happened without the efforts of Dr. Rami Al-Hadeethi.

I would also like to thank Dr. Ibrahim Al-Makky for the stimulating discussions and invaluable IT-related input. Also, thanks to Dr. Romali Biswal for her contagious positive research vibe and her encouragement to push forward. My gratitude extends to the rest of the tea-time group: Dr. Muhammad Shamir, Dr. Rohit Kshirsagar, Dr. Mohamed Marei and Dr. Shirine El Zaatari. Special thanks to Ali Saad for his proofreading efforts. This journey would have been bland without my friends: Faris Madi, Mahmoud Al-Alusi and Khaled Abu-Halaweh. Each added a flavour to this journey.

Last but not least, I'm grateful to Maya Amer, my pen pal since the early stages of this work and now she is my wife, for her input in proofing this thesis and who also kept encouraging me to give my utmost potential.

Abstract

Planning vehicles' routes in collection and delivery operations are commonly formulated into the Vehicle Routing Problem (VRP) or one of its variants. However, the fixed plan's assumption is no longer valid given dynamic updates; hence the Dynamic variant (DVRP) is evolved; however, limited to dynamic customer updates. Resource updates are rarely tackled, particularly vehicle breakdowns requiring adequate workload distribution measures across other vehicles. Traditional optimisation approaches are deemed inappropriate due to their inflexibility and time-consuming in producing optimal routes. Therefore, this thesis explores the emerging agent-based optimisation approach for dynamic problems and proposes an agent-based conceptual model to represent, simulate and optimise the problem.

The novelty of this thesis lies in designing the proper agents' interactions to optimise the problem, two of which are proposed based on a different degree of centralisation of the agents' interactions. First, a distributed interaction approach is proposed to construct routes sequentially, dubbed hybrid, given its slight centralisation with priority rules. Unique feasibility evaluations are proposed to address each vehicle agent's unique attributes. The second is a centralised approach that performs extensive global and multiple objective improvements aided by a population-based metaheuristic framework. A distinction is made between problem-dependent and the multi-objective metaheuristic framework components for which novel agent-based problem representation and evaluation are proposed. A Pareto dominance sorting is implemented without prioritising any of the objectives.

For verification and validation, tests on benchmark instances were run, resulting in a reduction of around 5% in vehicles used (hybrid) and 2.20-time units in total waiting times (centralised) at the expense of the total distance travelled. Furthermore, benchmark instances are modified to tackle the breakdown instant problem by randomising locations in addition to capacities and operating shifts and run to justify the applicability of the proposed model. Finally, a case study is adopted to validate the proposed approaches for a multiple breakdown case. A significant reduction in distance of around 68% and 100% elimination of constraint violations have resulted in the static scenario. Furthermore, the disrupted workload can be efficiently re-optimised with minimum deviation from the original static planned routes experimented under three different dynamic breakdown scenarios.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

ACL	Agent Communication Language
ACO	Ant Colony Optimisation
ACS	Ant Colony System
ALNS	Adaptive Large Neighbourhood Search
ANS	Adaptive Neighbourhood Selection
BCRC	Best Cost Route Crossover
BD	Breakdown
CG	Column Generation
CNP	Contract Net Protocol
CPU	Central Processing Unit
CX	Cycle Crossover
DAI	Distributed Artificial Intelligence
DARP	Dial-A-Ride Problem
DP	Dynamic Programming
DVRP	Dynamic Vehicle Routing Problem
DYCOL	Dynamic Column generation
GA	Genetic Algorithm

GVRP	Green Vehicle Routing Problem
HPC	High Performance Computer
ILP	Integer Linear Programming
ILS	Iterated Local Search
INE	Iterative Neighbourhood Exploration
IoT	Internet of Things
KM	Kilometre
KPI	Key Performance Indicator
LP	Linear Programming
MDP	Markovian Decision Process
MDVRP	Multiple Depot Vehicle Routing Problem
MDVRPSDP	Multiple Depot Vehicle Routing Problem with Simultaneous Delivery and Pickups
MDVRPTW	Multiple Depot Vehicle Routing Problem with Time Window
MIP	Mixed Integer Programming
MPHO	Messaging Protocol-Based Heuristic Optimisation
NNH	Nearest Neighbour Heuristic
OR	Operational Research
OX	Order Crossover
PDVRPTW	Pickup and Delivery Vehicle Routing Problem with Time Window
PF	Push Forward
PMX	Partially-Mapped Crossover
PVRPTW	Periodic Vehicle Routing Problem with Time Window
RAM	Random Access Memory

RBX	Route-Based Crossover
RC	Route Crossover
SA	Simulated Annealing
SBX	Sequence-Based Crossover
SDVRP	Site-Dependent Vehicle Routing Problem
SDVRPTW	Site-Dependent Vehicle Routing Problem with Time Window
SNS	Stochastic Neighbourhood Selection
TOP	Team Orienteering Problem
TS	Tabu Search
TSP	Travelling Salesman Problem
UAV	Unmanned Aerial Vehicle
VFA	Value Function Approximation
VNS	Variable Neighbourhood Search
VRP	Vehicle Routing Problem
VRPSDP	Vehicle Routing Problem with Simultaneous Delivery-Pickup
VRPTW	Vehicle Routing Problem with Time Window
VRSP	Vehicle Rescheduling Problem

VRP Notations

$\Delta t_{u\,ij}$	difference in time when customer u is inserted between i and j
b_i	customer i arrival time
b_{j_u}	new arrival time at customer j when customer u is inserted right before it in a route
C	a set of all customers
d_{uh}	distance between customer u and vehicle's home location h

d_{vh}	distance between vehicle v and vehicle's home location h
d_{vu}	distance between vehicle v and customer u
dep_i	departure time from customer i
dep_v	vehicle v departure time from its current location
$dur_{v\ max}$	maximum duration limit for vehicle v
dur_{v_u}	new duration for vehicle v when customer u is inserted
e_i	customer i early time-window
e_v	vehicle v shift time start
i	customer index within a route, 1..C
j	customer index within a route, 1..C
l_i	customer i late time-window
l_u	customer u late time-window
l_v	vehicle v shift time end
q_i	customer i demanded quantity
Q_v	vehicle v total capacity
$Q_{v\ cur}$	vehicle v current occupied capacity
s_i	servicing time for customer i
S_v	total servicing times for vehicle v
T_v	total travel times for vehicle v
t_{ij}	travel time from customer i to customer j
t_{v_i}	travel time from the vehicle v current location to the customer i
u	inserted customer index, 1..C
V	a set of all vehicles
v	vehicle index, 1..V

w_i	waiting time at customer i
W_v	total waiting time for vehicle v
W_{v_u}	total waiting time for vehicle v when customer u is inserted
cur	current attribute

Hybrid Approach Symbols

α	Solomon's parametric weighting
λ	weight parameter of serving a particular customer solo in a route
μ	distance saving parameter
PF_i	push forward in time after insertion at customer i
Far_Avg	customer Furthest Average distance from all vehicles priority rule
Far_Min	customer Furthest Minimum distance from all vehicles priority rule
LTW	Late Time Window customer priority rule

Centralised Approach Symbols

γ	penalties multiply/divide factor
D_{v_r}	total distance for vehicle v given route r is provided
dur_{v_r}	duration for vehicle v given route r is provided
$F_{obj\ V_r}$	a single objective function for a particular solution set V_r
F_{V_r}	the overall objective function for a particular solution set V_r
h	generations/iterations to revisit the penalties
P_{dur}	penalty for duration violations for a particular solution set V_r
P_Q	penalty for capacity violations for a particular solution set V_r
P_{TW}	penalty for time window violations for a particular solution set V_r
P_{V_r}	total penalties for a particular solution set V_r
Q_{v_r}	occupied capacity for vehicle v given route r is provided

r	a route from V_r set
T_{v_r}	total time for vehicle v given route r is provided
V_r	routes set provided for each vehicle v
v_r	route r provided to vehicle v
$Vdur_{V_r}$	total duration violations for a particular solution set V_r
$Vdur_{v_r}$	duration violation for vehicle v given route r is provided
VQ_{V_r}	total capacity violations for a particular solution set V_r
VQ_{v_r}	capacity violation for vehicle v given route r is provided
VTW_{V_r}	total time window violations for a particular solution set V_r
VTW_{v_r}	time window violations for vehicle v given route r is provided
W_{v_r}	total waiting time for vehicle v given route r is provided
y_{v_r}	a binary variable indicating whether vehicle v is idle (0) or utilised (1) given route r is provided

Genetic Algorithm Symbols

g	current generation
Gen	total number of generations
Ind	a specific individual in population
$Ind_{fitness}$	calculated fitness for an individual
LS_{rate}	total number of generations
M_{rate}	total number of generations
$Offspring$	resultant individuals from crossover
P_{Size}	population size
$Parents$	subsequent population of size 2 to apply crossover to
Pop	metaheuristic population

Pop_{ranked}	ranked population
$Pop_{selected}$	selected sub population
$Rank_{current}$	current Pareto rank for the non-dominated set
$Rank_{Ind}$	calculated rank for an individual
$Rank_{max}$	maximum rank in a given population
$Rank_{min}$	minimum rank in a given population
Sel_{size}	selection size from the population
X_{count}	number of times to perform crossover per generation
X_{rate}	Crossover Rate

Breakdown Symbols

λ_{bd}	mean time between breakdown events
bd	breakdown subscript

Outputs Abbreviations / Symbols

C	the Centralised approach
C Rule	Customer Priority Rule
CM	Customers Missed
DM	Demand Missed
H	the Hybrid approach
Org	Original
TD	total Travelled Distance
V	Vehicles used
WT	total Waiting Time

Chapter 1

Introduction

1.1 Chapter Overview

This chapter introduces this thesis. It provides a background, highlights the motivation behind this study and defines the problem statement. Next, the research aim and objectives are stated, followed by the research design, techniques, deliverables and scope, while research benefits and contributions are discussed. Finally, this thesis outline is provided.

1.2 Brief on Last-Mile Distribution

Transportation and logistics are seen as vital business activities to ensure the availability of goods at the right time and place. Last-mile distribution, where goods are required to be delivered to end customers, is seen as an emerging problem, and its importance to logistics management is getting higher. According to Boysen et al. (2021), last-mile delivery is seeing immense development pressure due to market, environmental and economic factors. With the increasing role of e-commerce, the current market significantly raised the demand for such delivery services, which induces increased pollution and road traffic congestion challenges. In addition to these factors, more demand for a more quick delivery makes managing and optimising delivery operations more challenging (Archetti et al., 2021). Optimising delivery operations are mainly problems formulated into routing problems where specific vehicles are scheduled in various fields. For example, in freight transportation, cargo loading and goods delivery, such problems translate to one or more of the specific variants of the well-known logistical problem of Vehicle Routing Problem (VRP) (Zhang et al., 2022).

The area of transportation and logistics is evolving to cope with dynamic markets. Speranza (2018) argues that with the recent technological advances, the transportation industry is facing problems that are trending to be more dynamic. The emerging dynamic problems question the assumption of fixed plans and solutions are adaptive to new information updates. VRPs are not far from this transformational trend, and routing studies evolved into Dynamic VRP (DVRP), with the first case seen in Psaraftis (1980). Furthermore, this dynamic problem has seen increased attention in the recent years (Psaraftis et al., 2016).

The challenge lies in defining, formulating and solving the dynamic problem. Pillac et al. (2013) highlighted the computational complexity in solving the problem at the instant of the dynamic information arrival, given the adoption of the traditional static approaches. To address this complex issue, Kuhn et al. (1994) proposed the use of specific methods from Distributed Artificial Intelligence (DAI) to dynamic transportation problems by arguing their ability to reduce the problem complexity by breaking down a studied problem into agents and solve using the agents' cooperation and interactions.

1.3 Vehicle Routing: from Static to Dynamic

VRP was extended from the Travelling Salesman Problem (TSP) to accommodate additional constraints. The problem was first introduced by Dantzig and Ramser (1959) to provide routing plans for vehicles to visit customers' locations starting and ending at the same depot. VRP is proven to be an NP-hard problem (Lenstra and Kan, 1981). The basic problem was extended later to other variants to accommodate additional constraints. For example, Solomon (1987) solved the problem with Time Window constraint (VRPTW) and Hosny and Mumford (2010) further extended VRPTW to a Pickup and Delivery problem (PDVRPTW). Based on the most recent VRP review (Zhang et al., 2022), additional and various variants have been introduced, including, but not limited to, Green (GVRP), Multiple Depot (MDVRP) and Simultaneous Delivery-Pickup (VRPSDP).

Although VRP problems have been well-explored, it is only the static type of the problem that has been well researched, and the current research trend is shifting towards the online and dynamic problem (Rios et al., 2021). As a consequence, DVRP has emerged and is seen in several studies in the literature; however, it focused mainly on updates to customer orders by issuing new orders or cancelling others, rather than considering disruptions from an operating vehicle (Li et al., 2009b).

1.4 The Emergent Agent-based for Dynamic Optimisation

There are mainly two schools in modelling and solving VRPs. The first uses mathematical modelling to find the exact optimal solutions, while the second utilises approximation algorithms, such as heuristics and metaheuristics, to find near-optimal solutions. Such division is due to the increased computational complexity when this NP-Hard problem size gets larger, thus, favouring the approximation techniques (Laporte, 2009). However, given the practicality of utilising such approximation techniques, the traditional way of modelling is questioned (Mes et al., 2007). Therefore, new modelling and solving techniques are emerging, and the agent-based approach is one of them (Barbati et al., 2012). According to Barbati et al. (2012), the agent-based optimisation approach has seen significant and relatively recent adoption in solving complex optimisation problems not only in routing but also in scheduling, supply chain planning and transportation.

The agent-based approach can be seen earlier in solving static VRP instances utilising agents' messaging in producing feasible routes in a heuristic-like fashion. Example of such work can be seen in the work of Thangiah et al. (2001), Vokřínek et al. (2010) and Kalina and Vokřínek (2012). Later, when VRP evolved to a dynamic problem, more tendency to model the problem in agent-based by further questioning the traditional optimisation approaches and their applicability to dynamic problems due to their inflexibility to adopt changes (Mes et al., 2007). Modelling and optimising DVRP using agent-based is not new and goes back to the 90s is evidenced by the work of Kuhn et al. (1994) and Fischer et al. (1996). Barbucha (2020) demonstrate the most recent study. The critical aspect in utilising this approach in optimisation is the design of the proper agents' messaging (Davidsson et al., 2007).

Therefore, at its core, this thesis is concerned with designing the proper agents' interactions in generating optimal or near-optimal routes.

1.5 Motivation

For any logistical and transportation company, satisfying customers is essential regardless of any sudden technical failures or even road situations/accidents faced during operations. If vehicle disruptions are faced with disrupted customer orders, the damaging consequences can be significant and can directly affect the company's image and order loss. Accommodating such problems is complex and adds pressure on logistic and transportation planners to perform quick solutions with limited quality. As a result, this problem drives the researcher to uniquely model such problems and develop a novel solution approach that adapts to these real-life dynamic problems, especially under dynamically occurring vehicle disruptions. The aim is to maximise customers' coverage while considering other routing costs. The contributions of this research would be significant to knowledge, and the techniques used could inspire other researchers to optimise other problems in transportation and other domains.

1.6 Problem Statement

In the classical Vehicle Routing Problem (VRP), delivery or collection vehicles are routed to certain customer locations, starting and ending at their representative depots. However, operating vehicles could face breakdown(s) at any time and location while in service, hindering their ability to service the remaining workload. As a result, an issue would arise regarding sharing the disrupted vehicle workload with the remaining operating vehicles. Therefore, the other in-service vehicles' routes could be possibly re-routed to reduce the impact of such problems. Figure 1.1 illustrates such a re-routing delivery problem.

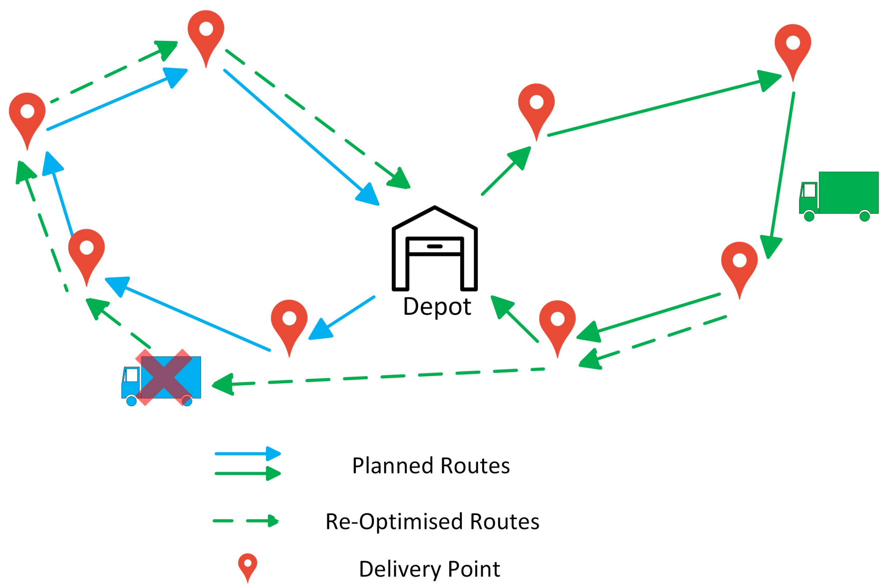


Fig. 1.1 Problem Visualisation

Figure 1.1 shows a set of in-service vehicles' scheduled routes, represented by the solid arrows, while the dotted arrows indicate the rescheduled route. In a delivery case, the problem dictates a visit to the location of the disrupted vehicle to collect the required load, assuming it is always allowed to transfer loads, either parcels or passengers, between the disrupted vehicle and any operating vehicle, fully or partially. The goal is to minimise the number of missed customers, utilised vehicles, total distance travelled and waiting time. The optimisation problem is subject to deliveries within customer time windows and vehicle capacity and duration constraints. Further complications will arise if another breakdown event occurs after producing the new routes.

1.7 Research Aim and Objectives

This thesis aims to develop agent-based optimisation approaches to solve the vehicle routing problem under random vehicle breakdowns in continuous time to reduce the impact of such dynamic disruptions. The following objectives are composed to fulfil the aim of this thesis:

- To review previous and recent literature concerning vehicle routing focusing on the dynamic problem with vehicle breakdown and their adopted solution approaches.
- To identify and comprehend the logic behind the dynamic vehicle routing in the case of both delivery and collection with multiple random vehicle breakdowns that might occur in continuous time.
- To develop an agent-based optimisation architecture embedded with the suitable agent interactions to emerge in optimal/feasible routes. The architecture will also mimic the operations of delivery or collection with the introduction of vehicle disruptions.
- To design and develop an efficient distributed/hybrid agent interaction to perform a quick search to produce near-optimal routes based on predefined rules.
- To design and develop an extensive centralised agent interaction that utilises a meta-heuristic framework and multi-objective non-dominance sorting to perform an in-depth routing.
- To verify, validate and test the efficiency of the proposed optimisation techniques against benchmarked instances.
- To run a case study to compare the output of the optimisation techniques with a real-life scenario.

1.8 Research Design

This research project is mainly an exploratory study that aims to propose and develop an optimisation model and put it to the test. It will mainly utilise quantitative and numerical techniques. Therefore, a single method research design (quantitative) will be adopted. The two research strategies adopted are:

- Experiment: This quantitative approach is considered the project's primary strategy that may include tools such as modelling and experimental design/analysis on the developed model and tests on benchmark instances available from the literature.

- A case study: This is an in-depth insight into the proposed optimisation model in a real-life setting to measure the extent of its applicability in producing optimal vehicle routing decisions at an overseas case company in collaboration with the university. Required and relevant data will be asked for from the company.

1.9 Research Tools and Techniques

The way and manner in which this research is carried out are considered critical aspects of the project. The following research tools and techniques are to be exploited to ensure the effectiveness of the project:

- A literature review that reviews previously faced problems and used tools and techniques within the area of vehicle routing, focusing on the dynamic case of vehicle breakdown.
- An agent-based approach to mimic the dynamic behaviour of vehicle collection and delivery operations over time.
- Appropriate agent interactions will be developed and embedded into a proposed agent-based architecture to find optimal/near-optimal solutions for the problem under study.
- An experimental design technique will test the proposed approaches against benchmark instances and generate different scenarios while comparing outputs with different approaches.
- A case study approach will be adopted to test the applicability of the optimisation approach(es) and compare it to a real-life situation.
- A selection of tools and techniques such as interviews will be utilised to collect the required data.

1.10 Research Deliverables

The following are details of the project deliverables, which are the objectives' intended outcome. The success of the project is determined by achieving these deliverables.

- A comprehensive review that analyses previous problems encountered within the field of dynamic vehicle routing along with the tools and techniques implemented to solve such problems.

- An agent-based architecture mimics the investigated problem and provides routing decisions.
- Agent interaction designs were developed to govern the routing decisions' logic to provide optimal/near-optimal routes.
- A verified and validated model has improved the solution using the proposed approach(es).
- A relevant real-life case study that further validates the developed optimisation approaches in optimising collection and/or delivery operations.

1.11 Research Scope

The scope of this research is limited to a specific transportation problem, VRPTW, under dynamic vehicle breakdowns that may occur randomly during operation. A delivery problem is considered by taking into consideration the disrupted workload pickup, as demonstrated in Figure 1.1. The focus will be on finding a way to reduce the impact of such dynamic disruptions. Maximising the number of served customers that resulted in such disruptions is prioritised. At the same time, other performance indicators are considered, including minimising the number of utilised vehicles, total travelled distance and total waiting time.

1.12 Research Benefits and Contributions

This research benefits the logistics industry in adapting their delivery operations to dynamic vehicle disruption events considering multiple criteria: customer coverage, vehicles used, distance travelled and waiting time. As a result of modelling the breakdown instant problem, this research benefits crowd-shipping applications to match and route supply (vehicles) with demand (customers) given that the supply is individuals with unique attributes (Le et al., 2019).

The academic contributions of this thesis can be summarised as follows:

- Fills in the gap in the existing academic body of knowledge in the area of vehicle routing under dynamic vehicle disruptions (Chapter 2).
- A novel agent-based conceptual model for solving VRPTW with dynamic breakdown along with its sub-variant of VRPTW with unique vehicles that represents the breakdown instant (Chapter 3, section 3.3).

- A novel agent messaging protocol-based heuristics optimisation model, following the hybrid cooperation approach (Chapter 3, section 3.4).
- A novel agent messaging protocol, following the centralised cooperation approach integrated with a new customisable multi-objective metaheuristic framework (Chapter 3, section 3.5).
- A new and generalised way of representing and evaluating VRPTW solutions in agent-based to make it applicable to apply any appropriate metaheuristic framework (Chapter 3, section 3.5.1).
- New best solutions on MDVRPTW benchmarks in terms of minimisation of the number of vehicles (Hybrid) and waiting times (Centralised) (Chapter 4, sections 4.3 & 4.4).
- A new real-life time window data set (Chapter 4, section 4.5, Appendix A).

1.13 Research Dissemination

Based on the research work conducted for this thesis, the following articles have been published:

- Abu-Monshar, A.M., Al-Bazi, A. F., and Alsalami, Q. H. (2021). On the Development of a Multi-Layered Agent-Based Heuristic System for Vehicle Routing Problem under Random Vehicle Breakdown. *Cihan University-Erbil Scientific Journal*, 5(1), pp.1-10. doi:10.24086/cuesj.v5n1y2021.pp1-10.
- Abu-Monshar, A., Al-Bazi A., and Vasile, P. (2022) An Agent-Based Optimisation Approach for Vehicle Routing Problem with Unique Vehicle Location and Depot. *Expert Systems with Applications*, p.116370. doi:10.1016/j.eswa.2021.116370.

While the following are still in the publication process:

- Abu-Monshar, A. and Al-Bazi A. (2022) A Multi-Objective Centralised Agent-Based Optimisation Approach for Vehicle Routing Problem with Unique Vehicles. *Applied Soft Computing* (*Under Revision*)
- Abu-Monshar, A. and Al-Bazi A. Comparison between Hybrid and Centralised Agent-Based Optimisation Models for Vehicle Routing Problems with Unique Vehicles. (*Submitted*)
- An Agent-Based Optimisation Approach for Dynamic Vehicle Routing Problem under Random Vehicle Breakdowns. (*Draft*)

1.14 Thesis Structure

This thesis is organised into five chapters as follows:

Chapter 1 introduces the study by providing general background, stating the problem and defining the research aim and objectives along with their respective research design, techniques, deliverables and scope. This chapter further states the benefits of this research by highlighting its contributions and listing the disseminated work.

Chapter 2 reviews the relevant literature on dynamic routing problems. It also reviews the previously adopted solution approaches to such dynamic problems. The chapter then highlights the critical points from previous literature and identifies the knowledge gap.

Chapter 3 documents the proposed solution approaches implemented in this study. It first introduces the conceptual model for the problem under study and then presents the two proposed agent optimisation approaches.

Chapter 4 reports, analyses and discusses the experimentation results of the proposed approaches. Tests were conducted against benchmark instances and modified benchmark instances to suit the dynamic breakdown instant and on a case study. The latter is where breakdown scenarios are generated and then solved.

Finally, chapter 5 summarises and concludes the study. It highlights the lessons learnt and addresses the limitation of the study. It further states recommendations for future research.

1.15 Chapter Summary

This chapter presented a background on the current research trend in routing problems and the limited ability of the traditional optimisation approaches to address the problem under a dynamic context, highlighting the motivation behind this research. The research problem was then defined, followed by stating the research aim, objectives, design, techniques, deliverables and scope. Next, the benefits and contributions of this research have been stated, followed by a list of articles based on the work conducted for this thesis. Finally, this thesis structure was outlined by providing a brief introduction to each chapter.

Chapter 2

Literature Review

2.1 Chapter Overview

In recent years, researchers have been concerned about managing real-time disruptions to the pre-optimised plans of the Vehicle Routing Problem (VRP). Such a problem is labelled as the Dynamic VRP (DVRP), where a disruption event occurs during the executions of the optimised plans while the vehicles are in-route. This chapter reviews different DVRP variants, including various problems faced, and then all related methodologies used in solving these variants/problems will be presented and discussed. Given the traditional (static) approaches adopted in DVRP, agent-based modelling in DVRP is sought and critiqued. Finally, this chapter highlights the critical missed factors from previous studies to be explored in this research.

2.2 Relevant Static VRP

A static VRP assumes that all the information about the problem is explicitly provided in the planning stage and performs the optimisation solution accordingly. A change in the optimised solution is not possible as the problem is considered to be off-line (Gendreau et al., 1999). Research papers in this area are extensive; therefore, a sample of such papers is chosen and critically appraised in this section. Many other interesting VRP papers can be found. However, they were disregarded as the focus of this research is on DVRP. In this section, the samples selected are based on the key authors who introduced benchmark instances for the time window problem and authors who adopted the agent-based approach in optimising the static problem given its good adaptation to solve the dynamic case.

The first who solved vehicle routing problems and with time windows (VRPTW) heuristically is Solomon (1987) who modified previously developed saving heuristic algorithms (Clarke and Wright, 1964) and introduced the Push Forward (PF) mathematical constraint that calculates the time shift at later customers in route. Solomon introduced new benchmark problems widely used to test even newly developed methods. Problem instances' nodes were either randomly generated (R), clustered (C) or a hybrid between the two (RC). In addition, a problem can be limited with a short operating time denoted as "1" while "2" for longer scheduling time. The insertion heuristic has proved its significance by quickly generating the best solutions against the benchmarked instances compared to other approaches, Nearest Neighbour Heuristic (NNH) and Sweep Algorithm. Although Solomon made a considerable breakthrough in introducing benchmark instances for VRPTW and finding near-optimal solutions, the introduced instances assume vehicles to be similar and start/end at the same depot. Accordingly, the method proposed is not adapted to heterogeneous vehicles with different locations. As in the dynamic problem or crowd shipping case, considering unique vehicles, especially with different locations that may also end at different depots, is essential to optimise a dynamic case (Le et al., 2019).

Cordeau et al. (2001) adopted a metaheuristic approach to solve VRP variants, the Multiple Depot and Periodic, both with time windows (MDVRPTW) (PVRPTW). MDVRPTW is where vehicles can start and end their routes at multiple depot locations, while PVRPTW is when customers require frequent visits within different periods. Tabu Search (TS) was proposed with given an initial solution generated either by assigning customers to the nearest depot (MDVRPTW) or random periodic combination (PVRPTW). Then each vehicle is routed using a sweep algorithm (Gillett and Miller, 1974) which assigns customers to vehicles starting from the smallest polar angle. Route improvements are performed by exchanging nodes within and between routes across all depots/periods while minimising the total cost (time or distance) and violating each load capacity, route duration and time window constraints. Computational experiments were conducted on Solomon's benchmarks, a case study in the US for fast food routing, and specially designed benchmarks for MDVRPTW and PVRPTW, which later became notable benchmark instances for these variants. Results are compared against other approaches. In Solomon's benchmarked instances and the food delivery case, the proposed TS resulted in competing solutions and outperformed previous methods in some instances. On generated instances, the algorithm produced feasible results in a reasonable amount of time compared to longer iterations. They extended their work to consider the Site-Dependent VRP (SDVRP) variant with time window (SDVRPTW) (Cordeau and Maischberger, 2012). The Site-Dependent variant is where customers require services with a specific type of vehicle. An initial solution is generated with an insertion heuristic, while a sweep heuristic is used for the problem without time windows. TS then seeks improvements with further perturbation using Iterated Local Search (ILS) to explore a wider solution space. Experiments are implemented parallel with different parametric settings while the best is chosen. Incorporating ILS within TS resulted in quick, high-quality solutions tested on Solomon's instances. However, the generated test instances still assume identical vehicles grouped at their start/end locations; therefore, adaptation is needed to heterogeneous vehicles with completely different locations for its applicability in optimising the dynamic breakdown instance.

Thangiah et al. (2001) adopted the agent-based approach with two-agent types: auctioneer and vehicle, where bidding and negotiations method is adopted to solve the routing problem. The first agent deals with order data, announces bids on its behalf and allocates it to the suitable bidder vehicle. The second agent, the vehicle agent, performs cost calculations of announced bids with possible insertion of the order using the Clarke-Wright saving heuristic (1964) and makes a bid offer to the auctioneer. The work was tested on only static problems of VRP and resulted in reasonable solutions with deviations from best-known solutions. However, the study was limited to decentralised negotiation interactions that generate routes and tested on the instances without time windows. Therefore, further adaptation to the decentralised negotiation is needed to accommodate time window constraints.

Vokřínek et al. (2010) considered an architecture of a task and vehicle agents, where the latter generates its routes while applying priority rules (improvement policies) in the agents' interactions directed by a higher-level agent dubbed allocation agent. Additional ordering rules have been accommodated in later studies to consider the time-window problem (Kalina and Vokřínek, 2012). The priority rules policy is proven efficient in generating optimal solutions. However, given the slightly centralised cooperation through the allocation agent, this study considers a standardised evaluation of each vehicle's constraints without considering the uniqueness of locations and other vehicle agent attributes.

Martin et al. (2016) considered a standard VRP problem without time windows. They proposed a cooperative approach between the agent-based model and metaheuristics. Information about solution parameters is shared, and each agent performs a different metaheuristic combination. The approach is applied to two static problems: VRP and flow-shop problems. Their agent-based architecture consists of two types of agents: launcher and metaheuristic agents. The first configures the other agent types for a particular problem and gathers solutions from those agents. The other performs the predefined metaheuristic differently from the agents of the same type and then communicates the best solution elements to be further searched. The proposed approach was tested on benchmarked VRP problems compared with their best-known solution. It resulted in solutions mostly in around 1% costly deviation from the best-known solutions, while few have even better solutions by increasing the number of agents. However, the study assumes that vehicles are identical in their attributes and locations, hindering their applicability to the breakdown instant. Furthermore, the cooperative agent approach adopted is strictly centralised with a metaheuristic. A comparison against distributed and hybrid approaches would be beneficial to see their suitability for a particular application.

In summary, this section looked into static VRP literature to find benchmark instances that are closely related to the problem under study and studies that adopt the agent-based approach in optimising closely related problems. However, it is concluded that the benchmark instances consider homogeneous vehicles with similar start/end locations that may require adaptation, and possible randomisation of such attributes, to benchmark a developed algorithm that optimises the breakdown instant. In addition, a mixture of different approaches within the agent-based was adopted in solving variants of static VRP. They range from decentralised to centralised agent cooperation, and none yet made a comparison or stated the benefits/drawbacks of each. Furthermore, the agent heterogeneity, particularly vehicles, is not captured in agent-based studies in VRP, which may require specialised evaluation within each agent. In the next section and its respective subsections, DVRP is investigated, aiming to lookup for responsive solution methodologies given disruptive events in VRP.

2.3 Dynamic VRP

DVRP studies question the assumption of keeping routes fixed during execution by assuming the possibility of communicating such disruption and communicating back with the re-optimised routes that can be solved using different optimisation strategies: Periodic and Continuous (Abbatecola et al., 2016). Previous studies that adopt either strategy are explored and categorised in the following subsections 2.3.1 and 2.3.2, respectively.

2.3.1 Periodic Optimisation of DVRP

Periodic optimisation of a DVRP means that solutions are revised periodically in fixed intervals where interruptions are accumulated throughout the period and solved as a static problem (Pillac et al., 2013). The main limitation behind the periodic strategy is that it is not agile and responsive to dynamic events as they delay the optimisation process and accumulate multiple dynamic events to be optimised at once. Furthermore, the implementation of this approach was only limited to dynamic order VRPs, as evidenced in the following studies due to the low costs associated with accumulating few customer orders, rather than a sudden disruption of plenty of customer orders due to a breakdown. Periodic DVRP literature is critically appraised below.

One of the earliest studies found in periodic DVPR is attributed to Yang et al. (2004) who investigated real-time updates of new customers' orders with time windows using an exact approach. A single depot case is considered, and a vehicle handles one order at a time. A Mixed Integer Programming (MIP) model was formulated to optimise the accumulated dynamic orders periodically. Order arrivals and their locations are randomised. The proposed approach resulted in a better solution for minimising number of vehicles and distances in most scenarios than another simple greedy heuristic. However, given that an exact approach is used, it may not be suitable for a practical sized application which may be computationally expensive.

Chen and Xu (2006) studied only a collection problem with a single depot and proposed a Dynamic Column generation (DYCOL) method for a dynamic customer VRPTW to minimise the total travelled distance. A set-partitioning model was formulated where a single vehicle route is represented as a column. Routes are generated periodically where the served customers are removed while inserting newly accumulated orders into the solution space for re-optimisation. The approach was compared to results from an insertion-based heuristic and validated against Solomon's benchmarks. The benchmarks are modified to a dynamic problem by revealing customers randomly through the simulation. The proposed DYCOL outperforms the insertion-based heuristic. However, the proposed method can be computationally expensive for a practical case.

Murray and Karwan (2010) considered routing Unmanned Aerial Vehicles (UAV) that require mission re-planning given a new sudden dynamic task, taking into consideration the UAV fuel range constraint. The objective is to maximise the mission effectiveness by incorporating all tasks while minimising deviations from the original plan and travel time. An exact approach is adopted by formulating a MILP solved periodically. A military field of application has been considered. For a small-sized problem with around 15 tasks and three resources, an optimal solution was obtained. However, this is not considered practical for a large-sized problem given the adoption of a computationally expensive exact approach. They further extended their study (2013) by proposing a modified branch-and-bound algorithm that utilises constraints relaxation that reduces the solution space in order to reduce the time in optimisation. The proposed algorithm outperforms the former MILP solution significantly for large-sized problems with 50 tasks. However, the problem is still solved as a periodic problem focused on dynamism from the order side.

Ferrucci et al. (2013) simulated arrival of delivery orders with time windows using a discrete event to minimise customer waiting time. A proactive dummy node is generated for potential demand to divert a vehicle to it based on estimated stochastic data. A metaheuristic approach is adopted, and TS is run at every period provided with an initial solution from a least cost insertion heuristic. A case study of a German newspaper publishing company was adopted. Instances with different numbers of available vehicles were generated with/without a proactive strategy. The proactive insertion strategy improves customer satisfaction through earlier service delivery, especially in limited resources cases.

Ghannadpour et al. (2014) investigated a delivery problem with real-time arrival of orders with fuzzy time windows to minimise the number of vehicles used, travelled distance and time as well as maximising the customer service level. They adopted an evolutionary approach using GA with a periodic optimisation strategy and are provided with an initial solution using Solomon's insertion heuristic (1987) that is further improved by λ-interchange (Osman, 1993) that exchanges λ customer(s) among routes. They proposed a specially designed GA operator that handles the fuzzy time windows. The proposed approach was tested on modified Solomon's benchmarked problems to accommodate fuzzy time windows and a case study of blood distribution where emergency orders are dynamically revealed. The algorithm was able to generate optimal solutions compared to static benchmarked instances and produced feasible routes in the case study problem.

Albareda-Sambola et al. (2014) adopted an exact approach for a dynamic PVRPTW where customer requests can occur dynamically provided by stochastic data. A compatibility index between customers is proposed to calculate the potential cost saving of pairing customers served in the same period. Furthermore, a profit function has been proposed to determine the attractiveness of serving a customer in a given period based on its urgency and compatibility compared to other customers. The proposed exact formulation was modelled in CPLEX and tested on modified Solomon benchmarks for the dynamic and periodic problems. At the same time, a Variable Neighbourhood Search (VNS) has been implemented for large instances. The proposed method improved solutions more than others, while VNS reduced the computational time with near-optimal solution quality.

Barkaoui et al. (2015) investigated a problem with multiple visits given that the number of required visits is dynamically revealed and follows historical stochastic data aiming to minimise total travelled distance and time window violations. They adopted an evolutionary approach using GA run periodically with a crossover utilising an insertion heuristic and mutation that randomly swaps customers and can utilise customer visit data to proactively plan routes for potential visits. Solomon benchmarks were modified to adopt dynamic requests and test the approach compared with a greedy heuristic method. The proactive planning of future customer visits still resulted in time window violations. The GA has significantly outperformed the greedy method in minimising time window violations, while the greedy method minimised the distance significantly compared to GA.

Sarasola et al. (2016) adopted a metaheuristic approach using an adapted VNS for DVRP with dynamic request and stochastic demanded quantity aiming to minimise the total distance. Benchmark problems were modified to accommodate the dynamic requests and uncertain demand. The proposed algorithm produced feasible routes and showed improved results than the traditional VNS.

Ulmer et al. (2018) considered the PVRP with initial stochastic information about the customer demand as a Markovian Decision Process (MDP) approximated using the dynamic programming approach to pro-actively make decisions on which dynamic customers to be accepted at the current period or postponed for later periods. Decision points of the MDP are set at the start of every period, which makes it equivalent to periodic optimisation, and an insertion heuristic is adopted at these points to generate routes. Value Function Approximation (VFA) was used to predict the transition space (future customer requests). In a later study, Ulmer (2020) adopted the same methodology, however, with a proposed dynamic pricing strategy to encourage customers to choose delivery slots that are easy to deliver by the fleet and an additional problem objective of maximising revenues. The approach was tested on randomly generated instances with up to 50 periods, 100 customers and 75% degree of dynamism. The degree of dynamism is a problem parameter representing the percentages of customers occurring dynamically. Compared to other heuristics approaches, the method with the proactive prediction of the customer request has resulted in better solutions in terms of maximising the served customers. Later, Ulmer et al. (2021) investigated a dynamic pickup and delivery problem with stochastic data of customer arrival and availability of commodity being delivered to minimise customer waiting time. An assignment heuristic is proposed to anticipate customer demand based on stochastic data. The proposed method increased the service level compared to the traditionally implemented method, tested on real-life data. However, all these studies limited the source of dynamism to customer orders.

Zou and Dessouky (2018) investigated DVRP with time windows by minimising the total travelled distance. Customers request orders based on stochastic data and are utilised pro-actively to route vehicles for potential future customers. A heuristic approach has been adopted with a periodic optimisation strategy. A parallel construction heuristic is used to generate an initial solution to be further improved by SA, which utilises local search operators. The generated solution goes further improvement by slacking the time for every route based on the probability of accommodating future customers. The method was tested on Solomon's benchmarks by solving it dynamically where customers' requests are unknown ahead and statically where they are known. In the dynamic problem instances, the method resulted in distance savings close to the benchmarked static ones.

Ninikas and Minis (2018) considered a DVRP variant with transferable load among vehicles and customers requesting orders to be either collected or delivered to minimise the total travelled distance. A metaheuristic approach with periodic optimisation strategy was adopted, and applied Clarke and Wright (1964) saving heuristic and improving it using TS adapted to insert nodes where vehicles transfer their load among each other. The approach was tested on randomly generated data with/without load transfer and with/without time windows. The study resulted in around 20% cost reductions in terms of distance if the transfer took place, and it could easily happen if the time windows were widened.

Alisoltani et al. (2021) considered a DARP with dynamic travel times based on traffic congestion aiming to minimise such congestions. A simulation framework is proposed for the ride-sharing process with a MILP formulation to solve the optimisation problem periodically. A heuristic approach is adopted, and a construction heuristic is proposed to adapt to the dynamic problem that is further improved by clustering the dynamic requests received and force sharing rides across multiple passengers. The proposed method was tested on two cities' case studies, medium and large. It significantly reduced traffic congestion, especially in the large city with more ride-sharing demands, compared to traditional taxi and dial-a-ride services. However, although the study considered dynamic customer and travel times, it misses another source of dynamism from the resources (vehicles) perspective.

Dayarian and Savelsbergh (2020) investigated a crowdshipping problem equivalent to VRPTW with dynamic customer arrival in real-time, and stochastic customer data are utilised for robust optimisation to minimise the operational costs and delays at customers. A metaheuristic approach is adopted by periodically utilising a TS to solve the problem. Instances are generated to test the proposed method with different scenarios of company fleet size and outsourced ones. The model resulted in significant cost savings when the company fleet size is considered small by benefiting from the outsourcing to crowd-shippers.

Based on the previously surveyed periodic DVRP studies, it can be concluded that all these studies considered only dynamism from the order/customer side. No study considered dynamism from the resource/vehicle perspective as the latter is more applicable to the problem under study. Furthermore, the periodic optimisation strategy, in its nature, is not as responsive as the continuous strategy, which can be more agile in tackling disruptions by providing re-optimised routes quickly as it initiates the optimisation process the moment a dynamic event arrives. Disruptions may also include multiple random vehicle breakdowns that occur at different times.

2.3.2 Continuous Optimisation of DVRP

Contrary to periodic optimisation of DVRP, a continuous strategy performs the optimisation at the instant of the dynamic event to be more responsive and agile (Pillac et al., 2013). The following studies adopt such a strategy in routing or scheduling. Furthermore, studies that consider dynamism from the vehicle's perspective are further classified.

The dynamic order arrival was first discussed by Gendreau et al. (1999). They adopted a metaheuristic approach and adapted a TS from Taillard et al. (1997) provided with an initial solution from an insertion heuristic that starts by randomly selecting the first seed customer to run the algorithm in parallel. The aim is to minimise the total distance travelled and time window violations. In case of dynamic order arrival, it will be checked to be feasibly inserted in an initial solution and rerun the Tabu Search. The algorithm was tested on Solomon's benchmark problems (1987) along with a discrete-time simulator to generate new requests. Compared to simple insertion heuristics, the proposed algorithm minimised the costs and the number of rejected orders. However, this study limits the source of dynamism from the order side.

Haghani and Jung (2005) investigated a pickup and delivery problem with dynamic order and travel times due to traffic uncertainty to minimise vehicles used, travelled distance and time window violations. A MILP was formulated and solved using GA with a randomised initial population with simulation to mimic the dynamic network operations and continuously optimise the problem when changes happen. Test problems with different sizes were randomly generated, and results were compared to an exact solution from the lower-bound solution method. The exact solution resulted in the minimum cost results. At the same time, the GA was able to produce results that are very close to the exact method with significantly less computational time, making it more useful for large-sized problems. Although their study considered a pickup and delivery problem that is applicable, partially, to the dynamic vehicle problem, this study only considered a dynamic order problem. It can be further extended to a dynamic vehicle one.

Similarly, Potvin et al. (2006) considered dynamic order and travel times for a collection problem with neglected vehicles' capacities problem to minimise the total travelled time and time window violations. A heuristic approach is adopted and implemented as an insertion heuristic, later improved by node exchange and CROSS arc exchange (Taillard et al., 1997). A discrete-event simulation method introduces the dynamic events, and re-optimisation is sought by finding the best possible position for the order across all routes. At the same time, travel times uncertainty is dealt with by only updating the planned arrival times of the vehicles at the node. Solomon's benchmarks were modified by introducing two parameters: route time limits and normal standard deviation to reveal dynamic orders. Results show that the method generated solutions with low or no deviations from the best solution with a higher time tolerance limit and low standard deviation. On the other hand, solutions resulted in high costs. This study, however, lacks the dynamism in vehicle availability.

Cheung et al. (2008) investigated a pickup and delivery problem with uncertain travel times and orders with time windows to minimise the travel time. GA is used and provided with an initial population generated by a sequential insertion heuristic adapted for the pickup and delivery problem that starts with a random seed customer. When a node is selected, its complement node will be inserted before (pickup) or after (delivery). However, only a simple insertion is used for dynamic orders to find its best position. GA was used statically and tested on a randomly generated problem with a size of up to 200 customers. The dynamic insertion was tested on the same problem but with hidden orders revealed dynamically. The dynamic insertion heuristic resulted in slight deviations from the benchmarked static solution and produced results in a reasonable amount of time. However, the dynamism in this study is still limited to dynamic orders only.

Barkaoui and Gendreau (2013) considered a continuous dynamic order DVRP with time windows to explore the different combinations of adaptive operators. A proposed two-level GA evolves both the solution and operator combinations (population selection, crossover and mutation). Tests were conducted on Solomon's benchmarks adapted for the dynamic problem. The algorithm produced the best dynamic results close to the optimum static results. Also, it has the best results in minimisation time window violation and number of missed customers. However, although the approach optimised the dynamic order problem, it does not consider vehicle breakdown.

Spliet et al. (2014) investigated a delivery problem with dynamic customer orders/cancellations formulated using MILP to reduce the travelled distance and deviations from the original plan. A two-phase heuristic is proposed to remove costly arcs and then insert them to minimise the objective function. A continuous optimisation strategy is adopted by manually calling the method at the dynamic instance. Tests were conducted on randomly generated problems compared to a computationally expensive exact branch-and-cut approach. The proposed algorithm produced very close to optimal results in most problems with significantly low computational time. Although a dynamic order problem is considered, vehicle dynamism is missed in this study.

Schyns (2015) considered an aircraft refuelling problem that is delivered by trucks in a dynamic airport environment to minimise travelled distance. A VRPTW with split deliveries, as trucks' capacity, may be lower than the demanded fuel quantity, and demand is dynamic given changes in flight schedules. The dynamic problem is simulated and continuously solved using a proposed Ant Colony Optimisation (ACO). A case study is adopted from

Liege Airport in addition to tests on Solomon's benchmarked problems. The proposed method was responsive to dynamic orders and provided near-optimal solutions for the distance travelled in minimal computational time. However, this study did not consider dynamic resource availability (vehicles).

Euchi et al. (2015) studied dynamic order DVRP with pickup and delivery to minimise the total travelled distance. A metaheuristic approach was adopted and proposed an Ant Colony System (ACS) to solve the problem continuously with an initial solution provided from an insertion heuristic improved with Local Search (LS) using arc exchange with the route (2-opt). The algorithm was tested on a set of delivery/collection instances for VRP and solution and then compared to the best-known solution. The proposed work provided the best results in these instances to minimise the travelled distance. Although this work provided the best-known solutions, it did not consider the problem with time windows and dynamic vehicle availability.

Bopardikar and Srivastava (2020) investigated a DVRP with a single vehicle and multiple trips considering dynamic order arrival with known stochastic distribution to minimise the expected service time per customer. A Spatio-temporal stochastic model is proposed and then solved using a heuristic used in solving Travel Salesperson Problem (TSP), dubbed as TSP-based policy, that routes customers within a vehicle. Based on random numerical analysis and comparison against NNH, the proposed heuristic outperforms NNH in finding better optimal solutions. However, this study does not consider the time window constraint and limits dynamism from the customer side.

Vinsensius et al. (2020) studied a dynamic and stochastic customer problem for VRPTW where customers choose the preferred delivery slot to maximise revenues and minimise travel costs. An approximate dynamic programming approach is adopted to estimate routing costs, while an incentive-based method is proposed for the availability of dynamic delivery slots to increase routing profitability. Tests were conducted on generated instances, and the method resulted in significant savings compared to methods without intensives, where customers do not choose slots. Although the study provides a demand-driven incentive approach in the optimisation, it does not consider dynamic vehicle availability.

Wang et al. (2021) considered a real-time customer arrival in a VRPTW case to minimise total distance and waiting time. An evolutionary approach is adopted with ensemble learning that utilises previous population data to produce high-quality solutions efficiently. The proposed method was tested on generated dynamic instances and outperforms other metaheuristics in solution quality, diversity of the Pareto front and efficiency. However, this study does not consider dynamism from the resources side (vehicles).

Based on the previously reviewed papers in continuous DVRP, only order arrivals or uncertainties are considered the source of dynamism. Therefore, the studies that consider vehicle dynamism under the continuous optimisation strategy are further categorised.

Vehicle Dynamism/Breakdown in Routing/Scheduling

Since only order-related dynamism is surveyed under continuous DVRP, the following further surveys studies that consider vehicle breakdown(s) under continuous optimisation strategy regardless of the problem to be routing or scheduling.

The first study on vehicle breakdown in VRP was done by Li et al. (2009b) who considered a vehicle disruption within a delivery and collection problem. In a delivery problem, the disrupted vehicle must be visited to collect its load to minimise service cancellations and the total distance travelled. A set covering VRPTW is formulated and solved heuristically with a Lagrangian relaxation covering all customers. An initial solution is obtained through a dynamic programming-based heuristic with eliminated two cycles to decrease computational time. A similar formulation is used in their second paper (2009a) but for a scheduling problem, Vehicle Rescheduling Problem (VRSP), that is solved by a column generation method. Both studies produce solutions that may result in inadmissible paths and uncovered customers; therefore, an insertion heuristic was adopted to improve solutions further. The earlier study was tested on Solomon's benchmarks, and the second was tested on randomly generated problems, both of which specified a vehicle to breakdown and its time and place. Methods were compared against a developed method in intuitive human rescheduling from an earlier study (Li et al., 2007). The proposed algorithms produced more cost-effective results than the human intuitive approach to every problem instance. However, performance on a large-scale problem is low and high computational complexity and costly. Although this is the earliest study considering a disruption from the vehicle side in routing, it is only limited to one breakdown with a pre-specified time and location. Therefore, vehicle disruptions can be further randomised and occur more than once per problem instance.

Wang et al. (2009) considered a VRPTW case with a pre-specified breakdown. An evolutionary approach for a MILP formulation was solved using a modified GA by proposing a specially designed crossover for the time window problem and adapting the initial population provided with a saving heuristic, then improved using a neighbourhood search. Tests were conducted on one instance of small-sized data with four pre-specified breakdown scenarios, one vehicle per scenario. It resulted in better distance and time savings. However, the problem is not considered dynamic and does not randomise breakdowns in continuous time.

Mu et al. (2011) considered a commodity delivery VRP with vehicle breakdown that does not require to be visited due to the delivery of the same commodity to minimise the number of vehicles, and the total distance travelled. A heuristic approach is adopted with an initial solution proposed by modifying the original route plan by inserting the disrupted customers into the routes and then improving it through Tabu Search. Tests were conducted on VRP benchmarks (VRP-REP, 2014) and modified by introducing pre-specified disruption. The heuristic algorithm provides a slight costly deviation from the optimal solution provided by the exact method. However, the study does not consider a time window problem which needs a different heuristic method. In addition, it does not consider randomised vehicle breakdowns for both delivery and collection routing problems.

Minis et al. (2012) investigated the Team Orienteering Problem (TOP) with a single-vehicle breakdown, similar to VRPTW but differed in its objective to maximise the customer coverage. An insertion heuristic is proposed by randomly inserting disrupted customers to routes based on a selection probability and then improving the solution by exchanging arcs (2-opt exchange). Comparison is made against GA, and tests were conducted on modified benchmark problems and newly generated large instances. The proposed heuristic approach produced results close to the ones resulting from the computationally expensive GA approach with only a 3% deviation. Similar results were achieved when changing the problem to the same product delivery, which allows the operating vehicles to replenish their capacities from depots and disabled vehicles (Mamasis et al., 2013). However, this study only considers a single static breakdown at a specified time per instance.

Monroy-Licht et al. (2017) investigated a Rescheduling-Arc Routing Problem (R-ARP) where arcs, or streets, need to be serviced given a vehicle breakdown aiming to minimise distance travelled and the disruption cost. A MILP was formulated and solved using a greedy heuristic algorithm, then tested on generated instances and produced optimal solutions with a slight trade-off between the conflicting objectives. However, this study considers the vehicle breakdown predetermined and applied once per instance.

Amrouss et al. (2017) investigated a rescheduling problem with dynamic disruptions from both the order, including cancellations and demand and vehicle with possible delays and breakdowns. A MILP was formulated with a time-space network presentation solved in real-time and produced optimised solutions in seconds for a case study in the forest industry. However, the study is limited to a scheduling problem with only one breakdown.

Dávid and Krész (2017) considered the Dynamic Vehicle Rescheduling Problem (DVRSP) that schedules trips given multiple vehicle breakdowns to minimise travelled distance, deviations from the original plan and delays. A trip is an arc that has a fixed travel distance and time and consists of two nodes, each with specified arrival and departure time. Connection-based and time-space networks were adopted to model the problem. Two heuristic methods were proposed, recursive and local search algorithms, and tested by considering their running time and deviations from the static solution knowing all disruptions ahead. A real-life bus scheduling problem has been considered, and randomly generated problems of different sizes. Disruptions were introduced by randomly specifying a vehicle and its inactive period. Both methods have resulted in computationally efficient solutions that are slightly deviated from the static solution of test instances. Similarly, efficient results were generated for the real-life problem. However, solution deviations were not provided. Although this study is the first to consider multiple dynamic vehicle disruptions, it was limited to a scheduling problem with pre-specified trips.

Van der Merwe et al. (2017) adapted the VRP problem to a dynamic wildfire response vehicle with a breakdown by proposing a modified MILP formulation from their previous static one (Van der Merwe et al., 2015). A case with changes in weather conditions was considered, as wind speed and direction may affect the spread rate and direction of a wildfire, making it equivalent to a dynamic order DVRP. Additionally, a single-vehicle breakdown is considered and rerouted to maximise the coverage and minimise the deviations from the original plan. A bi-objective MILP was formulated and solved with a time limit of 30 minutes. A real-life wildfire scenario was considered in South Hobart, Tasmania, Australia, and instances were generated with 30 to 60 locations served by ten vehicles considering one vehicle breakdown chosen arbitrarily. Optimal solutions were found for small-sized problems; however, large-sized problems from 40 service points and above needed more time and could not be solved. This study, however, is computationally expensive given its exact approach and did not consider multiple and random vehicle breakdowns, where disrupted vehicles and their time of breakdown occurrence are randomised.

Seyyedhasani and Dvorak (2018) investigated the dynamic agricultural vehicle availability as changes in the number of vehicles, the expected service time or the working area could occur. A metaheuristic approach was adopted using TS with an initial solution adapted from the saving algorithm (Clarke and Wright, 1964). A real case from an agricultural field with three vehicles was adopted with a dynamic vehicle event introduced when the land was 50% completed. The method was able to generate optimal solutions by minimising the completion time. Although the study considered dynamic vehicle availability, it did not consider multiple random vehicle breakdowns.

Van Lieshout et al. (2018) considered the VRSP of Li et al. (2009a) but with re-timing to possibly delay trips to maximise the customers served and minimise travel costs. The MILP formulation has been adapted to the re-timing constraint undergoing Lagrangian relaxation. A metaheuristic approach was adopted using an Iterative Neighbourhood Exploration (INE) and tested on Li et al. (2009a) generated problems consisting of 700 trips. The concept of "re-timing" and delaying services for only two trips for an average of 8 minutes reduced service cancellations by around 60%. However, given that this work studied the same problem as Li et al. (2009a) studied, it similarly lacks in considering multiple random vehicle breakdowns.

Guedes and Borenstein (2018) investigated a rescheduling problem with pickup and delivery and heterogeneous vehicles that may face breakdowns that occur at pre-specified point time affecting specific trips to minimise travelled distance and deviations from original plans. A time-space network is used to generate the possible list of arcs while eliminating infeasible ones (Guedes and Borenstein, 2015). Accordingly, a MILP was formulated and solved using a heuristic algorithm with truncated Column Generation (CG) techniques that do not repeatedly generate columns when the solution is not improving. A case study of the bus transit system in Santa Maria, Brazil, was adopted as well as randomly instances were generated with up to 2500 trips. The method was tested under different experimental settings, with/without arc elimination and modified column generation, and resulted in solutions for large instances in less than 150 seconds with minimum deviation. However, this study did not randomise the breakdown event and introduced the disruption at a specific time and place for the vehicle(s). In case of multiple breakdowns, they are introduced at the same instant.

Pandi et al. (2020) considered a Dial-A-Ride Problem (DARP), equivalent to dynamic to pickup and delivery but with passengers, and introduced a single breakdown in the schedule at a specific time aiming to minimise the number of vehicles used. An Adaptive Large Neighbourhood Search (ALNS) is proposed to be processed in graphical processing units to maximise the efficiency in continuously optimising the breakdown problem. The method improved vehicle utilisation and reduced operational costs under disruption with efficient computing compared to traditional approaches implemented in central processing units. However, the study pre-specified the time of the single breakdown targeting a specific vehicle. This differs from the problem under study where disrupted vehicles and their time of breakdowns is fully randomised which requires an adoption of a simulation approach.

Upon reviewing all studies related to dynamic vehicle availability, it is found that it is still a relatively new problem and rarely tackled with just above a dozen studies. The earliest study dates back to 2007 compared to the mature literature body in traditional VRP dating back six decades earlier (Dantzig and Ramser, 1959). All the studies in this area have considered breakdowns to be pre-specified at fixed points in the route or schedule. Furthermore, cases considering multiple breakdowns are studied in scheduling rather than routing problems. Bodin and Golden (1981) highlighted the difference between scheduling and routing problem, scheduling is when the arrival times at customers are fixed while in routing, they are unspecified which increasing the complexity of the problem with a bigger solutions space. A close view of these studies is detailed in section 2.5.

2.3.3 Tools and Techniques Used in Dynamic VRP

Based on the previously reviewed DVRP studies, a survey has been done based on the solution approaches adopted by each paper. Table 2.1 provides a summary of these approaches ranging from exact solutions to metaheuristics, in addition to some authors who adopt a simulation technique to capture the dynamism of the problem. Agent-based modelling and simulation technique has also been taken into consideration; however, they are not included in this table summary as all agent-based studies in DVRP are surveyed in a dedicated subsection 2.3.4 following Table 2.1. It is worth mentioning that some of the studies adopted a robust optimisation approach. The robust approach in VRP utilises stochastic data to route vehicles pro-actively for potential cost savings (Zou and Dessouky, 2018) and are amended later based on the actual updated information.

Table 2.1 DVRP Solution Tools and Techniques

Technique Type	Technique Used	Periodic/ Cont.	Robust	By Author(s)
Exact	MILP	P		Yang et al. (2004)
		P		Murray and Karwan (2010)
		P	✓	Albareda-Sambola et al. (2014)
		C		Van der Merwe et al. (2017)
		C		Monroy-Licht et al. (2017)
		C		Amrouss et al. (2017)
	CG	C		Guedes and Borenstein (2018)
	DYCOL	P		Chen and Xu (2006)
		C		Li et al. (2009b)
		C		Li et al. (2009a)
	Branch-and-Bound	P		Murray and Karwan (2013)
	Branch-and-Cut	C		Spliet et al. (2014)
	MDP	P	✓	Ulmer et al. (2018)
		P	✓	Ulmer (2020)
		P	✓	Ulmer et al. (2021)
	Dynamic Programming	C		Vinsensius et al. (2020)
Heuristics	Local Search	C		Spliet et al. (2014)
		C		Dávid and Krész (2017)
		C		Guedes and Borenstein (2018)
	Recursive Search	C		Dávid and Krész (2017)
	Insertion	C		Potvin et al. (2006)
		C		Cheung et al. (2008)
		C		Li et al. (2009b)
		C		Li et al. (2009a)
		C		Minis et al. (2012)
		C		Mamasis et al. (2013)
	Lagrangian	C		Li et al. (2009b)
		C		Li et al. (2009a)
		C		Van Lieshout et al. (2018)
	Greedy	C		Monroy-Licht et al. (2017)
		C		Bopardikar and Srivastava (2020)

Table 2.1 continued from previous page

Technique Type	Technique Used	Periodic/ Cont.	Robust	By Author(s)
	Construction	P	✓	Zou and Dessouky (2018)
		P		Alisoltani et al. (2021)
	Saving	P		Ninikas and Minis (2018)
		C		Seyyedhasani and Dvorak (2018)
	Assignement	P	✓	Ulmer et al. (2021)
Metaheuristics	TS	C		Gendreau et al. (1999)
		C		Mu et al. (2011)
		P	✓	Ferrucci et al. (2013)
		P		Ninikas and Minis (2018)
		C		Seyyedhasani and Dvorak (2018)
		P	✓	Dayarian and Savelsbergh (2020)
	GA	C		Haghani and Jung (2005)
		C		Cheung et al. (2008)
	GA	C		Wang et al. (2009)
		C		Barkaoui and Gendreau (2013)
		P		Ghannadpour et al. (2014)
		P		Barkaoui et al. (2015)
		C		Wang et al. (2021)
Metaheuristics	SA	P	✓	Zou and Dessouky (2018)
	ACO	C		Schyns (2015)
	ACS	C		Euchi et al. (2015)
	LS	C		Euchi et al. (2015)
	INE	C		Van Lieshout et al. (2018)
	VNS	P	✓	Albareda-Sambola et al. (2014)
		P		Sarasola et al. (2016)
	G-ALNS	C		Pandi et al. (2020)
Simulation	DES	C		Gendreau et al. (1999)
		P		Yang et al. (2004)
		C		Haghani and Jung (2005)
		C		Potvin et al. (2006)
		P	✓	Ferrucci et al. (2013)
		C		Barkaoui and Gendreau (2013)
		P		Ghannadpour et al. (2014)
		C		Schyns (2015)
		C		Euchi et al. (2015)

It can be concluded from Table 2.1 that some studies have adopted hybrid techniques, while the most common approach used is the metaheuristic. Although many papers adopted the periodic optimisation strategy, most of the strategies adopted were under the continuous optimisation category due to the higher demand for a responsive solution for the problem. The robust strategy is associated only with periodic optimisation, and few authors have implemented it.

2.3.4 Agent-based Approach in Dynamic VRP

Contrary to the traditional approaches in DVRP surveyed earlier, the agent-based approach is seen to be emerging dynamic problems (Barbati et al., 2012). The agent-based approach in DVRP is not new and is mainly adopted using a dynamic continuous optimisation strategy to improve routing operations' agility. Distributed Artificial Intelligence (DAI), which agent-based is part of, delegates to an agent a rational sense and autonomy (Wooldridge and Jennings, 1995). Studies in DVRP that utilised this approach were concerned with capturing the dynamism of customer requests during the execution of the routes, which has proven its superiority compared to the traditional Operational Research due to their limitation to static optimisations (Fischer et al., 1996). The following review consists of literature that uses the agent-based approach in DVRP.

The earliest agent-based study found was for Kuhn et al. (1994), considered a dynamic order problem modelled in a multi-agent system consisting of two agent types: shipping and truck agents. The first represents customers and their demands based on the supplied database, while the latter performs route planning. Orders are assigned to trucks using a form of cooperative interaction dubbed Contract Net Protocol (CNP). Two forms of CNP interactions are proposed, one between a shipping agent and their truck agents (vertical) while the other among shipper agents to exchange orders (horizontal). Preliminary results on a small-sized problem were generated, and results with both cooperation types showed cost reductions and increased utilisation. Fischer et al. (1996) extended the CNP interaction by consolidating the agent architecture with company and truck agents where companies receive orders and extend the bidding process to its trucks. Then it evaluates the bid to make an allocation decision. Their work was tested on static benchmark problems with 100 orders modified to accommodate dynamism in the orders. The system resulted in acceptable solutions compared to the traditional heuristic approach. However, these studies report preliminary results and do not report the efficiency of their proposed methods. In addition, their proposed communication protocols do not consider other problem constraints such as customer time window and unique vehicle constraints, e.g. capacity and shift. Those require a more detailed definition of agents and adapt their evaluation and interactions accordingly. If such constraints are added, a challenge arises in the computing efficiency to producing optimal results.

Kohout and Erol (1999) also utilised CNP but with a different agent architecture consisting of customer, vehicle and verifier agents, where the latter type manages data of the modelling. Customers issue orders with time windows and are considered through a bidding processing between the customer and vehicle agents using CNP aided with modified Solomon's insertions for the pickup and delivery problem. The approach was tested on data from an airport shuttle company in Washington, DC. The proposed approach was only tested on the static problem and resulted in solutions comparable to the original heuristics of the problem. However, solution time is not reported for the proposed method in order to test its applicability to the dynamic problem. Furthermore, the study lacks the vehicle heterogeneity that requires further definitions within the vehicle agent and adapting its local evaluation considering efficient methods.

A similar work adopted by Zeddini et al. (2008) considered an auction negotiations interactions involving two agents: Client representing the customer and vehicle agents. The latter performs bids based on the Clarke-Wright saving algorithm (1964) and Bidder, which acts as a medium between the Client and Vehicle agents where it broadcasts a client request to all vehicles. The approach was tested on modified benchmarks to accommodate the dynamic problem and resulted in slightly costly solutions. However, their results are preliminary and do not report the efficiency of their proposed method. Moreover, practical problem constraints are not considered; adding more customer and vehicle constraints may increase the computational complexity.

Mes et al. (2007) adopted a more agent-based decentralised approach for a DARP, equivalent to dynamic pickup and delivery but with passengers. They also considered random servicing and travel times to increase vehicle utilisation and server level. Furthermore, they have adapted an auction mechanism, adapted from Vickrey (1961) between the orders (shippers) and the bidding vehicles (fleet), where the latter calculates bids based on adding a job to the end of their route, insertion or constructing a new route. An experimental setting is based on Amsterdam Airport Schiphol's case study to manage its internal transportation network. The proposed model with insertion heuristic was able to generate reasonable solutions in terms of vehicle utilisation and service level and outperformed traditional heuristics in its computational complexity. However, a modelling challenge would arise in accommodating additional constraints and evaluating them if vehicles are considered heterogeneous, each having unique attributes, which may require a unique local bid evaluation technique by the vehicle agent.

Barbucha and Jedrzejowicz (2007) (2009) also adopted CNP but with different agent architecture and adopted further route improvement heuristics considering a hybrid search approach that is between the decentralised and centralised searches in agent-based. CNP is involved with Company, Vehicle, Request Generator and Request Manager agents. The agent-based system runs in three phases: allocating static customers using a sweep algorithm (Gillett and Miller 1974), allocating dynamic requests using CNP, and route improvement heuristics within and across other routes. Solutions resulted in up to 8% deviations from best known benchmarked static solutions. In later studies (Barbucha, 2012) (2013) (2020), they extended their distributed search to a more centralised one by extending the improvement heuristics to metaheuristics with guided local search and population-based optimisation and experimented with their parameters and managed to increase the efficiency of the approach further. Although these studies showed promising adoption of the agent-based approach in DVRP using hybrid and centralised approaches, a more challenging modelling case is required if additional constraints are considered. As a result, a customised agent evaluation and interactions is needed for both centralised and decentralised approaches.

Maciejewski and Nagel (2012) developed an architecture which is centralised around a Dynamic Optimiser agent that utilises a memetic algorithm and the problem-specific agents: Customer, Traffic Monitor and Vehicle Fleet. The approach was tested on specially generated problems and compared to solutions to the static problem, meaning that all orders are known in advance. The approach produced feasible solutions; however, slight cost increase compared to the static solution. The approach could be further elaborated to a decentralised approach for a dynamic vehicle availability problem.

Gath et al. (2013) adopted the k-means clustering approach for static VRP, then adopted CNP to deal with dynamic orders and implemented depth-first branch-and-bound routing within each vehicle agent to route and calculate the bid. The approach was tested on a real transportation case where order data were collected from an industrial company. The model quickly generated feasible solutions that maximised the serving of dynamic order; however, with a slight percentage of missed ones given the hard constraint strategy adopted while the efficiency of the method is not reported.

Nambiar and Idicula (2013) considered a waste collection problem that requires re-routing in case a vehicle is nearly full to minimise return trips to the disposal site. They have adopted a knowledge base architecture with three main agents: Master Data, Master Control and Vehicle agents. Master Data constructs the geographical data while the control agent optimises based on k-means clustering and ACO. Vehicle agents send real-time data about executing their routes, and in the instance, a vehicle exceeds its threshold limit, it reports to the Master Control for re-routing. The modelling approach decreased the total distance travelled, making sense because of the re-routing to minimise vehicles' trips to the disposal site. However, the efficiency of the adopted method was not reported. Moreover, a challenge would arise if adopting this method to a more highly constrained problem that considers specific constraints per vehicle agent. It may require a unique design of a local evaluation at the vehicle agent level.

Fonseca-Galindo et al. (2022) investigated the DVRP with dynamic stochastic customers aiming to minimise the total distance travelled. Their multi-agent implementation is based on the hybrid implementation from Barbucha and Jędrzejowicz (2009); however, with trajectory data mining techniques to utilise stochastic customer data in distributing packages. Furthermore, the data mining technique generates territorial patterns to be used as an agent "bet" to improve its accommodation of dynamic orders. The proposed method was compared to other (meta)heuristics methods, resulting in significant cost reductions and algorithmic efficiency. Although the study proved the suitability of trajectory data mining within hybrid agent-based implementation of DVRP, the interaction and messaging adopted are still limited to a customised, per agent, constrained problem.

To summarise, the agent-based modelling approach in optimisation can be seen emerging in DVRP; however, the implementation, problem wise, was limited to dynamic order problems. The previous agent-based VRP implementations considered suitable agent interaction designs that managed to produce feasible solutions. However, most previous studies were mainly focused on proposing a working solution rather than studying the efficiency of the proposed methods in terms of their computational complexity. Furthermore, if additional constraints are considered, a generalised agent evaluation is needed to consider the agents' different attributes, for example, different vehicle capacities, shifts, and locations, which would make the problem more suitable for the dynamic vehicle availability.

2.4 Overall Literature Review Matrix

In order to identify the research gap, Table 2.2 was constructed to provide a complete overview of all the literature explored in this chapter based on the type of dynamic problem considered. All of the above-studied papers, dynamic only, are classified based on the VRP variant as indicated by classifying them into two categories based on the nature of the problem dynamism. The two horizontal sections in the table represent this dynamism classification and are detailed as follows:

- Dynamic order VRP: The source of dynamism is limited only to the customers' side (temporary entity). For example, the issue or cancellation of orders while in operations.
- Dynamic vehicle VRP: The source of dynamism is only from the vehicle's side (permanent entity), for example, a vehicle breakdown.

Table 2.2 Comprehensive Literature Review Matrix

Problem Type	Author	Scheduling	Capacitated	Time Window	Multiple Depot	Hetero-Vehicles	Periodic	Pick & Delivery	Site-Dependent	Dial-A-Ride	Stochastic Demand	Multiple Trip
Dynamic Order	Kuhn et al. (1994)		✓									
	Fischer et al. (1996)		✓									
	Kohout and Erol (1999)			✓				✓	✓			
	Gendreau et al. (1999)		✓	✓								
	Yang et al. (2004)		✓	✓				✓				
	Haghani and Jung (2005)		✓	✓		✓		✓		✓		
	Potvin et al. (2006)			✓						✓		
	Chen and Xu (2006)		✓	✓				✓				
	Mes et al. (2007)			✓					✓	✓		
	Zeddini et al. (2008)		✓	✓								
	Cheung et al. (2008)		✓	✓				✓				
	Barbucha and Jedrzejowicz (2007)(2009)(2012)(2013)(2020)		✓									
	Murray and Karwan (2010)			✓								
	Maciejewski and Nagel (2012)			✓								
	Ferrucci et al. (2013)			✓						✓		
	Gath et al. (2013)		✓	✓		✓		✓				
	Nambiar and Idicula (2013)		✓			✓						
	Barkaoui and Gendreau (2013)		✓	✓								
	Spliet et al. (2014)		✓									
	Albareda-Sambola et al. (2014)		✓	✓			✓			✓		
	Ghannadpour et al. (2014)		✓	✓								
	Barkaoui et al. (2015)		✓	✓								
	Euchi et al. (2015)		✓					✓				
	Schyns (2015)		✓	✓							✓	✓
	Sarasola et al. (2016)									✓		
	Ulmer et al. (2018)						✓			✓	✓	
	Ulmer (2020)						✓				✓	
	Ulmer et al. (2021)		✓					✓			✓	
	Zou and Dessouky (2018)		✓	✓						✓		
	Ninikas and Minis (2018)		✓	✓								
	Van der Merwe et al. (2017)			✓								
	Seyyedhasani and Dvorak (2018)			✓								
	Bopardikar and Srivastava (2020)		✓								✓	✓
	Wang et al. (2021)		✓	✓								
	Vinsensius et al. (2020)		✓	✓							✓	
	Alisoltani et al. (2021)		✓	✓		✓		✓		✓		
	Dayarian and Savelsbergh (2020)		✓	✓		✓						
	Fonseca-Galindo et al. (2022)											
Dynamic Vehicle	Li et al. (2009b)		✓	✓				✓				
	Li et al. (2009a)	✓	✓	✓				✓				
	Wang et al. (2009)		✓	✓	✓							
	Mu et al. (2011)		✓									
	Minis et al. (2012)		✓	✓								
	Mamasis et al. (2013)		✓	✓								
	Dávid and Krész (2017)	✓			✓							
	Monroy-Licht et al. (2017)	✓										
	Amrouss et al. (2017)	✓										
	Van der Merwe et al. (2017)			✓								
	Seyyedhasani and Dvorak (2018)			✓								
	Van Lieshout et al. (2018)	✓		✓								
	Guedes and Borenstein (2018)	✓	✓		✓		✓					
	Pandi et al. (2020)		✓	✓						✓		

From Table 2.2, it can be seen that a research paper could consider multiple variants of the problem. The stochastic demand variant represents customers' order uncertainty in terms of their time of occurrence. Furthermore, a paper could explore two of the dynamic classifications, both dynamic order and vehicle, which were seen only in Van der Merwe et al. (2017) and Seyyedhasani and Dvorak (2018). The dynamic problem was researched and concluded that most of the authors who look into DVRP limit the source of dynamism to the customer. However, few papers considered dynamic vehicle availability and not all considered it a routing problem, while all considered vehicles to be homogeneous. If heterogeneous vehicles are considered, the proposed method needs to have a generalised evaluation procedure across all vehicles. The agent-based method can aid in developing such an evaluation procedure given its ability to customise the agent's attributes and consider them as constraints. A detailed problem critique is provided in the next section 2.5.

2.5 Close View at the Related Literature

Upon reviewing the related literature, most of the DVRP studies focused on routing under dynamic order arrival/cancellation, while only a few papers have considered a vehicle breakdown case. However, they have made some limited assumptions and do not apply to the case problem of this research. Dynamism from the vehicle side was taken into consideration when Li et al. (2007) brought attention to this problem. Li et al. (2009b), Minis et al. (2012), Mamasis et al. (2013), Van der Merwe et al. (2017) and Seyyedhasani and Dvorak (2018) considered only one vehicle breakdown per schedule and pre-specified the time and location of the breakdown. Mu et al. (2011) further limited the problem to a single commodity delivery, where the only broken down vehicle (specified previously) in the schedule does not have to be revisited to collect the load. Similarly, only one-vehicle breakdown problem was tackled in Li et al. (2009a) and Van Lieshout et al. (2018), however, for a vehicle scheduling problem. According to Bodin and Golden (1981) vehicle scheduling differs from routing as it specifies explicitly when the vehicle should arrive at a node. As a result, the routing problem would be more complex due to the bigger solution space to be searched because of the arrival time flexibility. Multiple breakdown cases have been only considered in scheduling problems in Guedes and Borenstein (2018), Monroy-Licht et al. (2017) and Amrouss et al. (2017), where up to three pre-specified vehicle breakdowns that occur almost simultaneously, and Dávid and Krész (2017)), where breakdowns happen over time. It is only in the work of Wang et al. (2009) and Pandi et al. (2020) who considered a single breakdown for a dynamic routing problem; however, they are limited to be one time per schedule. A more detailed critique of the papers mentioned above is shown below:

- Li et al. (2009b) (2009a) tested their developed algorithm in different problem settings based on the time and location of a pre-specified vehicle being disrupted. For example, it could be on the 20%, 50% or 80% of this vehicle route. Therefore, the problem is considered a deterministic disruption, not dynamic. They have also limited their problem to only one vehicle breakdown in the entire schedule. Therefore, their approach was limited to only one-time re-routing. In addition, they assumed that every vehicle has the same capacity in a single depot scenario.

- Wang et al. (2009) considered a breakdown problem. However, not a dynamic one as the formulated model is only designed for one breakdown that is not even random.
- Mu et al. (2011) also pre-specified when the vehicle will break down and perform a one-time re-optimisation. The authors also simplified their problem from Li et al. (2009b) (2009a) for a unique case problem without time windows.
- Minis et al. (2012) and in their later work (Mamasis et al., 2013) assumed that the breakdown point is in the centre location of the pre-planned route of a pre-specified vehicle and performed their approach accordingly. Also, the breakdown vehicle is manually pre-determined, for which is limited for a full randomised testing which requires an adoption of a simulation approach.
- Monroy-Licht et al. (2017) and Amrouss et al. (2017) considered a single pre-specified breakdown in schedule considered for a non-VRP scheduling problem.
- Van der Merwe et al. (2017) were the first to consider a random selection of vehicle breakdowns. However, it was only limited to one single vehicle at the beginning of its route, and the re-optimisation was done only once due to the adopted approach inability to mimic multiple breakdowns and check the overall customer satisfaction.
- Dávid and Krész (2017) were the only authors to consider multiple breakdowns that could happen over the time horizon of the problem. However, the problem considered is a scheduling problem rather than routing.
- Seyyedhasani and Dvorak (2018) limited their experiment to either adding or removing a pre-specified vehicle when the overall completion of the routes is halfway through completion and implemented their proposed approach only once in this problem.
- Van Lieshout et al. (2018) made their assumption of a breakdown scenario to be also in a pre-specified vehicle upon its 80% completion of the schedule and performed the rescheduling accordingly.
- Guedes and Borenstein (2018) introduced failures at a specific time early in the route and considered up to three simultaneous failures on arbitrarily chosen vehicles and implemented their developed algorithm only once.

- Pandi et al. (2020) considered a DARP case with pre-specified time of the occurrence of the single breakdown.

Although the limitations seen in the above-mentioned critiqued studies, the studies are helpful in initial modelling of the breakdown instant as per only one vehicle that can be extended to multiple random ones and accordingly tailoring a suitable optimisation method for it.

Gap in Knowledge

DVRP with time window and multiple random vehicle breakdowns is not well explored in literature. In general, the routing problem with breakdowns is not a well-explored area; as per a recent review DVRP (Rios et al., 2021), it has been reported that there are only three studies related to the availability of vehicles, one of which (Monroy-Licht et al., 2017) relating to breakdown. All of the dynamic vehicle routing problems explicitly specify when the event of the vehicle disruption will occur and distribute the workload accordingly. However, capturing the randomness of such events will be more practical. Furthermore, all the previous studies only consider one-time re-routing per problem and do not implement their developed approach in continuous time. Therefore, there is a need to investigate multiple random numbers of vehicle breakdowns within a working shift, day, week or any pre-specified period while considering variants such as heterogeneous vehicles. A heterogeneous fleet is not only different in capacities but also unique shifts and locations that are highly applicable for crowd shipping problems (Le et al., 2019). Such investigation has not been considered in any of the DVRP with vehicle dynamism from previous studies. Therefore, the problem may require redesigning the methods and adopting an appropriate modelling approach. As reviewed in this chapter, the agent-based modelling technique would be promising in dynamic problems, given its flexibility in accommodating unique agent structures and adapting and reacting to dynamic events.

2.6 Chapter Summary

This chapter provided an extensive literature review of DVRP cases considering its main two types, dynamic order and vehicle. In both cases(Meta)Heuristics were the widely implemented techniques due to their computational efficiency. Solution approaches were divided into two solution strategies depending on when to optimise the problem, periodic and continuous. Modelling and simulation approaches are usually required to capture the randomness of the dynamic case and act as a framework for the optimisation approach to work. Upon studying the literature, a potential research gap in DVRP that considers multiple random vehicle breakdowns that may occur at any point of time during the routing period and optimise the routes in a continuous matter accordingly.

Chapter 3

Methodology

3.1 Chapter Overview

This chapter presents the agent-based approach to modelling the Dynamic Vehicle Routing Problem (DVRP) with repeatedly/continuous vehicle breakdowns. It highlights the importance of designing agents and their interactions to design an optimisation solution. This chapter proposes the agent-based conceptual model for the problem under study and defines the model's scope with the related Key Performance Indicators (KPI). Two agent interaction approaches have been adopted, hybrid and centralised. The first construct routes based on localised agent rules, and accordingly, the appropriate Messaging Protocol-Based Optimisation (MPHO) is proposed. While the latter performs an extensive search governed by a centralised agent aided with a metaheuristic. Necessary adjustments are made to the problem module to work with two additional optimisation centralised modules, metaheuristic and multi-objective modules.

3.2 Agent-Based Approach Applicability for DVRP

The agent-based approach can be utilised mainly in behavioural and optimisation modelling. Macal (2016) highlighted the differences based on the studied application. Behavioural simulation modelling studies the emergent end behaviours of a particular model or system based on different micro agent interactions. In contrast, optimisation modelling provides certain decisions, such as scheduling or routing, within a system measured through specific criteria. This research problem does not consider the emergent end behaviour; therefore, it considers the optimisation approach of the agent-based to optimise the DVRP.

The optimisation agent-based approach has been previously adapted to DVRP. Kuhn et al. (1994) and Fischer et al. (1996) were the first to use such an approach in DVRP where customers' orders arrive dynamically on a real-time basis. Fischer et al. (1996) also discussed the limitation of traditional operational research methods, such as Linear Programming (LP) or Dynamic Programming (DP), in terms of their agility to respond to dynamic events. Mes et al. (2007) mentioned that the traditional approaches require all information to be available prior to their application, and this may not apply to dynamic problems where solutions are sensitive to the information updates. If the traditional approaches are applied, it will be time-consuming due to their inflexibility and computational expense, justifying the applicability of non-traditional approaches such as the agent-based approach.

3.2.1 The Role of Agents Interaction Design in System Optimisation

The use of the agent-based approach in optimisation is not new. It has already been addressed previously. Barbati et al. (2012) discussed the design of the agents' communications, including interactions and their roles in the optimisation of manufacturing and transportation systems. They mentioned that such communications could be classified into either competitive or cooperative interactions. In competition, agents will act according to their greedy objective, while in collaboration, they will act according to global objective(s). Cooperative interactions can also be categorised into two: distributed and centralised. In the distributed approach, agents govern their behaviours based on their goals' perspective. On the other hand, the centralised approach adds an extra agent, dubbed a manager or mediator, that supervises and governs certain aspects of the sub-agents interactions to benefit the global objectives. Centralisation in agent-based is introduced to overcome the challenges in achieving the global objective(s) given the original distributed nature of the approach (Monostori et al., 2006).

Monostori et al. (2006) explained the centralised/distributed coordination mechanisms by elaborating more on the levels or degrees of the centralisation. They classified the coordination mechanism into three levels: centralised, decentralised and hybrid. In the centralised approach, the manager or super agent instructs all of its sub-agent tasks to accommodate a global objective. In contrast, in the decentralised approach, all agents interact and emerge into a solution. However, the approach does not assure the benefit of the agents as a whole. On the other hand, a hybrid approach considers both by defining specific responsibilities for a manager agent to guide/select sub-agents in their task execution.

In Figure 3.1, the type of interactions adopted in agent-based optimisations are summarised. Choosing the right interaction design is vital for a problem under study. Generally, the research problem seeks a careful allocation of customers to a vehicle; as a result, agent

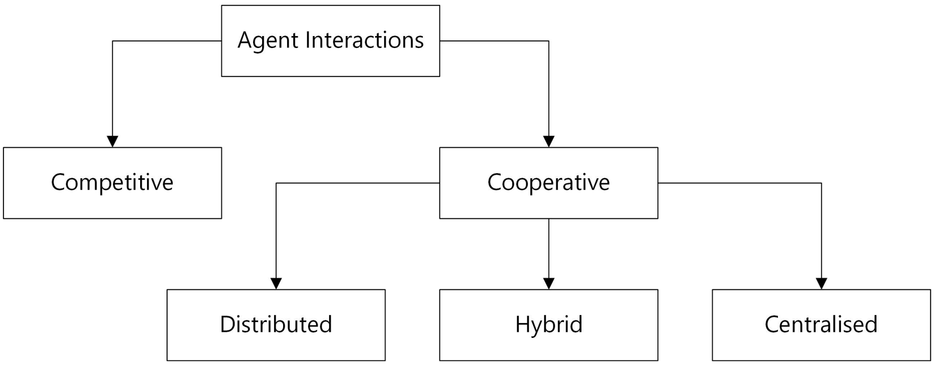


Fig. 3.1 Types of Agent Interactions in Optimisation

competitiveness should be eliminated as it will result in so many conflicting objectives. Consequently, the obvious choice is to adopt a cooperative approach; however, choosing any of the three types of cooperation has its benefits and drawbacks.

3.2.2 Cooperative Approaches Trade-off

Davidsson et al. (2007) compared the classical optimisation techniques that are centralised to agent-based approaches given their distributed nature. In their comparison, they neglected specific centralisation features in agent-based for the sake of the study. They compared both approaches for nine properties, six of which are critically related to optimisation. As highlighted by Barbati et al. (2012), they are the problem size, solution quality, solution time, computational complexity, adaptability and modularity. On the other hand, the remaining comparison properties are related to networking cost, reliability and security that are not relevant to the course or scope of this research.

The decentralised way of agent-based provides a localised solutions' decision calculation at the individual agent level. This benefits large-sized problems by reducing its solution time through its ability to modularise and divide the problem into sub-problems. Moreover, its ability to flexibly adapt and change its state by adding/eliminating agents based on a dynamic situation with a significant drawback of not assuring the optimality of the solution quality compared to traditional centralised optimisation approaches (Davidsson et al., 2007). Figure 3.2 illustrates the trade-off between the centralised and the distributed approaches.

By reflecting on the research problem, a good solution approach is considered adaptable to dynamic events and minimum run time while considering specific global objectives that need to be met, at least in a near-optimal way. A fully decentralised approach has not been seen in DVRP; as a result, a hybrid approach in cooperation is most favourable while also considering a centralised approach for comparison.

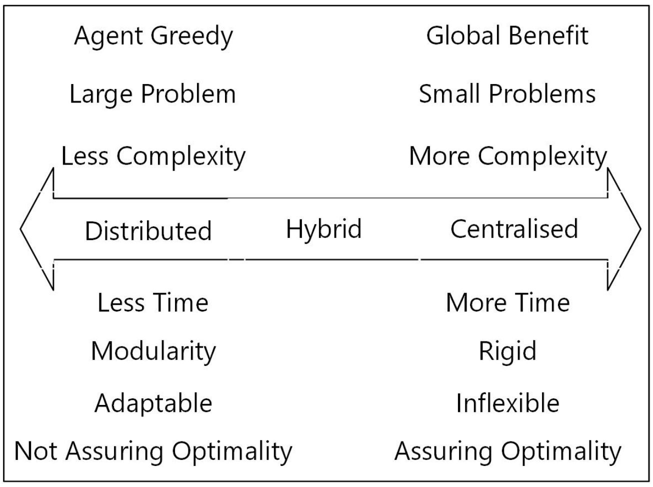


Fig. 3.2 Distributed-Centralised Trade off

3.3 The Agent-Based Conceptual Model

Before adopting any agent interaction strategies, it is essential to define the agent-based model and its architecture along with its inputs and resulting outcomes to define the scope of the model's process. Figure 3.3 shows all the given inputs categorised in terms of customers, vehicles and the resulted outputs.

The input data for each customer consist of a unique identification number, specific location, time window at which the customer is available to be served, service time needed per visit and service type, whether it is a delivery or collection. On the other hand, the vehicle inputs define the different attributes of the resources on hand, including vehicle capacity, its current location, home location or depot where it should end its route, its availability provided by its operating shift and the maximum duration it can operate.

The agent-based module comprises the assignment agent, customer agent and vehicle agent. The assignment agent is designed to centrally control the search for an optimal solution. In this study, it is designed to work in a hybrid and centralised manner by dictating the type of interactions among the other agents given specific global objectives. On the other hand, the customer agent initiates requests and evaluates the responses from the vehicle agents. The vehicle agent performs specific optimisation tasks by conducting local feasibility evaluations. This module can be applied to both static and dynamic VRP. Given the aim of this work to accommodate vehicle breakdowns (BD), a randomised breakdown event can face a vehicle agent, forcing it to report its unavailability to the assignment agent in order to perform the necessary re-routing, as highlighted in red in Figure 3.3. In a static problem, inputs are typically provided, and optimal routes will be produced accordingly. On the other

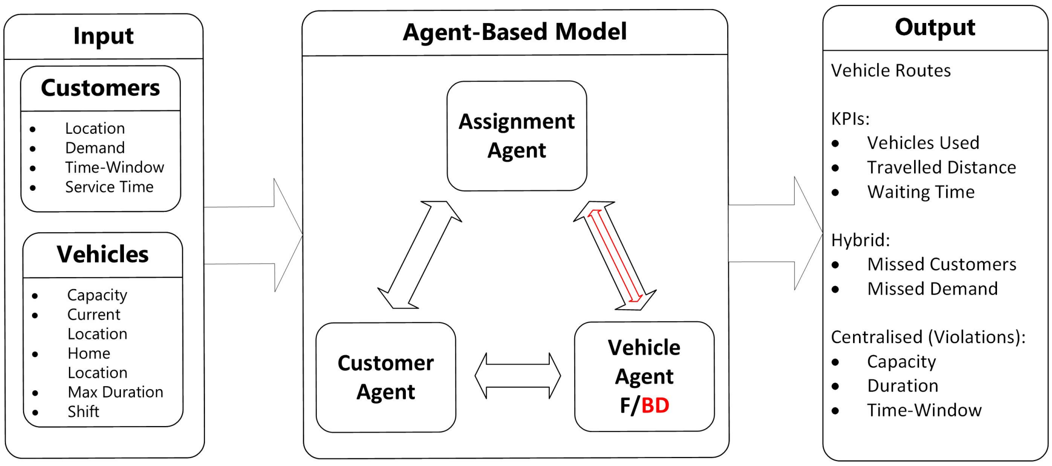


Fig. 3.3 Model Inputs and Outputs

hand, when a breakdown occurs, the inputs of the problem are provided in a similar structure; however, with updated data as time has passed and specific customers have been already served. The module does not apply only to the disrupted/affected customers but also to all not served customers, which neglects any previous routes to ensure the possibility of exploring better alternatives. This module would be beneficial for the studied problem as it flexibly accommodates the updated list of agents, especially resources or vehicles, given a particular disruption, considering the different vehicle attributes such as locations, home locations, shifts and current capacities.

The primary model outcome is to generate feasible routes by allocating customers to vehicles and their visit sequence by each vehicle. The model will also generate specific Key Performance Indicators (KPIs) to evaluate the generated routes' performance. Those KPIs are dependent on the adopted agents' interactions in optimisation, hybrid and centralised, with three common.

Different KPIs are generated based on the interaction due to different constraints handling strategies by deciding which of the constraints to violate its feasible domain. Every constraint has a specific feasible domain. The number of vehicles used and the customers served should not exceed the available vehicles and customers, respectively. Allocations of customers to a vehicle should not exceed the vehicle capacity and duration constraints. Finally, arrival time at a customer should not exceed its late time window constraint. The hybrid relaxes the coverage of customers while the centralised relaxes the vehicle's capacity constraints and durations as well as customers' late time window. In this kind of NP-hard problems, violation in one or more constraints may be needed to further investigate solution space for efficient solution emergence. The KPIs studied are as follows:

- Main KPIs:
  - The total number of vehicles used, as it is unnecessary to use all available vehicles, the minimum used the better.
  - Total travelled distance which is the summation of all the distances travelled by each vehicle given their travelled routes.
  - Total waiting time is the summation of all the waiting time of each vehicle given if it arrived before the opening time window of the customer.
- Hybrid KPIs:
  - Total customers missed (coverage) given the relaxation of the coverage in the hybrid approach
  - The total missed demanded quantity from the unmet customers.
- Centralised KPIs:
  - Capacity violations as vehicles may exceed their capacity with mathematical penalties.
  - Duration violations as vehicles may exceed their duration limit with mathematical penalties.
  - Time window violations as vehicles may arrive late at the customer location and its home depot, given its shift, with mathematical penalties.

Besides their different constraints' relaxation, the hybrid and the centralised approaches also differ in their solution evaluations and degree of centralisation. The hybrid approach evaluates solutions locally at the vehicle agent level while the centralised performs it globally at the assignment agent level. With respect to their degree of centralisation, the centralised approach performs an extensive routes evaluations and alterations while the hybrid is only limited to simple customers prioritisation and sorting. The key differences between the approaches are summarised in Table 3.1.

Table 3.1 Differences between the Hybrid and the Centralised Approaches

	Hybrid Approach	Centralised Approach
Relaxed Constraints	Customers coverage	Capacity, duration and time window
Solution Evaluation	At the vehicle agent level	At the assignment agent level
Degree of Centralisation	Limited to customers prioritisation	Global routes evaluation and variation

3.4 The Hybrid Approach

The hybrid cooperative approach, as agent interactions, requires a degree of centralisation. This can be found in mediator architectures presented by Barbati et al. (2012). Such architectures provide a level of tracking of global objectives through certain negotiation protocols with a mediator or manager agent. Good examples of such cooperation protocols was presented by Mes et al. (2007) and Martin et al. (2016). The first example adopted an auction mechanism where a requester agent issues a bid while resource agents evaluate the bid and make an offer, and then the requester agent chooses the best. The second example provides a similar interaction; however, the agents here represent metaheuristics that exchange specific moves rather than order-resource agents that exchange cost information. In this section, the approach of the first example is adopted as evidenced in VRP studies that utilise the agent-based approach specifically, see (Barbucha, 2016) (Kalina and Vokřínek, 2012). The messaging protocol for such cooperative protocol has been standardised and dubbed Agent Communication Language (ACL) (FIPA, 2000).

3.4.1 The Messaging Protocol-Based Heuristic Optimisation

This section presents the proposed messaging model to achieve sequential routes construction that aims to route every possible customer and allocate them to the available vehicle resources. The decentralisation in this model comes from the optimisation objectives that are evaluated at the vehicle agent level. However, the messaging model is not entirely decentralised as specific priority rules are applied globally at the assignment agent level. As a result, a Messaging Protocol-Based Heuristic Optimisation (MPHO) has been proposed. This protocol is believed to be less computationally expensive due to its high level of decentralisation, which makes it suitable for the dynamic problem, as evidenced in the next chapter in sections 4.3 4.4.

The MPHO model solves the DVRP with vehicle disruption/breakdown, based on Solomon's Time-Windows Push Forward feasibility checking as well as the insertion method

(Solomon, 1987) that accommodates, implicitly, maximising the coverage of every customer under the minimisation the number of vehicles and total distances. However, in the proposed model, the insertion heuristic is extended to agent negotiation-based optimisation to accommodate more priority rules. The priority rules are not limited to only prioritising customers based on their distance or earliest time window. Given that at the breakdown instant, there is no central depot for customer distances to be compared to all vehicles' locations as a whole. With such a negotiation approach, the disruption problem can be optimised given the various attributes of each agent, customer and vehicle.

At the start of the MPHO presented in Figure 3.4, the assignment agent initiates the routes construction process by sorting customers based on the priority rules where additional rules have been adapted to DVRP compared to Solomon's as vehicles have different locations and home locations. A seed customer is then selected with the privilege to initiate a vehicle's route; therefore, this stage of routing is dubbed as **Priority Routing** phase. The seed customer will issue requests to all vehicle agents with empty routes; accordingly, each vehicle agent performs feasibility evaluation, detailed in subsection 3.4.2, to check for any constraints violation. The vehicle agents then return their evaluations with costs to the seed, and the latter selects the nearest vehicle if provided a feasible solution. However, if no feasible solution is provided, the seed customer is considered as missed. If the seed customer selects a vehicle agent, the latter assigns the customer in its route and notify back to both assignment and customer agents.

However, prioritising every customer will not be efficient in minimising the vehicles; therefore, the **Non-Priority Routing** phase is followed after a seed customer is allocated to a vehicle agent to possibly route the remaining customers. It starts by providing a sorted list of all the unrouted customers, based on the priority rule, from the assignment agent to the vehicle agent that has its route recently initiated by the seed customer, to consider routing them as much as possible without any constraints' violations. Then, by looping through this unrouted list of customer agents, each customer attribute is requested by the vehicle agent to be feasibly evaluated as in 3.4.2 and routed if proven feasible. If a customer is routed, its agent is notified and the assignment agent. However, if the vehicle has been exhausted with no more customers can be considered, the vehicle agent returns the remaining unrouted customer to the assignment agent. Accordingly, the assignment agent repeats the priority cycle for the unrouted customers and selects another seed again until all vehicle agents are exhausted.

To further accommodate this messaging to dynamic breakdown in a delivery problem, where the breakdown instant can be seen as a pickup and delivery problem, the evaluation of a customer insertion request can be divided into two evaluation steps: the pickup first

Fig. 3.4 Messaging Protocol-Based Heuristic Optimisation Model (MPHO) (Abu-Monshar et al., 2022)

followed by delivery. Similar implementations can be seen in Kohout and Erol (1999) and Hosny and Mumford (2010). The pickup node is created artificially at a disrupted vehicle location with the same customer time window, a negative demand and zero service time (assumed). It is mandatory to follow up with a delivery node, the actual customer location, in the same route.

The priority rules for sorting the unrouted customer agents by the assignment agent can be based on three scenarios:

- Earliest Deadline: to prioritise customers based on their late time window
- Furthest Minimum Distance: to prioritise customers based on their furthest minimum distance from all vehicles
- Furthest Average Distance: to prioritise customers based on their furthest average distance from all vehicles

The earliest deadline priority will prioritise a customer based on its time window. For example, if a customer i has a time window of (e_i, l_i) then a priority customer u is selected given l_u, its late time window, is minimum. On the other hand, distance priority has two measures because vehicles are not associated with a depot to start/end their routes. One of the proposed measures is to check each customer's distance from all vehicles, and the minimum for every customer is recorded; as a result, a customer with the recorded highest minimum distance value is prioritised first. Similarly, instead of selecting the minimum, the furthest average distance measure averages all distances across all vehicles for a particular customer.

It can be seen that the MPHO model as a whole is an adaptation of Solomon's insertion heuristic given its sequential routes construction; as a result, it is expected that the computational efficiency is similar to a heuristic algorithm.

3.4.2 Vehicle Agent Evaluation

When a vehicle agent is directed to evaluate a customer agent request to consider in its route, the vehicle agent firstly checks if the customer demand and time window are within its capacity and operating shift constraints according to the following conditions:

$$Capacity: \quad q_i + Q_{v\ cur} <= Q_v \tag{3.1}$$

$$Time: \quad e_v <= e_i \quad l_i <= l_v \tag{3.2}$$

where q_i is customer i demanded quantity, $Q_{v\ cur}$ is the current occupied capacity of vehicle v, Q_v is the total capacity of the vehicle while (e_i, l_i) and (e_v, l_v) are customer i time-window and the vehicle operating shift, respectively. If both conditions are satisfied, further checks are sought by the vehicle agent to seek the best position to insert the customer within its route. The insertion utilises an adapted Solomon's insertion with its Push Forward (*PF*) technique due to its high effectiveness in the time window problems (Solomon, 1987). The *PF* method is used when a customer is inserted to check later customers' time window feasibility given the updated times due to the insertion while also tracking local objectives of distance saving and delay (urgency) in the next customer. This *PF* technique has been tailored further to be utilised in the feasibility evaluation within a vehicle agent to consider its unique attributes, different shifts and unique start/ending route locations.

The calculation of *PF* is done in sequence for every customer in route after the insertion as shown in Equation 3.3 while taking into consideration the new arrival time at customer i

to be less than its late time window as seen in Equation 3.4.

$$PF_i = b_i^{new} - b_i \tag{3.3}$$

$$b_i^{new} <= l_i \tag{3.4}$$

where b_i^{new}, b_i are the new and original arrival time at customer i next in route, respectively. Given that customers next in route depend on this change, PF is recursively implemented as represented in Algorithm 1. w_i is the waiting time for customer i.

Algorithm 1: Push Forward Recursive

```
Data: customer i, Previous PF_{i-1}
Result: feasible or not
PF_i = max(0, PF_{i-1} - w_i);
if PF_i = 0 then
    stop, feasible;
else
    if b_i + PF_i > l_i then
        stop, violation;
    else
        if customer i is Home Location then
            stop, feasible;
        else
            Push Forward Recursive (next customer i, PF_i);
        end
    end
end
```

The parameters provided to Algorithm 1 are the next customer i in route and the previously calculated PF to check the time window feasibility of customer i. If the last customer i in route happens to be the last visit location (vehicle home depot), then the recursion stops. Furthermore, if the recursive function resulted in any violation in time windows, it stops and indicates the infeasibility of the insertion.

Although that capacity and vehicle shifts are assessed in Equations 3.1 and 3.2, as well as Algorithm 1, checks the time window feasibility of the insertion, such methods do not yet evaluate the cost or saving of a customer insertion in a specific position in comparison to other positions in route. Therefore, the saving function is introduced in Equation 3.5 to allow each vehicle agent to assess the customer u insertion position through to select a position with maximum saving distance value.

$$\frac{\lambda}{2}(d_{vu}+d_{uh})-\alpha_1(d_{vu}+d_{uh}-\mu d_{vh})-\alpha_2(b_{j_u}-b_j) \tag{3.5}$$

$$\lambda,\mu >= 0 \tag{3.6}$$

$$\alpha_1+\alpha_2=1 \quad \alpha_1,\alpha_2 >= 0 \tag{3.7}$$

where λ, μ, α_1 and α_2 are non-negative parameters while d_{vu}, d_{uh} and d_{vh} are distances between vehicle to customer, customer to vehicle's home location and vehicle and its home location, respectively. b_{j_u} is the the new arrival time at customer j when customer u is inserted between i and j.

The first term in Equation 3.5 is an adaptation from Solomon, originally λd_{vu}, to accommodate the problem under study as vehicles may have different starting and ending locations. As a result, the λ parameter is halved, and the distances from vehicle v to the inserted customer u and from customer u to home depot h are added. Contrary to previous Solomon's implementation that assumes these two distances are equal given the start and end locations are the same. The remaining second and third terms of the equation are the distance saving from Clarke and Wright (1964) and the urgency of pushing forward of customer j, respectively, that are weighted and normalised by α_1 and α_2 parameters.

However, the traditional Solomon's insertion method assumes that all vehicles start their routes at the moment their shift starts. As a result, this would lead to unnecessary waiting times, especially for the first customer to visit in a vehicle's route (Chiu et al., 2006) which may increase the total time in route, including the waiting time, of each vehicle and may exhaust the maximum duration constraint. The Solomon benchmarks are without such constraint; however, in multiple depots benchmarks with time-window by Cordeau et al. (2001) the constraint was introduced. Therefore, eliminating such unnecessary waiting times becomes essential. The proposed MPHO method overcomes this issue by calculating the waiting times sequentially for every node in the route starting from the vehicle. Firstly, the departure time from the vehicle's current location to the first customer in route is calculated t_{v_i} as shown in Equation 3.8, where $i=1$ indicating the time needed to visit the first customer in route. The departure time should be either zero or the difference between the first customer's early time window e_i and the travel time, whichever is maximum, to eliminate the waiting time for the first customer.

$$dep_v = max(0, e_i - t_{vi}) \quad i = 1 \tag{3.8}$$

Next, the arrival time at every customer is calculated given the previous node departure time, servicing time and travel time. The previous node could be a previous customer in route or the vehicle's initial location. In the case of the latter, then the servicing time is set to zero. The calculation of the arrival time is shown in Equation 3.9, where dep_{i-1} and t_{i-1i} are the previous node departure time and the travel time, respectively. Finally, with Equation 3.10, the waiting time at customer i can be calculated, which is the difference between the arrival time and the early time window e_i or zero if the difference results in a negative value.

$$b_i = dep_{i-1} + t_{i-1i} \tag{3.9}$$

$$w_i = max(0, e_i - b_i) \tag{3.10}$$

The total waiting time for a route can be calculated by firstly calculating every customer in route waiting time. Algorithm 2 is developed to calculate the total waiting time W_{v_u} for a vehicle v given the insertion of customer u in its route.

Algorithm 2: Calculating Total Waiting Time

Data: customer u, position (i, j)
Result: route total waiting time W_{v_u}
Add customer u between position (i, j);
Calculate departure time dep_v from the vehicle;
while *remaining customers in route* **do**
    Calculate arrival time b_i for customer i;
    Calculate waiting time w_i for customer i;
    Add w_i to the total W_{v_u};
end

Calculating the total waiting time of a particular route makes it possible to check the duration constraint feasibility given a particular insertion. Route duration consists of the travel, servicing and waiting times. When a customer u is inserted between i and j, additional travel and servicing time should be added while waiting time has to be recalculated as previously shown in Algorithm 2. Additional travelling time can be calculated similarly to the distance saving from Clarke and Wright (1964) but with times instead, represented in

Equation 3.11.

$$\Delta t_{u\ ij} = t_{iu} + t_{uj} - t_{ij} \tag{3.11}$$

where t_{iu}, t_{uj} and t_{ij} are the travel times between customers (i, u), (u, j) and (i, j), respectively. The time-saving equation adds the extra travel times due to the inserted node while eliminating the previously defined time between (i, j). Given that all the elements of change in route duration are now calculated, the new route duration can be calculated, as in Equation 3.12 given the previous route total travel T_v and serving times S_v.

$$dur_{v_u} = (T_v + \Delta t_u) + (S_v + s_u) + W_{v_u} \tag{3.12}$$

The equation adds the differences in time Δt, servicing time of customer u (s_u) to the original total time T_v and servicing time S_v, respectively, in the first and second terms of the equation. In contrast, the last term is the total waiting time calculated according to Algorithm

Finally, the duration constraint can be checked by comparing the new duration time of the route dur_{v_u}, given the insertion of customer u, against the duration limit of vehicle v as seen in Equation 3.13, where $dur_{v\ max}$ is its maximum duration.

2.

$$dur_{v_u} <= dur_{v\ max} \tag{3.13}$$

3.5 The Centralised Approach

In the hybrid approach, the proposed vehicle agent evaluation, shown in subsection 3.4.2, provides a method for objective calculation; however, such evaluation is localised from the vehicle agent perspective. Therefore, the assignment agent and priority rules are proposed to overcome local greediness by governing agents' cooperation. However, such a technique does not calculate any global objectives, such as minimising the number of vehicles used, distance or waiting times to direct the search accordingly. In other words, it is not flexible enough to tackle the specific optimisation criteria and control the search in routes' construction or alteration. Therefore, the assignment agent needs to be equipped with the necessary techniques to calculate such objectives and govern the cooperation for strategic route building and alteration.

Due to its rigid algorithmic nature, heuristics are inflexible in tackling the combinatorial optimisation problem, such as VRP. As a result, they have evolved into metaheuristics in order to provide a suitable algorithmic framework(Glover, 1986). However, challenges

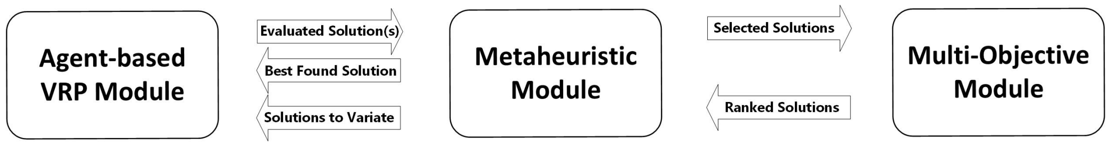


Fig. 3.5 Metaheuristic-Based Modules Workflow

arise in applying such algorithmic frameworks to the proposed agent-based model of the studied DVRP. Talbi (2009) implied that, when designing and implementing a metaheuristic, it is essential to differentiate between problem-dependent and metaheuristic-specific components to design a reusable code. Problem-dependent components include the solution representation design, evaluation and variation, while metaheuristic-specific components consist of solution selection criteria, replacement and the algorithm stopping conditions. In addition, as categorised by Talbi (2009), multi-objective components can be considered in the implementation by considering a specific fitness assignment and solution preservation techniques.

Initial problem components have already been proposed in the original agent-based module, shown in Figure 3.3, by proposing the agents and their constraints as well as solution evaluation through a decentralised vehicle agent evaluation, discussed in subsection 3.4.2, however, such components are still incomplete for implementation within a metaheuristic framework. Therefore, further problem components are sought in this section. In addition, metaheuristic-specific and multi-objective components are yet to be initiated and proposed independently of the original agent-based module.

Given the new structure of implementing the centralised approach using a metaheuristic, two additional core modules are proposed, the Metaheuristic and Multi-Objective Modules. At the same time, further developments are sought in the original agent-based VRP module. The communication among these core modules is illustrated in Figure 3.5.

The process starts in the agent-based module, where solutions are represented, evaluated and altered, and sends evaluated solution(s) to the metaheuristic framework to perform a global search. Next, selected solutions are passed to the Multi-Objective module to be ranked using a Pareto non-dominance sorting. After sorting, solutions are returned to the metaheuristic module to decide further to either request more solution variations from the problem module or return the fittest solution found. The latter is decided if the stopping condition is satisfied. In the following subsections, additional development to the agent-based module are shown in subsection 3.5.1 and the metaheuristic framework and adaptation to the problem is explained in subsection 3.5.2. Finally, the multi-objective components are detailed in subsection 3.5.3.

3.5.1 The Agent-based Module

Referring to the agent-based module in the conceptual model shown in Figure 3.3, the agent interactions, in a centralised approach context, will be centred around the assignment agent to govern global interactions between the agents for the optimisation process to diverge away from the local optima. This form of agent interactions is explained later in 3.5.1.2, where messages are exchanged between the agents and the necessary local evaluations are performed. This section aims to utilise a metaheuristic framework to aid in the search. Therefore, the agent-based module must adapt to accommodate the necessary metaheuristic problem-dependent Representation, Evaluation, and Variation components.

3.5.1.1 Representation

The earliest VRP formulation (Dantzig and Ramser, 1959) was in Integer Linear Programming (ILP), where a binary variable represents a connection between two nodes (edge/arc) which was later dubbed as "edge representation". However, exact methods for such NP-hard problems are proven impractical for large-sized problems (Laporte, 2009). Therefore, solution methods have evolved to heuristics and approximation methods (Lenstra and Kan, 1981). As a result, classical mathematical models, such as ILP, have been challenged, and new techniques have emerged, such as the saving (Clarke and Wright, 1964), sweep (Gillett and Miller, 1974) and the two-phase (Christofides et al., 1979) heuristics. However, only the first heuristic kept the edge representation. At the same time, the rest started to use ordered customers' permutations as routes to implement the heuristics procedures, thus shifting to "path representation", which originated from early studies in solving the Travelling Salesman Problem (TSP) using genetic operators (Michalewicz, 1996).

Such representations have been extended to metaheuristics in VRP such as Tabu Search (TS) (Rochat and Semet, 1994) as well as Genetic Algorithm (GA) (Maeda et al., 1999). According to Talbi (2009), a solution representation in a metaheuristic should be complete in representing all possible solutions, efficient in its alteration and manipulation as well as being connective, meaning every two solutions must have a search path between them. In the path representation, it can be deduced that any customer permutation that includes all customers can be considered a solution, regardless of its feasibility, as some constraints can be relaxed.

However, dealing with multiple vehicles with different attributes and locations requires adaptations to such representation. This adaptation can overcome this issue by specifying routes for every vehicle agent with its unique starting and home location, therefore, omitting the depot representation, as it will be implicitly known from its predefined attributes. As a

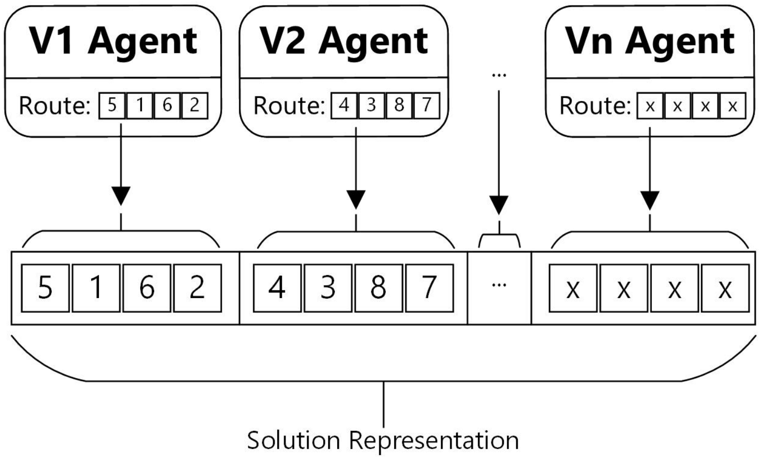


Fig. 3.6 Agent-based Path Solution Representation

result, the agent-based solution representation is shown in Figure 3.6, where vehicle agents (V1, V2, ..., Vn) are provided with route attributes of sequences of customers, given that all customers are sequenced and appear only once per solution.

3.5.1.2 Evaluation

Evaluating solutions in routing problems, TSP specifically, is usually aimed at measuring only the total distance by taking the summation of all the distances between nodes in a committed permutation, and the objective is to minimise it (Larrañaga et al., 1999). Since VRP is a generalisation of a TSP, it inherited this total distance evaluation (Bektas, 2006); nevertheless, additional objectives have also been taken into consideration, such as minimising the vehicles used (Ghoseiri and Ghannadpour, 2010) and waiting time (Chiu et al., 2006). Consequently, multi-objective methods have been proposed to solve the problem (Ombuki et al., 2006). Furthermore, the generalised problem incorporated additional constraints such as capacity, customer time window and route duration limit. As a result, solution evaluation in metaheuristics had to be adapted to check solution feasibility with respect to these additional constraints. However, this has resulted in a highly constrained problem that increased the difficulty of exploring the feasible region (Cordeau et al., 2001). Consequently, solution evaluation in metaheuristic approaches for the problem has been adapted, and two different constraints handling strategies have emerged, the hard and relaxed constraints strategies.

In the proposed centralised approach, evaluation occurs mainly at the vehicle agent level where a route is provided, and accordingly, checks are performed against the vehicle constraints. Every customer, in route, information is requested, including the time window, demand, service time, and location to be included in the localised vehicle agent evaluation.

Accordingly, local route measures such as distance, waiting time, and constraint violations are determined. However, global measures such as the total of these measures and the number of vehicles used are determined globally at the assignment agent level to perform a more centralised assessment on the evaluations returned from all vehicle agents for a particular solution.

Figure 3.7 illustrates the centralised agent messaging for evaluating a solution. It starts with the assignment agent providing a potential route for a particular vehicle agent to check its feasibility with respect to the vehicle's and customer's constraints. Accordingly, the vehicle agent requests from every customer in route with their attributes to be included in its localised feasibility evaluation and then returns the evaluation to the assignment agent. In the end, the assignment agent performs a global evaluation of all routes across all vehicle agents.

In this work, the objectives considered are:

- Minimisation of the total distance: $min \ \sum_{v_r=1}^{V_r} D_{v_r}$
- Minimisation of the total waiting time: $min \ \sum_{v_r=1}^{V_r} W_{v_r}$
- Minimisation of the number of vehicles used: $min \ \sum_{v_r=1}^{V_r} y_{v_r}$

where V_r is a set of different routes provided to each vehicle while The subscript v_r indicates a selected route r provided to vehicle v from the set V_r. D_{v_r} and W_{v_r} are the total distance and waiting time for every route r provided to vehicle v while y_{v_r} indicates whether vehicle v is idle (0) or utilised (1) if route r is provided. A vehicle route's distance and waiting time measures are calculated at the vehicle agent level. Then, the summation of the distances and times are calculated at the assignment agent level. The measure of the number of vehicles used can be implicitly determined from the route provided to the vehicle; if empty, then the vehicle is not utilised; otherwise, it is. If an empty route is provided to a vehicle agent, distance and time measures are null.

Concerning constraint handling, Talbi (2009) has specified different strategies for problems solved using metaheuristics, two of which are used in VRPTW problems: reject and penalised strategy. The first resembles a hard constraint implementation, meaning if a constraint violation appears in a particular solution of a problem, the solution is rejected and discarded. On the other hand, the penalised strategy is more pragmatic that explores such infeasibility by applying a quantified penalty on the resulted amount of violation in a particular solution and adding to the overall objective function.

VRP started as a hard constraint problem; however, when more additional constraints have been introduced, especially time window constraints, solution approaches tended to shift to this relaxed strategy of handling constraints to explore the infeasible regions. This

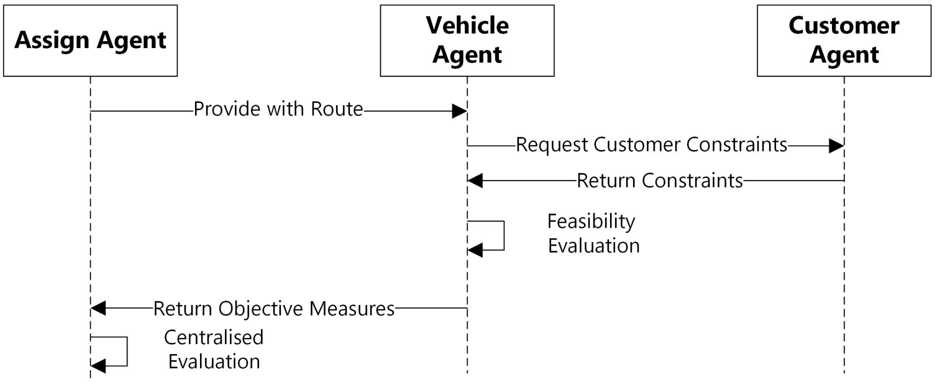


Fig. 3.7 Centralised Evaluation

is seen in Gendreau et al. (1994) where they relaxed vehicle's capacity and route duration constraints, Rochat and Semet (1994) and Cordeau et al. (2001) further relaxed the time-window constraint by penalising late arrivals, and finally Taillard et al. (1997) only relaxed the time-window constraint. As a result, the penalised strategy is adopted for the time window variant under study.

It is vital to choose the correct penalty values in the penalised strategy. If they are too low, the final solution might be infeasible, or if they are too high, the final solution might be costly. Therefore, Talbi (2009) indicated three strategies for applying penalties: static, dynamic and adaptive. In static, penalties are fixed beforehand and remain constant during the algorithmic implementation. At the same time, the dynamic adjusts these penalties based on the iteration allowing for early iterations with lower penalties to consider none feasible solutions, then increasing these penalties iteratively to encourage finding non-violating solutions towards the end. On the other hand, the adaptive strategy incorporates solution information in adjusting these penalties by checking previous iteration solutions concerning each constraint. A penalty is adjusted based on its constraints violations in previous solutions, increased when violated while reduced when feasible. Referring to the VRPTW problem, static and adaptive are used when adopting the penalised strategy; in most cases, the adaptive while static is only seen in Taillard et al. (1997).

In this work, the penalised constraints are capacity, route duration and time-window constraints. When evaluating a route provided to a vehicle agent, it can accept any route with these three constraints violations while measuring the degree of violation for each of the constraints as per the following equations:

$$VQ_{v_r} = max(0, Q_{v_r} - Q_v) \tag{3.14}$$

$$Vdur_{v_r} = max(0, dur_{v_r} - dur_v) \tag{3.15}$$

$$VTW_{v_r} = \sum_{i \in v_r} (max(0, b_i - l_i)) \tag{3.16}$$

where VQ_{v_r}, $Vdur_{v_r}$ and VTW_{v_r} are violations for capacity, duration and time-window constraints, respectively, for a route r provided to vehicle v. Q_{v_r} is the occupied capacity and dur_{v_r} is the resulted duration, each for for route v_r. Since these measures are localised at the vehicle agent measure, the total violations for each constraint are then calculated by the assignment agent for a particular solution of routes V_r as per the following equations:

$$VQ_{V_r} = \sum_{v_r \in V_r} VQ_{v_r} \tag{3.17}$$

$$Vdur_{V_r} = \sum_{v_r \in V_r} Vdur_{v_r} \tag{3.18}$$

$$VTW_{V_r} = \sum_{v_r \in V_r} VTW_{v_r} \tag{3.19}$$

where VQ_{V_r}, $Vdur_{V_r}$ and VTW_{V_r} are total violations for each capacity, duration and time windows for a particular solution set V_r. In order to fully apply the penalised strategy, penalties should be multiplied for every total violation to influence the algorithmic evaluation of a particular solution V_r. The following equation represents the total violation for a solution V_r:

$$P_{V_r} = P_Q \times VQ_{V_r} + P_{dur} \times Vdur_{V_r} + P_{TW} \times VTW_{v_r} \tag{3.20}$$

where P_Q, P_{dur} and P_{TW} are non-negative parameters representing penalties for each of the capacity, route duration and time-window constraints, respectively. P_{V_r} is the resulted penalty value of the solution V_r. In order to overcome the hassle of setting values for these penalty parameters, Gendreau et al. (1994) proposed an adaptive technique by checking certain previous solutions if they have violated these constraints. They specify an integer parameter h for which the metaheuristic checks the previous h solutions, every h iteration or generation, for violation for each of the relaxed constraints and update the penalties accordingly. If all the previous h solutions violated a constraint, the corresponding penalty is doubled; otherwise, it is halved. However, if the previous h solutions are mixed with feasible and infeasible solutions, the corresponding penalties remain the same. Rochat and Semet (1994) further randomised the update by multiplying or dividing the penalties by a random factor γ, between 1.5 and 2.0. The resulted adaptive penalty when solutions do not violate the respected constraint is shown as follows:

$$P_Q := P_Q/\gamma \tag{3.21}$$

$$P_{dur} := P_{dur}/\gamma \tag{3.22}$$

$$P_{TW} := P_{TW}/\gamma \tag{3.23}$$

While the resulted adaptive penalty when solutions violate the respected constraint is shown below:

$$P_Q := P_Q \times \gamma \tag{3.24}$$

$$P_{dur} := P_{dur} \times \gamma \tag{3.25}$$

$$P_{TW} := P_{TW} \times \gamma \tag{3.26}$$

Given the provided objectives and constraints evaluation, the overall utility or objective function is still needed to represent an evaluation of a particular solution that combines the objectives and the penalties of the relaxed constraints. Since no multiple objective methods have been considered yet, the single objective function $F_{obj\ V_r}$ is assumed to be based on only one of the provided objectives for a particular solution set V_r. Accordingly, the overall utility function F_{V_r} will be the summation of the objective and the resulted solution penalties as shown below:

$$F_{V_r} = F_{obj\ V_r} + P_{V_r} \tag{3.27}$$

3.5.1.3 Variation

Altering solutions that are path represented in metaheuristic frameworks are dependent on the type of the framework used, single solution-based or population-based. Single solution-based variations originate from traditional local improvement heuristics in TSP that are used to improve a single route (intra-route). At the same time, also newer techniques have been developed to improve multi-route (inter-route) solutions (Laporte and Semet, 2002). Intra-route improvements originate from TSP edge exchanges λ-opt, where λ is the number of edges connected between nodes of a current solution. However, they have been later adapted to the exchange of nodes as seen in Or-opt with up to three-node relocations (Or, 1976) which makes it applicable to the adopted path representation. This work adopts the Or-opt; however, the three-node limit is lifted, and the selection of nodes to be relocated is random based on the adopted metaheuristic. Unlike TSP single route improvements, VRP needs improvement moves involving more than one route. Such moves are exchanges of vertices or edges between two routes, where the routes and vertices are selected mostly at random, depending on the metaheuristic used. Since this work is focused on the path representation, inter-route moves studied here are related to the exchange of vertices. The adopted inter-route move in this study is the CROSS exchange, where a specific sequence of customers from two routes are exchanged, originally proposed by Taillard et al. (1997).

On the other hand, various solution alteration methods have been implemented in population-based metaheuristics. For example, evolutionary Algorithms such as Genetic Algorithms (GA) have been proven their applicability in solving complex problems, particularly routing such as TSP and VRP (Goldberg, 1989). However, Michalewicz (1996) highlights the problem when altering solutions in such population-based search algorithms and applying the original recombination operators, particularly crossovers, to routing problems, they would result in duplicates and omissions of vertices in path-represented solutions. Therefore, it contradicts the problem constraint that each city must be visited only once. As a result, modifications to such operators are essential, making them routing specific operators. Consequently, unique operators for TSP in path representation have been developed, such as partially-mapped (PMX), order (OX) and cycle (CX) crossovers (Michalewicz, 1996).

The population-based variation process in routing problems is further complicated when time window constraint is introduced. Potvin and Bengio (1996), being one of the earliest VRPTW studies in GA. They designed two types of crossovers, Sequence-Based (SBX) and Route-Based (RBX) crossovers, with route repair procedures that resolve missing vertices by reinserting them at a feasible insertion position. Insertion aims to minimise additional cost and resolve redundancies by simply removing old repeated vertices. Later, Ombuki et al. (2002) proposed a different recombination operator, Route Crossover (RC), that generates

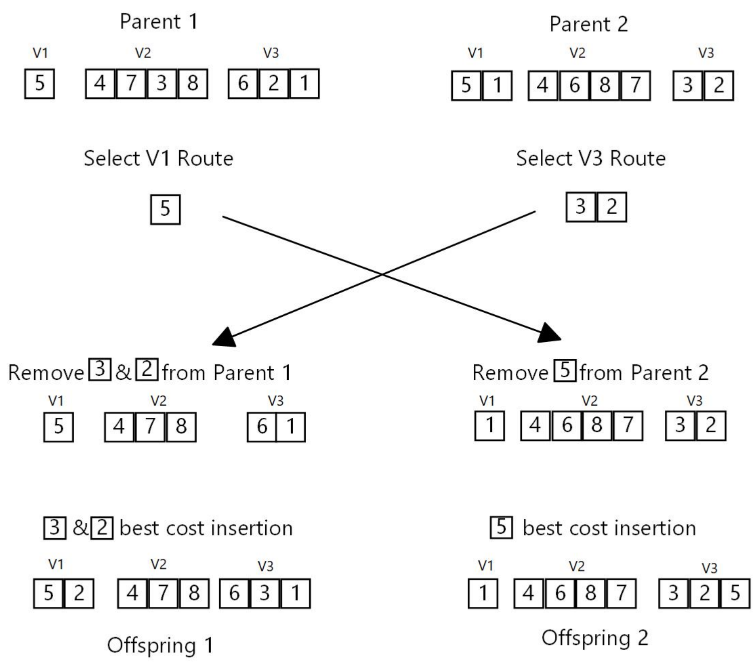


Fig. 3.8 Modified BCRC Crossover Example

a binary mask for each parent that dictates which of the routes of that parent to be fixed or not. Customers in the non-fixed routes are sorted based on the other parent in the selected crossover. Each is sought for a least-cost feasible insertion position in the fixed routes, and if no such position exists, the customer will be listed as missed with a penalty. In a later study (Ombuki et al., 2006), they have modified RC to Best Cost RC (BCRC), where only one random route is selected from each parent and customers are inserted in arbitrary order. However, if non-feasible insertion positions occur, the corresponding customer is routed by initiating a new route. This work modifies the BCRC operator to be compatible with the relaxed constraints. The cost of re-insertions considers penalties instead of initiating routes that are previously assumed similar due to the homogeneous vehicle problem. Costs for route initiations are calculated per the vehicle that this route is provided; consequently, more computational effort is more likely.

Figure 3.8 demonstrates an example of the modified BCRC. Starting with two parents, each with 3 vehicle routes, vehicle 1 route is selected from parent 1 and vehicle 2 route from the other parent. The customers that occur in the selected route are removed from the other parent. In this example, Customer 5 is removed from Parent 2 and Customers 3 and 2 are removed from parent 1. The removed customers are then inserted on the best cost insertion position while also considering constraints violations. In the example, customer 5 has been inserted at the end of vehicle 3 route in parent 2 generating the offspring 2. On the other hand, customers 3 and 2 are inserted within vehicle 3 route and at the end of vehicle 1 route, respectively, generating offspring 1. In case of multiple customer insertions per parent as the latter case, the choice of which customer to be inserted first is done arbitrary.

3.5.2 The Metaheuristic Module

By their definition, metaheuristics are algorithmic frameworks, or "recipes", that provide high-level solution search strategies independent of the problem (Sörensen, 2015) and when it comes to VRP problems, it is evident that these recipes are the most favoured when solving the problem (Montoya-Torres et al., 2015). It is essential to differentiate between their two main types, population-based metaheuristics and single solution-based (Talbi, 2009). Concerning VRP literature, both types have been implemented, with Genetic Algorithm (GA) and Tabu Search (TS) being the most used in each type, respectively (Elshaer and Awad, 2020). The type of metaheuristics to implement for a particular VRP is usually based on the researcher's preferences. However, since the adopted problem considers optimising multiple objectives, it is evident that the population-based evolutionary algorithms, particularly GA, along with Pareto dominance sorting, are most widely used for a problem similar to the problem under study (Jozefowiez et al., 2008). As the Pareto dominance sorting methods do not favour objectives over another. As a result, the GA metaheuristic framework is the primary focus of this implementation. It is adopted with its components examined and recreated to be compatible with the proposed agent-based model.

GA is an algorithmic framework that was first introduced by Holland (1975). It explores the solution space through a pool of solutions (population) by altering selected individual solutions (parents) based on reproduction operators known as crossover and mutation in order to generate new solutions (offspring) that replace the current population for another evolutionary round to be performed. GA is a stochastic process as the selection is applied randomly and skewed to better solutions. Furthermore, the reproduction operators are also applied randomly given their rate of occurrence, and the selection of altering positions within the individuals is also random.

Before presenting the overall algorithm of the adopted GA, it is essential to explain the GA adaptation to the agent-based VRP formulation. Since GA is a population-based metaheuristic, different copies of individuals need to be stored; therefore, memory adaptation is needed for the proposed agent-based VRP formulation. This subsection also explains the choice of the initial solution and other metaheuristic-specific components from selection and recombination. A summary of the overall algorithmic framework is provided at the end of this subsection.

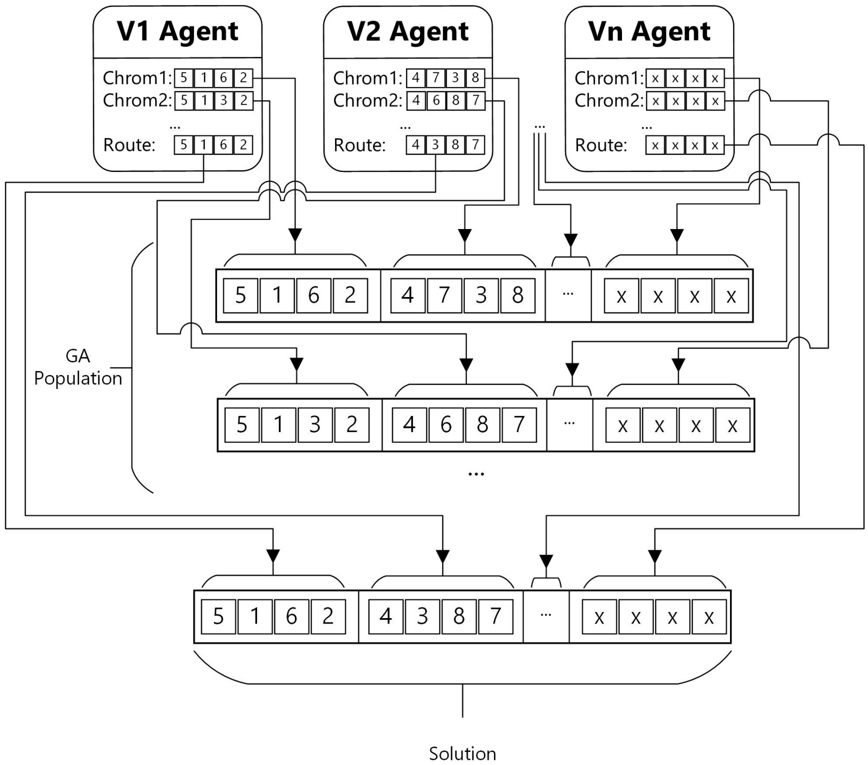


Fig. 3.9 GA Population Representation Adapted to the Agent-Based VRP

3.5.2.1 Agent-based Memory Adaptation

When applying this algorithmic framework to work with the proposed single solution path representation, shown in Figure 3.6, some essential modifications are needed in order to represent the whole population. A new memory structure is proposed to accommodate multiple solutions generated during the evolutionary process. Figure 3.9 shows the solution representation of the population, which is represented by structuring more vehicles' route attributes, dubbed chromosomes, as being part of a whole individual solution in the overall individual representation. Individuals are selected based on their unique numbers during selection, then generate new individuals with unique identification numbers after applying recombination operations. The top individual found throughout the process can be copied in the solution memory composed of the best route parts stored in each vehicle agent attribute.

3.5.2.2 Initial Population

The evolutionary algorithm needs to start with an initial population to initiate the evolutionary search. There have been different implementations for generating the initial population; however, most of them are mixed with randomly generated individuals based on customers' permutations despite constraints violations and heuristics. Such random diversity in the population is desired for a better solution exploration (Ghoseiri and Ghannadpour, 2010). The adopted heuristics in generating the initial population are divided into two, either a greedy heuristic or Solomon's insertion, as evidenced in Ombuki et al. (2006) Ghoseiri and Ghannadpour (2010), respectively. However, such heuristics, if implemented in the proposed agent-based model, will be considered hybrid in their approach, contrary to the goal of the development of the centralised approach to be fully compared against the hybrid approach proposed in section 3.4. Therefore, a fully randomised initial population is considered by randomising sequences of customers, and the number of customers is equally distributed across the vehicles for a particular individual solution.

3.5.2.3 Selection and Recombination

The selection for recombination in this GA is based on Roulette Wheel Selection, where individuals' selection probabilities are based on their fitness value concerning their whole population. Accordingly, their probability of selection is calculated. Individuals with better fitnesses will have a high probability of being selected as a parent. Regarding solutions recombination, the modified BCRC is adopted as crossover and Or-opt, with three nodes limit, as mutation (single solution alteration) as described in 3.5.1.3. Furthermore, a greedy local search is implemented using CROSS-exchange to improve specific individuals with a pre-defined probability. A similar greedy approach was previously adopted for VRPTW in Ghoseiri and Ghannadpour (2010) work as well as in Vidal et al. (2012) as an education operator.

3.5.2.4 The Overall Algorithm

The adapted overall framework of the adopted GA is shown in Algorithm 3. g and Gen represent the current generation number and the total number of generations, respectively. X_{rate}, M_{rate} and LS_{rate} are the crossover, mutation and local search rates, respectively. Pop represents the populations of size P_{Size}. Sel_{size} determines the selection size from the population by creating a subsequent subpopulation dubbed as $Pop_{selected}$ while X_{count} dictates the number of times to perform crossover per generation on $Pop_{selected}$. $Parents$ are a subsequent population of size two generated to apply the crossover operator between two individuals to generate $Offspring$. Ind is a notation to represent a specific individual in the population.

Algorithm 3: Genetic Algorithm

Data: Gen, X_{rate}, M_{rate}, LS_{rate}, P_{Size}
Result: Best found V_r
Generate Pop randomly;
Sel_{size} = max(int((1 - X_{rate}) x P_{size}), 2);
$X_{count} = P_{Size} - Sel_{size}$;
$g = 0$;
while $g < Gen$ **do**
    $g := g + 1$;
    $Pop_{selected}$ = Roulette Wheel Selection(Pop, Sel_{size});
    $Parents$ = Roulette Wheel Selection($Pop_{selected}$, 2);
    for X_{count} **do**
        Apply Crossover on $Parents$ and generate $Offspring$;
        Add $Offspring$ to $Pop_{selected}$;
    end
    Set Pop as $Pop_{selected}$;
    for *Ind in Pop* **do**
        Apply Mutation on Ind with a probability of M_{rate};
        Apply Local Search on Ind with a probability of LS_{rate};
    end
    Calculate fitnesses for every Ind in Pop;
    Rank every Ind in Pop based on fitness function;
    Determine the fittest V_r;
end

Algorithm 3 represents the implementation of the GA for the problem under study. The evolutionary process starts by defining the maximum number of generations Gen, probabilities of crossover X_{rate}, mutation M_{rate} and local search LS_{rate} and population size P_{Size}. Prior to the evolutionary search, the initial population are firstly generated as well as calculation of the population selection size Sel_{size} and number of crossovers X_{count} are calculated based on the parameters provided. At the start of every generation, a subsequent subpopulation $Pop_{selected}$ is selected of size Sel_{size} as well as a subsequent subpopulation of size two $Parents$ using Roulette Wheel Selection and for a number of counts X_{count}, crossovers are performed, and offspring are added to the population $Pop_{selected}$. At the end of the crossover operation, a new population Pop is set to $Pop_{selected}$. The mutation and local search are applied to every individual with probabilities of M_{rate} and LS_{rate}, respectively. Towards the end of a specific generation, fitnesses are calculated for every individual, ranking them accordingly while the best individual is sought and checked with the best found. This evolutionary process is repeated until the maximum number of generations is reached.

3.5.3 The Multi-Objective Module

According to Talbi (2009), metaheuristics can be designed with three multi-objective search components, which are fitness assignment, diversity preserving and elitism. Fitness assignment is concerned with strategies of assigning single value quality measure to a vector of objective functions, and diversity preservation emphasises generating diverse sets of solutions. Finally, elitism focuses on preserving the best solutions across iterations. The choice of strategies in each of these components is explained in the following subsections.

3.5.3.1 Fitness Assignment

Talbi (2009) stated that there are four general types of fitness assignment; however, only two have been previously considered based on VRP metaheuristic implementations, scalar and dominance-based strategies. Scalar approaches directly transform the multi-objective function into a mono one by aggregating and weighting the different functions into one simplified function. However, this requires assuming weights and parameters of the aggregation based on the previous problem-specific knowledge from the modeller or decision-maker. On the other hand, dominance-based approaches utilise the Pareto optimality in sorting solutions based on their objectives' vectors to avoid weighting the objectives and not combining them into a mono function rather than dealing with an objective vector as a whole.

Jozefowiez et al. (2008) argues that for routing problems, the dominance-based approaches are favoured to be implemented with population-based metaheuristics, particularly evolutionary algorithms. Their ability to deal with multiple sets of solutions to find their Pareto optimality makes them applicable. In contrast, scalar approaches are primarily used in single solution metaheuristics, such as local search or even simpler heuristics. Therefore, in this work, the dominance-based Pareto ranking approach is adopted. The implementation is similar to what has been implemented by Ombuki et al. (2006) and Ghoseiri and Ghannadpour (2010). Given that the adopted evaluation method considers the relaxed constraint approach, penalties must be considered when evaluating the objectives. As a result, adding penalties to the objectives is required to represent a relaxed evaluation. Traditionally, penalties are added to only one objective in the scalar method. Given the case of multiple objectives, the total penalties from Equation 3.20 are added to the set of the objectives resulting in the following overall Pareto set:

$$Overall\ \ Objective\ \ Vector = [P_{V_r} + \sum_{v_r=1}^{V_r} D_{v_r}, P_{V_r} + \sum_{v_r=1}^{V_r} W_{v_r}, P_{V_r} + \sum_{v=1}^{V_r} y_{v_r}] \tag{3.28}$$

Upon calculating the individual objective vector with total penalties addition, individuals are ranked accordingly based on Algorithm 4, where Pop_{ranked} is the resulted ranked population and $Rank_{current}$ is the current rank that increments at each Non-dominated Pareto set.

Algorithm 4: Pareto Ranking

```
Data: Pop
Result: Pop_ranked
Rank_current = 1;
while Pop ≠ φ do
    for Ind in Pop do
        if Ind is non-dominated in Pop then
            Ind Rank = Rank_current;
        end
    end
    for Ind in Pop do
        if Ind Rank = Rank_current then
            Remove Ind from Pop;
            Add Ind to Pop_ranked;
        end
    end
    Rank_current := Rank_current + 1;
end
```

Algorithm 4 ranks a specific population based on their Pareto dominance. It starts by initialising the first rank of 1 and loops through all the individuals of the population *Pop* and checks which of them are not dominated by any other individual. If they are not dominated, their rank is set to the current rank. Upon determining this rank of non-dominated individuals, they are then removed from the unsorted population *Pop* and added to the ranked population set Pop_{ranked} and incremented the current rank $Rank_{current}$ by 1. The process is repeated to determine the next non-dominated rank set until no further individuals are left in the unsorted population *Pop*. Algorithm 4 is applied when determining the fitnesses of all the individuals in a population for a particular generation. Fitness is the complement of the normalised ranks across the population to reflect their probability of selection. The calculation of fitness based on the individual rank in a population is shown by Equation 3.29.

$$Ind_{fitness} = 1 - (Rank_{Ind} - Rank_{min})/(Rank_{max} - Rank_{min}) \tag{3.29}$$

where $Ind_{fitness}$ and $Rank_{Ind}$ are the calculated fitness and the rank given to the individual, respectively, while $Rank_{min}$ and $Rank_{max}$ are the minimum and maximum ranks in a given population, respectively.

3.5.3.2 Preserving Diversity and Elitism

Metaheuristic algorithms tend to degrade their solutions at higher iterations and may result in stagnating search, especially when multiple objectives are considered. Therefore, elitism is introduced to prevent the degrading performance and diversity is used to minimise the stagnation effect (Talbi, 2009). Since these approaches are only used in population-based metaheuristics, their implementation is limited to the GA adopted. Elitism is adopted by passing a certain number of best fit individuals to the next generation unaltered. This can be seen in Algorithm 3, where only two individuals ($Parents$) are selected from $Pop_{selected}$ for recombination while the remaining in $Pop_{selected}$ are passed to the next generation population (Pop). On the other hand, diversity can not be implemented in a parameterless way due to the discrete objective function, minimisation the number of vehicles. This discrete objective hinders maintaining a proper preserved distance, in the objective space, between individuals in a population (Ombuki et al., 2006).

3.6 Chapter Summary

This chapter proposes an agent-based model for DVRP with vehicle breakdowns that hinder vehicles' ability to execute their assigned route. By highlighting the importance of agents' interactions and cooperation, distributed and centralised, in optimisation and their applicability in VRP, two cooperative agents approaches have been adopted, hybrid and centralised. As a result, the proposed agent-based model consists of vehicle and customer agents and a super agent, dubbed the assignment agent, to carry out centralised tasks. In the hybrid approach, a Messaging Protocol-Based Heuristic Optimisation (MPHO) has been proposed to construct vehicles' routes sequentially based on pre-specified priority rules. Feasibility evaluation techniques have been adapted to the unique vehicle attribute problem that arises from the breakdown instant by modifying the Push Forward feasibility to be performed within each vehicle agent. The modified Push Forward check has been adopted to evaluate insertions given the vehicles' unique attributes, including route duration constraints. Additional reduced total waiting time calculation is also adopted to overcome Solomon's assumption that a route is initiated the moment the vehicle is available. On the other hand, the agent-based model is adapted to a more extensive centralised search aided by a metaheuristic framework in the centralised approach. Differentiation is needed between problem-dependent components to solve the problem with a metaheuristic framework flexibly. Problem-dependent and metaheuristic components are represented in the agent-based and metaheuristic modules, respectively. Solution representation is adapted to a population-based metaheuristic by providing additional memories within each vehicle agent to represent the pool of solutions that can undergo alterations and variations adapted from literature. Solution evaluation is performed through a centralised messaging where the assignment agents add up all the localised customer and vehicle evaluations considering constraints relaxation strategies. Such a generalised way of representing, altering, and evaluating solutions apply to any appropriate metaheuristic framework for which the Genetic Algorithm is adopted in this work. Finally, a multi-objective search design is proposed by adopting the dominance-based Pareto sorting on the specially designed Pareto objective set that considers the relaxed constraints.

Chapter 4

Results Analysis and Discussions

4.1 Chapter Overview

This chapter verifies and validates the proposed approaches against benchmark instances, modified benchmark instances, and a case study, including the analysis and discussions. Before presenting the results, the hardware and software of the implementation are reported, in addition to the parametric settings for both the hybrid and centralised approaches. The output KPIs are also abbreviated and summarised for ease of results interpretation. First, experiments are conducted on MDVRPTW benchmark instances, the closest instances to the breakdown instant problem compared to the best-known solutions, followed by further modifications of these benchmark instances to appropriately represent the breakdown problem and compare the outputs of both the hybrid and centralised approaches. However, the dynamic theme of breakdowns was not compared with appropriate benchmark instances due to its absence. A case study is also adopted for a particular day of operations solved statically and compared against the original routing solution adopted by the company. Breakdowns are randomly introduced for this particular day of operations, generating three different dynamic breakdown problem scenarios that are solved using the most efficient proposed approach. These generated scenarios can benefit future studies in this domain of study.

4.2 Experimental Settings

This section presents the hardware, software and parametric settings for the hybrid and centralised approaches while highlighting the abbreviated output KPIs. Both approaches are implemented using the same hardware and software. Subsection 4.2.1 reports on the hardware and software of the implementation while subsections 4.2.2 and 4.2.3 report on the parametric settings of the hybrid and centralised approaches, respectively. Finally, the output abbreviated KPIs are explained in subsection 4.2.4.

4.2.1 Hardware and Software

The agent-based system, with both of its approaches, hybrid and centralised, is programmed in Python (Python Software Foundation, 2021). No optimisation libraries were used as the model was coded from scratch, including the metaheuristic framework constrained by the proposed customised agent structures. Every model run is implemented on Linux based High Performance Computer (HPC) using an Intel(R) Xeon(R) Broadwell CPU E5-2683 v4 @ 2.10GHz (32CPU-cores/node) with 128GB of RAM available. Multiple experiment runs are conducted in parallel, using the multiple cores of each CPU node.

4.2.2 Hybrid Approach Parametric Setting

The hybrid approach considered the range and values of each parameter as implemented originally by Solomon (1987). The parameters are: μ is either 1 or 2, α_1, and α_2 are between 0.0 and 1.0 while λ is always set to 1. On the other hand, the Customer Priority Rules (C Rule) were either by latest deadline (LTW) or farthest distance, which is also split into two rules either average (Far_Avg) or minimum (Far_Min) distance of all vehicles. Therefore, the hybrid approach experiments are conducted by considering all the combinations of these parameters and rules. The best solutions are reported for every instance based on prioritising, customer coverage, the minimum number of vehicles, distance travelled, and waiting times in order. The hybrid approach parameters are shown in Table 4.1.

Table 4.1 Hybrid Approach Parametric Settings

Parameter	Set Value
μ	0 or 1
α_1	range(0.0, 1.0)
α_2	range(0.0, 1.0)
λ	set to 1
C Rule	LTW, Far_Avg & Far_Min

4.2.3 Centralised Approach Parametric Setting

The adopted evolutionary metaheuristic parametric settings are mainly adopted from Ghoseiri and Ghannadpour (2010) by considering the number of runs, population size, crossover and mutation rates. The number of generations, however, is reduced given the implementation of an additional greedy local search with a probability of 10%, similar to the education parameter from Vidal et al. (2012) where they set the number of generation parameter to be 150. Additional relaxed constraint handling parameters are considered, h and γ, which are the number of iterations to revisit penalty calculations and the factor of multiplying/dividing these penalties. h is adapted from Gendreau et al. (1994), where they experimented with this parameter within the range 5 to 20 and resulted in no very low sensitivity to such parameter for the problem. Therefore, we arbitrarily chose 5. γ is adapted from Rochat and Semet (1994) where they firstly randomised it to be between 1.5 and 2. The centralised stochastic GA experiment is repeated ten times, and the dominant solution is reported. The centralised approach parameters are shown in Table 4.2.

Table 4.2 Centralised Approach Parametric Settings

Parameter	Description	Set Value
P_{Size}	Population size	100
Gen	Number of generations	150
X_{rate}	Crossover rate	80%
M_{rate}	Mutation rate	20%
LS_{rate}	Greedy local search rate	10%
h	Generations to revisit the penalties	5
γ	Penalties multiply/divide factor	Uniform(1.5, 2.0)
-	Number of parallel experiments	10

4.2.4 Output KPIs

The main output of the model, as stated in Figure 3.3, are the Vehicles used (V), total Travelled Distance (TD) and total Waiting Time (WT), in addition to CPU core timing in seconds as a measure of complexity. Furthermore, the Hybrid approach (H) will have added two performance measures: the total Customers (CM) and their Demands (DM) Missed due to the adopted hard constraint strategy. On the other hand, the Centralised approach (C) relaxed three constraints and reported their violations: capacity VQ_{V_r}, duration $Vdur_{V_r}$ and time window VTW_{V_r}. The output abbreviations and symbols are summarised in Table 4.3.

Table 4.3 Results KPIs; H: Hybrid, C: Centralised

KPIs	Description	Approach
V	Number of Vehicles Used	Any
TD	Total Distance Travelled	Any
WT	Total Waiting Time	Any
CM	Total Missed Customers	H
DM	Total Missed Demand	H
VQ_{V_r}	Total Capacity Violation	C
$Vdur_{V_r}$	Total Duration Violation	C
VTW_{V_r}	Total Time Window Violation	C

4.3 Results on Benchmark Instances

This section validates and shows the superiority of both the hybrid and centralised approaches against the MDVRPTW benchmark instances (Cordeau et al., 2001). It is found that these instances are the nearest to the problem under study, as other problem settings would not have direct comparison factors that suit the problem's nature nor provide validated benchmark results for comparison. The characteristics of each MDVRPTW benchmark instance are summarised in Table 4.4. For every instance, the number of available vehicles is presented along with the vehicle duration and capacity constraints. Number of customers is also shown along with their generated time window width, tight or wide. The instances are grouped into two based on their customers time windows, pr01-10 and pr10-20.

The proposed approach was first tested on these instances towards initial verification, although their original solution approach minimises the travelled distance. In a later study by Chiu et al. (2006), they generated multiple criteria from Cordeau et al. (2001) work, including total travelled distance, waiting times and the number of vehicles, and the results are compared against them to ensure a fair multiple criteria comparison study.

Tests on MDVRPTW benchmark instances have been conducted. Results are presented and discussed in subsections 4.3.1 and 4.3.2 and compared to best-known solutions as reported by Chiu et al. (2006). They adopted a two-phase (route construction and improvement) heuristic approach to minimise vehicle waiting times leading for which it may reduce the total number vehicles used. Their results were compared Cordeau et al. (2001) Tabu Search (Cordeau). Results' comparisons are made in deviation, except for waiting times, which are made in differences due to deviation may result in division by zero. Comparisons between the hybrid and centralised approaches are presented and discussed in subsection 4.3.3.

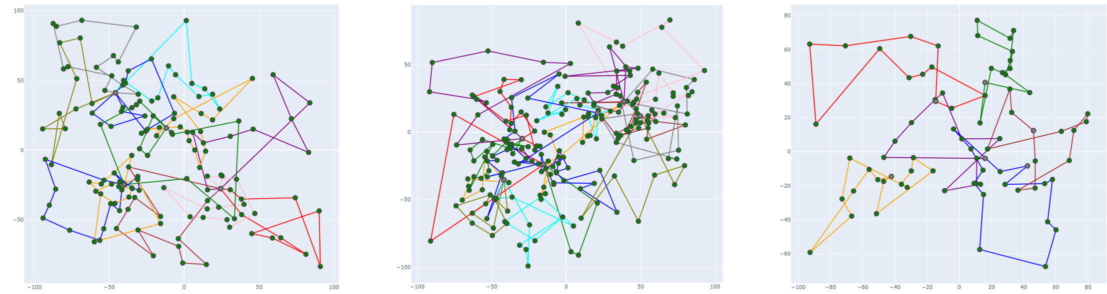
Table 4.4 Characteristics of MDVRTW Benchmark Instances

	Vehicles			Customers	
Inst.	Count	Duration	Capacity	Count	Time Window
pr01	8	500	200	48	tight
pr02	12	480	195	96	tight
pr03	16	460	190	144	tight
pr04	20	440	185	192	tight
pr05	24	420	180	240	tight
pr06	28	400	175	288	tight
pr07	12	500	200	72	tight
pr08	18	475	190	144	tight
pr09	24	450	180	216	tight
pr10	30	425	170	288	tight
pr11	4	500	200	48	wide
pr12	8	480	195	96	wide
pr13	12	460	190	144	wide
pr14	16	440	185	192	wide
pr15	20	420	180	240	wide
pr16	24	400	175	288	wide
pr17	42	500	200	72	wide
pr18	12	475	190	144	wide
pr19	18	450	180	216	wide
pr20	24	425	170	288	wide

4.3.1 Hybrid Approach

The results for the Hybrid (H) approach are presented in Table 4.5. It can be seen that the most common rule to emerge to the best solutions is the Far_Avg which means that prioritisations based on customers' urgency are not fruitful. Regarding the other three parameters, μ seems to be favoured to be 2 in 11 out of the 20 instances, meaning it has a neutral effect on solutions. At the same time, α_1 is favoured to be greater than 0.7 in all cases and above 0.9 in 17 instances out of 20. This behaviour indicates that distance insertions are favoured compared to time insertions. Sampled result maps with their routes are shown in Figure 4.1, where a unique colour of arcs represents a route in a map.

Concerning the output, the tight time window problems pr01-10, the average number of used vehicles is 16, which has considerably reduced from Cordeau by around 18% and around 6% reduction compared to Chiu. The slight reduction is seen in the wide time window instances pr11-20 with 3.5% compared to Cordeau and Chiu. However, despite the decrease in the number of vehicles, an increase in the travelled distance is incurred. An average of



(a) pr03 (b) pr09 (c) pr17

Fig. 4.1 Hybrid Approach Sampled Routes' Maps

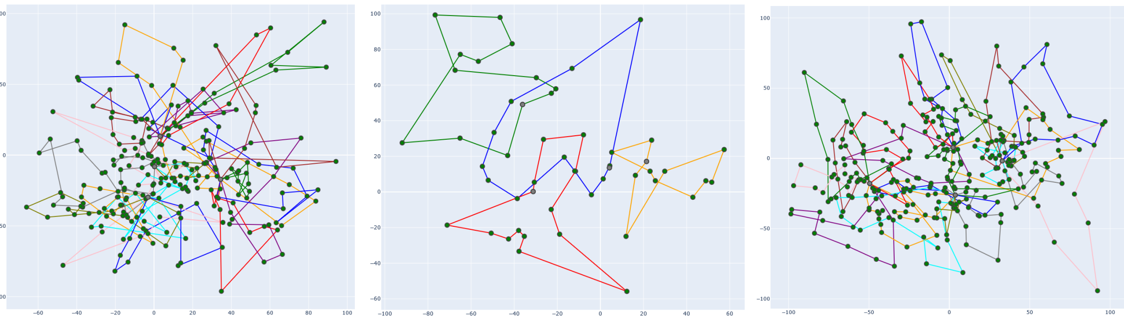
3641.19 distance units for tight time window instances increased by around 44% and 6% from Cordeau and Chiu.

Wide time window instances, pr11-20, total distance averaged to 2813.45 with an increase of 34.3% and 7.36% from Cordeau and Chiu. Waiting times have averaged 354.33 and 197-time units for tight and wide time window instances, respectively, where it has considerably reduced by 1623.77-time units from Cordeau and increased by 287.14 from Chiu in the tight time window instances and similar behaviour seen in the wide time window instances where it decreased by 289.11 and increased by 193.21 in Cordeau and Chiu, respectively. Finally, CPU times of the hybrid methods are considerably low, averaged below 10 seconds for both problem types, compared to Chiu's, as only reported the heuristic timing, with above 90% reduction in both instance types.

The superiority of the hybrid approach in achieving routes with less number of vehicles is attributed to the adopted route construction approach that firstly priorities the minimisation of the number of vehicles by not initiating a new route until no further insertions are possible, as evidenced in the messaging in Figure 3.4. Contrary to Cordeau, they implemented a metaheuristic with only one objective considered: the total distance travelled. On the other hand, Chiu considered a similar route construction heuristic, however, with a different priority of minimising the total waiting times, arguing that it would minimise the total utilised vehicles. However, as demonstrated in this study, this is not the case as the resulted routed achieved fewer utilised vehicles with higher waiting times than Chiu. Furthermore, the experimentation with additional sorting rules, given the unique modelling of the vehicle's locations, may have contributed to achieving better results in minimising the number of vehicles.

Table 4.5 Hybrid Approach Results on MDVRPTW Instances

	Parameters				V	V%		TD	TD%		WT	WT(Diff)		CPU(s)	CPU%
Inst.	C Rule	μ	α_1	α_2	H	Cordeau	Chiu	H	Cordeau	Chiu	H	Cordeau	Chiu	H	Chiu
pr01	LTW	2	0.9	0.1	6	-25.00%	0.00%	1493.99	37.82%	-3.04%	211.18	-390.92	200.28	0.29	-97.81%
pr02	Far_Min	1	0.9	0.1	9	-25.00%	0.00%	2470.84	40.14%	7.02%	104.00	-1024.70	77.30	1.26	-96.05%
pr03	Far_Avg	1	0.8	0.2	13	-18.75%	-13.33%	3376.52	40.20%	4.79%	174.55	-2067.05	82.25	2.90	-95.80%
pr04	Far_Avg	2	1.0	0.0	17	-15.00%	-5.56%	4096.62	38.48%	7.42%	443.04	-1393.36	258.34	5.41	-95.42%
pr05	Far_Avg	1	0.9	0.1	22	-4.35%	-4.35%	4383.22	39.86%	-1.61%	421.05	-1795.45	372.75	7.72	-95.64%
pr06	Far_Avg	2	1.0	0.0	27	-3.57%	-3.57%	5362.25	37.35%	6.21%	626.01	-1879.99	557.31	11.70	-95.94%
pr07	Far_Avg	2	0.9	0.1	7	-30.00%	-12.50%	2091.43	46.93%	7.91%	37.45	-1336.25	28.55	0.70	-96.84%
pr08	Far_Avg	1	1.0	0.0	13	-23.53%	-7.14%	3312.84	54.07%	13.04%	333.65	-2090.75	212.35	3.00	-95.46%
pr09	Far_Avg	2	1.0	0.0	18	-21.74%	-10.00%	4394.68	55.08%	9.50%	518.92	-2169.28	461.12	7.04	-95.21%
pr10	Far_Avg	2	1.0	0.0	26	-10.34%	-7.14%	5429.50	46.06%	4.66%	673.45	-2089.95	621.35	10.42	-95.57%
				Avg	16	-17.73%	-6.36%	3641.19	43.60%	5.59%	354.33	-1623.77	287.16	5.04	-95.97%
pr11	Far_Min	2	0.7	0.3	4	0.00%	0.00%	1222.68	18.53%	-3.41%	48.36	-68.64	38.46	0.47	-96.07%
pr12	Far_Min	1	0.7	0.3	8	0.00%	0.00%	1974.59	31.60%	11.73%	204.77	-89.23	204.77	2.33	-93.33%
pr13	Far_Min	2	0.9	0.1	11	-8.33%	-8.33%	2723.80	34.80%	6.61%	142.34	-169.46	142.34	6.13	-91.61%
pr14	Far_Avg	2	0.9	0.1	15	-6.25%	-6.25%	3045.28	35.51%	11.11%	173.88	-519.42	173.88	12.06	-89.86%
pr15	Far_Min	2	1.0	0.0	19	-5.00%	-5.00%	3513.45	39.99%	11.09%	223.44	-490.26	223.44	13.59	-92.23%
pr16	Far_Avg	1	0.9	0.1	23	-4.17%	-4.17%	4028.20	36.83%	10.61%	330.93	-575.77	330.93	20.21	-92.78%
pr17	Far_Avg	2	0.9	0.1	6	0.00%	0.00%	1626.72	30.13%	1.54%	89.41	26.81	82.61	1.11	-94.47%
pr18	Far_Avg	1	1.0	0.0	12	0.00%	0.00%	2532.35	39.96%	6.80%	145.23	-102.37	135.83	4.68	-92.90%
pr19	Far_Avg	1	1.0	0.0	16	-11.11%	-11.11%	3283.09	42.07%	14.91%	205.54	-562.36	205.54	11.74	-92.01%
pr20	Far_Avg	1	1.0	0.0	24	0.00%	0.00%	4184.30	33.60%	2.61%	406.09	-340.41	394.29	17.37	-93.00%
				Avg	14	-3.49%	-3.49%	2813.45	34.30%	7.36%	197.00	-289.11	193.21	8.97	-92.83%



(a) pr05 (b) pr11 (c) pr16

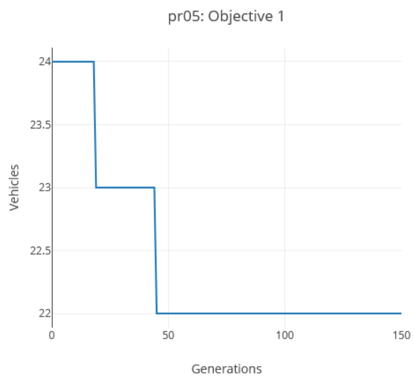
Fig. 4.2 Centralised Approach Sampled Routes' Maps

4.3.2 Centralised Approach

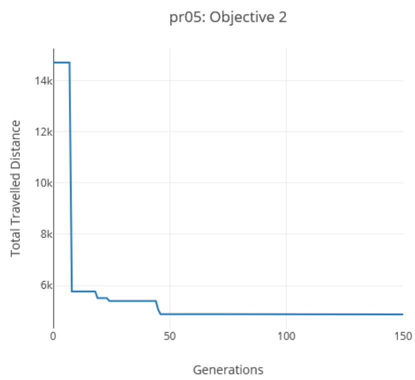
The Centralised (C) approach results for all instances are summarised in Table 4.6. The approach has demonstrated the same output, compared to the hybrid, in terms of the number of vehicles used and further decreased the waiting times. However, at the expense of increasing distances, explained by the adopted multi-objective technique by not favouring any objective over another.

A selected sample of the resulted maps with their routes is shown in Figure 4.2. The sub-figures 4.2a, 4.2b and 4.2c show the maps for the resulted routes in pr05, pr11 and pr16 instances, respectively. A sample GA objectives improvement run of instance pr05 is illustrated in Figure 4.3. Figure 4.3a shows the number of vehicles improvement across the generations, Figure 4.3b shows total travelled distance improvement while Figure 4.3c shows waiting times improvements. All figures indicate no more improvements found after around generation 50 for this particular instance run. It is worth mentioning that these evolutionary generation charts update when the best non-dominated solutions are found, in other words, when all objectives are improved.

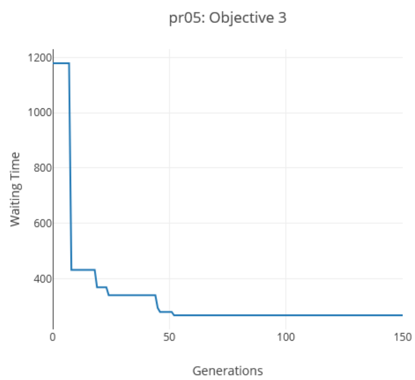
Referring to the reported results in Table 4.6, the centralised search technique was computationally expensive due to its centralised evaluation. The complexity of the centralised evaluation is due to the implementation of the adopted hybrid metaheuristic (GA and LS). It took around 4 and 5 hours on average for the respective tight and wide time window instances, which have risen around 15 and 20 times. The resulting distance increase was around 72% and 27% in the tight time window instances and around 42% and 14% in the wide time window instances compared to Cordeau and Chiu, respectively. Nevertheless, the centralised approach reached zero waiting times in almost half the instances. Tight time window instances waiting times averaged 61.77, reduced by around 1916 and 5-time units from Cordeau and Chiu, respectively. On the other hand, the wide time window



(a) V



(b) TD



(c) WT

Fig. 4.3 Centralised GA Run for Each Objective, Instance pr05

instances averaged only 4.78-time units, decreased from Cordeau by around 481-time units and increased by around 1-time unit from Chiu.

The superiority of the centralised approach in achieving better waiting times than the previous methods is attributed to the adopted multiple-objective approach using non-dominance based sorting considering all three objectives. This sorting technique does not prioritise any objective over the other. Contrary to what has been implemented by Cordeau and Chiu. Cordeau only optimised for distance, while Chiu prioritised the waiting time objective over the others in their developed heuristic.

The extensive search in the education operator using a greedy local search has considerably improved the solution quality, however, at the expense of the computational time. The limited stopping conditions can also explain the expensive computational time, which is only when the specified number of generations, 150 in this case, is achieved. Other stopping conditions could have been implemented, such as terminating the algorithm when no more improvements are achieved in a certain number of consecutive generations. Figure 4.3 and its sub-figures demonstrates the unnecessary search after around generation 50.

Table 4.6 Centralised Approach Results on MDVRPTW Instances

	V	V%		TD	TD%		WT	WT(Diff)		CPU(s)	CPU%
Inst.	C	Cordeau	Chiu	C	Cordeau	Chiu	C	Cordeau	Chiu	C	Chiu
pr01	6	-25.00%	0.00%	1834.91	69.27%	19.09%	0.00	-602.10	-10.90	1904.83	14552.54%
pr02	9	-25.00%	0.00%	2654.15	50.54%	14.96%	0.66	-1128.04	-26.04	8660.86	26965.17%
pr03	13	-18.75%	-13.33%	3534.00	46.74%	9.68%	79.03	-2162.57	-13.27	12304.66	17732.84%
pr04	17	-15.00%	-5.56%	5276.27	78.36%	38.35%	22.78	-1813.62	-161.92	17797.88	14982.95%
pr05	22	-4.35%	-4.35%	4862.08	55.14%	9.14%	267.71	-1948.79	219.41	21369.77	11973.31%
pr06	27	-3.57%	-3.57%	6180.17	58.30%	22.41%	89.86	-2416.14	21.16	23869.13	8187.89%
pr07	7	-30.00%	-12.50%	2601.83	82.79%	34.25%	0.00	-1373.70	-8.90	3143.87	14190.30%
pr08	13	-23.53%	-7.14%	3990.90	85.61%	36.18%	92.05	-2332.35	-29.25	10683.54	16087.19%
pr09	18	-21.74%	-10.00%	5739.30	102.53%	43.00%	39.81	-2648.39	-17.99	21443.41	14487.36%
pr10	26	-10.34%	-7.14%	7187.03	93.35%	38.53%	25.80	-2737.60	-26.30	24675.62	10400.26%
Avg	16	-17.73%	-6.36%	4386.06	72.26%	26.56%	61.77	-1916.33	-5.40	14585.36	14955.98%
pr11	4	0.00%	0.00%	1137.32	10.26%	-10.15%	0.00	-117.00	-9.90	3352.02	27833.50%
pr12	8	0.00%	0.00%	1972.70	31.47%	11.62%	0.00	-294.00	0.00	8137.50	23149.99%
pr13	11	-8.33%	-8.33%	2566.60	27.02%	0.46%	0.00	-311.80	0.00	15146.84	20649.09%
pr14	15	-6.25%	-6.25%	3251.16	44.68%	18.62%	0.00	-693.30	0.00	22343.77	18676.27%
pr15	19	-5.00%	-5.00%	3841.19	53.05%	21.46%	0.00	-713.70	0.00	25202.29	14301.31%
pr16	23	-4.17%	-4.17%	4395.81	49.32%	20.70%	28.15	-878.55	28.15	28889.73	10217.76%
pr17	6	0.00%	0.00%	1583.12	26.64%	-1.18%	5.32	-57.28	-1.48	5248.43	26142.16%
pr18	12	0.00%	0.00%	2964.17	63.82%	25.01%	0.00	-247.60	-9.40	15126.68	22819.21%
pr19	16	-11.11%	-11.11%	3683.63	59.40%	28.93%	0.00	-767.90	0.00	24952.40	16874.42%
pr20	24	0.00%	0.00%	4881.18	55.85%	19.70%	14.29	-732.21	2.49	30344.10	12135.52%
Avg	14	-3.49%	-3.49%	3027.69	42.15%	13.52%	4.78	-481.33	0.99	17874.38	19279.92%

4.3.3 Comparison

By comparing the outputs of both the hybrid and centralised approaches, Table 4.7 summarises the per cent deviation in the KPIs between the two approaches. The average number of vehicles used is the same as the previous hybrid approach; however, significant differences between the methods in terms of the total distance and waiting time. Compared to the hybrid approach, the centralised approach managed to decrease the waiting time by around 293 and 192-time units on average, however, at the expense of increasing the total distance by around 17% and 7%, for the tight and wide time window instances, respectively. This behaviour is explained by the dominance-based multi-objective approach that does not weigh any of the objectives. In addition, it considers the waiting time in the objective evaluation.

Table 4.7 Compared Results on MDVRPTW Instances

Inst.	V%	TD%	WT (Diff)
pr01	0.00%	-18.58%	211.18
pr02	0.00%	-6.91%	103.34
pr03	0.00%	-4.46%	95.52
pr04	0.00%	-22.36%	420.26
pr05	0.00%	-9.85%	153.34
pr06	0.00%	-13.23%	536.15
pr07	0.00%	-19.62%	37.45
pr08	0.00%	-16.99%	241.60
pr09	0.00%	-23.43%	479.11
pr10	0.00%	-24.45%	647.65
Avg	0.00%	-16.98%	292.56
pr11	0.00%	7.51%	48.36
pr12	0.00%	0.10%	204.77
pr13	0.00%	6.12%	142.34
pr14	0.00%	-6.33%	173.88
pr15	0.00%	-8.53%	223.44
pr16	0.00%	-8.36%	302.78
pr17	0.00%	2.75%	84.09
pr18	0.00%	-14.57%	145.23
pr19	0.00%	-10.87%	205.54
pr20	0.00%	-14.28%	391.80
Avg	0.00%	-7.08%	192.22

The significant difference in the reported results between the hybrid and centralised approaches is attributed to the different optimisation strategies adopted in each. The hybrid is a route construction algorithm mainly aimed at minimising the number of vehicles used while taking into consideration the other objectives, as evidenced in section 3.4. On the other hand, the centralised approach, reported in section 3.5, is a route improving search that starts with randomly generated solutions that starts in constraints violations. Given the initial start of violations, the approach prioritises overcoming the resulting violation penalties and seeks further route improvement. Furthermore, the centralised approach adopted a non-dominance based multiple objective sorting that does not prioritise any objective over the other. As a result, it seeks to balance the conflicting objectives resulting in the significant difference reported.

4.4 Results on Modified Benchmark Instances

Given that this study aims to model the dynamic breakdown problem, modifications are proposed to the existing MDVRPTW benchmark instances to resemble the optimisation of one breakdown instant. The rationale behind the proposed modifications is illustrated in Figure 4.4.

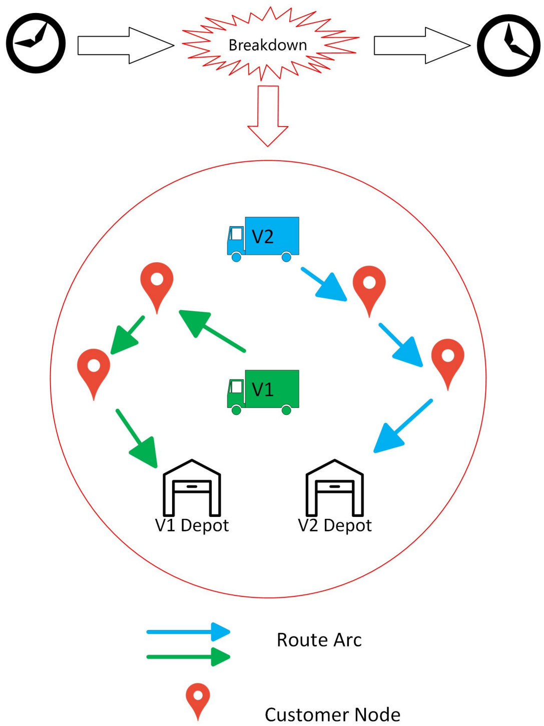
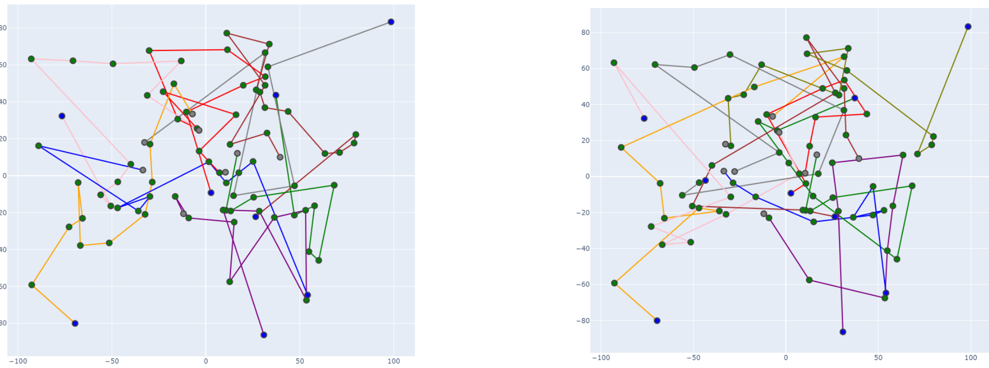


Fig. 4.4 The Rationale for Modifying VRP Benchmark Instances

Figure 4.4 resembles a continuous dynamic environment where breakdowns could occur in continuous time. As vehicles progress in their routes, each would be located at a particular location on the map. Therefore, every vehicle would be located at a specific location when a



(a) Hybrid (b) Centralised

Fig. 4.5 Sample Routes for the Modified pr07 Instance

breakdown occurs. Hence the MDVRPTW benchmark instances are modified by changing each vehicle's locations, starting and ending. Capacities and shifts for every vehicle are further randomised given the ability of the proposed agent-based model to capture the uniqueness of agents, particularly vehicles, as stated in Figure 3.3.

When these benchmarks were introduced (Cordeau et al., 1997), customers' locations are randomly generated within a square of coordinates $[-100, 100]^2$. Depots are generated within $[-50, 50]^2$; therefore, location modifications to the instances are proposed by randomising every vehicle's starting location to be the same as the customers' range while their end locations to be of a similar range as the depots'. Furthermore, capacities are changed by randomising every original vehicle capacity within $\pm 10\%$. Finally, shifts are randomly selected to be reduced by 25% by reducing their ending time or remaining at full availability.

Given the proposed modification, a new set of proposed instances are generated. The resulting data on the modified benchmark instances for both the hybrid and centralised approaches and their comparison are summarised in Table 4.8. A sample solution map of the location modified pr07 instance is shown in Figure 4.5, where the starting node of the vehicles are in blue, and their home depots are in grey.

The main models' outputs are reported, resulting in missed customers (CM) and demand (DM) in the hybrid approach. On the other hand, missing customers is not allowed in the centralised approach given the chromosome design that must route every customer but with constraints violations. Though such violations are allowed, none are reported in the centralised approach.

The hybrid approach has averaged around nine missed customers with the demanded quantity of 128 in tight time window instances while significant missed customers and demand in the wide time window instances of around 71, and 980, respectively. This result is attributed to the hard constraint routing strategy that does not reshuffle the routes to consider these customers. In contrast, the centralised relaxed constraint approach further explores the infeasible region with solution alterations to emerge better solutions.

Regarding the other KPIs, the number of vehicles used in the hybrid approach is higher by 3.28% than its centralised counterpart in the tight time window instance. At the same time, it considerably decreased by 39.37% in the other instances. Similar behaviour is reported concerning the total distance objectives where the centralised approach decreased the distances by 7.48% while increasing by 34.21% in the tight and wide time window instances, respectively. The significant reduction in the vehicles used and distance travelled in the hybrid approach on the wide time window instances is attributed to the large number of missed customers that will not require additional utilised vehicles and therefore travel less distance.

Waiting times in the centralised approach, on the other hand, have demonstrated its superiority in minimising them or even achieving zero waiting time. This achievement is attributed to the explicit objective definition in a multi-objective approach, contrary to the previous VRP implementations. As a result, it has reduced with an average of around 92 and 32-time units in the respective instances.

Finally, the computational cost is also compared where the hybrid approach can be executed on average within around 12 seconds while the centralised within 4.5 hours. The extensive search proposed in the centralised approach is the main factor in increasing the computational cost. It operates through all the 150 iterations with a high probability of implementing the greedy local search. Moreover, this implementation is without a stopping condition of terminating the search if no new best is found for many consecutive iterations. The idea behind running such an approach with the extensive computational cost is to test its utmost potential in emerging better solutions considering multiple objectives.

To sum up, the hybrid approach is sensitive to the tightness of the customer time windows when constructing the problem routes. The tighter the time window is, the more likely it will generate a solution by not missing customers. This behaviour is due to the lack of the hybrid approach's ability to further search the solution space in wider time window instances after constructing the routes. In other words, once the routes are generated following the pre-set rules, the approach can not improve and alter the routes further as it is only limited to route constructions. On the other hand, the centralised approach has this ability and managed to achieve feasible solutions with no violations; however, it is more computationally complex due to the high probability of implementing a greedy education operator.

Table 4.8 Hybrid Vs. Centralised on Modified MDVRPTW Instances

	V			TD			WT			H Missed		CPU(s)		
Inst.	H	C	%	H	C	%	H	C	Diff	CM	DM	H	C	%
pr01	6	5	20.00%	1924.80	1482.7	29.82%	82.24	81.5	0.74	1	19	0.50	2090.92	-99.98%
pr02	10	9	11.11%	2912.02	2510.66	15.99%	121.15	28.91	92.24	3	27	1.84	6972.82	-99.97%
pr03	16	15	6.67%	4378.17	3928.42	11.45%	256.59	47.9	208.69	1	8	4.31	12201.50	-99.96%
pr04	16	18	-11.11%	4249.38	4565.55	-6.93%	113.07	97.63	15.44	31	421	5.66	14737.53	-99.96%
pr05	24	24	0.00%	6273.64	5720.85	9.66%	175.10	206.28	-31.18	15	215	5.98	17585.23	-99.97%
pr06	27	28	-3.57%	6291.80	6461.29	-2.62%	350.02	265.68	84.34	28	394	19.12	20860.76	-99.91%
pr07	8	9	-11.11%	2413.27	2626.7	-8.13%	246.71	34.08	212.63	0	0	1.53	3229.73	-99.95%
pr08	15	14	7.14%	3874.21	3560.31	8.82%	185.40	86.52	98.88	8	109	4.86	11217.11	-99.96%
pr09	22	20	10.00%	5605.78	5217.26	7.45%	345.80	166.75	179.05	0	0	13.10	19740.31	-99.93%
pr10	28	27	3.70%	6945.39	6355.32	9.28%	236.47	179.2	57.27	7	88	16.41	22756.68	-99.93%
Avg	17.2	16.9	3.28%	4486.85	4242.91	7.48%	211.26	119.45	91.81	9.4	128.1	7.33	13139.26	-99.95%
pr11	2	4	-50.00%	639.83	1332.26	-51.97%	0.00	1.61	-1.61	20	301	2.09	3473.54	-99.94%
pr12	5	8	-37.50%	1317.39	2009	-34.43%	50.39	0	50.39	31	483	4.53	9155.87	-99.95%
pr13	7	12	-41.67%	2097.08	3232.5	-35.13%	65.62	3	62.62	61	749	7.27	13398.83	-99.95%
pr14	10	16	-37.50%	2366.60	3451.49	-31.43%	83.58	18.83	64.75	68	906	13.44	17382.29	-99.92%
pr15	14	20	-30.00%	3241.30	4539.28	-28.59%	62.84	22.97	39.87	80	1162	15.49	20220.87	-99.92%
pr16	11	24	-54.17%	2762.09	5489.9	-49.69%	37.76	36.08	1.68	172	2268	19.74	23965.08	-99.92%
pr17	4	6	-33.33%	1063.42	1613.88	-34.11%	28.90	0	28.90	26	301	2.21	5705.17	-99.96%
pr18	7	12	-41.67%	1992.36	3147.68	-36.70%	33.90	0	33.90	67	996	7.57	17478.54	-99.96%
pr19	9	17	-47.06%	2617.90	3791.05	-30.95%	40.68	0.67	40.01	105	1445	17.05	22000.43	-99.92%
pr20	19	24	-20.83%	5097.14	5609.63	-9.14%	24.81	25.74	-0.93	81	1192	25.94	23213.84	-99.89%
Avg	8.8	14.3	-39.37%	2319.51	3421.67	-34.21%	42.85	10.89	31.96	71.1	980.3	11.53	15599.45	-99.93%

Regarding the parameters of the solutions reported on the modified instances using the hybrid approach, Table 4.9 summarises these parameters that are selected based on the solutions that minimise the missed customers. The far distance rule is the most common with selection based on the minimum distance used in 11 cases, while 7 have the average distance, and only two are reported with the late time window rule. The favouring of distance prioritisation may be attributed to the modified vehicles' locations. The μ parameter was reported to be set to 2 in 12 out of the 20 instances showing a similar neutral effect as shown in the solutions to the original benchmark instances. α parameters have seen a mixture of settings. More cases here are seen here with α_1 less than or equal to 0.5, compared to the original instances solution with the hybrid approach in Table 4.5, with 8 cases. This behaviour could be explained by the higher urgency of insertions when constructing routes in these modified instances.

Table 4.9 Hybrid Best Parameters on Modified MDVRPTW Instances

Inst.	C Rule	μ	α_1	α_2
pr01	Far_Min	2	0.3	0.7
pr02	Far_Min	2	0.5	0.5
pr03	Far_Avg	2	0.7	0.3
pr04	Far_Avg	1	0.5	0.5
pr05	Far_Min	2	0.2	0.8
pr06	Far_Min	1	0.9	0.1
pr07	Far_Min	1	0.8	0.2
pr08	Far_Avg	2	0.5	0.5
pr09	LTW	1	0.6	0.4
pr10	Far_Min	2	0.5	0.5
pr11	LTW	2	0.9	0.1
pr12	Far_Min	2	1	0
pr13	Far_Min	2	0.6	0.4
pr14	Far_Min	2	0.9	0.1
pr15	Far_Min	2	0.7	0.3
pr16	Far_Avg	1	0.8	0.2
pr17	Far_Min	2	1	0
pr18	Far_Avg	1	0.4	0.6
pr19	Far_Avg	1	0.3	0.7
pr20	Far_Avg	1	0.2	0.8

4.5 Case Study

This section tests the proposed agent-based model on a real-life case study provided by a collaborator. The collaborator, Aramex, is multinational logistics, courier and package delivery company based in Dubai, United Arab Emirates. Their services expand to over 60 countries with 353 office branches worldwide. It offers a range of logistics services, domestically and internationally, including express delivery, freight forwarding, e-commerce, retail customer services, logistics solutions, and management. Such services are performed in multiple countries and their respective cities while maintaining a proper database for its historical data. The express delivery service is the fundamental interest of this research, mainly their domestic services. This service translates into a routing optimisation problem that is very close to this project's scope. Typical delivery service is mainly about managing the delivery of parcels from a depot to respective customers.

Data of a representative day of operations has been provided for a particular city to be put under study. The provided data translates to a single depot VRPTW case, given that all vehicles start and end their routes from a central depot and the time window availability of the customer. However, the problem slightly defers from VRPTW with vehicles' capacity assumed infinite and no route duration limit. The problem considers 300 customers, each with a 3 minute serving time, with the availability of 4 vehicles.

The data provided is summarised in Appendix A. The customer attributes are reported in Table A.1. For every customer, data provided includes the allocated vehicle identification, preferred delivery time window and the actual delivery time if the service was successful. Customer locations are provided in KM, converted from coordinates and altered for data protection. Furthermore, additional data are collected during multiple interviews with key contacts at Aramex: operations director and operation manager. An interview with the operations director was held on the 21^{st} of August 2019. Two interviews were held with the operation manager on the 22^{nd} and 28^{th} of June 2021. Each interview lasted for 1 hour. The key interview questions are summarised and reported in Table A.2. The questions, along with their responses, address additional problem attributes not reported in the customer data in Table A.1 which include:

- Optimisation approach adopted for the provided case
- Expert estimates on breakdowns
- Vehicle average speed estimation
- Vehicle shift and capacity limits

- Customer servicing time
- Depot location

The remainder of this section is structured as follows: subsection 4.5.1 focuses on the static implementation by reporting the quality of the original static routing solution compared to the results of the proposed approaches. While subsection 4.5.2 performs dynamic implementations of the case study by defining dynamic breakdown scenarios then optimised.

4.5.1 Static Implementation

In this subsection, the outputs of the originally implemented routes are compared against the outputs of the generated routes using the proposed approaches. Originally, the routing process at Aramex followed a cluster-first route-second approach, as reported in the first interview question in Table A.2. In other words, the customers are firstly clustered based on their geographical area and then routed based on the driver's experience in the area.

4.5.1.1 Original Static Implementation

As reported in the customer data reported in Table A.1, the actual time of service delivery per customer is reported. As a result, the actual vehicles' routes can be deduced based on the reported measure. A vehicle's original reported route is the order set of its allocated customers based on their service delivery time, while customers with no actual delivery time are considered missing. The resulted outputs for this particular day using this traditional method are summarised in Table 4.10. The resulted vehicle routes are summarised in Table A.3 and plotted in Figure 4.6 representing the missed customers as red dots on the map.

Table 4.10 Original KPIs of Static Routes implemented by Aramex

KPI	Org
V	4
TD (KM)	856.1
WT (Hr)	0
CM	53
DM	53
VQ_{V_r} (Hr)	N/A
$Vdur_{V_r}$ (Hr)	N/A
VTW_{V_r} (Hr)	993.20

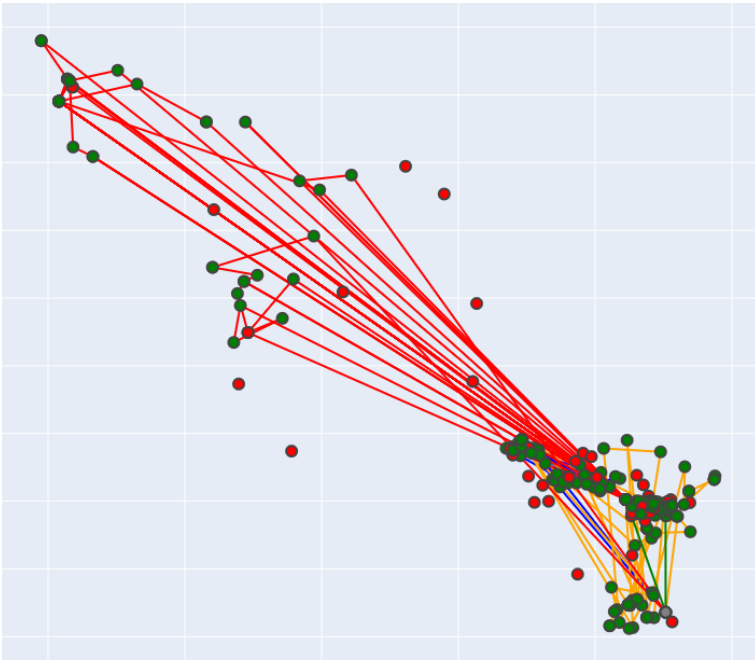


Fig. 4.6 Original Static Routes Implemented by Aramex

From Table A.3, the routes deduced from the traditional approach have utilised all four vehicles which reflects the highly constrained problem. Moreover, from the routes plot shown in Figure 4.6, it can be seen that some customers have been missed that are not included in any vehicle route, while routes arcs are crossing, showing unnecessary cross paths that increase travelled distance. Vehicle routes are broken down into arcs. Arcs of specific colour represent a single-vehicle route from Table A.3, blue for Vehicle 1, yellow for Vehicle 2, green for Vehicle 3 and red for Vehicle 4.

For the solution quality, Table 4.10 presents outputs related to the KPIs considered in this study. As all vehicles have been utilised, V is considered to be 4. TD is calculated Euclidean distances between customers in the provided original customer sequences per vehicle and then added up across all vehicles. WT is assumed to be zero as there is no measure to collect and provide this data. The traditional delivery method resulted in 53 missed customers (CM); this could be attributed to various reasons, including unavailable customers at collection or unsuccessful reach to the customer location.

Given that the considered case problem is not capacitated, each customer demanded quantity is assumed to be 1; as a result, the missed demand (DM) is 53 as well. In addition, capacity violations VQ_{V_r} are not applicable in this problem. Similarly, vehicles are not constrained by any duration limit; therefore, duration violations $Vdur_{V_r}$ are also not applicable. Finally, time window violations VTW_{V_r} were calculated based on the cumulative differences across all served customers, from the actual delivery time and the preferred customer time window, which resulted in 993.20 hours.

As a general insight into the traditional routing approach implemented by Aramex, service delivery is underperformed by missing customers or serving them out of their preferred time window. This behaviour is explained by routing decisions made by the individual drivers who perform customer sequencing without a clear optimisation rule or method. This inefficiency made the case to test the proposed approaches on this static implementation.

4.5.1.2 Hybrid and Centralised Static Implementations

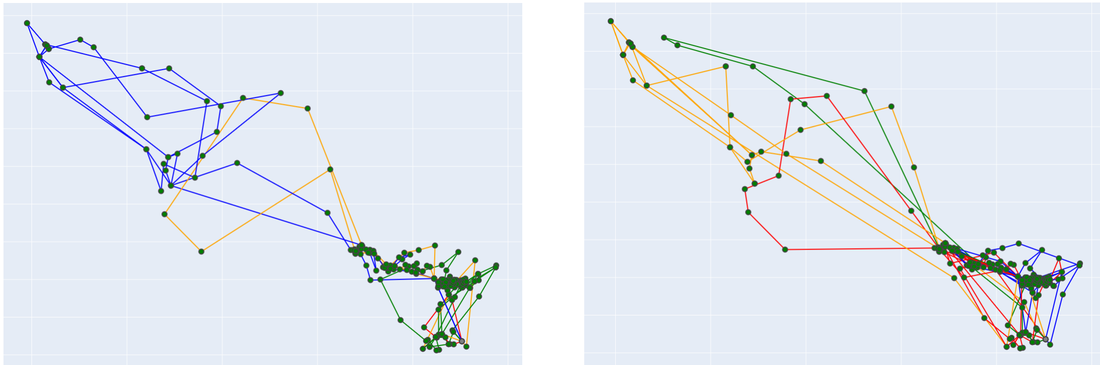
The proposed approaches, hybrid and centralised, are tested against this static case. The resulted routes' KPIs are compared with the traditional solution outputs, provided in Table 4.11. Table 4.11 reports the resulting best solution on the case study instance for the hybrid approach and averaged across five runs for the centralised approach. Their improvements to the original solution method are also reported. Given that the Original (Org) solution is determined intuitively, not computed, by each driver, CPU comparison is considered not applicable. Solution maps for both approaches are presented in Figure 4.7 while their best routes are provided in Tables B.3 and B.4

Table 4.11 Hybrid and Centralised Results of Static Case Study

Objective	H	H-Org%	C	C-Org%
V	4	0.00%	4	0.00%
TD	269.99	-68.46%	403.66	-52.85%
WT (Hr)	1.49	1.49	0.13	0.13
CM	0	-100.00%	0	-100.00%
VTW_{V_r}	0	-100.00%	0	-100.00%
CPU(Hr)	0.20	N/A	91.75	N/A

From Tables B.3 and B.4, it can be seen that the hybrid approach has exhausted the first three vehicles while providing Vehicle 4 with a very light workload of 8 customers. On the contrary, the centralised approach managed to allocate more balanced across all vehicles. This behaviour of the hybrid approach is attributed to its route building approach that iteratively exhausts a vehicle and then initiates another route. Concerning the solution maps representing the routes from both approaches, no red dots are present, meaning no customers were missed. Furthermore, route arc crossing is reduced compared to the traditional solution indicating total distance reduction in both approaches.

Based on the reported quality outputs in Table 4.11, the hybrid approach resulted in routes that utilised all the four vehicles, which remained similar to the original problem



(a) Hybrid (b) Centralised

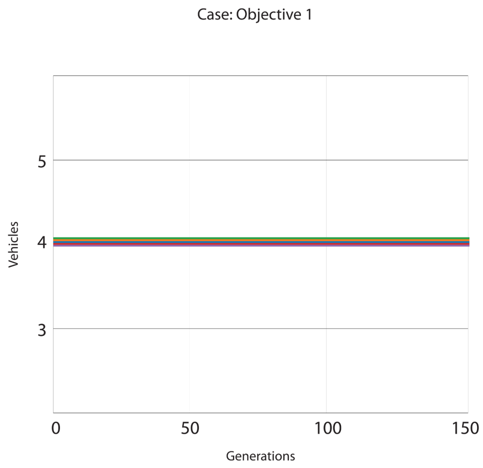
Fig. 4.7 Resulted Routes for the Case Study using the Proposed Approaches

Table 4.12 Hybrid Best Parameters on Case Study

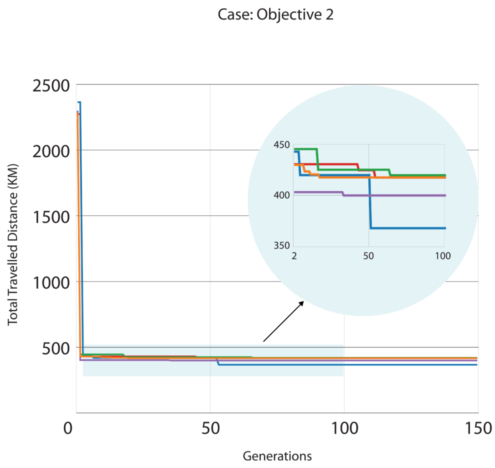
C Rule	μ	α_1	α_2
Far_Avg	1	1.0	0.0

solution. A significant reduction in the total distance resulted in 269.99 KM with a 68.46% reduction compared to the traditional routing approach. On the contrary, waiting times have increased by 1.49 hours compared to the original routing solution. The hybrid approach resulted in no time window violations and included all the customers resulting in a 100% reduction in both measures. Finally, the CPU time for this run was 0.20 hours which equals 12 minutes. The hybrid parameters resulted in this best run are summarised in Table 4.12 that is based on all runs reported in Table B.1. Those resulted best parameters indicate that the best solutions emerge for this problem by prioritising customers and customers' insertions based on distances, for which the Far_Avg rule is favoured, and α_1 is set to 1. The solution map for this approach is illustrated in Figure 4.7a.

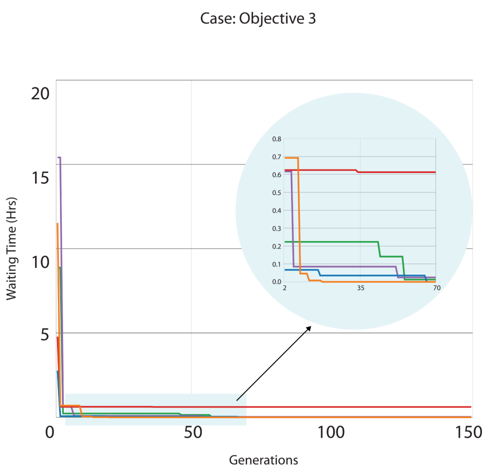
On the other hand, the centralised approach showed similar improvement in behaviour. Its averaged results resulted in 4 utilised vehicles with no improvement from the traditional solution and the hybrid approach, implying a highly constrained problem. Total distance measure has resulted in 403.66 with around 52.85% reduction from the original solution. Waiting time has been nearly eliminated, resulting in 0.13 hours, equivalent to around 8 minutes. No violations and no missed customers have resulted from this approach which has been eliminated compared to the original solution. The approach has seen a significant run time of 91.75 hours, equivalent to around four days. Such a high run time for the centralised approach questions its applicability for its use in a dynamic context. The best solution map for this approach is illustrated in Figure 4.7b with the resulted KPIs of V = 4, TD = 366.58



(a) V



(b) TD



(c) WT

Fig. 4.8 Centralised GA Runs for Each Objective, Case Study

KM, WT = 0.03 Hours. The GA objectives improvement across all the five implemented runs are illustrated in Figure 4.8.

Figure 4.8 illustrates the objectives' improvement run of each GA run for the Case Study. Figure 4.8a shows that all runs have not reduced the number of vehicles, and all were fixed on four utilised vehicles for every generation. Figures 4.3b and 4.3c demonstrate improvements in total travelled distance and waiting times where they considerably reduce after the first or second generations. The runs may seek further improvements in distance and waiting times objectives after generation 5 until generation 55, then remain steady until the 150 generations. The significant improvement in the first two iterations is attributed to the education operator, which utilises a greedy local search that can not seek further improvements in later generations.

The overall behaviour of both approaches is similar compared to the traditional solution. No reductions reported in the number of vehicles explain the highly constrained problem that requires all vehicles to be utilised to cover all customers. Significant distance reduction is explained by implementing a proper optimisation method compared to the traditional implementation based on drivers' routing decisions. The waiting time increase is attributed to the original data's lack of total waiting time reports. Covering all customers with no violations in the resulted routes of the proposed approaches is attributed to the suitable implementation of a proper optimisation method compared to the traditional driver intuitive solution.

Solutions resulting from both approaches differed in the total distance and waiting time KPIs due to the different optimisation strategies implemented. The hybrid approach first seeks to minimise vehicles used and then minimise the total distance in its route construction. At the same time, the latter considers multiple-objective optimisation with no objective priorities. The hybrid approach managed to decrease the total distance by 133.67 KM; however, at the expense of the waiting time, that resulted in a 1.46-hour difference, for which the centralised approach has nearly eliminated. This similar behaviour is seen when both approaches are applied and compared on the benchmark instances in 4.3.3 and explained by the non-dominance sorting of the centralised approach. Finally, run times have seen a significant difference; the hybrid approach has proved its superiority in this problem and makes it applicable for the dynamic problem for continuous optimisation that must be implemented quickly. The high run times provided by the centralised approach can be explained by the company's time data, which are in seconds and applied here accordingly.

4.5.2 Dynamic Breakdown Implementation

This section further tests the proposed agent-based model for the adopted case study problem. The aim is to optimise any possible breakdown event(s) that could occur during the execution of the routes. Based on the interview question in Table A.2, the case company reports that they may face breakdown(s) while vehicles are in-service; however, breakdown logs are not maintained. Therefore, random estimate scenarios are provided based on expert knowledge. This subsection explains the set-up of the dynamic breakdown experimental settings while reporting their results with analysis and discussions.

4.5.2.1 Dynamic Breakdown Problem Settings

In order to simulate the case study instance to a dynamic problem, the routes are simulated, and breakdown randomisation is proposed to target operating vehicles randomly selected at random times. Breakdown events could occur multiply per day, meaning multiple in-service vehicles could face a breakdown at different times during the day.

The breakdown event is defined by its mean inter-arrival time λ_{bd} that follows an exponential distribution, approximated to Poisson as the model deal with discrete-time ticks. Once a breakdown event occurs, as per the specified randomisations, an operating vehicle is arbitrary selected to face a breakdown. However, a key question arises on how to estimate λ_{bd}. Based on expert knowledge, they are estimated into three scenarios:

- Scenario 1: breakdowns occur early in the shift, and more frequent
- Scenario 2: breakdowns that occur mid-shift with intermediate frequency
- Scenario 3: breakdowns occur later in the shift but are less frequent

To parametrise these scenarios, λ_{bd} is set to three numerical settings: quarter ($^1/_4$), half ($^1/_2$) and three-quarter ($^3/_4$) of the shift. Accordingly, the breakdown events are triggered, and the optimisation process initiates. Given that the centralised approach is costly computational and is proven unsuitable for the dynamic case, as proven in 4.5.1, only the hybrid approach is used in setting the initially planned routes and in solving the breakdown event with the parameters provided in Table 4.12.

Given the multiple possible occurrences of breakdown events per scenario, as a result, multiple optimisation processes are performed accordingly. Therefore, the output of every scenario is summarised as the overall actual achieved KPIs. In other words, every KPI reported is based on the actual delivery of serviced customers in the scenario. In addition, given the random breakdown occurrences per scenario, each scenario is run five times while the average overall KPIs are reported.

4.5.2.2 Results on the Dynamic Breakdown Case Study

This subsection applies the suitable hybrid approach to the case study under dynamic vehicle breakdowns. The resulted routes' KPIs will be compared with the results provided by the hybrid approach for the static problem, as reported by Table 4.11. The resulting averaged data on the dynamic case study settings for both the hybrid approach is summarised in Table 4.13.

Table 4.13 reports the overall results on three scenarios based on a set value of λ_{bd}. Every scenario has reported at least one breakdown. The most scenario with breakdowns is scenario

Table 4.13 Dynamic Breakdown Scenarios - Average Results

Scenario			KPIs				CPU (hr)
No.	λ_{bd}	BD	V	TD	WT	CM	-
1	$1/4$	3.0	4	199.01	1.05	119.60	0.20
2	$1/2$	1.6	4	251.37	1.45	27.40	0.19
3	$3/4$	1.0	4	265.19	1.49	8.40	0.19

1, with the shortest breakdown inter-arrival times, resulting in three average breakdowns. Followed by scenario 2, which resulted, on average, in 1.6 breakdowns. While the least breakdown events occurred in scenario 3, with the highest λ_{bd} set to three-quarters of a shift with one breakdown on average. This behaviour is expected because as the inter-arrival times increase, the breakdown events become distant, resulting in fewer breakdowns in the same shift.

For the resulted outputs, all scenarios have utilised all of the four vehicles, given that their initial solution has utilised these four vehicles initially, as reported by the hybrid approach in Table 4.11. It is difficult to reduce the vehicle used, especially after a breakdown event. On the other hand, the total distance travelled and waiting times have reduced. At the same time, some customers are evidenced to be missed in such dynamic scenarios compared to the hybrid static results reported in Table 4.11.

Scenario 1, with the most frequent breakdowns, has seen the most significant missed customers averaging 119.6 customers, followed by a reduction in travelled distance and waiting times with around 71 KM and 0.4 hours reduced, respectively, compared to the optimised hybrid solution. The second scenario has seen fewer changes with an average of 27.4 customers missed and reductions in 18.62 KM and 0.04 hours in total travelled distance and waiting times, respectively. Finally, the slightest changes are reported in the third scenario with the least frequent breakdowns with an average of 8.4 missed customers and distance reduction of 4.8 KM and no change in waiting times.

As a general insight on the breakdown scenarios, it can be seen that the decrease in the inter-arrival times of breakdown events results in more breakdowns. Accordingly, the results' output highly deviates from the originally planned, compared to the static hybrid implementation in Table 4.11, and vice versa.

The significant reduction in the travelled distance in the more frequent breakdown problem is explained by the infeasibility to serve the missed customers that vehicles can not serve on time given the new dynamic situation. As a result, not including these customers in the vehicles' routes; hence decreasing the travelled distance. On the contrary, waiting times have seen slight reductions given the need for more utilisation of vehicles' time to adapt to the dynamic change.

Regarding the CPU times, all scenarios resulted in average optimisation time, for generating routes only, in 0.2 hours, which is mainly attributed to the initial routes generation prior breakdowns given the problem sized is reduced as customers are served not included in the breakdown problem.

4.5.2.3 Routes Visualisation

To further verify the resulted dynamic solutions provided by the hybrid approach, route visualisations are provided. As stated in the problem statement section 1.6 and illustrated in Figure 1.1, the dynamic breakdown optimisation for a delivery problem requires a demand pick up first at the disrupted vehicle location. Therefore, a highlighted route section, shown in Figure 4.9, for a particular breakdown section is highlighted to demonstrate this pickup node.

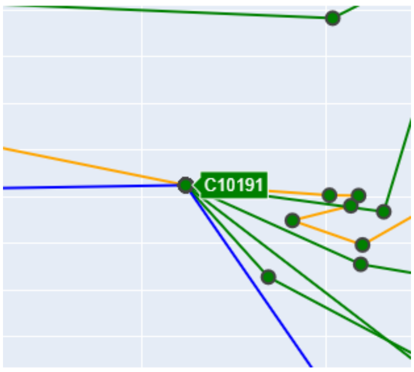


Fig. 4.9 A Pickup Node Example for Customer 191 represented as C10191

Figure 4.9 highlights a particular pickup node at a disrupted vehicle location where customer 191 demanded quantity is located. The collection node identification number is added by 10,000 to distinguish this dummy collection agent from its original delivery agent.

To further visualise the overall breakdown events, selected runs, one for every scenario, are visualised in Figures 4.11, 4.12 and 4.13. The initial solution for every case, prior the breakdowns, is the same solution provided by the hybrid approach with the resulted routes shown in Table B.3 and illustrated in Figure 4.7a. In every sampled scenario run, solution maps after the breakdown events are illustrated, in addition to the over all solution map. The solution routes for every breakdown event for these runs are provided in Appendix B, Tables B.5 through B.13.

Table 4.14 and Figure 4.10 show an example from scenario 1 for which only vehicle 3 route generated after the third breakdown. The table presents the sequence of customer identification numbers starting from customer 133 and ending at customer 88. This route is represented in the figure highlighting the route (blue) connecting the served customers' nodes (green) which highlighting the missed customers in red.

Table 4.14 Vehicle Route Example of Scenario 1 after the 3rd Breakdown

Veh.	Route
3	[133 124 152 135 109 101 104 89 148 127 136 90 88]

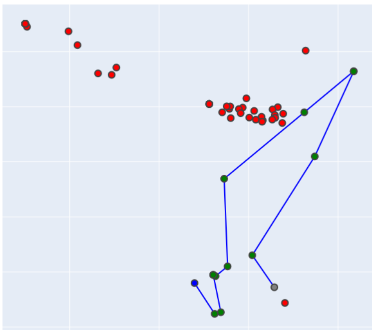
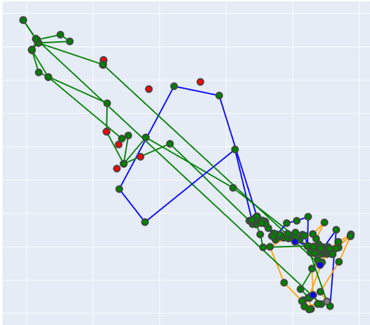


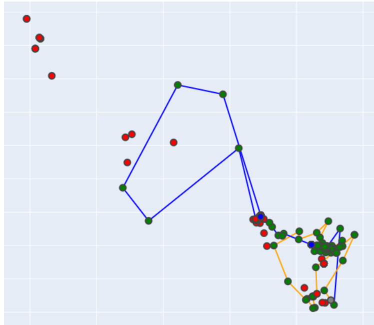
Fig. 4.10 Map example of Scenario 1 after the 3rd Breakdown

Figure 4.11 illustrates solution maps for a sampled run of the first scenario where the inter-arrival times are the shortest resulting in more frequent breakdowns. Figure 4.11a, where the first breakdown occurred, shows that the method managed to route most of the customers but missed 15 and started to heavily miss customers in the following second and third breakdowns, as shown in Figures 4.11b and 4.11c by missing 42 and 36 customers, respectively. Finally, the actual routes are represented in Figure 4.11d with 93 missed customers out of 300.

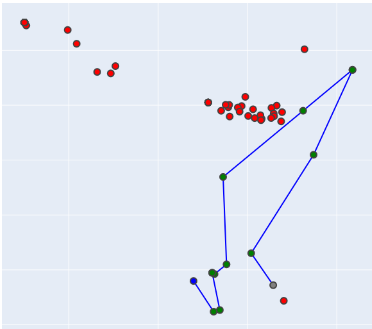
The routes that represent these illustrates, in their respective order, are summarised in Tables B.5, B.6, B.7 and B.8. Table B.5 shows the routes after the first breakdown faced by Vehicle 1, which was missing from the resulted routes. The optimised routes indicate that Vehicles 2 and 3 have progressed with their routes and did not accommodate any disrupted customers. In contrast, Vehicle 4 managed to accommodate most of the disrupted customers and insert them before and after its original route as compared with the original solution routes shown in Table B.3. For the second breakdown, Table B.6 shows missing Vehicle 4, meaning it was disrupted and Vehicles 2 and 3 and just continuing with their progressed routes from the earlier breakdown event from Table B.5 and similarly for the third, in Table B.7, where Vehicle 2 is disrupted. Vehicle 3 is just continuing its route from before.



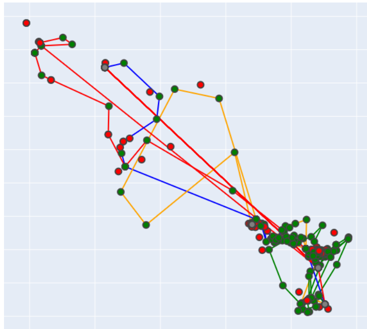
(a) 1st Breakdown, 15 missed customers



(b) 2nd Breakdown, 42 missed customers



(c) 3rd Breakdown, 36 missed customers



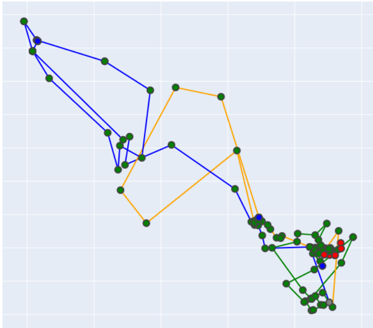
(d) Overall Routes, 93 missed customers

Fig. 4.11 Scenario 1 Solution Maps, λ_{bd} = $^1/_4$ of the Shift

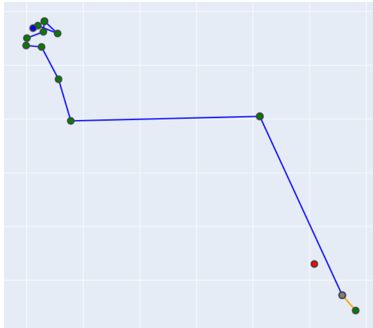
Finally, Table B.8 shows the overall routes for the day, which is what each vehicle has served, including the disrupted vehicles.

The inability to accommodate the disrupted customers in later breakdowns is attributed to the infeasibility to serve customers within their time window and possible violation of each of the vehicle shifts. Furthermore, the inability to serve these customers lies in the capacity reduction due to the breakdown, given that the problem considered is originally tight in serving these customers. This is evidenced by the original case solution, Table 4.10, and the outputs are based on the proposed approaches, Table 4.11, where all vehicles have been utilised.

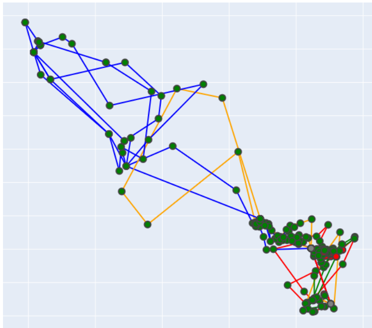
Based on the illustrations in Figure 4.12 for the case with intermediate level of inter-arrival times, this run resulted in two breakdowns. The routes generated after the first breakdown have resulted in 11 missed customers while the second resulted in only 1 missed customer, as shown in Figures 4.12a and 4.12b, totalling to 12 missed customers. The resulting overall routes are presented in Figure 4.11d. The routes representing these solution maps are summarised in Tables B.9, B.10 and B.11, respectively.



(a) 1st Breakdown, 11 missed customers



(b) 2nd Breakdown, 1 missed customer

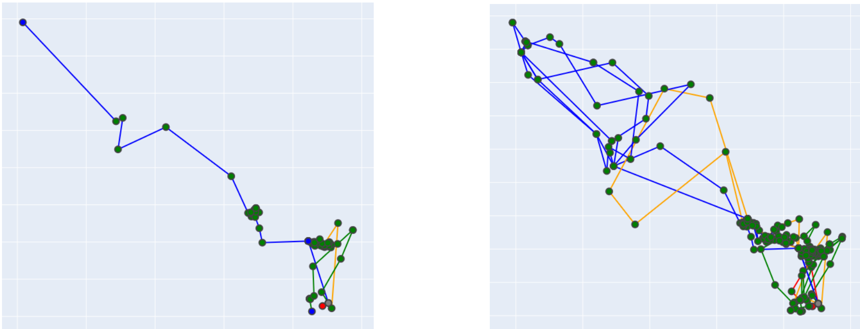


(c) Overall Routes, 12 missed customers

Fig. 4.12 Scenario 2 Solution Maps, λ_{bd} = 1/2 of the Shift

The routes representing these solution maps are summarised in Tables B.9, B.10 and B.11, respectively. Table B.9 summarises the optimised routes for the first breakdown event, where Vehicle 3 is disrupted. Vehicles 1 and 2 have progressed with their routes and did not accommodate any disrupted customers, while again, Vehicle 4 managed to reshuffle its route sequence to consider most of the disrupted customers. Table B.10 summarises the routes after the second breakdown where Vehicle 4 is disrupted. Vehicles 1 and 2 did not accommodate any disruptions, possibly due to the late breakdown and existing workload, and continued with their respective routes. The overall scenario routes are summarised in Table B.11. This run has resulted in fewer breakdowns and, accordingly, fewer missed customers due to better time availability of the vehicles. Fewer breakdowns are attributed to the increased inter-arrival times compared to the previous run.

Finally, Figure 4.13 illustrates the run where the breakdown events inter-arrival times are the longest and result in less frequent breakdowns. It can be seen in Figure 4.13a, where the first and only breakdown occurred, that the hybrid approach managed to route nearly all of the customers but missed only one customer. The overall routes are represented in



(a) 1st Breakdown, 1 missed customer

(b) Overall Routes, 1 missed customer

Fig. 4.13 Scenario 3 Solution Maps, λ_{bd} = $^3/_4$ of the Shift

Figure 4.11d totalling only one missed customer out of 300, resulting in the minimum routes changes compared to the previous dynamic scenarios. Contrary to the previous scenarios, the less frequent breakdowns due to the longest disruption inter-arrival times have maximised the availability of vehicles to route nearly all the customers in this heavily constrained problem.

The routes representing the solution maps for this case are summarised in Tables B.12 and B.13, respectively. Table B.12 represents the optimised routes after the only breakdown, where Vehicle 4 is disrupted. The remaining vehicles have progressed with their routes and did not accommodate the only disrupted customer, the last customer in Vehicle's 4 route. The missing customer is evidenced in the overall routes represented in Table B.13 where the only difference between the original planned routes from Table B.3 is the missed customer 96 from the end of Vehicle 4 route.

Overall, the case problem considered is a highly constrained one, and any sudden unavailability of a vehicle, the model would result in routes missing some of the disrupted customers. The service level, or customer coverage, is inversely proportional to the breakdowns' inter-arrival times. The more distant the breakdown event, the more time the vehicles become available to service the customers and less likely to miss any disrupted ones, and vice versa. For this particular problem instance, Vehicle 4 is considered a backup vehicle. If disrupted, the solution approach starts to miss customers. An introduction of an additional backup vehicle would considerably improve the service level.

4.6 Chapter Summary

This chapter reported the experimental settings: the used hardware/ software and the parametric settings for each of the proposed approaches, then presented the analysis and discussion of the experiments performed. The program is implemented on a High-Performance Computer (HPC) coded in Python. The hybrid parametric setting is summarised in Table 4.1 that considers a range of rules and Solomon's insertion heuristic parameters, while the centralised parametric setting, summarised in Table 4.2 considers evolutionary metaheuristic parameters adapted from literature. Finally, the output KPIs of an experiment is summarised and abbreviated in Table 4.3.

Firstly, experiments were conducted on MDVRPTW benchmark instances to test the quality and performance of the proposed approaches against the best-known solutions. Tables 4.5 and 4.6 summarises the respective results of the hybrid and centralised then compared in Table 4.7. The hybrid approach resulted in new best solutions in terms of minimising the number of vehicles that was further reduced by around 5% on average. In contrast, the centralised approach further minimised waiting times and reduced by an average of 2.21-time units, given its non-dominance based objective sorting, however, at the expense of other objectives and the computational time.

Secondly, those MDVRPTW instances are further modified to verify the suitability of the proposed approaches to the vehicle breakdown problem under study. The hybrid and centralised approaches are compared in Table 4.8. Each resulted in different behaviour given their different constraint handling strategies. The hybrid approach would rather miss a customer than violate its constraints and vice versa for the centralised. Results in missing customers are evident in the reported results where the hybrid approach resulted in missed customers while the centralised approached managed to serve every customer without any violations. The hybrid approach is seen to miss more customers in the tighter time window instances and accordingly reduce distances and vehicles as these customers are not included in the plan; as a result, they are not included in these KPIs calculations. The main benefit of the hybrid approach is its significant efficiency in generating routes, in seconds, compared to the inefficient centralised approach, in hours.

Thirdly, a case study for a particular day of operations is applied. The original company's traditional static solution is reported in Table 4.10 then optimised using both proposed approaches compared to the original solution, as reported in Table 4.11. Both approaches resulted in significant improvements in eliminating time window violations, serving all customers and considerably reducing the distance travelled, 68% and 57% distance reduction using the respective hybrid and centralised approaches. A similar computational issue is faced in the benchmark experiments with the centralised. It took around four days to run for this particular case compared to only 12 minutes for the hybrid, which makes the centralised not efficient and unsuitable for the dynamic case to be solved continuously.

Finally, the same case study is simulated as a dynamic vehicle breakdown problem where breakdown events are randomised in their time of occurrence. Three random scenarios are generated based on three set values for the average inter-arrival times of such breakdown events: quarter ($1/4$), half ($1/2$) and three-quarter ($3/4$) of the operating shift. Each scenario is run five times and solved using the hybrid approach only, and the average results are reported in Table 4.13. All scenarios have utilised all available vehicles, given the difficulty of minimising them after a breakdown event where the capacity is reduced. Given the highly constrained problem after the breakdown, several customers are missed. As a consequence, vehicles are not visiting those customers, thus, reducing the total distance travelled and waiting times. The degree of route deviations correlates to the breakdown parameter as the inter-arrival times between breakdowns decrease, resulting in deviations in routes and their KPIs increase. Routes visualisations of sampled dynamic runs of each scenario are provided, analysed and discussed for further model verification for the dynamic vehicle breakdown delivery problem.

Chapter 5

Conclusion

5.1 Chapter Overview

This chapter concludes the context of this thesis sequentially. It firstly concludes the literature review chapter, which relates to the identification of the gap in knowledge and theoretical background of related techniques. It then relates to the methodology chapter by providing key conclusions from previously adopted optimisation approaches in agent-based VRP and the proposed conceptual model while summarising both proposed approaches. Next, knowledge gained and conclusions from the provided results are listed for every experimentation type: benchmarks, modified benchmarks and a case study. Finally, the limitations of this research are addressed with prospects for future research.

5.2 Conclusion from Literature Review

This thesis literature review, Chapter 2, has reviewed the relevant routing problems under dynamic real-time updates while surveying their specific problem variants and their adopted solution methods. This review has supported this study in positioning itself to the existing body of knowledge by identifying a knowledge gap in time window routing problems where vehicles may face multiple and random disruptions during the execution of planned routes. In addition, the survey of tools and techniques implemented in the previous studies has provided this research with the necessary technical and theoretical background in addressing the knowledge gap and the unique problem under study.

From the previously adopted techniques in similar problems, approximation algorithms, such as (meta)heuristics, are mainly used due to their ability to scale to large applications by finding near-optimal solutions. However, even such pragmatic approaches are still mainly adapted from static problems, thus, lacking the ability to adapt to dynamic problems, especially for a dynamic vehicle breakdown problem. The breakdown problem requires modelling updated positions of vehicles by considering unique locations of vehicles; therefore, a heterogeneous problem is considered with its entities having unique attributes, locations in particular. The agent-based modelling approach is emerging in optimising dynamic routing problems, attributed to its ability to simulate a dynamic problem, capture entity uniqueness through its agents' definitions, and produce feasible solutions by utilising proper agents' interactions.

Furthermore, based on the review of the VRP cases, the optimisation objectives are generally concluded and measured when testing the quality of the produced routes. Solution quality is measured against three primary KPIs: the number of utilised vehicles, total distance travelled, and total waiting times. Any additional VRP objectives are extended from those three. For example, a VRP study considers cost minimisation; it refers to the distances and vehicles multiplied by their unit costs. Additional KPIs may be required in case of any constraints relaxations as the majority of studies in time window considered relaxing of customers' time window constraints, while few studies considered relaxing coverage of all customers and other vehicle constraints, capacity and duration limits. Usually, such relaxations are just to ease the algorithmic search and are not reported at the end, given that methods overcome them and produce solutions with no violations.

As a result, this chapter has provided the required knowledge from routing models, simulation methods, performance measures and optimisation techniques to propose a solution approach for this study.

5.3 Conclusion from Agent-based Optimisation Approach for VRP

Chapter 3, methodology, has provided justifications for the adoption of the agent-based approach for the problem under study. Section 3.2 highlights key studies that adopt this approach in optimisation for similar problems. It is concluded that this approach is adaptable to dynamic disruption events compared to the traditional static approaches that are inflexible in tackling updates because of the latter's rigid modelling nature and is computationally expensive. These reasons make the traditional static approaches inefficient in solving sudden dynamic updates and make them unable to produce solutions quickly.

Furthermore, Section 3.2.1 lists the different types of agent-based optimisation approaches that are based on the types of agents' interactions, distributed and centralised. Section 3.2.2

concludes their trade-off for which the centralised is more complex and time-consuming but better in assuring optimality. At the same time, the distributed approach is quick and less complex but does not assure optimality. This study labels the distributed approach as hybrid as it is not fully evidenced to be fully decentralised in VRP implementations.

This trade-off is evidenced in this study's results as both approaches are developed and tested on benchmark instances, modified instances and a case study. The hybrid approach produced quick and good solutions in minimising the number of vehicles used, with a 5% reduction, in benchmarked instances within 20 seconds. On the other hand, the centralised further improved waiting times, with a 2.20-time units reduction, due to its multi-objective approach; however, it took hours. Similar behaviour is seen in modified benchmark instances and the adopted case study. Detailed conclusions on each are listed in 5.7.

The experimentations with both proposed approaches have benefited the study in exploring the right one for the problem considered. It is concluded that the hybrid approach is the most suitable given its efficiency in producing high-quality solutions for all experiments compared to the best results in benchmarked instances and compared to the centralised approach in the modified instances and the case study. The justification for this conclusion is evidenced in the static implementation of the case study, concluded in 5.7.3, where the centralised approach produced results within days, hindering their applicability for a dynamically responsive solution method.

5.4 Conclusion from the Agent-based Conceptual Model

Given the decision to implement two different solution approaches, designing a flexible conceptual model is necessary to accommodate both approaches. Based on the in-depth understanding of the problem entities, a model, shown in 3.3, is proposed that consists of different agent types that resemble the problem under study. Customer and Vehicle agent types are considered; each has its unique attributes and constraints, which allowed the modelling of unique vehicle problems, where each vehicle has its starting and ending locations, shifts and capacities. Potential breakdowns from vehicles are made possible. Furthermore, a super agent, an assignment agent, is also introduced to govern the agents' interactions based on the adopted approach.

Based on the literature review conclusions in 5.2 concerning the relevant VRP KPIs, the model's output routes are measured against those defined KPIs. However, each adopted approach has a different constraint handling strategy; as a result, they differ in the reported violations. The different relaxation strategies are justified based on the implemented approaches, each summarised and concluded in 5.5 and 5.6.

As evidenced in the implementation of the modified benchmark instances in 4.4 and the dynamic case study in 4.5.2, the model's dynamic input data have been successfully accommodated to the breakdown instant problem and then solved accordingly. The model simulates the planned routes until the breakdown instant in a dynamic delivery scenario, translating into a specific pickup and delivery problem for the disrupted customers. This problem has been successfully modelled and optimised, as evidenced in Figure 4.9 where the pickup precedence is illustrated and verified. Furthermore, the model is applied to solve multiple breakdowns per shift and the respective breakdown routes are evidenced in the Appendix B, Tables B.5 to B.13. This evidence demonstrates the achievement of the research's aim of solving the routing problem under multiple breakdowns.

5.5 Adoption of the Hybrid Approach

Given that there is no evidence of distributed optimisation in routing problems, a higher degree of centralisation is required for which this hybrid approach was proposed. This approach, proposed in section 3.4, adopts a sequential route construction method for the time window problem.

Given that this thesis aims to model breakdowns in a delivery problem, the previously implemented route construction methods are modified accordingly. Two critical factors need modifications to the existing route construction approach. The first is the unique vehicles' location at the instant of the breakdowns, and the second is the collected precedence requirement for the disrupted customers.

The unique location problem is that each vehicle could have different starting and ending route positions. The problem is tackled by proposing an evaluation procedure done at the vehicle agent level, as shown in 3.4.2. Furthermore, the definition of more rules for customer prioritisation classifies distances across all vehicles, averaged or minimum, given that distance measures between vehicles and customers differ from one vehicle to another. Contrary to previous work, where a customer to all vehicles measures are the same given that vehicles start and end their routes at a centralised depot.

The collection precedence part of the problem is where a visit to the broken down vehicle is mandatory to collect the disrupted customers' orders prior to delivery. This part translates into a pickup and delivery problem. Therefore, the method is further adapted by considering every disrupted customer with two orders: the first is a mandatory prerequisite collection at the disrupted vehicle location, and the second is the actual delivery at the customer location.

The approach has successfully modelled the breakdown delivery problem by accommodating these two significant modifications, as demonstrated in 4.5.2 for the dynamic breakdown implementation of the case study. To sum up the procedure of this approach, the overall agent interactions, dubbed Messaging Protocol-Based Heuristic Optimisation (MPHO), where the process is divided into two stages. The first stage starts with the assignment agent, who prioritises customers based on pre-specified rules. The most priority customer is privileged to select and initiate the vehicle route while the vehicle evaluates the feasibility of the customer; upon routing, the second stage initiates when the assignment agent provides a list of the remaining customers to the vehicle allocated in stage one to consider routing them at once. The process is repeated until no more customers remain or vehicles are exhausted. If there is any constraint violation at any stage, customers will be missed.

5.6 Adoption of the Centralised Approach

From its name, centralised, this approach optimises the problem while looking at the problem from a central perspective, in this case, at the assignment agent level. This thesis also aims to reduce the costs associated with routes; therefore, further optimisation of the KPIs is sought. The drawback behind the previously developed hybrid approach is that it is based on the localised measures in evaluating routes locally at the vehicle agent level, limiting the approach to global objective search given the absence of a centralised evaluation. Section 3.5 proposes this centralised approach where the assignment agent is aided with a Genetic Algorithm (GA) and dominance-based Pareto ranking in searching for a better global optimum without prioritising any of the objectives or KPIs.

The critical aspect of this implementation is to capture the logic behind the problem-dependent components as the other metaheuristic and multi-objective components can be easily implemented by adopting a proper metaheuristic framework. The problem-dependent components consist of solution representation, evaluation and variations. The metaheuristic solutions are represented in "path representation", which has been extended to population-based representation where every vehicle agent can have multiple route attributes to relate to the overall population. Solutions are evaluated centrally by the assignment agent to ensure the calculation of the global objectives aided with localised evaluation at the vehicle agent level where capacity and duration limits constraint violations are calculated, later penalised by the metaheuristic framework. Finally, solution variations are adapted from the literature.

Concerning the solution quality produced by this approach, it has further improved waiting times in all static implementations and covered all customers without any violations. Waiting times have been reduced by 2.20-time units compared to best-known solutions in benchmarked instances, by around 62-time units compared to the hybrid approach in modified instances and 1.36-hours in the adopted static case study. However, these improvements resulted in a slight increase in distances compared to the hybrid approach, with around a 13% increase in benchmark and modified benchmark instances and a 50% increase in the adopted static case study. This trade-off between the distance and waiting time objectives is attributed to the Pareto sorting that does not prioritise any objective over the other. Computationally, this approach has been costly, which took hours to produce solutions in benchmark and modified benchmark instances, and it took days for the static case study, which deemed this approach inapplicable for a dynamic scenario.

5.7 Knowledge Gained From Experimentations

This thesis further verifies and validates the proposed approaches in Chapter 4. The chapter firstly reports the settings of the implementations in section 4.2 by summarising the parametric settings for both the hybrid and centralised approaches, which are adapted from literature. Given the uniqueness of the problem and the modelling approach, a customised implementation is needed; therefore, the model was programmed entirely using an object-oriented programming language (Python).

Given the lack of dynamic benchmark instances, the proposed approaches are firstly tested against static instances, then modified to resemble a breakdown problem instant and finally on a case study. Conclusions of each are summarised in the following subsections 5.7.1, 5.7.2 and 5.7.3.

5.7.1 Conclusion from Benchmark Experimentations

Section 4.3 compares the quality of the solutions produced by the proposed approaches against best-known solutions for Multiple Depot VRP with Time Window MDVRPTW benchmark instances. This variant is judged to be the closest time window variant for a vehicle breakdown problem where vehicles are not in one location. Key conclusions are obtained from those benchmark instances experimentations:

- Both proposed approaches are proven superior in minimising the number of vehicles compared to previous best solutions. The hybrid approach explains its superiority by its adopted route construction method that does not initiate additional routes until previous vehicles are exhausted. While in the centralised, the explicit definition of the minimum number of vehicles objective contrary to the previously best two approaches from literature, that only considered one of the other objectives. Both proposed approaches reduced the number of utilised vehicles by around 5% on average.

- The centralised approach is proven superior in minimising the total waiting times given its multiple-objective approach in Pareto sorting that does not prioritise any of the objectives. Contrary to the previous implementations, the proposed hybrid approach adopts a single objective function or a localised evaluation within a heuristic. As a result, the proposed approach reduced waiting times by 2.20-time units on average.

- By comparing both proposed approaches, they resulted in solutions that differ in the total travelled distance and waiting time objectives. Compared to the centralised approach, the hybrid approach reduced the travelled distance by an average of 12.03% while increasing waiting times by 242.39-time units. This behaviour is explained by the ability of the centralised approach to consider centralised evaluation of the objectives and favour solutions with Pareto dominance rather than just running a localised heuristic rule as implemented in the hybrid approach.

- The hybrid approach has demonstrated its efficiency in producing high-quality solutions with up to 20 seconds of computational time. The quick implementation is attributed to its distributed nature in agents' interactions. Contrary to the centralised approach, it takes hours to produce solutions because of its centralised evaluation and multiple objective sorting.

5.7.2 Conclusion from Modified Benchmark Experimentations

Section 4.4 compares the proposed approaches against modified MDVRPTW instances. Modifications are mainly targeting the unique vehicle locations due to the breakdown problem instant, where every vehicle starting and ending locations are randomised. In addition, randomisations of other vehicle attributes are considered. As a result, the following conclusions are obtained:

- Both proposed approaches captured the uniqueness of vehicles' attributes; therefore, they are deemed applicable for optimising the vehicle breakdown problem instant.

- With the proposed modifications, the instances became highly constrained. As a result, the hybrid approach is reported to miss customers given its rigidity in constructing routes and does not improve them further. It is reported that the approach missed, on average, around 40 customers with around 554 of their average demanded quantity.

- The hybrid approach demonstrated its sensitivity to the width of customers' time windows. The wider a customer time window is, the more likely the approach will result in missed customers due to the early commitments to prioritised customers that hinders the ability to alter the solution to later commitments for the same customer, thus, missing others. On average, the approach missed 9.4 customers in tight time window instances while around 71 customers in wide time window instances.

- The centralised approach is proven superior in servicing all customers with no violations due to its ability to alter solutions.

- The centralised approach has proven its superiority in all KPIs for the tight time window instances with 3.28%, 7.48% and 91.81 reductions in the number of vehicles, distances and waiting times, respectively. It has also seen waiting time improvements of around 32-time units in wider time window instances. However, costly deviations were reported in vehicles utilised and distances compared to the hybrid approach with around 39% and 34%, respectively. Those costly deviations are attributed to the missed customers in the hybrid approach by not including them in routes, thus, reducing the cost of these objectives.

- The centralised approach is still considered significantly costly in terms of its computational complexity by requiring hours to generate solutions compared to the computationally efficient hybrid approach within seconds.

5.7.3 Conclusion from Case Study Experimentations

Section 4.5 adopts a case study where data for a representative day of operations is provided. Original actual static routes are provided for this particular day, and their KPIs are measured and compared to results generated from the proposed approaches. Based on the static implementation of this case study, the following conclusions are obtained:

- Both of the proposed approaches managed to feasibly route all customers with no violations compared to the actual routes where customers were missed in addition to significant time window violations. Furthermore, the travelled distance was reduced by around 68% and 53% in both the hybrid and centralised approaches, respectively. This reduction is attributed to systematic routing methods proposed compared to the cluster first-route second approach traditionally adopted with routing done intuitively by drivers.

- Given its multiple objective optimisation ability, the centralised approach managed to produce routes with minimum waiting time with a slight increase of 0.13 hours, on average, compared to the original solutions, which are estimated to be zero. It has also proven its consistency in producing routes across multiple runs as the original solutions.

- The hybrid approach has proven its superiority in producing optimal routes efficiently with computational time within minutes compared to its counterpart, which produced routes within days. This superiority is attributed to the extensive search proposed by the centralised with its greedy local search in addition to the time window search for which time ticks are provided in seconds. Therefore, the hybrid approach is deemed applicable for this case study under dynamic breakdowns that require quick re-optimisation.

Furthermore, the dynamic breakdown problem is simulated under different breakdown settings based on three values of the breakdown mean inter-arrival times. The values of the inter-arrival time are made relative to the overall problem shift. It experimented with a quarter, half and three-quarters of the shift. The following conclusions are obtained:

- The mandatory pickup before the actual delivery has been successfully modelled and verified through the routes produced and their visualisations.
- As the inter-arrival times of breakdown increase, it results in less frequent breakdowns and accordingly decreases the deviations from the initial plan. From lowest to highest, breakdown inter-arrival times resulted, on average, in 3, 1.6 and 1 breakdowns, respectively.
- Less frequent breakdowns decrease the chance of missing customers. In the respective breakdown cases, they missed 119.6, 27.4 and 8.4 customers. Missing customers here is attributed to the reduced capacity due to the sudden unavailability of vehicles.
- Total distance travelled and waiting times decrease when more customers are missed. Distances decreased by 71, 18.62 and 4.8 kilometres, while waiting times decreased by 0.4, 0.04 and zero hours, all in their respective order of the breakdown scenarios. This behaviour is due to the customer prioritisation adopted that results in missing non-priority customers and according excluded from routes, thus, reducing those two KPIs measures.
- The overall number of vehicles across the whole shift has not reduced and concludes the highly constrained problem that demands the utilisation of all vehicles.
- Compared to the static implementation, the computational times saw negligible differences. Therefore, it is concluded that most of the time utilised is attributed to the

initial static routes generation. The highly similar implementation time is attributed to the problem reduction after the breakdown, given that some customers are already served, thus, requiring less computational time.

5.8 Limitations and Prospects for Future Research

The main limitation of this research is the lack of dynamic breakdown benchmark instances and the explicit definitions of how to model vehicle breakdowns. This study derived the breakdown instant problem by modifying existing benchmark instances. However, it is limited as it does not represent a dynamic case with possible multiple breakdowns per shift. An introduction of standardised benchmark instances would be beneficial to test any proposed approach systematically.

Given the previous limitation, this study was motivated to adopt a case study to gather and model data related to breakdowns. However, collaborators could not provide historical vehicle breakdown logs as the company's databases do not maintain such data. Potential future research may include case study data with more breakdown data provided by the same or another case company.

Another constraint is to validate the proposed approaches for actual real-time implementation, which requires further integration of the developed model into a case company routing system. In addition, real-time implementations require further considerations of another source of dynamism, for example, time-dependent travel times. Those limitations could not be addressed in this research as they are beyond the defined scope.

The centralised approach faced significant computational expense, which may be attributed to the adapted GA parameters from literature or the improper selection of the metaheuristic components. However, the algorithmic complexity could be reduced by redesigning the components of the algorithm, performing proper tuning of the GA parameters or adopting another metaheuristic framework such as Tabu Search and Ant Colony Optimisation. Alternatively, innovative handling of the GA parameters could be proposed, which could be achieved by designing a hyper-heuristic or a machine learning method.

Both proposed approaches were applied individually to test their effect on the problem considered. However, it would be beneficial to combine them in one approach. The hybrid approach could provide an initial solution to be further altered in the centralised approach. In addition, solution variation within the centralised approach could be implemented with a distributed agents' interactions.

Finally, this work could be extended to include other VRP variants, such as split delivery, periodic and stochastic. A stochastic problem can include uncertainty in the customer demand, in terms of time window and demanded quantity. Furthermore, the agent-based optimisation approach is not limited to routing only but also to scheduling and rostering problems.

Appendix A

Case Study Collected Data

Table A.1 Customer Data

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
1	1	03:36:00	PM	03:00 PM - 05:00 PM	2838.81	5616.50
2	1	03:17:00	PM	11:00 AM - 01:00 PM	2838.63	5616.53
3	1	01:48:00	PM	01:00 PM - 03:00 PM	2838.77	5616.40
4	1	10:54:00	AM	03:00 PM - 05:00 PM	2837.50	5623.83
5	1	10:02:00	AM	11:00 AM - 01:00 PM	2838.10	5619.80
6	1	10:02:00	AM	03:00 PM - 05:00 PM	2838.10	5619.80
7	1	10:02:00	AM	01:00 PM - 03:00 PM	2838.10	5619.80
8	1	NA	NA	11:00 AM - 01:00 PM	2838.67	5620.80
9	1	05:43:00	PM	03:00 PM - 05:00 PM	2837.99	5619.59
10	1	09:42:00	AM	11:00 AM - 01:00 PM	2837.87	5619.19
11	1	03:09:00	PM	01:00 PM - 03:00 PM	2838.74	5616.58
12	1	11:18:00	AM	11:00 AM - 01:00 PM	2837.88	5622.19
13	1	01:56:00	PM	All Day	2838.61	5616.98
14	1	09:35:00	AM	03:00 PM - 05:00 PM	2838.43	5618.44
15	1	11:43:00	AM	All Day	2838.39	5620.54
16	1	NA	NA	01:00 PM - 03:00 PM	2838.81	5616.50
17	1	NA	NA	11:00 AM - 01:00 PM	2838.08	5621.02
18	1	04:02:00	PM	03:00 PM - 05:00 PM	2838.78	5616.17
19	1	12:34:00	PM	09:00 AM - 11:00 AM	2838.92	5617.12
20	1	03:02:00	PM	11:00 AM - 01:00 PM	2838.69	5616.64
21	1	10:15:00	AM	All Day	2837.86	5621.24
22	1	12:29:00	PM	11:00 AM - 01:00 PM	2838.93	5617.13

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
23	1	12:23:00	PM	All Day	2838.99	5617.16
24	1	12:12:00	PM	01:00 PM - 03:00 PM	2838.79	5617.84
25	1	11:08:00	AM	11:00 AM - 01:00 PM	2837.38	5623.86
26	1	09:24:00	AM	03:00 PM - 05:00 PM	2838.01	5619.10
27	1	03:02:00	PM	01:00 PM - 03:00 PM	2838.69	5616.64
28	1	09:20:00	AM	All Day	2838.02	5618.93
29	1	09:54:00	AM	11:00 AM - 01:00 PM	2838.99	5617.16
30	1	03:09:00	PM	11:00 AM - 01:00 PM	2838.74	5616.58
31	1	12:56:00	PM	03:00 PM - 05:00 PM	2838.92	5616.96
32	1	10:20:00	AM	09:00 AM - 11:00 AM	2837.77	5621.60
33	1	11:18:00	AM	11:00 AM - 01:00 PM	2837.88	5622.19
34	1	12:27:00	PM	11:00 AM - 01:00 PM	2838.96	5617.21
35	1	NA	NA	All Day	2838.47	5620.29
36	1	12:30:00	PM	All Day	2838.99	5617.16
37	1	NA	NA	03:00 PM - 05:00 PM	2838.81	5617.50
38	1	NA	NA	All Day	2838.59	5621.29
39	1	09:13:00	AM	01:00 PM - 03:00 PM	2838.05	5618.78
40	1	NA	NA	09:00 AM - 11:00 AM	2838.12	5619.45
41	1	03:09:00	PM	01:00 PM - 03:00 PM	2838.74	5616.58
42	1	NA	NA	All Day	2838.77	5616.40
43	1	10:44:00	AM	01:00 PM - 03:00 PM	2837.57	5623.10
44	1	10:10:00	AM	01:00 PM - 03:00 PM	2837.96	5620.22
45	1	12:05:00	PM	11:00 AM - 01:00 PM	2838.75	5618.13
46	1	11:28:00	AM	01:00 PM - 03:00 PM	2838.06	5622.88
47	1	11:31:00	AM	11:00 AM - 01:00 PM	2838.11	5622.62
48	1	11:37:00	AM	09:00 AM - 11:00 AM	2837.94	5621.75
49	1	09:20:00	AM	01:00 PM - 03:00 PM	2838.02	5618.93
50	1	02:17:00	PM	09:00 AM - 11:00 AM	2838.66	5617.79
51	1	NA	NA	All Day	2838.63	5616.53
52	1	03:09:00	PM	All Day	2838.74	5616.58
53	1	NA	NA	11:00 AM - 01:00 PM	2838.10	5619.80
54	1	12:03:00	PM	All Day	2838.74	5618.11
55	1	03:09:00	PM	11:00 AM - 01:00 PM	2838.74	5616.58
56	1	09:29:00	AM	03:00 PM - 05:00 PM	2838.17	5619.09

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
57	1	03:09:00	PM	11:00 AM - 01:00 PM	2838.74	5616.58
58	1	NA	NA	01:00 PM - 03:00 PM	2837.53	5618.41
59	1	NA	NA	All Day	2837.50	5617.55
60	1	NA	NA	All Day	2838.10	5619.80
61	1	10:25:00	AM	09:00 AM - 11:00 AM	2838.15	5620.75
62	1	05:43:00	PM	All Day	2837.95	5619.68
63	1	03:09:00	PM	11:00 AM - 01:00 PM	2838.74	5616.58
64	1	NA	NA	09:00 AM - 11:00 AM	2837.57	5623.10
65	1	03:09:00	PM	01:00 PM - 03:00 PM	2838.74	5616.58
66	1	12:40:00	PM	11:00 AM - 01:00 PM	2838.92	5617.12
67	1	NA	NA	11:00 AM - 01:00 PM	2838.99	5617.16
68	1	NA	NA	All Day	2838.12	5617.35
69	1	12:18:00	PM	03:00 PM - 05:00 PM	2838.63	5618.16
70	1	12:29:00	PM	01:00 PM - 03:00 PM	2837.82	5621.58
71	1	01:30:00	PM	01:00 PM - 03:00 PM	2838.21	5621.77
72	1	02:37:00		All Day	2838.83	5617.09
73	1	02:43:00		09:00 AM - 11:00 AM	2838.99	5617.16
74	1	NA	NA	09:00 AM - 11:00 AM	2837.50	5623.83
75	1	NA	NA	09:00 AM - 11:00 AM	2837.91	5618.15
76	1	09:48:00		All Day	2837.99	5619.41
77	1	NA	NA	11:00 AM - 01:00 PM	2838.10	5619.80
78	1	02:05:00		11:00 AM - 01:00 PM	2838.67	5617.66
79	2	09:46:00		03:00 PM - 05:00 PM	2838.99	5617.16
80	2	NA	NA	All Day	2834.67	5625.15
81	2	NA	NA	01:00 PM - 03:00 PM	2837.49	5626.96
82	2	05:36:00		All Day	2837.49	5625.43
83	2	03:10:00		03:00 PM - 05:00 PM	2834.96	5621.87
84	2	01:54:00		01:00 PM - 03:00 PM	2837.17	5626.14
85	2	03:15:00		03:00 PM - 05:00 PM	2834.91	5621.71
86	2	NA	NA	03:00 PM - 05:00 PM	2837.65	5624.50
87	2	03:05:00		01:00 PM - 03:00 PM	2835.49	5621.68
88	2	02:55:00		03:00 PM - 05:00 PM	2835.36	5624.12
89	2	02:19:00		03:00 PM - 05:00 PM	2836.48	5623.37
90	2	02:02:00		03:00 PM - 05:00 PM	2836.80	5626.81

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
91	2	NA	NA	09:00 AM - 11:00 AM	2837.27	5624.50
92	2	11:24:00		01:00 PM - 03:00 PM	2836.65	5624.42
93	2	09:33:00		All Day	2837.57	5623.10
94	2	11:15:00		09:00 AM - 11:00 AM	2836.88	5624.20
95	2	NA	NA	01:00 PM - 03:00 PM	2838.14	5623.92
96	2	09:23:00		01:00 PM - 03:00 PM	2834.76	5624.07
97	2	02:13:00		11:00 AM - 01:00 PM	2836.78	5624.70
98	2	12:25:00		11:00 AM - 01:00 PM	2838.13	5628.65
99	2	10:45:00		09:00 AM - 11:00 AM	2835.07	5622.60
100	2	12:44:00		All Day	2838.35	5626.88
101	2	10:46:00		03:00 PM - 05:00 PM	2835.06	5622.68
102	2	NA	NA	09:00 AM - 11:00 AM	2838.10	5621.48
103	2	10:45:00		11:00 AM - 01:00 PM	2835.07	5622.60
104	2	03:40:00		03:00 PM - 05:00 PM	2835.20	5623.16
105	2	09:25:00		09:00 AM - 11:00 AM	2834.77	5623.63
106	2	11:15:00		01:00 PM - 03:00 PM	2836.88	5624.20
107	2	03:40:00		11:00 AM - 01:00 PM	2835.20	5623.16
108	2	12:01:00		01:00 PM - 03:00 PM	2837.76	5626.98
109	2	10:45:00		03:00 PM - 05:00 PM	2835.07	5622.60
110	2	03:40:00		11:00 AM - 01:00 PM	2837.93	5620.82
111	2	10:47:00		09:00 AM - 11:00 AM	2835.06	5622.68
112	2	10:45:00		09:00 AM - 11:00 AM	2835.07	5622.60
113	2	03:48:00		11:00 AM - 01:00 PM	2834.65	5621.94
114	2	10:45:00		09:00 AM - 11:00 AM	2835.07	5622.60
115	2	03:40:00		01:00 PM - 03:00 PM	2835.20	5623.16
116	2	03:40:00		11:00 AM - 01:00 PM	2838.98	5623.55
117	2	11:15:00		09:00 AM - 11:00 AM	2836.88	5624.20
118	2	10:45:00		11:00 AM - 01:00 PM	2835.07	5622.60
119	2	03:35:00		09:00 AM - 11:00 AM	2835.06	5623.43
120	2	11:15:00		11:00 AM - 01:00 PM	2836.88	5624.20
121	2	12:01:00		09:00 AM - 11:00 AM	2837.76	5626.98
122	2	NA	NA	All Day	2837.06	5624.14
123	2	10:45:00		All Day	2835.07	5622.60
124	2	09:33:00		03:00 PM - 05:00 PM	2834.53	5622.74

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
125	2	10:55:00		09:00 AM - 11:00 AM	2835.17	5622.86
126	2	NA	NA	01:00 PM - 03:00 PM	2837.91	5624.26
127	2	12:19:00		03:00 PM - 05:00 PM	2838.05	5628.59
128	2	NA	NA	03:00 PM - 05:00 PM	2835.80	5619.72
129	2	10:45:00		09:00 AM - 11:00 AM	2835.07	5622.60
130	2	03:40:00		01:00 PM - 03:00 PM	2834.78	5623.73
131	2	03:40:00		All Day	2835.20	5623.16
132	2	02:27:00		03:00 PM - 05:00 PM	2838.99	5617.16
133	2	03:26:00		03:00 PM - 05:00 PM	2834.51	5622.50
134	2	10:45:00		01:00 PM - 03:00 PM	2835.07	5622.60
135	2	10:45:00		03:00 PM - 05:00 PM	2835.07	5622.60
136	2	12:19:00		03:00 PM - 05:00 PM	2838.04	5628.58
137	2	12:01:00		11:00 AM - 01:00 PM	2837.76	5626.98
138	2	NA	NA	09:00 AM - 11:00 AM	2836.24	5623.14
139	2	09:51:00		09:00 AM - 11:00 AM	2835.06	5622.61
140	2	03:21:00		All Day	2834.57	5621.34
141	2	12:19:00		11:00 AM - 01:00 PM	2838.04	5628.58
142	2	10:45:00		09:00 AM - 11:00 AM	2835.07	5622.60
143	2	10:45:00		09:00 AM - 11:00 AM	2835.06	5622.68
144	2	03:40:00		01:00 PM - 03:00 PM	2838.70	5625.51
145	2	10:45:00		09:00 AM - 11:00 AM	2835.07	5622.60
146	2	03:40:00		11:00 AM - 01:00 PM	2838.78	5622.08
147	2	02:58:00		09:00 AM - 11:00 AM	2835.29	5624.18
148	2	12:30:00		03:00 PM - 05:00 PM	2837.45	5626.59
149	2	10:45:00		All Day	2835.07	5622.60
150	2	11:42:00		01:00 PM - 03:00 PM	2837.17	5626.09
151	2	11:33:00		11:00 AM - 01:00 PM	2838.99	5617.16
152	2	10:45:00		03:00 PM - 05:00 PM	2835.07	5622.60
153	2	12:56:00		11:00 AM - 01:00 PM	2834.77	5623.63
154	2	10:45:00		09:00 AM - 11:00 AM	2835.07	5622.60
155	2	10:45:00		01:00 PM - 03:00 PM	2835.07	5622.60
156	2	03:21:00		09:00 AM - 11:00 AM	2834.57	5621.34
157	3	01:25:00		01:00 PM - 03:00 PM	2837.18	5625.47
158	3	12:12:00		09:00 AM - 11:00 AM	2837.29	5624.99

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
159	3	10:06:00		03:00 PM - 05:00 PM	2837.38	5624.99
160	3	12:12:00		11:00 AM - 01:00 PM	2837.38	5623.42
161	3	03:18:00		All Day	2837.52	5624.86
162	3	12:14:00		01:00 PM - 03:00 PM	2837.29	5623.82
163	3	12:52:00		03:00 PM - 05:00 PM	2837.45	5623.55
164	3	03:06:00		03:00 PM - 05:00 PM	2837.31	5625.00
165	3	10:06:00		01:00 PM - 03:00 PM	2837.39	5624.58
166	3	NA	NA	03:00 PM - 05:00 PM	2837.34	5625.38
167	3	NA	NA	09:00 AM - 11:00 AM	2837.18	5623.34
168	3	NA	NA	09:00 AM - 11:00 AM	2837.57	5625.83
169	3	12:14:00		11:00 AM - 01:00 PM	2837.44	5625.39
170	3	03:36:00		03:00 PM - 05:00 PM	2837.43	5624.23
171	3	03:36:00		09:00 AM - 11:00 AM	2837.46	5625.57
172	3	01:04:00		01:00 PM - 03:00 PM	2837.18	5625.41
173	3	09:52:00		11:00 AM - 01:00 PM	2837.27	5625.29
174	3	12:14:00		All Day	2837.44	5624.46
175	3	01:09:00		01:00 PM - 03:00 PM	2837.57	5623.10
176	3	01:14:00		All Day	2837.57	5623.10
177	3	09:27:00		01:00 PM - 03:00 PM	2837.51	5625.02
178	3	02:40:00		03:00 PM - 05:00 PM	2837.57	5623.10
179	3	12:42:00		11:00 AM - 01:00 PM	2837.45	5624.21
180	3	12:18:00		03:00 PM - 05:00 PM	2837.57	5623.10
181	3	12:14:00		03:00 PM - 05:00 PM	2837.57	5623.10
182	3	09:39:00		11:00 AM - 01:00 PM	2837.23	5623.78
183	3	11:46:00		09:00 AM - 11:00 AM	2837.54	5624.53
184	3	10:06:00		03:00 PM - 05:00 PM	2837.52	5624.33
185	3	11:33:00		11:00 AM - 01:00 PM	2837.29	5625.82
186	3	12:42:00		03:00 PM - 05:00 PM	2837.29	5625.74
187	3	10:06:00		01:00 PM - 03:00 PM	2837.48	5623.97
188	3	03:10:00		All Day	2837.51	5624.93
189	3	09:33:00		11:00 AM - 01:00 PM	2837.57	5623.10
190	3	12:12:00		11:00 AM - 01:00 PM	2837.57	5623.10
191	3	02:04:00		01:00 PM - 03:00 PM	2837.57	5623.10
192	3	03:56:00		03:00 PM - 05:00 PM	2837.43	5625.81

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
193	3	12:14:00		03:00 PM - 05:00 PM	2837.37	5625.50
194	3	NA	NA	01:00 PM - 03:00 PM	2837.25	5624.16
195	3	02:57:00		09:00 AM - 11:00 AM	2837.28	5624.51
196	3	12:12:00		01:00 PM - 03:00 PM	2837.57	5623.10
197	3	12:14:00		01:00 PM - 03:00 PM	2837.57	5623.10
198	3	04:35:00		01:00 PM - 03:00 PM	2837.19	5624.79
199	3	NA	NA	All Day	2837.52	5625.63
200	3	10:06:00		All Day	2837.41	5625.49
201	3	NA	NA	All Day	2837.36	5623.84
202	3	12:14:00		All Day	2837.38	5624.69
203	3	12:12:00		01:00 PM - 03:00 PM	2837.35	5625.54
204	3	11:19:00		11:00 AM - 01:00 PM	2837.42	5625.68
205	3	NA	NA	All Day	2837.53	5623.74
206	3	09:36:00		03:00 PM - 05:00 PM	2837.33	5625.02
207	3	NA	NA	03:00 PM - 05:00 PM	2837.34	5624.77
208	3	09:14:00		11:00 AM - 01:00 PM	2837.43	5625.87
209	3	02:57:00		03:00 PM - 05:00 PM	2837.37	5624.53
210	3	NA	NA	01:00 PM - 03:00 PM	2837.25	5623.39
211	3	12:12:00		01:00 PM - 03:00 PM	2837.17	5626.09
212	3	12:14:00		03:00 PM - 05:00 PM	2837.49	5624.18
213	3	12:12:00		03:00 PM - 05:00 PM	2837.57	5623.10
214	3	12:12:00		All Day	2837.57	5623.10
215	3	12:52:00		03:00 PM - 05:00 PM	2837.57	5623.10
216	3	NA	NA	09:00 AM - 11:00 AM	2837.25	5624.47
217	3	01:21:00		All Day	2837.41	5623.44
218	3	03:56:00		All Day	2837.57	5623.10
219	3	10:06:00		All Day	2837.57	5623.10
220	3	12:42:00		03:00 PM - 05:00 PM	2837.53	5623.87
221	3	NA	NA	09:00 AM - 11:00 AM	2837.40	5624.08
222	3	01:16:00		01:00 PM - 03:00 PM	2837.39	5624.65
223	3	03:31:00		11:00 AM - 01:00 PM	2837.22	5623.92
224	3	09:19:00		03:00 PM - 05:00 PM	2837.47	5624.74
225	4	11:55:00		03:00 PM - 05:00 PM	2842.89	5602.16
226	4	NA	NA	01:00 PM - 03:00 PM	2844.82	5614.00

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
227	4	10:45:00		All Day	2837.57	5623.10
228	4	09:57:00		11:00 AM - 01:00 PM	2837.57	5623.10
229	4	NA	NA	01:00 PM - 03:00 PM	2842.22	5615.28
230	4	11:33:00		All Day	2841.53	5601.23
231	4	10:46:00		09:00 AM - 11:00 AM	2842.17	5600.95
232	4	01:59:00		09:00 AM - 11:00 AM	2846.52	5602.39
233	4	NA	NA	01:00 PM - 03:00 PM	2840.31	5600.36
234	4	NA	NA	11:00 AM - 01:00 PM	2845.47	5611.83
235	4	03:20:00		09:00 AM - 11:00 AM	2847.02	5591.21
236	4	02:46:00		01:00 PM - 03:00 PM	2837.57	5623.10
237	4	04:59:00		01:00 PM - 03:00 PM	2847.50	5591.99
238	4	02:46:00		11:00 AM - 01:00 PM	2837.57	5623.10
239	4	04:12:00		All Day	2847.55	5591.87
240	4	NA	NA	09:00 AM - 11:00 AM	2837.57	5623.10
241	4	04:55:00		11:00 AM - 01:00 PM	2847.35	5592.12
242	4	02:10:00		11:00 AM - 01:00 PM	2847.75	5594.97
243	4	02:59:00		11:00 AM - 01:00 PM	2845.93	5591.79
244	4	02:03:00		All Day	2837.57	5623.10
245	4	02:03:00		11:00 AM - 01:00 PM	2837.57	5623.10
246	4	04:19:00		01:00 PM - 03:00 PM	2848.45	5590.52
247	4	02:03:00		01:00 PM - 03:00 PM	2847.02	5591.21
248	4	05:04:00		11:00 AM - 01:00 PM	2847.43	5596.05
249	4	03:20:00		01:00 PM - 03:00 PM	2837.57	5623.10
250	4	04:58:00		11:00 AM - 01:00 PM	2847.50	5591.96
251	4	03:03:00		01:00 PM - 03:00 PM	2847.50	5591.99
252	4	NA	NA	01:00 PM - 03:00 PM	2838.72	5603.14
253	4	02:46:00		01:00 PM - 03:00 PM	2837.57	5623.10
254	4	02:10:00		11:00 AM - 01:00 PM	2847.50	5591.96
255	4	04:54:00		09:00 AM - 11:00 AM	2847.35	5592.12
256	4	05:13:00		01:00 PM - 03:00 PM	2846.53	5600.02
257	4	11:44:00		09:00 AM - 11:00 AM	2842.74	5601.31
258	4	NA	NA	All Day	2847.02	5591.21
259	4	03:03:00		03:00 PM - 05:00 PM	2847.55	5591.87
260	4	02:46:00		All Day	2845.71	5592.93

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
261	4	10:15:00		09:00 AM - 11:00 AM	2841.53	5601.23
262	4	11:44:00		11:00 AM - 01:00 PM	2837.57	5623.10
263	4	02:10:00		All Day	2837.57	5623.10
264	4	11:44:00		03:00 PM - 05:00 PM	2842.74	5601.31
265	4	05:02:00		01:00 PM - 03:00 PM	2847.02	5591.21
266	4	01:37:00		01:00 PM - 03:00 PM	2845.13	5605.31
267	4	NA	NA	11:00 AM - 01:00 PM	2844.45	5599.93
268	4	05:13:00		01:00 PM - 03:00 PM	2846.53	5600.02
269	4	12:45:00		All Day	2843.82	5605.83
270	4	NA	NA	All Day	2840.37	5614.56
271	4	11:11:00		01:00 PM - 03:00 PM	2841.87	5603.41
272	4	12:23:00		11:00 AM - 01:00 PM	2843.08	5599.50
273	4	10:55:00		01:00 PM - 03:00 PM	2841.30	5600.32
274	4	04:47:00		03:00 PM - 05:00 PM	2837.57	5623.10
275	4	02:04:00		03:00 PM - 05:00 PM	2847.02	5591.21
276	4	04:19:00		All Day	2848.45	5590.52
277	4	NA	NA	01:00 PM - 03:00 PM	2847.02	5591.21
278	4	NA	NA	09:00 AM - 11:00 AM	2847.35	5592.12
279	4	02:46:00		11:00 AM - 01:00 PM	2837.57	5623.10
280	4	03:03:00		11:00 AM - 01:00 PM	2837.57	5623.10
281	4	10:42:00		11:00 AM - 01:00 PM	2837.57	5623.10
282	4	NA	NA	11:00 AM - 01:00 PM	2841.53	5601.23
283	4	05:13:00		09:00 AM - 11:00 AM	2837.57	5623.10
284	4	11:37:00		01:00 PM - 03:00 PM	2842.46	5600.84
285	4	04:58:00		09:00 AM - 11:00 AM	2837.57	5623.10
286	4	03:20:00		All Day	2847.02	5591.21
287	4	05:31:00		09:00 AM - 11:00 AM	2844.91	5606.47
288	4	03:40:00		03:00 PM - 05:00 PM	2847.55	5591.87
289	4	01:13:00		01:00 PM - 03:00 PM	2845.27	5608.49
290	4	10:07:00		11:00 AM - 01:00 PM	2842.80	5604.32
291	4	12:23:00		01:00 PM - 03:00 PM	2843.08	5599.50
292	4	02:04:00		All Day	2837.57	5623.10
293	4	02:46:00		01:00 PM - 03:00 PM	2845.71	5592.93
294	4	02:24:00		09:00 AM - 11:00 AM	2847.02	5591.21

Table A.1 continued from previous page

Cust.	Veh.	Delivery Time	AM/PM	Time Window	Lat	Lon
295	4	03:03:00		03:00 PM - 05:00 PM	2847.02	5591.21
296	4	05:13:00		03:00 PM - 05:00 PM	2847.50	5591.96
297	4	01:41:00		01:00 PM - 03:00 PM	2837.57	5623.10
298	4	03:08:00		All Day	2847.50	5591.99
299	4	NA	NA	03:00 PM - 05:00 PM	2842.49	5607.25
300	4	05:13:00		11:00 AM - 01:00 PM	2847.02	5591.21

Table A.2 Interview Questions

Q.	Question	Interviewee	Response	Action
1	How do you usually model and optimise your operations?	Operation Director	We do allocations to drivers based on key geographical location, and the driver delivers them based on their familiarity with the area	The traditional optimisation approach noted
2	Do you face breakdowns or sudden unavailability of vehicle(s) during operations?	Operation Director	Yes, we face such disruptions and could be multiple once per shift	Noted the possibility of breakdowns
3	How do you usually handle breakdowns during operations?	Operation Director	We do not handle them and consider the allocated customers as missed ones	The traditional dynamic optimisation approach noted
4	Do you maintain vehicle breakdown logs?	Operation Manager	Unfortunately not, but we may expect a breakdown or multiple ones early, mid, or late in the shift	Utilised this expert information in modelling random vehicle breakdowns, an exponential random variable is estimated concerning the shifts
5	What time does a shift start, and how long is it? Is there a maximum duration for the vehicles to operate?	Operation Manager	Shifts start at 8 am and end at 6 pm, with no maximum duration except the shift length	Vehicles' shifts are adapted, and modelled vehicles with duration limit considered to be infinite
6	Based on your data, what day setting is closest to a breakdown problem?	Operation Manager	There is a particular setting we can provide where possible breakdowns are faced. They could occur early, mid, or late in the shift. These disruptions events may occur multiple times	Data received and post-processed. Post-processing includes accommodating to the required model data structure and deducing the original problem routes (statically) based on the actual delivery of services
7	For this day, where do vehicles start and end their routes?	Operation Manager	They start from one central depot. Located (Lat: 2834.89km, Lon: 5624.82km)	Updated case study data with the depot information
8	What is the average estimated vehicle speed for this day?	Operation Manager	We can estimate the speed across all vehicles by checking the distance travelled per shift	24km/h is provided as an average estimated vehicle speed is accommodated when modelling the case study

Table A.2 continued from previous page

Q.	Question	Interviewee	Response	Action
9	Is there a vehicle capacity limit?	Operation Manager	We assume vehicles can accommodate any order provided	Modelled the case study assuming infinite vehicle capacities
10	The customer addresses you provided are text descriptions. Can you please provide them as coordinates?	Operation Manager	Yes, I believe it is possible. We can deduce them based on the vehicle location at the actual delivery time. Missed customers can be estimated based on the randomised average of the region.	Adapted every customer location for the case study data
11	What is the service time for every customer?	Operation Manager	We usually service customers within minutes. 3 minutes is more than enough to service a customer	Servicing time for every customer is assumed to be 3 minutes

Table A.3 Case Study Actual Implemented Routes

Veh.	Route
1	[39 28 49 26 56 14 10 29 5 6 7 44 21 32 61 43 4 25 12 33 46 47 48 15 54 45 24 69 23 34 22 70 36 19 66 31 71 3 13 78 50 72 73 20 27 11 30 41 52 55 57 63 65 2 1 18 9 62 76]
2	[84 90 97 89 132 88 147 87 83 85 140 156 133 119 104 107 110 115 116 130 131 144 146 113 82 96 105 93 124 79 139 99 103 109 112 114 118 123 129 134 135 142 143 145 149 152 154 155 101 111 125 94 106 117 120 92 151 150 108 121 137 127 136 141 98 148 100 153]
3	[172 175 176 222 217 157 191 178 195 209 164 188 161 223 170 171 192 218 198 208 224 177 189 206 182 173 159 165 184 187 200 219 204 185 183 158 160 190 196 203 211 213 214 162 169 174 181 193 197 202 212 180 179 186 220 163 215]
3	[289 266 297 232 244 245 247 275 292 242 254 263 294 236 238 253 260 279 293 243 251 259 280 295 298 235 249 286 288 239 246 276 274 255 241 250 285 237 265 248 256 268 283 296 300 287 228 290 261 281 227 231 273 271 230 284 257 262 264 225 272 291 269]

Appendix B

Case Study Results

Table B.1 Hybrid Approach Runs on Case Study

Run	C Rule	μ	α_1	α_2	V	TD (KM)	WT (Hr)	CM	DM	VTW_{V_r} (Hr)	CPU (Hr)
1	Far_Avg	1	0.0	1.0	4	443.19	0.50	5	5	0	0.30
2	Far_Avg	1	0.1	0.9	4	429.24	0.76	4	4	0	0.31
3	Far_Avg	1	0.2	0.8	4	428.54	1.00	1	1	0	0.30
4	Far_Avg	1	0.3	0.7	4	428.16	0.97	1	1	0	0.31
5	Far_Avg	1	0.4	0.6	4	415.2	1.44	5	5	0	0.33
6	Far_Avg	1	0.5	0.5	4	415.29	1.44	5	5	0	0.33
7	Far_Avg	1	0.6	0.4	4	414.48	1.20	1	1	0	0.35
8	Far_Avg	1	0.7	0.3	4	418.51	1.21	2	2	0	0.34
9	Far_Avg	1	0.8	0.2	4	399.61	1.71	1	1	0	0.35
10	Far_Avg	1	0.9	0.1	4	391.59	2.10	1	1	0	0.35
11	Far_Avg	1	1.0	0.0	4	269.99	1.49	0	0	0	0.21
12	Far_Avg	2	0.0	1.0	4	443.19	0.50	5	5	0	0.31
13	Far_Avg	2	0.1	0.9	4	425.75	0.80	3	3	0	0.32
14	Far_Avg	2	0.2	0.8	4	426.75	0.88	3	3	0	0.29
15	Far_Avg	2	0.3	0.7	4	422.58	0.96	1	1	0	0.29
16	Far_Avg	2	0.4	0.6	4	425.03	1.27	7	7	0	0.33
17	Far_Avg	2	0.5	0.5	4	412.97	1.63	5	5	0	0.33
18	Far_Avg	2	0.6	0.4	4	411.03	1.57	2	2	0	0.34
19	Far_Avg	2	0.7	0.3	4	419.18	1.12	1	1	0	0.35
20	Far_Avg	2	0.8	0.2	4	412.56	2.05	2	2	0	0.32
21	Far_Avg	2	0.9	0.1	4	412.2	1.17	1	1	0	0.39
22	Far_Avg	2	1.0	0.0	4	304.17	2.83	0	0	0	0.22
23	Far_Min	1	0.0	1.0	4	443.19	0.50	5	5	0	0.30
24	Far_Min	1	0.1	0.9	4	429.24	0.76	4	4	0	0.31
25	Far_Min	1	0.2	0.8	4	428.54	1.00	1	1	0	0.30
26	Far_Min	1	0.3	0.7	4	428.16	0.97	1	1	0	0.31
27	Far_Min	1	0.4	0.6	4	415.2	1.44	5	5	0	0.33

Table B.1 continued from previous page

Run	C Rule	μ	α_1	α_2	V	TD (KM)	WT (Hr)	CM	DM	VTW_{V_r} (Hr)	CPU (Hr)
28	Far_Min	1	0.5	0.5	4	415.29	1.44	5	5	0	0.33
29	Far_Min	1	0.6	0.4	4	414.48	1.20	1	1	0	0.35
30	Far_Min	1	0.7	0.3	4	418.51	1.21	2	2	0	0.35
31	Far_Min	1	0.8	0.2	4	399.61	1.71	1	1	0	0.34
32	Far_Min	1	0.9	0.1	4	391.59	2.10	1	1	0	0.36
33	Far_Min	1	1.0	0.0	4	269.99	1.49	0	0	0	0.20
34	Far_Min	2	0.0	1.0	4	443.19	0.50	5	5	0	0.31
35	Far_Min	2	0.1	0.9	4	425.75	0.80	3	3	0	0.31
36	Far_Min	2	0.2	0.8	4	426.75	0.88	3	3	0	0.29
37	Far_Min	2	0.3	0.7	4	422.58	0.96	1	1	0	0.29
38	Far_Min	2	0.4	0.6	4	425.03	1.27	7	7	0	0.32
39	Far_Min	2	0.5	0.5	4	412.97	1.63	5	5	0	0.33
40	Far_Min	2	0.6	0.4	4	411.03	1.57	2	2	0	0.34
41	Far_Min	2	0.7	0.3	4	419.18	1.12	1	1	0	0.36
42	Far_Min	2	0.8	0.2	4	412.56	2.05	2	2	0	0.32
43	Far_Min	2	0.9	0.1	4	412.2	1.17	1	1	0	0.39
44	Far_Min	2	1.0	0.0	4	304.17	2.83	0	0	0	0.22
45	LTW	1	0.0	1.0	4	417.69	2.82	0	0	0	0.54
46	LTW	1	0.1	0.9	4	412.46	2.83	0	0	0	0.56
47	LTW	1	0.2	0.8	4	412.19	2.83	0	0	0	0.54
48	LTW	1	0.3	0.7	4	412.67	2.87	0	0	0	0.54
49	LTW	1	0.4	0.6	4	411.91	2.83	0	0	0	0.54
50	LTW	1	0.5	0.5	4	416.02	2.80	0	0	0	0.56
51	LTW	1	0.6	0.4	4	412.66	2.88	0	0	0	0.55
52	LTW	1	0.7	0.3	4	416.17	2.81	0	0	0	0.57
53	LTW	1	0.8	0.2	4	411.71	1.91	0	0	0	0.55
54	LTW	1	0.9	0.1	4	408.27	1.98	0	0	0	0.55
55	LTW	1	1.0	0.0	4	378.81	2.64	0	0	0	0.29
56	LTW	2	0.0	1.0	4	417.69	2.82	0	0	0	0.54
57	LTW	2	0.1	0.9	4	414.76	2.83	0	0	0	0.55
58	LTW	2	0.2	0.8	4	414.76	2.83	0	0	0	0.55
59	LTW	2	0.3	0.7	4	414.76	2.83	0	0	0	0.54
60	LTW	2	0.4	0.6	4	413.91	2.83	0	0	0	0.58
61	LTW	2	0.5	0.5	4	413.62	2.84	0	0	0	0.55
62	LTW	2	0.6	0.4	4	414.9	2.89	0	0	0	0.54
63	LTW	2	0.7	0.3	4	414.93	2.89	0	0	0	0.54
64	LTW	2	0.8	0.2	4	412.54	1.91	0	0	0	0.55
65	LTW	2	0.9	0.1	4	376.7	2.84	0	0	0	0.55
66	LTW	2	1.0	0.0	4	374.21	3.25	0	0	0	0.35

Table B.2 Centralised Approach GA Runs' Average on Case Study

Gen	V	TD (KM)	WT (Hr)	VTW_{V_r} (Hr)
0	4	2290.12	8.67	3432.19
1	4	1182.89	5.13	1359.40
2	4	429.79	0.44	0.00
3	4	429.79	0.44	0.00
4	4	429.79	0.44	0.00
5	4	429.79	0.44	0.00
6	4	425.14	0.34	0.00
7	4	425.14	0.34	0.00
8	4	425.14	0.34	0.00
9	4	423.84	0.21	0.00
10	4	423.84	0.21	0.00
11	4	423.84	0.21	0.00
12	4	423.84	0.21	0.00
13	4	423.25	0.20	0.00
14	4	423.25	0.20	0.00
15	4	423.25	0.20	0.00
16	4	423.25	0.20	0.00
17	4	423.25	0.20	0.00
18	4	419.15	0.20	0.00
19	4	418.55	0.19	0.00
20	4	418.55	0.19	0.00
21	4	418.55	0.19	0.00
22	4	418.55	0.19	0.00
23	4	418.55	0.19	0.00
24	4	418.55	0.19	0.00
25	4	418.55	0.19	0.00
26	4	418.55	0.19	0.00
27	4	418.55	0.19	0.00
28	4	418.55	0.19	0.00
29	4	418.55	0.19	0.00
30	4	418.55	0.19	0.00
31	4	418.55	0.19	0.00
32	4	418.55	0.19	0.00

Table B.2 continued from previous page

Gen	V	TD	WT	VTW_{V_r}
33	4	418.55	0.19	0.00
34	4	418.55	0.19	0.00
35	4	417.90	0.19	0.00
36	4	417.90	0.19	0.00
37	4	417.90	0.19	0.00
38	4	417.90	0.19	0.00
39	4	417.90	0.19	0.00
40	4	417.90	0.19	0.00
41	4	417.90	0.19	0.00
42	4	417.90	0.19	0.00
43	4	417.90	0.19	0.00
44	4	417.90	0.19	0.00
45	4	416.71	0.17	0.00
46	4	416.71	0.17	0.00
47	4	416.71	0.17	0.00
48	4	416.71	0.17	0.00
49	4	416.71	0.17	0.00
50	4	416.71	0.17	0.00
51	4	416.71	0.17	0.00
52	4	416.71	0.17	0.00
53	4	406.19	0.16	0.00
54	4	406.19	0.16	0.00
55	4	406.19	0.16	0.00
56	4	404.75	0.14	0.00
57	4	404.75	0.14	0.00
58	4	404.75	0.14	0.00
59	4	404.75	0.14	0.00
60	4	404.75	0.14	0.00
61	4	404.75	0.14	0.00
62	4	404.75	0.14	0.00
63	4	404.75	0.14	0.00
64	4	404.75	0.14	0.00
65	4	404.75	0.14	0.00
66	4	403.66	0.13	0.00

Table B.2 continued from previous page

Gen	V	TD	WT	VTW_{V_r}
67	4	403.66	0.13	0.00
68	4	403.66	0.13	0.00
69	4	403.66	0.13	0.00
70	4	403.66	0.13	0.00
71	4	403.66	0.13	0.00
72	4	403.66	0.13	0.00
73	4	403.66	0.13	0.00
74	4	403.66	0.13	0.00
75	4	403.66	0.13	0.00
76	4	403.66	0.13	0.00
77	4	403.66	0.13	0.00
78	4	403.66	0.13	0.00
79	4	403.66	0.13	0.00
80	4	403.66	0.13	0.00
81	4	403.66	0.13	0.00
82	4	403.66	0.13	0.00
83	4	403.66	0.13	0.00
84	4	403.66	0.13	0.00
85	4	403.66	0.13	0.00
86	4	403.66	0.13	0.00
87	4	403.66	0.13	0.00
88	4	403.66	0.13	0.00
89	4	403.66	0.13	0.00
90	4	403.66	0.13	0.00
91	4	403.66	0.13	0.00
92	4	403.66	0.13	0.00
93	4	403.66	0.13	0.00
94	4	403.66	0.13	0.00
95	4	403.66	0.13	0.00
96	4	403.66	0.13	0.00
97	4	403.66	0.13	0.00
98	4	403.66	0.13	0.00
99	4	403.66	0.13	0.00
100	4	403.66	0.13	0.00

Table B.2 continued from previous page

Gen	V	TD	WT	VTW_{V_r}
101	4	403.66	0.13	0.00
102	4	403.66	0.13	0.00
103	4	403.66	0.13	0.00
104	4	403.66	0.13	0.00
105	4	403.66	0.13	0.00
106	4	403.66	0.13	0.00
107	4	403.66	0.13	0.00
108	4	403.66	0.13	0.00
109	4	403.66	0.13	0.00
110	4	403.66	0.13	0.00
111	4	403.66	0.13	0.00
112	4	403.66	0.13	0.00
113	4	403.66	0.13	0.00
114	4	403.66	0.13	0.00
115	4	403.66	0.13	0.00
116	4	403.66	0.13	0.00
117	4	403.66	0.13	0.00
118	4	403.66	0.13	0.00
119	4	403.66	0.13	0.00
120	4	403.66	0.13	0.00
121	4	403.66	0.13	0.00
122	4	403.66	0.13	0.00
123	4	403.66	0.13	0.00
124	4	403.66	0.13	0.00
125	4	403.66	0.13	0.00
126	4	403.66	0.13	0.00
127	4	403.66	0.13	0.00
128	4	403.66	0.13	0.00
129	4	403.66	0.13	0.00
130	4	403.66	0.13	0.00
131	4	403.66	0.13	0.00
132	4	403.66	0.13	0.00
133	4	403.66	0.13	0.00
134	4	403.66	0.13	0.00

Table B.2 continued from previous page

Gen	V	TD	WT	VTW_{V_r}
135	4	403.66	0.13	0.00
136	4	403.66	0.13	0.00
137	4	403.66	0.13	0.00
138	4	403.66	0.13	0.00
139	4	403.66	0.13	0.00
140	4	403.66	0.13	0.00
141	4	403.66	0.13	0.00
142	4	403.66	0.13	0.00
143	4	403.66	0.13	0.00
144	4	403.66	0.13	0.00
145	4	403.66	0.13	0.00
146	4	403.66	0.13	0.00
147	4	403.66	0.13	0.00
148	4	403.66	0.13	0.00
149	4	403.66	0.13	0.00
150	4	403.66	0.13	0.00

Table B.3 Case Study Best Routes from the Hybrid Approach, Run 11

Veh.	Route
1	[214 176 93 21 62 60 15 38 35 76 28 54 75 50 19 73 261 231 257 269 287 232 260 294 235 278 255 241 242 248 267 234 290 282 272 243 300 298 254 250 251 237 239 268 256 266 271 284 273 291 293 277 265 247 246 276 288 259 296 295 286 275 258 264 225 230 299 270 18 42 1 31 37 132 79 36 23 72 52 51 13 68 59 227 219 218]
2	[131 149 123 140 117 94 122 216 91 195 202 158 171 168 188 161 174 183 221 74 167 217 292 285 283 263 244 240 64 32 48 102 61 44 10 7 5 40 8 146 116 189 43 46 47 33 12 17 110 77 53 49 39 56 45 24 78 2 20 63 57 55 30 66 22 34 151 67 29 226 289 233 252 229 16 3 65 41 11 27 69 14 26 9 6 274 215 213 181 180 178 205 220 4 163 201 212 86 184 170 209 224 166 193 200 186 192 199 82 159 206 164 207 100 80]
3	[147 105 119 143 111 139 156 154 145 142 129 114 112 99 125 138 121 107 153 113 118 103 97 173 185 141 98 137 208 204 169 120 223 182 179 25 160 281 280 279 262 245 238 228 190 297 253 249 236 197 196 191 175 210 162 194 198 222 172 157 84 211 150 81 108 203 177 165 187 126 144 95 70 71 58 128 85 83 133 124 152 135 109 101 104 89 148 127 136 90 88]
4	[92 106 87 155 134 115 130 96]

Table B.4 Case Study Best Routes from the Centralised Approach

Veh.	Route
1	[121 168 200 171 199 82 117 94 32 102 61 10 39 73 50 28 49 62 240 64 283 285 48 122 74 205 174 223 182 120 131 107 245 280 228 279 281 189 262 33 17 77 53 8 146 116 141 98 161 202 169 108 81 84 211 150 203 157 172 198 92 106 194 162 210 24 71 197 196 175 95 144 126 177 188 222 165 209 170 163 274 213 178 215 181 180 220 212 184 86 224 207 159 206 164 166 193 186 192 148 127 136 90 80]
2	[201 217 218 219 244 263 214 292 227 93 140 59 260 235 294 276 239 255 278 257 231 230 261 282 272 243 300 250 254 241 267 290 55 57 229 226 269 284 291 268 256 293 298 237 251 246 286 247 265 277 275 258 295 288 259 296 264 225 299 79 132 37 9 89 88]
3	[147 105 119 111 139 145 154 142 99 112 129 125 138 26 40 5 287 232 248 242 234 63 30 2 20 66 22 151 67 29 34 45 6 7 69 297 43 253 236 191 249 4 187 87 134 155 123 115 130 96]
4	[21 52 51 23 36 54 76 13 72 68 35 38 176 221 183 158 195 91 216 167 143 114 156 75 60 15 44 56 14 19 149 113 103 118 153 97 173 185 137 100 208 204 179 25 160 46 47 12 110 190 238 78 70 58 27 11 65 41 3 42 16 270 289 266 271 273 233 252 18 1 31 128 85 83 133 124 109 152 135 101 104]

Table B.5 Scenario 1 Sampled Solution Routes after the 1st Breakdown

Veh.	Route
2	[48 102 61 44 10 7 5 40 8 146 116 189 43 46 47 33 12 17 110 77 53 49 39 56 45 24 78 2 20 63 57 55 30 66 22 34 151 67 29 226 289 233 252 229 16 3 65 41 11 27 69 14 26 9 6 274 215 213 181 180 178 205 220 4 163 201 212 86 184 170 209 224 166 193 200 186 192 199 82 159 206 164 207 100 80]
3	[107 153 113 118 103 97 173 185 141 98 137 208 204 169 120 223 182 179 25 160 281 280 279 262 245 238 228 190 297 253 249 236 197 196 191 175 210 162 194 198 222 172 157 84 211 150 81 108 203 177 165 187 126 144 95 70 71 58 128 85 83 133 124 152 135 109 101 104 89 148 127 136 90 88]
4	[10227 10219 10218 10255 10059 10068 10037 10013 10072 10079 10132 10023 10036 10031 10051 10052 10001 10042 10278 10018 10270 10282 10299 10290 10230 10225 10264 10272 10267 10243 10300 10241 10254 10250 10242 10248 10260 10237 10258 10295 10286 10275 10298 10296 10259 10288 10239 10276 10246 255 278 248 242 254 250 241 300 243 267 272 282 290 270 218 92 106 87 155 134 115 130 96 237 298 239 276 246 259 288 296 258 295 286 275 260 264 225 230 299 18 42 1 31 37 79 132 23 36 72 52 51 13 68 59 227 219]

Table B.6 Scenario 1 Sampled Solution Routes after the 2nd Breakdown

Veh.	Route
2	[22 34 151 67 29 226 289 233 252 229 16 3 65 41 11 27 69 14 26 9 6 274 215 213 181 180 178 205 220 4 163 201 212 86 184 170 209 224 166 193 200 186 192 199 82 159 206 164 207 100 80]
3	[297 253 249 236 197 196 191 175 210 162 194 198 222 172 157 84 211 150 81 108 203 177 165 187 126 144 95 70 71 58 128 85 83 133 124 152 135 109 101 104 89 148 127 136 90 88]

Table B.7 Scenario 1 Sampled Solution Routes after the 3rd Breakdown

Veh.	Route
3	[133 124 152 135 109 101 104 89 148 127 136 90 88]

Table B.8 Scenario 1 Sampled Solution Overall Routes

Veh.	Route
1	[214 176 93 21 62 60 15 38 35 76 28 54 75 50 19 73 261 231 257 269 287 232]
2	[131 149 123 140 117 94 122 216 91 195 202 158 171 168 188 161 174 183 221 74 167 217 292 285 283 263 244 240 64 32 48 102 61 44 10 7 5 40 8 146 116 189 43 46 47 33 12 17 110 77 53 49 39 56 45 24 78 2 20 63 57 55 30 66 22 34 151 67 29 226 289 233 252 229 16 3 65 41 11]
3	[147 105 119 143 111 139 156 154 145 142 129 114 112 99 125 138 121 107 153 113 118 103 97 173 185 141 98 137 208 204 169 120 223 182 179 25 160 281 280 279 262 245 238 228 190 297 253 249 236 197 196 191 175 210 162 194 198 222 172 157 84 211 150 81 108 203 177 165 187 126 144 95 70 71 58 128 85 83 133 124 152 135 109 101 104 89 148 127 136 90 88]
4	[10227 10219 10218 10255 10059 10068 10037 10013 10072 10079 10132 10023 10036 10031 10051 10052 10001 10042 10278 10018 10270 10282 10299 10290 10230 10225 10264 10272 10267 10243 10300 10241 10254 10250 10242 10248 10260 10237 10258 10295 10286 10275 10298 10296 10259 10288 10239 10276 10246 255 278 248 242 254 250 241 300 243 267 272 282 290 270 218]

Table B.9 Scenario 2 Sampled Solution Routes after the 1st Breakdown

Veh.	Route
1	[250 251 237 239 268 256 266 271 284 273 291 293 277 265 247 246 276 288 259 296 295 286 275 258 264 225 230 299 270 18 42 1 31 37 132 79 36 23 72 52 51 13 68 59 227 219 218]
2	[226 289 233 252 229 16 3 65 41 11 27 69 14 26 9 6 274 215 213 181 180 178 205 220 4 163 201 212 86 184 170 209 224 166 193 200 186 192 199 82 159 206 164 207 100 80]
4	[10089 10127 10136 10148 10090 10194 10088 10165 10104 10162 10210 162 194 165 92 106 210 10187 10101 10109 10135 10152 10144 10126 10095 10124 10133 10058 10071 10070 10297 10196 297 10197 196 10253 197 10249 253 10236 249 10191 236 10175 191 10083 175 10085 10128 187 126 144 95 71 70 58 87 155 134 115 130 96 124 133 83 85 128 89 148 127 136 90 109 135 152 101 104 88]

Table B.10 Scenario 2 Sampled Solution Routes after the 2nd Breakdown

Veh.	Route
1	[31 37 132 79 36 23 72 52 51 13 68 59 227 219 218]
2	[80]

Table B.11 Scenario 2 Sampled Solution Overall Routes

Veh.	Route
1	[214 176 93 21 62 60 15 38 35 76 28 54 75 50 19 73 261 231 257 269 287 232 260 294 235 278 255 241 242 248 267 234 290 282 272 243 300 298 254 250 251 237 239 268 256 266 271 284 273 291 293 277 265 247 246 276 288 259 296 295 286 275 258 264 225 230 299 270 18 42 1 31 37 132 79 36 23 72 52 51 13 68 59 227 219 218]
2	[131 149 123 140 117 94 122 216 91 195 202 158 171 168 188 161 174 183 221 74 167 217 292 285 283 263 244 240 64 32 48 102 61 44 10 7 5 40 8 146 116 189 43 46 47 33 12 17 110 77 53 49 39 56 45 24 78 2 20 63 57 55 30 66 22 34 151 67 29 226 289 233 252 229 16 3 65 41 11 27 69 14 26 9 6 274 215 213 181 180 178 205 220 4 163 201 212 86 184 170 209 224 166 193 200 186 192 199 82 159 206 164 207 100 80]
3	[147 105 119 143 111 139 156 154 145 142 129 114 112 99 125 138 121 107 153 113 118 103 97 173 185 141 98 137 208 204 169 120 223 182 179 25 160 281 280 279 262 245 238 228 190]
4	[10089 10127 10136 10148 10090 10194 10088 10165 10104 10162 10210 162 194 165 92 106 210 10187 10101 10109 10135 10152 10144 10126 10095 10124 10133 10058 10071 10070 10297 10196 297 10197 196 10253 197 10249 253 10236 249 10191 236 10175 191 10083 175 10085 10128 187 126 144 95 71 70 58 87 155 134 115 130 96 124 133 83 85 128 89 148 127 136 90 109 135 152 101 104]

Table B.12 Scenario 3 Sampled Solution Routes after the Only Breakdown

Veh.	Route
1	[264 225 230 299 270 18 42 1 31 37 132 79 36 23 72 52 51 13 68 59 227 219 218]
2	[215 213 181 180 178 205 220 4 163 201 212 86 184 170 209 224 166 193 200 186 192 199 82 159 206 164 207 100 80]
3	[152 135 109 101 104 89 148 127 136 90 88]

Table B.13 Scenario 3 Sampled Solution Overall Routes

Veh.	Route
1	[214 176 93 21 62 60 15 38 35 76 28 54 75 50 19 73 261 231 257 269 287 232 260 294 235 278 255 241 242 248 267 234 290 282 272 243 300 298 254 250 251 237 239 268 256 266 271 284 273 291 293 277 265 247 246 276 288 259 296 295 286 275 258 264 225 230 299 270 18 42 1 31 37 132 79 36 23 72 52 51 13 68 59 227 219 218]
2	[131 149 123 140 117 94 122 216 91 195 202 158 171 168 188 161 174 183 221 74 167 217 292 285 283 263 244 240 64 32 48 102 61 44 10 7 5 40 8 146 116 189 43 46 47 33 12 17 110 77 53 49 39 56 45 24 78 2 20 63 57 55 30 66 22 34 151 67 29 226 289 233 252 229 16 3 65 41 11 27 69 14 26 9 6 274 215 213 181 180 178 205 220 4 163 201 212 86 184 170 209 224 166 193 200 186 192 199 82 159 206 164 207 100 80]
3	[147 105 119 143 111 139 156 154 145 142 129 114 112 99 125 138 121 107 153 113 118 103 97 173 185 141 98 137 208 204 169 120 223 182 179 25 160 281 280 279 262 245 238 228 190 297 253 249 236 197 196 191 175 210 162 194 198 222 172 157 84 211 150 81 108 203 177 165 187 126 144 95 70 71 58 128 85 83 133 124 152 135 109 101 104 89 148 127 136 90 88]
4	[92 106 87 155 134 115 130]

References

Abbatecola, L., Fanti, M. P., and Ukovich, W. (2016). A review of new approaches for dynamic vehicle routing problem. *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 361–366.

Abu-Monshar, A., Al-Bazi, A., and Palade, V. (2022). An agent-based optimisation approach for vehicle routing problem with unique vehicle location and depot. *Expert Systems with Applications*, 192:116370.

Albareda-Sambola, M., Fernández, E., and Laporte, G. (2014). The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers and Operations Research*, 48:31–39.

Alisoltani, N., Leclercq, L., and Zargayouna, M. (2021). Can dynamic ride-sharing reduce traffic congestion? *Transportation Research Part B: Methodological*, 145:212–246.

Amrouss, A., El Hachemi, N., Gendreau, M., and Gendron, B. (2017). Real-time management of transportation disruptions in forestry. *Computers & Operations Research*, 83:95–105.

Archetti, C., Guerriero, F., and Macrina, G. (2021). The online vehicle routing problem with occasional drivers. *Computers & Operations Research*, 127:105144.

Barbati, M., Bruno, G., and Genovese, A. (2012). Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications*, 39(5):6020–6028.

Barbucha, D. (2012). Agent-based guided local search. *Expert Systems with Applications*, 39(15):12032–12045.

Barbucha, D. (2013). Experimental study of the population parameters settings in cooperative multi-agent system solving instances of the VRP. In *Transactions on Computational Collective Intelligence IX*, volume 7770, pages 1–28. Springer Berlin Heidelberg.

Barbucha, D. (2016). An improved agent-based approach to the dynamic vehicle routing problem. In *Intelligent Decision Technologies*, pages 361–370. Springer.

Barbucha, D. (2020). Multi-agent approach to the DVRP with GLS improvement procedure. In Czarnowski, I., Howlett, R. J., and Jain, L. C., editors, *Intelligent Decision Technologies*, Smart Innovation, Systems and Technologies, pages 117–126. Springer.

Barbucha, D. and Jedrzejowicz, P. (2007). An agent-based approach to vehicle routing problem. *International Journal of Computer and Information Engineering*, 1(2):6.

Barbucha, D. and Jędrzejowicz, P. (2009). Agent-based approach to the dynamic vehicle routing problem. In *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*, pages 169–178. Springer.

Barkaoui, M., Berger, J., and Boukhtouta, A. (2015). Customer satisfaction in dynamic vehicle routing problem with time windows. *Applied Soft Computing*, 35:423–432.

Barkaoui, M. and Gendreau, M. (2013). An adaptive evolutionary approach for real-time vehicle routing and dispatching. *Computers and Operations Research*, 40(7):1766–1776.

Bektas, T. (2006). The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega*, 34(3):209–219.

Bodin, L. and Golden, B. (1981). Classification in vehicle routing and scheduling. *Networks. An International Journal*, 11(2):97–108.

Bopardikar, S. D. and Srivastava, V. (2020). Dynamic vehicle routing in presence of random recalls. *IEEE Control Systems Letters*, 4(1):37–42.

Boysen, N., Fedtke, S., and Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43(1):1–58.

Chen, Z.-L. and Xu, H. (2006). Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, 40(1):74–88.

Cheung, B. K.-S., Choy, K. L., Li, C.-L., Shi, W., and Tang, J. (2008). Dynamic routing model and solution methods for fleet management with mobile technologies. *International Journal of Production Economics*, 113(2):694–705.

Chiu, H. N., Lee, Y. S., and Chang, J. H. (2006). Two approaches to solving the multi-depot vehicle routing problem with time windows in a time-based logistics environment. *Production Planning & Control*, 17(5):480–493.

Christofides, N., Mingozzi, A., and Toth, P. (1979). The vehicle routing problem. In *Combinatorial Optimization*, pages 315–338. Wiley.

Clarke, G. and Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581.

Cordeau, J.-F., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks. An International Journal*, 30(2):105–119.

Cordeau, J.-F., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, 52(8):928–936.

Cordeau, J. F. and Maischberger, M. (2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers and Operations Research*, 39(9):2033–2050.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

Davidsson, P., Persson, J. A., and Holmgren, J. (2007). On the integration of agent-based and mathematical optimization techniques. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, volume 4496, pages 1–10. Springer Berlin Heidelberg.

Dayarian, I. and Savelsbergh, M. (2020). Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29(9):2153–2174.

Dávid, B. and Krész, M. (2017). The dynamic vehicle rescheduling problem. *Central European Journal of Operations Research*, 25(4):809–830.

Elshaer, R. and Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140:106242.

Euchi, J., Yassine, A., and Chabchoub, H. (2015). The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. *Swarm and Evolutionary Computation*, 21:41–53.

Ferrucci, F., Bock, S., and Gendreau, M. (2013). A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. *European Journal of Operational Research*, 225(1):130–141.

FIPA (2000). Specifications. http://www.fipa.org/specifications/index.html [31 March 2020].

Fischer, K., Müller, J., and Pischel, M. (1996). Cooperative transportation scheduling: An application domain for DAI. *Applied Artificial Intelligence*, 10(1):1–34.

Fonseca-Galindo, J. C., de Castro Surita, G., Neto, J. M., de Castro, C. L., and Lemos, A. P. (2022). A multi-agent system for solving the dynamic capacitated vehicle routing problem with stochastic customers using trajectory data mining. *Expert Systems with Applications*, 195:116602.

Gath, M., Edelkamp, S., and Herzog, O. (2013). Agent-based dispatching in groupage traffic. In *2013 IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS)*, pages 54–60. IEEE.

Gendreau, M., Guertin, F., Potvin, J.-Y., and Taillard, E. D. (1999). Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4):381–390.

Gendreau, M., Hertz, a., and Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290.

Ghannadpour, S. F., Noori, S., Tavakkoli-Moghaddam, R., and Ghoseiri, K. (2014). A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Applied Soft Computing Journal*, 14:504–527.

Ghoseiri, K. and Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing*, 10(4):1096–1107.

Gillett, B. E. and Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, 13 edition.

Guedes, P. C. and Borenstein, D. (2015). Column generation based heuristic framework for the multiple-depot vehicle type scheduling problem. *Computers and Industrial Engineering*, 90:361–370.

Guedes, P. C. and Borenstein, D. (2018). Real-time multi-depot vehicle type rescheduling problem. *Transportation Research Part B: Methodological*, 108:217–234.

Haghani, A. and Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers and Operations Research*, 32(11):2959–2986.

Holland, J. (1975). Adaptation in artificial and natural systems. *Ann Arbor: The University of Michigan Press*, page 232.

Hosny, M. I. and Mumford, C. L. (2010). The single vehicle pickup and delivery problem with time windows: intelligent operators for heuristic and metaheuristic algorithms. *Journal of Heuristics*, 16(3):417–439.

Jozefowiez, N., Semet, F., and Talbi, E.-G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309.

Kalina, P. and Vokřínek, J. (2012). Parallel solver for vehicle routing and pickup and delivery problems with time windows based on agent negotiation. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1558–1563.

Kohout, R. and Erol, K. (1999). In-time agent-based vehicle routing with a stochastic improvement heuristic. *AAAI/IAAI*, pages 864–869.

Kuhn, N., Müller, J. P., and Müller, J. (1994). Simulating cooperative transportation companies. In *Simulation als betriebliche Entscheidungshilfe*, pages 263–274. Vieweg+Teubner Verlag.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.

Laporte, G. and Semet, F. (2002). Classical heuristics for the capacitated VRP. In *The Vehicle Routing Problem*, Discrete Mathematics and Applications, pages 109–128. Society for Industrial and Applied Mathematics.

Larrañaga, P., Kuijpers, C., Murga, R., Inza, I., and Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170.

Le, T. V., Stathopoulos, A., Van Woensel, T., and Ukkusuri, S. V. (2019). Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. *Transportation Research Part C: Emerging Technologies*, 103:83–103.

Lenstra, J. K. and Kan, A. H. G. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks. An International Journal*, 11(2):221–227.

Li, J.-Q., Mirchandani, P. B., and Borenstein, D. (2007). The vehicle rescheduling problem: Model and algorithms. *Networks. An International Journal*, 50(3):211–229.

Li, J. Q., Mirchandani, P. B., and Borenstein, D. (2009a). A lagrangian heuristic for the real-time vehicle rescheduling problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):419–433.

Li, J. Q., Mirchandani, P. B., and Borenstein, D. (2009b). Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research*, 194(3):711–727.

Macal, C. M. (2016). Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, 10(2):144–156.

Maciejewski, M. and Nagel, K. (2012). Towards multi-agent simulation of the dynamic vehicle routing problem in MATSim. In *Parallel Processing and Applied Mathematics*, volume 7204, pages 551–560. Springer Berlin Heidelberg.

Maeda, O., Nakamura, M., Ombuki, B. M., and Onaga, K. (1999). A genetic algorithm approach to vehicle routing problem with time deadlines in geographical information systems. In *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, volume 4, pages 595–600. IEEE.

Mamasis, K., Minis, I., and Dikas, G. (2013). Managing vehicle breakdown incidents during urban distribution of a common product. *Journal of the Operational Research Society*, 64(6):925–937.

Martin, S., Ouelhadj, D., Beullens, P., Ozcan, E., Juan, A. A., and Burke, E. K. (2016). A multi-agent based cooperative approach to scheduling and routing. *European Journal of Operational Research*, 254(1):169–178.

Mes, M., van der Heijden, M., and van Harten, A. (2007). Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research*, 181(1):59–75.

Michalewicz, Z. (1996). The travelling salesman problem. In *Genetic algorithms + data structures = evolution programs*, pages 209–237. Springer-Verlag, 3 edition.

Minis, I., Mamasis, K., and Zeimpekis, V. (2012). Real-time management of vehicle breakdowns in urban freight distribution. *Journal of Heuristics*, 18(3):375–400.

Monostori, L., Váncza, J., and Kumara, S. (2006). Agent-based systems for manufacturing. *CIRP Annals*, 55(2):697–720.

Monroy-Licht, M., Amaya, C. A., Langevin, A., and Rousseau, L.-M. (2017). The rescheduling arc routing problem. *International Transactions in Operational Research*, 24(6):1325–1346.

Montoya-Torres, J. R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129.

Mu, Q., Fu, Z., Lysgaard, J., and Eglese, R. (2011). Disruption management of the vehicle routing problem with vehicle breakdown. *Journal of the Operational Research Society*, 62(4):742–749.

Murray, C. and Karwan, M. (2013). A branch-and-bound-based solution approach for dynamic rerouting of airborne platforms. *Naval Research Logistics*, 60(2):141–159.

Murray, C. C. and Karwan, M. H. (2010). An extensible modeling framework for dynamic reassignment and rerouting in cooperative airborne operations. *Naval Research Logistics*, 57(7):634–652.

Nambiar, S. K. and Idicula, S. (2013). A multi-agent vehicle routing system for garbage collection. In *2013 Fifth International Conference on Advanced Computing (ICoAC)*, pages 72–76.

Ninikas, G. and Minis, I. (2018). Load transfer operations for a dynamic vehicle routing problem with mixed backhauls. *Journal on Vehicle Routing Algorithms*, 1(1):47–68.

Ombuki, B. M., Nakamura, M., and Osamu, M. (2002). A hybrid search based on genetic algorithms and tabu search for vehicle routing. *6th IASTED Intl. Conf. On Artificial Intelligence and Soft Computing (ASC 2002)*, pages 176–181.

Ombuki, B. M., Ross, B. J., and Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, 24(1):17–30.

Or, I. (1976). *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking.* PhD thesis, Department of Industrial Engineering and Management Sciences Northwestern University.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4):421–451.

Pandi, R. R., Ho, S. G., Nagavarapu, S. C., Tripathy, T., and Dauwels, J. (2020). Disruption management for dial-a-ride systems. *IEEE Intelligent Transportation Systems Magazine*, 12(4):219–234.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11.

Potvin, J.-Y. and Bengio, S. (1996). The vehicle routing problem with time windows part II: Genetic search. *INFORMS Journal on Computing*, 8(2):165–172. Publisher: INFORMS.

Potvin, J.-Y., Xu, Y., and Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, 33(4):1129–1137.

Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2):130–154. Publisher: INFORMS.

Psaraftis, H. N., Wen, M., and Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31.

Python Software Foundation (2021). The Python Language Reference — Python 3.8.12 documentation. https://docs.python.org/3.8/reference/index.html [06 September 2021].

Rios, B. H. O., Xavier, E. C., Miyazawa, F. K., Amorim, P., Curcio, E., and Santos, M. J. (2021). Recent dynamic vehicle routing problems: A survey. *Computers & Industrial Engineering*, 160:107604.

Rochat, Y. and Semet, F. (1994). A tabu search approach for delivering pet food and flour in switzerland. *Journal of the Operational Research Society*, 45(11):1233–1246.

Sarasola, B., Doerner, K. F., Schmid, V., and Alba, E. (2016). Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research*, 236(2):425–461.

Schyns, M. (2015). An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research*, 245(3):704–718.

Seyyedhasani, H. and Dvorak, J. S. (2018). Dynamic rerouting of a fleet of vehicles in agricultural operations through a dynamic multiple depot vehicle routing problem representation. *Biosystems Engineering*, 171:63–77.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.

Speranza, M. G. (2018). Trends in transportation and logistics. *European Journal of Operational Research*, 264(3):830–836.

Spliet, R., Gabor, A. F., and Dekker, R. (2014). The vehicle rescheduling problem. *Computers & Operations Research*, 43:129–136.

Sörensen, K. (2015). Metaheuristics-the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18.

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

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. John Wiley & Sons.

Thangiah, S. R., Shmygelska, O., and Mennell, W. (2001). An agent architecture for vehicle. *Proceedings of the 2001 ACM symposium on Applied computing*, pages 517–521.

Ulmer, M. W. (2020). Dynamic pricing and routing for same-day delivery. *Transportation Science*, 54(4):1016–1033.

Ulmer, M. W., Soeffker, N., and Mattfeld, D. C. (2018). Value function approximation for dynamic multi-period vehicle routing. *European Journal of Operational Research*, 269(3):883–899.

Ulmer, M. W., Thomas, B. W., Campbell, A. M., and Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, 55(1):75–100.

Van der Merwe, M., Minas, J. P., Ozlen, M., and Hearne, J. W. (2015). A mixed integer programming approach for asset protection during escaped wildfires. *Canadian Journal of Forest Research*, 45(4):444–451.

Van der Merwe, M., Ozlen, M., Hearne, J. W., and Minas, J. P. (2017). Dynamic rerouting of vehicles during cooperative wildfire response operations. *Annals of Operations Research*, 254(1):467–480.

Van Lieshout, R., Mulder, J., and Huisman, D. (2018). The vehicle rescheduling problem with retiming. *Computers and Operations Research*, 96:131–140.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., and Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624.

Vinsensius, A., Wang, Y., Chew, E. P., and Lee, L. H. (2020). Dynamic incentive mechanism for delivery slot management in e-commerce attended home delivery. *Transportation Science*, 54(3):567–587.

Vokřínek, J., Komenda, A., and Echouček, M. (2010). Agents towards vehicle routing problems. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 773–780. International Foundation for Autonomous Agents and Multiagent Systems.

VRP-REP (2014). Problem variant CVRP: Capacitated vehicle routing problem. http://www.vrp-rep.org/variants/item/cvrp.html [25 October 2018].

Wang, F., Liao, F., Li, Y., Yan, X., and Chen, X. (2021). An ensemble learning based multi-objective evolutionary algorithm for the dynamic vehicle routing problem with time windows. *Computers & Industrial Engineering*, 154:107131.

Wang, X., Wu, X., Wang, Z., and Hu, X. (2009). A model and an improved genetic algorithm for the vehicle routing problem with break-down vehicles. In *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*, pages 696–699. IEEE.

Wooldridge, M. and Jennings, N. R. (1995). Agent theories, architectures, and languages: A survey. In *Intelligent Agents*, volume 890, pages 1–39. Springer Berlin Heidelberg.

Yang, J., Jaillet, P., and Mahmassani, H. (2004). Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, 38(2):135–148.

Zeddini, B., Temani, M., Yassine, A., and Ghedira, K. (2008). An agent-oriented approach for the dynamic vehicle routing problem. In *2008 International Workshop on Advanced Information Systems for Enterprises*, pages 70–76. IEEE.

Zhang, H., Ge, H., Yang, J., and Tong, Y. (2022). Review of vehicle routing problems: Models, classification and solving algorithms. *Archives of Computational Methods in Engineering*, 29(1):195–221.

Zou, H. and Dessouky, M. M. (2018). A look-ahead partial routing framework for the stochastic and dynamic vehicle routing problem. *Journal on Vehicle Routing Algorithms*, pages 1–16.